

FLTK 1.3.8 Programming Manual



Revision 9.8 by F. Costantini, D. Gibson, M. Melcher,
A. Schlosser, B. Spitzak, and M. Sweet.

Copyright 1998-2022 by Bill Spitzak and others.

This software and manual are provided under the terms of the GNU Library General Public License.
Permission is granted to reproduce this manual or any portion for any purpose,
provided this copyright and permission notice are preserved.

Generated by Doxygen 1.9.4

September 8, 2022

1 FLTK Programming Manual	1
2 Preface	3
2.1 Organization	3
2.2 Conventions	4
2.3 Abbreviations	4
2.4 Copyrights and Trademarks	4
3 Introduction to FLTK	5
3.1 History of FLTK	5
3.2 Features	6
3.3 Licensing	6
3.4 What Does "FLTK" Mean?	7
3.5 Building and Installing FLTK Under UNIX and Apple OS X	7
3.6 Building FLTK Under Microsoft Windows	9
3.6.1 GNU toolsets (Cygwin or MinGW) hosted on Windows	9
3.6.2 Using the Visual C++ DLL Library	10
3.7 Internet Resources	10
3.8 Reporting Bugs	10
4 FLTK Basics	11
4.1 Writing Your First FLTK Program	11
4.1.1 Creating the Widgets	12
4.1.2 Creating Widget hierarchies	12
4.1.3 Get/Set Methods	12
4.1.4 Redrawing After Changing Attributes	13
4.1.5 Labels	13
4.1.6 Showing the Window	13
4.1.7 The Main Event Loop	13
4.2 Compiling Programs with Standard Compilers	14
4.3 Compiling Programs with Makefiles	14
4.4 Compiling Programs with Microsoft Visual C++	15
4.5 Naming	15
4.6 Header Files	15
5 Common Widgets and Attributes	17
5.1 Buttons	17
5.2 Text	18
5.3 Valuators	18
5.4 Groups	19
5.5 Setting the Size and Position of Widgets	20
5.6 Colors	20
5.7 Box Types	21
5.7.1 Making Your Own Boxtypes	22

5.8 Labels and Label Types	23
5.9 Callbacks	26
5.10 Shortcuts	26
6 Designing a Simple Text Editor	27
6.1 Determining the Goals of the Text Editor	27
6.2 Designing the Main Window	27
6.3 Variables	28
6.4 Menubars and Menus	28
6.5 Editing the Text	28
6.6 The Replace Dialog	29
6.7 Callbacks	29
6.7.1 changed_cb()	29
6.7.2 copy_cb()	29
6.7.3 cut_cb()	29
6.7.4 delete_cb()	30
6.7.5 find_cb()	30
6.7.6 find2_cb()	30
6.7.7 new_cb()	30
6.7.8 open_cb()	31
6.7.9 paste_cb()	31
6.7.10 quit_cb()	31
6.7.11 replace_cb()	31
6.7.12 replace2_cb()	31
6.7.13 replall_cb()	32
6.7.14 replcan_cb()	32
6.7.15 save_cb()	32
6.7.16 saveas_cb()	32
6.8 Other Functions	33
6.8.1 check_save()	33
6.8.2 load_file()	33
6.8.3 save_file()	33
6.8.4 set_title()	33
6.9 The main() Function	34
6.10 Compiling the Editor	34
6.11 The Final Product	34
6.12 Advanced Features	35
6.12.1 Syntax Highlighting	35
7 Drawing Things in FLTK	39
7.1 When Can You Draw Things in FLTK?	39
7.1.1 What Drawing Unit do FLTK drawing functions use?	39
7.2 Drawing Functions	40

7.2.1 Boxes	40
7.2.2 Clipping	41
7.3 Colors	42
7.3.1 Line Dashes and Thickness	44
7.3.2 Drawing Fast Shapes	45
7.3.3 Drawing Complex Shapes	47
7.3.4 Drawing Text	49
7.3.5 Fonts	51
7.3.6 Character Encoding	52
7.3.7 Drawing Overlays	52
7.4 Drawing Images	52
7.4.1 Direct Image Drawing	53
7.4.2 Direct Image Reading	54
7.4.3 Image Classes	55
7.4.4 Offscreen Drawing	56
8 Handling Events	59
8.1 The FLTK Event Model	59
8.2 Mouse Events	59
8.2.1 FL_PUSH	59
8.2.2 FL_DRAG	59
8.2.3 FL_RELEASE	60
8.2.4 FL_MOVE	60
8.2.5 FL_MOUSEWHEEL	60
8.3 Focus Events	60
8.3.1 FL_ENTER	60
8.3.2 FL_LEAVE	60
8.3.3 FL_FOCUS	60
8.3.4 FL_UNFOCUS	60
8.4 Keyboard Events	61
8.4.1 FL_KEYBOARD, FL_KEYDOWN, FL_KEYUP	61
8.4.2 FL_SHORTCUT	61
8.5 Widget Events	61
8.5.1 FL_DEACTIVATE	61
8.5.2 FL_ACTIVATE	61
8.5.3 FL_HIDE	62
8.5.4 FL_SHOW	62
8.6 Clipboard Events	62
8.6.1 FL_PASTE	62
8.6.2 FL_SELECTIONCLEAR	62
8.7 Drag and Drop Events	62
8.7.1 FL_DND_ENTER	63

8.7.2 FL_DND_DRAG	63
8.7.3 FL_DND_LEAVE	63
8.7.4 FL_DND_RELEASE	63
8.8 Other events	63
8.8.1 FL_SCREEN_CONFIGURATION_CHANGED	63
8.8.2 FL_FULLSCREEN	63
8.9 Fl::event_*() methods	64
8.10 Event Propagation	64
8.11 FLTK Compose-Character Sequences	65
9 Adding and Extending Widgets	67
9.1 Subclassing	67
9.2 Making a Subclass of Fl_Widget	67
9.3 The Constructor	67
9.4 Protected Methods of Fl_Widget	68
9.5 Handling Events	71
9.6 Drawing the Widget	72
9.7 Resizing the Widget	72
9.8 Making a Composite Widget	72
9.9 Cut and Paste Support	74
9.10 Drag And Drop Support	74
9.11 Making a subclass of Fl_Window	74
10 Using OpenGL	75
10.1 Using OpenGL in FLTK	75
10.2 Making a Subclass of Fl_Gl_Window	75
10.2.1 Defining the Subclass	76
10.2.2 The draw() Method	76
10.2.3 The handle() Method	76
10.3 Using OpenGL in Normal FLTK Windows	77
10.4 OpenGL Drawing Functions	78
10.5 Speeding up OpenGL	79
10.6 Using OpenGL Optimizer with FLTK	79
10.7 Using OpenGL 3.0 (or higher versions)	81
11 Programming with FLUID	83
11.1 What is FLUID?	83
11.2 Running FLUID Under UNIX	84
11.3 Running FLUID Under Microsoft Windows	84
11.4 Compiling .fl files	85
11.5 A Short Tutorial	85
11.5.1 The CubeView Class	86
11.5.2 The CubeViewUI Class	88

11.5.3 Adding Constructor Initialization Code	92
11.5.4 Generating the Code	92
11.6 FLUID Reference	92
11.6.1 The Widget Browser	92
11.6.2 Menu Items	93
11.6.3 The Widget Panel	102
11.7 GUI Attributes	102
11.7.1 Style Attributes	105
11.7.2 C++ Attributes	107
11.8 Selecting and Moving Widgets	110
11.9 Image Labels	110
11.10 Internationalization with FLUID	113
11.10.1 I18N Methods	113
11.10.2 Using GNU gettext for I18N	113
11.10.3 Using POSIX catgets for I18N	114
11.11 Known limitations	114
12 Advanced FLTK	115
12.1 Multithreading	115
12.2 FLTK multithread locking - Fl::lock() and Fl::unlock()	115
12.3 Simple multithreaded examples using Fl::lock	116
12.4 FLTK multithreaded "lockless programming"	118
12.5 FLTK multithreaded Constraints	119
13 Unicode and UTF-8 Support	121
13.1 About Unicode, ISO 10646 and UTF-8	121
13.2 Unicode in FLTK	123
13.3 Illegal Unicode and UTF-8 Sequences	124
13.4 FLTK Unicode and UTF-8 Functions	124
13.5 FLTK Unicode Versions of System Calls	128
14 FLTK Enumerations	129
14.1 Version Numbers	129
14.2 Events	130
14.3 Callback "When" Conditions	131
14.4 Fl::event_button() Values	131
14.5 Fl::event_key() Values	131
14.6 Fl::event_state() Values	132
14.7 Alignment Values	133
14.8 Fonts	133
14.9 Colors	134
14.9.1 Color Constants	134
14.10 Cursors	135

14.11	FD "When" Conditions	136
14.12	Damage Masks	136
15	GLUT Compatibility	137
15.1	Using the GLUT Compatibility Header File	137
15.2	Known Problems	137
15.3	Mixing GLUT and FLTK Code	138
15.4	class <code>Fl_Glut_Window</code>	139
15.4.1	Class Hierarchy	139
15.4.2	Include Files	139
15.4.3	Description	139
15.4.4	Members	139
15.4.5	Methods	140
16	Forms Compatibility	141
16.1	Importing Forms Layout Files	141
16.2	Using the Compatibility Header File	141
16.3	Problems You Will Encounter	142
16.4	Additional Notes	143
17	Operating System Issues	147
17.1	Accessing the OS Interfaces	147
17.2	The UNIX (X11) Interface	148
17.2.1	Handling Other X Events	148
17.2.2	Drawing using Xlib	149
17.2.3	Changing the Display, Screen, or X Visual	149
17.2.4	Using a Subclass of <code>Fl_Window</code> for Special X Stuff	151
17.2.5	Setting the Icon of a Window	152
17.2.6	X Resources	153
17.3	The Windows (WIN32) Interface	154
17.3.1	Using filenames with non-ASCII characters	154
17.3.2	Responding to <code>WM_QUIT</code>	154
17.3.3	Handling Other WIN32 Messages	154
17.3.4	Drawing Things Using the WIN32 GDI	155
17.3.5	Setting the Icon of a Window	155
17.3.6	How to Not Get a MSDOS Console Window	155
17.3.7	Known WIN32 Bugs and Problems	156
17.4	The Apple OS X Interface	156
17.4.1	Setting the icon of an application	158
17.4.2	Drawing Things Using Quartz	158
17.4.3	Internationalization	158
17.4.4	OpenGL and 'retina' displays	159
17.4.5	<code>Fl_Double_Window</code>	159

17.4.6 Mac File System Specifics	159
18 Migrating Code from FLTK 1.0 to 1.1	161
18.1 Color Values	161
18.2 Cut and Paste Support	161
18.3 File Chooser	161
18.4 Function Names	161
18.5 Image Support	162
18.6 Keyboard Navigation	162
19 Migrating Code from FLTK 1.1 to 1.3	163
19.1 Migrating From FLTK 1.0	163
19.2 Fl_Scroll Widget	163
19.3 Unicode (UTF-8)	163
19.4 Widget Coordinate Representation	164
20 Developer Information	165
20.1 Non-ASCII Characters	167
20.2 Document Structure	167
20.3 Creating Links	168
20.4 Paragraph Layout	169
20.5 Navigation Elements	170
21 Software License	171
22 Example Source Code	177
22.1 Example Applications	177
22.1.1 adjuster	177
22.1.2 arc	177
22.1.3 ask	177
22.1.4 bitmap	178
22.1.5 blocks	178
22.1.6 boxtype	178
22.1.7 browser	178
22.1.8 button	178
22.1.9 buttons	178
22.1.10 checkers	178
22.1.11 clock	178
22.1.12 colbrowser	178
22.1.13 color_chooser	179
22.1.14 cube	179
22.1.15 CubeView	179
22.1.16 cursor	179
22.1.17 curve	179

22.1.18 demo	179
22.1.19 device	179
22.1.20 doublebuffer	179
22.1.21 editor	179
22.1.22 fast_slow	180
22.1.23 file_chooser	180
22.1.24 fonts	180
22.1.25 forms	180
22.1.26 fractals	180
22.1.27 fullscreen	180
22.1.28 gl_overlay	180
22.1.29 glpuzzle	180
22.1.30 hello	180
22.1.31 help	180
22.1.32 iconize	181
22.1.33 image	181
22.1.34 inactive	181
22.1.35 input	181
22.1.36 input_choice	181
22.1.37 keyboard	181
22.1.38 label	181
22.1.39 line_style	181
22.1.40 list_visuals	181
22.1.41 mandelbrot	181
22.1.42 menubar	182
22.1.43 message	182
22.1.44 minimum	182
22.1.45 navigation	182
22.1.46 output	182
22.1.47 overlay	182
22.1.48 pack	182
22.1.49 pixmap_browser	182
22.1.50 pixmap	182
22.1.51 preferences	182
22.1.52 radio	183
22.1.53 resizebox	183
22.1.54 resize	183
22.1.55 scroll	183
22.1.56 shape	183
22.1.57 subwindow	183
22.1.58 sudoku	183
22.1.59 symbols	183

22.1.60 tabs	183
22.1.61 threads	184
22.1.62 tile	184
22.1.63 tiled_image	184
22.1.64 unittests	184
22.1.65 utf8	184
22.1.66 valuator	184
22.1.67 fluid	184
23 FAQ (Frequently Asked Questions)	185
23.1 Where do I start learning FLTK?	185
23.2 How do I make a box with text?	185
23.3 Can I use FLTK to make closed-source commercial applications?	185
23.4 Hitting the 'Escape' key closes windows - how do I prevent this?	186
24 Todo List	187
25 Deprecated List	191
26 Module Index	193
26.1 Modules	193
27 Hierarchical Index	195
27.1 Class Hierarchy	195
28 Class Index	199
28.1 Class List	199
29 File Index	207
29.1 File List	207
30 Module Documentation	213
30.1 Callback function typedefs	213
30.1.1 Detailed Description	214
30.1.2 Typedef Documentation	214
30.1.2.1 FI_Event_Dispatch	214
30.2 Windows handling functions	214
30.2.1 Detailed Description	214
30.2.2 Function Documentation	215
30.2.2.1 default_atclose()	215
30.2.2.2 first_window() [1/2]	215
30.2.2.3 first_window() [2/2]	215
30.2.2.4 grab() [1/2]	215
30.2.2.5 grab() [2/2]	215
30.2.2.6 modal()	215

30.2.2.7	next_window()	216
30.2.2.8	set_atclose()	216
30.2.3	Variable Documentation	216
30.2.3.1	atclose	216
30.3	Events handling functions	216
30.3.1	Detailed Description	219
30.3.2	Function Documentation	219
30.3.2.1	add_handler()	219
30.3.2.2	add_system_handler()	219
30.3.2.3	belowmouse() [1/2]	220
30.3.2.4	belowmouse() [2/2]	220
30.3.2.5	compose()	220
30.3.2.6	compose_reset()	220
30.3.2.7	disable_im()	221
30.3.2.8	enable_im()	221
30.3.2.9	event()	221
30.3.2.10	event_button()	221
30.3.2.11	event_button1()	221
30.3.2.12	event_button2()	221
30.3.2.13	event_button3()	221
30.3.2.14	event_buttons()	222
30.3.2.15	event_clicks() [1/2]	222
30.3.2.16	event_clicks() [2/2]	222
30.3.2.17	event_clipboard()	222
30.3.2.18	event_clipboard_type()	222
30.3.2.19	event_dispatch()	223
30.3.2.20	event_dx()	223
30.3.2.21	event_dy()	223
30.3.2.22	event_inside() [1/2]	223
30.3.2.23	event_inside() [2/2]	224
30.3.2.24	event_is_click() [1/2]	224
30.3.2.25	event_is_click() [2/2]	224
30.3.2.26	event_key() [1/2]	225
30.3.2.27	event_key() [2/2]	225
30.3.2.28	event_length()	225
30.3.2.29	event_original_key()	225
30.3.2.30	event_state() [1/2]	226
30.3.2.31	event_state() [2/2]	226
30.3.2.32	event_text()	226
30.3.2.33	event_x_root()	226
30.3.2.34	event_y_root()	227
30.3.2.35	focus() [1/2]	227

30.3.2.36	focus() [2/2]	227
30.3.2.37	get_key()	227
30.3.2.38	get_mouse()	227
30.3.2.39	handle()	227
30.3.2.40	handle_()	228
30.3.2.41	pushed() [1/2]	228
30.3.2.42	pushed() [2/2]	228
30.3.2.43	remove_handler()	229
30.3.2.44	remove_system_handler()	229
30.3.2.45	test_shortcut()	229
30.3.3	Variable Documentation	229
30.3.3.1	fl_eventnames	229
30.3.3.2	fl_fontnames	230
30.4	Selection & Clipboard functions	230
30.4.1	Detailed Description	231
30.4.2	Function Documentation	231
30.4.2.1	add_clipboard_notify()	231
30.4.2.2	clipboard_contains()	231
30.4.2.3	copy()	231
30.4.2.4	dnd()	232
30.4.2.5	paste() [1/2]	232
30.4.2.6	paste() [2/2]	232
30.4.2.7	selection()	233
30.4.2.8	selection_owner() [1/2]	233
30.4.2.9	selection_owner() [2/2]	233
30.5	Screen functions	233
30.5.1	Detailed Description	234
30.5.2	Function Documentation	234
30.5.2.1	screen_dpi()	234
30.5.2.2	screen_num() [1/2]	234
30.5.2.3	screen_num() [2/2]	236
30.5.2.4	screen_work_area() [1/3]	236
30.5.2.5	screen_work_area() [2/3]	236
30.5.2.6	screen_work_area() [3/3]	237
30.5.2.7	screen_xywh() [1/4]	237
30.5.2.8	screen_xywh() [2/4]	237
30.5.2.9	screen_xywh() [3/4]	238
30.5.2.10	screen_xywh() [4/4]	238
30.6	Color & Font functions	238
30.6.1	Detailed Description	240
30.6.2	Function Documentation	240
30.6.2.1	fl_color() [1/3]	240

30.6.2.2 fl_color() [2/3]	240
30.6.2.3 fl_color() [3/3]	240
30.6.2.4 fl_color_average()	241
30.6.2.5 fl_contrast()	241
30.6.2.6 fl_font() [1/2]	241
30.6.2.7 fl_font() [2/2]	241
30.6.2.8 fl_height() [1/2]	242
30.6.2.9 fl_height() [2/2]	242
30.6.2.10 fl_latin1_to_local()	242
30.6.2.11 fl_local_to_latin1()	242
30.6.2.12 fl_local_to_mac_roman()	243
30.6.2.13 fl_mac_roman_to_local()	243
30.6.2.14 fl_show_colormap()	243
30.6.2.15 fl_size()	244
30.6.2.16 fl_text_extents() [1/2]	244
30.6.2.17 fl_text_extents() [2/2]	245
30.6.2.18 fl_width()	245
30.6.2.19 fl_xpixel() [1/2]	245
30.6.2.20 fl_xpixel() [2/2]	245
30.6.2.21 free_color()	246
30.6.2.22 get_color() [1/2]	246
30.6.2.23 get_color() [2/2]	246
30.6.2.24 get_font()	246
30.6.2.25 get_font_name()	246
30.6.2.26 get_font_sizes()	247
30.6.2.27 set_color() [1/2]	247
30.6.2.28 set_color() [2/2]	247
30.6.2.29 set_font()	247
30.6.2.30 set_fonts()	247
30.7 Drawing functions	248
30.7.1 Detailed Description	252
30.7.2 Macro Definition Documentation	252
30.7.2.1 fl_clip	252
30.7.3 Enumeration Type Documentation	252
30.7.3.1 anonymous enum	252
30.7.4 Function Documentation	253
30.7.4.1 copy_offscreen()	253
30.7.4.2 fl_add_symbol()	253
30.7.4.3 fl_arc() [1/2]	253
30.7.4.4 fl_arc() [2/2]	254
30.7.4.5 fl_begin_complex_polygon()	254
30.7.4.6 fl_begin_offscreen()	255

30.7.4.7 fl_begin_points()	255
30.7.4.8 fl_can_do_alpha_blending()	255
30.7.4.9 fl_circle()	255
30.7.4.10 fl_clip_box()	255
30.7.4.11 fl_clip_region()	256
30.7.4.12 fl_copy_offscreen()	256
30.7.4.13 fl_create_offscreen()	256
30.7.4.14 fl_cursor()	257
30.7.4.15 fl_curve()	257
30.7.4.16 fl_delete_offscreen()	257
30.7.4.17 fl_draw() [1/4]	257
30.7.4.18 fl_draw() [2/4]	258
30.7.4.19 fl_draw() [3/4]	258
30.7.4.20 fl_draw() [4/4]	258
30.7.4.21 fl_draw_box()	258
30.7.4.22 fl_draw_image() [1/2]	259
30.7.4.23 fl_draw_image() [2/2]	259
30.7.4.24 fl_draw_image_mono() [1/2]	260
30.7.4.25 fl_draw_image_mono() [2/2]	260
30.7.4.26 fl_draw_pixmap() [1/2]	261
30.7.4.27 fl_draw_pixmap() [2/2]	261
30.7.4.28 fl_draw_symbol()	261
30.7.4.29 fl_expand_text()	262
30.7.4.30 fl_frame()	262
30.7.4.31 fl_frame2()	262
30.7.4.32 fl_gap()	263
30.7.4.33 fl_line_style()	263
30.7.4.34 fl_measure()	263
30.7.4.35 fl_measure_pixmap() [1/2]	264
30.7.4.36 fl_measure_pixmap() [2/2]	264
30.7.4.37 fl_mult_matrix()	264
30.7.4.38 fl_not_clipped()	265
30.7.4.39 fl_old_shortcut()	265
30.7.4.40 fl_pie()	266
30.7.4.41 fl_polygon() [1/2]	266
30.7.4.42 fl_polygon() [2/2]	266
30.7.4.43 fl_pop_clip()	267
30.7.4.44 fl_push_clip()	267
30.7.4.45 fl_push_matrix()	267
30.7.4.46 fl_read_image()	267
30.7.4.47 fl_rect()	268
30.7.4.48 fl_rectf()	268

30.7.4.49 fl_reset_spot()	268
30.7.4.50 fl_rotate()	268
30.7.4.51 fl_scale() [1/2]	268
30.7.4.52 fl_scale() [2/2]	269
30.7.4.53 fl_scroll()	269
30.7.4.54 fl_set_spot()	269
30.7.4.55 fl_set_status()	270
30.7.4.56 fl_shortcut_label() [1/2]	270
30.7.4.57 fl_shortcut_label() [2/2]	270
30.7.4.58 fl_transform_dx()	271
30.7.4.59 fl_transform_dy()	271
30.7.4.60 fl_transform_x()	271
30.7.4.61 fl_transform_y()	271
30.7.4.62 fl_transformed_vertex()	272
30.7.4.63 fl_translate()	272
30.7.4.64 fl_vertex()	272
30.8 Multithreading support functions	272
30.8.1 Detailed Description	273
30.8.2 Function Documentation	273
30.8.2.1 awake() [1/2]	273
30.8.2.2 awake() [2/2]	273
30.8.2.3 lock()	273
30.8.2.4 thread_message()	274
30.8.2.5 unlock()	274
30.9 Safe widget deletion support functions	274
30.9.1 Detailed Description	274
30.9.2 Function Documentation	275
30.9.2.1 clear_widget_pointer()	275
30.9.2.2 delete_widget()	275
30.9.2.3 do_widget_deletion()	276
30.9.2.4 release_widget_pointer()	276
30.9.2.5 watch_widget_pointer()	276
30.10 Cairo Support Functions and Classes	277
30.10.1 Detailed Description	277
30.10.2 Function Documentation	277
30.10.2.1 cairo_autolink_context() [1/2]	277
30.10.2.2 cairo_autolink_context() [2/2]	277
30.10.2.3 cairo_cc()	278
30.10.2.4 cairo_make_current()	278
30.11 Unicode and UTF-8 functions	278
30.11.1 Detailed Description	280
30.11.2 Macro Definition Documentation	280

30.11.2.1 ERRORS_TO_CP1252	280
30.11.2.2 ERRORS_TO_ISO8859_1	280
30.11.2.3 STRICT_RFC3629	280
30.11.3 Function Documentation	280
30.11.3.1 fl_access()	281
30.11.3.2 fl_chmod()	281
30.11.3.3 fl_fopen()	281
30.11.3.4 fl_getcwd()	282
30.11.3.5 fl_getenv()	282
30.11.3.6 fl_make_path()	282
30.11.3.7 fl_make_path_for_file()	282
30.11.3.8 fl_mkdir()	283
30.11.3.9 fl_nonspacing()	283
30.11.3.10 fl_open()	283
30.11.3.11 fl_rename()	284
30.11.3.12 fl_rmdir()	284
30.11.3.13 fl_stat()	284
30.11.3.14 fl_system()	285
30.11.3.15 fl_ucs_to_Utf16()	285
30.11.3.16 fl_unlink()	285
30.11.3.17 fl_utf8back()	286
30.11.3.18 fl_utf8bytes()	286
30.11.3.19 fl_utf8decode()	286
30.11.3.20 fl_utf8encode()	286
30.11.3.21 fl_utf8from_mb()	287
30.11.3.22 fl_utf8froma()	287
30.11.3.23 fl_utf8fromwc()	287
30.11.3.24 fl_utf8fwd()	288
30.11.3.25 fl_utf8len()	288
30.11.3.26 fl_utf8len1()	288
30.11.3.27 fl_utf8locale()	288
30.11.3.28 fl_utf8test()	289
30.11.3.29 fl_utf8to_mb()	289
30.11.3.30 fl_utf8toa()	289
30.11.3.31 fl_utf8toUtf16()	289
30.11.3.32 fl_utf8towc()	290
30.11.3.33 fl_utf_strcasecmp()	290
30.11.3.34 fl_utf_strncasecmp()	291
30.11.3.35 fl_utf_tolower()	291
30.11.3.36 fl_utf_toupper()	291
30.11.3.37 fl_wcwidth()	291
30.11.3.38 fl_wcwidth_()	292

30.12 Mac OS X-specific symbols	292
30.12.1 Detailed Description	293
30.12.2 Function Documentation	293
30.12.2.1 fl_mac_set_about()	293
30.12.2.2 fl_open_callback()	293
30.12.2.3 gl_texture_pile_height() [1/2]	293
30.12.2.4 gl_texture_pile_height() [2/2]	293
30.12.3 Variable Documentation	293
30.12.3.1 fl_mac_quit_early	294
30.13 Common Dialogs classes and functions	294
30.13.1 Detailed Description	295
30.13.2 Function Documentation	295
30.13.2.1 fl_alert()	295
30.13.2.2 fl_ask()	296
30.13.2.3 fl_beep()	296
30.13.2.4 fl_choice()	296
30.13.2.5 fl_choice_n()	298
30.13.2.6 fl_color_chooser() [1/2]	298
30.13.2.7 fl_color_chooser() [2/2]	299
30.13.2.8 fl_dir_chooser()	300
30.13.2.9 fl_file_chooser()	301
30.13.2.10 fl_file_chooser_callback()	302
30.13.2.11 fl_file_chooser_ok_label()	302
30.13.2.12 fl_input()	302
30.13.2.13 fl_message()	302
30.13.2.14 fl_message_hotspot() [1/2]	303
30.13.2.15 fl_message_hotspot() [2/2]	303
30.13.2.16 fl_message_icon()	303
30.13.2.17 fl_message_title()	304
30.13.2.18 fl_message_title_default()	304
30.13.2.19 fl_password()	304
30.13.3 Variable Documentation	305
30.13.3.1 error	305
30.13.3.2 fatal	305
30.13.3.3 warning	305
30.14 File names and URI utility functions	306
30.14.1 Detailed Description	306
30.14.2 Typedef Documentation	306
30.14.2.1 FI_File_Sort_F	307
30.14.3 Function Documentation	307
30.14.3.1 fl_decode_uri()	307
30.14.3.2 fl_filename_absolute()	307

30.14.3.3 fl_filename_expand()	307
30.14.3.4 fl_filename_ext()	309
30.14.3.5 fl_filename_free_list()	309
30.14.3.6 fl_filename_isdir()	309
30.14.3.7 fl_filename_list()	310
30.14.3.8 fl_filename_match()	310
30.14.3.9 fl_filename_name()	311
30.14.3.10 fl_filename_relative()	311
30.14.3.11 fl_filename_setext()	312
30.14.3.12 fl_open_uri()	312
31 Class Documentation	315
31.1 FI_Preferences::Entry Struct Reference	315
31.2 FI Class Reference	315
31.2.1 Detailed Description	323
31.2.2 Member Enumeration Documentation	323
31.2.2.1 FI_Option	323
31.2.3 Member Function Documentation	324
31.2.3.1 abi_check()	324
31.2.3.2 abi_version()	324
31.2.3.3 add_check()	324
31.2.3.4 add_fd()	325
31.2.3.5 add_idle()	325
31.2.3.6 add_timeout()	325
31.2.3.7 api_version()	325
31.2.3.8 arg()	326
31.2.3.9 args() [1/2]	326
31.2.3.10 args() [2/2]	327
31.2.3.11 background()	327
31.2.3.12 background2()	327
31.2.3.13 box_color()	328
31.2.3.14 box_dh()	328
31.2.3.15 box_dw()	328
31.2.3.16 box_dx()	328
31.2.3.17 box_dy()	328
31.2.3.18 check()	329
31.2.3.19 display()	329
31.2.3.20 dnd_text_ops() [1/2]	329
31.2.3.21 dnd_text_ops() [2/2]	329
31.2.3.22 draw_box_active()	329
31.2.3.23 flush()	329
31.2.3.24 get_system_colors()	329

31.2.3.25 <code>gl_visual()</code>	330
31.2.3.26 <code>is_scheme()</code>	330
31.2.3.27 <code>option()</code> [1/2]	330
31.2.3.28 <code>option()</code> [2/2]	331
31.2.3.29 <code>own_colormap()</code>	331
31.2.3.30 <code>readqueue()</code>	331
31.2.3.31 <code>ready()</code>	332
31.2.3.32 <code>release()</code>	332
31.2.3.33 <code>reload_scheme()</code>	332
31.2.3.34 <code>remove_check()</code>	332
31.2.3.35 <code>remove_timeout()</code>	332
31.2.3.36 <code>repeat_timeout()</code>	333
31.2.3.37 <code>run()</code>	333
31.2.3.38 <code>scheme()</code>	333
31.2.3.39 <code>scrollbar_size()</code> [1/2]	333
31.2.3.40 <code>scrollbar_size()</code> [2/2]	334
31.2.3.41 <code>set_box_color()</code>	334
31.2.3.42 <code>set_idle()</code>	334
31.2.3.43 <code>use_high_res_GL()</code> [1/2]	334
31.2.3.44 <code>use_high_res_GL()</code> [2/2]	335
31.2.3.45 <code>version()</code>	335
31.2.3.46 <code>visible_focus()</code> [1/2]	335
31.2.3.47 <code>visible_focus()</code> [2/2]	335
31.2.3.48 <code>visual()</code>	335
31.2.3.49 <code>wait()</code>	336
31.2.4 Member Data Documentation	336
31.2.4.1 <code>help</code>	336
31.2.4.2 <code>idle</code>	336
31.3 FI_Adjuster Class Reference	337
31.3.1 Detailed Description	337
31.3.2 Constructor & Destructor Documentation	338
31.3.2.1 <code>FI_Adjuster()</code>	338
31.3.3 Member Function Documentation	338
31.3.3.1 <code>draw()</code>	338
31.3.3.2 <code>handle()</code>	338
31.3.3.3 <code>soft()</code> [1/2]	339
31.3.3.4 <code>soft()</code> [2/2]	339
31.3.3.5 <code>value_damage()</code>	339
31.4 FI_Bitmap Class Reference	339
31.4.1 Detailed Description	340
31.4.2 Member Function Documentation	340
31.4.2.1 <code>copy()</code>	340

31.4.2.2	draw()	341
31.4.2.3	label() [1/2]	341
31.4.2.4	label() [2/2]	341
31.4.2.5	uncache()	341
31.5	FI_BMP_Image Class Reference	341
31.5.1	Detailed Description	342
31.5.2	Constructor & Destructor Documentation	342
31.5.2.1	FI_BMP_Image()	342
31.6	FI_Box Class Reference	342
31.6.1	Detailed Description	343
31.6.2	Constructor & Destructor Documentation	343
31.6.2.1	FI_Box()	343
31.6.3	Member Function Documentation	343
31.6.3.1	draw()	343
31.6.3.2	handle()	343
31.7	FI_Browser Class Reference	344
31.7.1	Detailed Description	347
31.7.2	Constructor & Destructor Documentation	347
31.7.2.1	FI_Browser()	347
31.7.3	Member Function Documentation	348
31.7.3.1	_remove()	348
31.7.3.2	add()	348
31.7.3.3	bottomline()	348
31.7.3.4	clear()	349
31.7.3.5	column_char() [1/2]	349
31.7.3.6	column_char() [2/2]	349
31.7.3.7	column_widths() [1/2]	349
31.7.3.8	column_widths() [2/2]	350
31.7.3.9	data() [1/2]	350
31.7.3.10	data() [2/2]	350
31.7.3.11	display()	350
31.7.3.12	displayed()	351
31.7.3.13	find_line()	351
31.7.3.14	format_char() [1/2]	351
31.7.3.15	format_char() [2/2]	352
31.7.3.16	full_height()	352
31.7.3.17	hide() [1/2]	352
31.7.3.18	hide() [2/2]	353
31.7.3.19	icon() [1/2]	353
31.7.3.20	icon() [2/2]	353
31.7.3.21	incr_height()	353
31.7.3.22	insert() [1/2]	354

31.7.3.23 insert() [2/2]	354
31.7.3.24 item_at()	354
31.7.3.25 item_draw()	355
31.7.3.26 item_first()	355
31.7.3.27 item_height()	355
31.7.3.28 item_last()	356
31.7.3.29 item_next()	356
31.7.3.30 item_prev()	356
31.7.3.31 item_select()	357
31.7.3.32 item_selected()	357
31.7.3.33 item_swap()	357
31.7.3.34 item_text()	358
31.7.3.35 item_width()	358
31.7.3.36 lineno()	358
31.7.3.37 lineposition()	359
31.7.3.38 load()	359
31.7.3.39 make_visible()	359
31.7.3.40 middleline()	360
31.7.3.41 move()	360
31.7.3.42 remove()	360
31.7.3.43 remove_icon()	361
31.7.3.44 select()	361
31.7.3.45 selected()	361
31.7.3.46 show() [1/2]	362
31.7.3.47 show() [2/2]	362
31.7.3.48 size()	362
31.7.3.49 swap() [1/2]	362
31.7.3.50 swap() [2/2]	362
31.7.3.51 text() [1/2]	363
31.7.3.52 text() [2/2]	363
31.7.3.53 textsize()	363
31.7.3.54 topline() [1/2]	363
31.7.3.55 topline() [2/2]	364
31.7.3.56 value() [1/2]	364
31.7.3.57 value() [2/2]	364
31.7.3.58 visible()	364
31.8 FI_Browser_ Class Reference	365
31.8.1 Detailed Description	368
31.8.1.1 Keyboard navigation of browser items	368
31.8.2 Member Enumeration Documentation	368
31.8.2.1 anonymous enum	368
31.8.3 Constructor & Destructor Documentation	369

31.8.3.1 FI_Browser_()	369
31.8.4 Member Function Documentation	369
31.8.4.1 bbox()	369
31.8.4.2 deleting()	369
31.8.4.3 deselect()	369
31.8.4.4 display()	370
31.8.4.5 displayed()	370
31.8.4.6 draw()	370
31.8.4.7 find_item()	370
31.8.4.8 full_height()	371
31.8.4.9 full_width()	371
31.8.4.10 handle()	371
31.8.4.11 has_scrollbar()	371
31.8.4.12 hposition() [1/2]	372
31.8.4.13 hposition() [2/2]	372
31.8.4.14 incr_height()	372
31.8.4.15 inserting()	373
31.8.4.16 item_at()	373
31.8.4.17 item_draw()	373
31.8.4.18 item_first()	373
31.8.4.19 item_height()	373
31.8.4.20 item_last()	374
31.8.4.21 item_next()	374
31.8.4.22 item_prev()	374
31.8.4.23 item_quick_height()	374
31.8.4.24 item_select()	375
31.8.4.25 item_selected()	375
31.8.4.26 item_swap()	375
31.8.4.27 item_text()	376
31.8.4.28 item_width()	376
31.8.4.29 leftedge()	376
31.8.4.30 new_list()	376
31.8.4.31 position() [1/2]	376
31.8.4.32 position() [2/2]	377
31.8.4.33 redraw_line()	377
31.8.4.34 redraw_lines()	377
31.8.4.35 replacing()	377
31.8.4.36 resize()	378
31.8.4.37 scrollbar_left()	378
31.8.4.38 scrollbar_right()	378
31.8.4.39 scrollbar_size() [1/2]	378
31.8.4.40 scrollbar_size() [2/2]	378

31.8.4.41 scrollbar_width() [1/2]	379
31.8.4.42 scrollbar_width() [2/2]	379
31.8.4.43 select()	379
31.8.4.44 select_only()	380
31.8.4.45 selection()	380
31.8.4.46 sort()	380
31.8.4.47 swapping()	380
31.8.4.48 textfont()	381
31.8.5 Member Data Documentation	381
31.8.5.1 hscrollbar	381
31.8.5.2 scrollbar	381
31.9 FI_Button Class Reference	381
31.9.1 Detailed Description	382
31.9.2 Constructor & Destructor Documentation	383
31.9.2.1 FI_Button()	383
31.9.3 Member Function Documentation	383
31.9.3.1 clear()	383
31.9.3.2 down_box() [1/2]	383
31.9.3.3 down_box() [2/2]	384
31.9.3.4 draw()	384
31.9.3.5 handle()	384
31.9.3.6 set()	385
31.9.3.7 shortcut() [1/2]	385
31.9.3.8 shortcut() [2/2]	385
31.9.3.9 value()	385
31.10 FI_Cairo_State Class Reference	386
31.10.1 Detailed Description	386
31.10.2 Member Function Documentation	386
31.10.2.1 cc()	386
31.11 FI_Cairo_Window Class Reference	387
31.11.1 Detailed Description	387
31.11.2 Member Function Documentation	387
31.11.2.1 draw()	388
31.11.2.2 set_draw_cb()	388
31.12 FI_Chart Class Reference	388
31.12.1 Detailed Description	389
31.12.2 Constructor & Destructor Documentation	390
31.12.2.1 FI_Chart()	390
31.12.3 Member Function Documentation	390
31.12.3.1 add()	390
31.12.3.2 autosize() [1/2]	390
31.12.3.3 autosize() [2/2]	390

31.12.3.4 bounds() [1/2]	391
31.12.3.5 bounds() [2/2]	391
31.12.3.6 draw()	391
31.12.3.7 insert()	391
31.12.3.8 maxsize()	392
31.12.3.9 replace()	392
31.13 FL_CHART_ENTRY Struct Reference	392
31.13.1 Detailed Description	393
31.14 FI_Check_Browser Class Reference	393
31.14.1 Detailed Description	394
31.14.2 Member Function Documentation	394
31.14.2.1 add() [1/2]	394
31.14.2.2 add() [2/2]	394
31.14.2.3 handle()	394
31.14.2.4 nchecked()	395
31.14.2.5 nitems()	395
31.14.2.6 remove()	395
31.14.2.7 set_checked()	395
31.15 FI_Check_Button Class Reference	395
31.15.1 Detailed Description	396
31.15.2 Constructor & Destructor Documentation	396
31.15.2.1 FI_Check_Button()	396
31.16 FI_Choice Class Reference	396
31.16.1 Detailed Description	397
31.16.2 Constructor & Destructor Documentation	398
31.16.2.1 FI_Choice()	398
31.16.3 Member Function Documentation	398
31.16.3.1 draw()	398
31.16.3.2 handle()	398
31.16.3.3 value() [1/3]	399
31.16.3.4 value() [2/3]	399
31.16.3.5 value() [3/3]	399
31.17 FI_Clock Class Reference	400
31.17.1 Detailed Description	400
31.17.2 Constructor & Destructor Documentation	401
31.17.2.1 FI_Clock() [1/2]	401
31.17.2.2 FI_Clock() [2/2]	401
31.17.3 Member Function Documentation	401
31.17.3.1 handle()	402
31.18 FI_Clock_Output Class Reference	402
31.18.1 Detailed Description	403
31.18.2 Constructor & Destructor Documentation	403

31.18.2.1 FI_Clock_Output()	403
31.18.3 Member Function Documentation	404
31.18.3.1 draw() [1/2]	404
31.18.3.2 draw() [2/2]	404
31.18.3.3 hour()	404
31.18.3.4 minute()	404
31.18.3.5 second()	405
31.18.3.6 value() [1/3]	405
31.18.3.7 value() [2/3]	405
31.18.3.8 value() [3/3]	405
31.19 FI_Color_Chooser Class Reference	406
31.19.1 Detailed Description	407
31.19.2 Constructor & Destructor Documentation	407
31.19.2.1 FI_Color_Chooser()	407
31.19.3 Member Function Documentation	408
31.19.3.1 b()	408
31.19.3.2 g()	408
31.19.3.3 hsv()	408
31.19.3.4 hsv2rgb()	408
31.19.3.5 hue()	408
31.19.3.6 mode() [1/2]	409
31.19.3.7 mode() [2/2]	409
31.19.3.8 r()	409
31.19.3.9 rgb()	409
31.19.3.10 rgb2hsv()	409
31.19.3.11 saturation()	411
31.19.3.12 value()	411
31.20 FI_Copy_Surface Class Reference	411
31.20.1 Detailed Description	412
31.20.2 Constructor & Destructor Documentation	412
31.20.2.1 FI_Copy_Surface()	412
31.20.3 Member Function Documentation	412
31.20.3.1 class_name()	412
31.20.3.2 draw()	413
31.20.3.3 draw_decorated_window()	413
31.20.3.4 set_current()	413
31.21 FI_Counter Class Reference	413
31.21.1 Detailed Description	414
31.21.2 Constructor & Destructor Documentation	415
31.21.2.1 FI_Counter()	415
31.21.3 Member Function Documentation	415
31.21.3.1 draw()	415

31.21.3.2 handle()	416
31.21.3.3 lstep()	416
31.21.3.4 step() [1/2]	416
31.21.3.5 step() [2/2]	416
31.22 FI_Device Class Reference	417
31.22.1 Detailed Description	417
31.22.2 Constructor & Destructor Documentation	417
31.22.2.1 ~FI_Device()	417
31.22.3 Member Function Documentation	418
31.22.3.1 class_name()	418
31.22.4 Member Data Documentation	418
31.22.4.1 class_id	418
31.23 FI_Device_Plugin Class Reference	418
31.23.1 Detailed Description	419
31.23.2 Member Function Documentation	419
31.23.2.1 print()	419
31.23.2.2 rectangle_capture()	419
31.24 FI_Dial Class Reference	419
31.24.1 Detailed Description	420
31.24.2 Constructor & Destructor Documentation	421
31.24.2.1 FI_Dial()	421
31.24.3 Member Function Documentation	421
31.24.3.1 angle1()	421
31.24.3.2 draw() [1/2]	421
31.24.3.3 draw() [2/2]	421
31.24.3.4 handle() [1/2]	421
31.24.3.5 handle() [2/2]	422
31.25 FI_Display_Device Class Reference	422
31.25.1 Detailed Description	422
31.25.2 Member Function Documentation	423
31.25.2.1 class_name()	423
31.26 FI_Double_Window Class Reference	423
31.26.1 Detailed Description	424
31.26.2 Constructor & Destructor Documentation	424
31.26.2.1 ~FI_Double_Window()	424
31.26.3 Member Function Documentation	424
31.26.3.1 flush() [1/2]	424
31.26.3.2 flush() [2/2]	424
31.26.3.3 hide()	424
31.26.3.4 resize()	425
31.26.3.5 show()	425
31.27 FI_End Class Reference	425

31.27.1 Detailed Description	426
31.28 FI_File_Browser Class Reference	426
31.28.1 Detailed Description	427
31.28.2 Constructor & Destructor Documentation	427
31.28.2.1 FI_File_Browser()	427
31.28.3 Member Function Documentation	427
31.28.3.1 filetype() [1/2]	427
31.28.3.2 filetype() [2/2]	427
31.28.3.3 filter() [1/2]	427
31.28.3.4 filter() [2/2]	427
31.28.3.5 iconsize() [1/2]	428
31.28.3.6 iconsize() [2/2]	428
31.28.3.7 load()	428
31.29 FI_File_Chooser Class Reference	428
31.29.1 Detailed Description	431
31.29.2 Constructor & Destructor Documentation	432
31.29.2.1 FI_File_Chooser()	432
31.29.3 Member Function Documentation	432
31.29.3.1 add_extra()	433
31.29.3.2 filter()	433
31.29.3.3 iconsize() [1/2]	433
31.29.3.4 iconsize() [2/2]	433
31.29.3.5 preview()	433
31.29.3.6 value()	433
31.29.4 Member Data Documentation	433
31.29.4.1 showHiddenButton	434
31.30 FI_File_Icon Class Reference	434
31.30.1 Detailed Description	435
31.30.2 Constructor & Destructor Documentation	435
31.30.2.1 FI_File_Icon()	435
31.30.3 Member Function Documentation	435
31.30.3.1 add()	435
31.30.3.2 add_color()	436
31.30.3.3 add_vertex() [1/2]	436
31.30.3.4 add_vertex() [2/2]	436
31.30.3.5 draw()	436
31.30.3.6 find()	437
31.30.3.7 label()	437
31.30.3.8 labeltype()	437
31.30.3.9 load()	438
31.30.3.10 load_fti()	438
31.30.3.11 load_image()	438

31.30.3.12 load_system_icons()	438
31.30.3.13 next()	438
31.30.3.14 type()	439
31.31 FI_File_Input Class Reference	439
31.31.1 Detailed Description	440
31.31.2 Constructor & Destructor Documentation	440
31.31.2.1 FI_File_Input()	440
31.31.3 Member Function Documentation	440
31.31.3.1 down_box()	440
31.31.3.2 draw()	441
31.31.3.3 errorcolor()	441
31.31.3.4 handle()	441
31.31.3.5 value() [1/2]	441
31.31.3.6 value() [2/2]	441
31.32 FI_Fill_Dial Class Reference	442
31.32.1 Detailed Description	442
31.33 FI_Fill_Slider Class Reference	442
31.33.1 Detailed Description	443
31.34 FI_Float_Input Class Reference	443
31.34.1 Detailed Description	443
31.34.2 Constructor & Destructor Documentation	443
31.34.2.1 FI_Float_Input()	443
31.35 FI_FLTK_File_Chooser Class Reference	444
31.36 FI_Font_Descriptor Class Reference	445
31.36.1 Detailed Description	445
31.37 FI_Fontdesc Struct Reference	445
31.38 FI_FormsBitmap Class Reference	445
31.38.1 Detailed Description	446
31.38.2 Member Function Documentation	446
31.38.2.1 draw()	446
31.38.2.2 set()	446
31.39 FI_FormsPixmap Class Reference	446
31.39.1 Detailed Description	447
31.39.2 Constructor & Destructor Documentation	447
31.39.2.1 FI_FormsPixmap()	447
31.39.3 Member Function Documentation	447
31.39.3.1 draw()	447
31.39.3.2 Pixmap()	448
31.39.3.3 set()	448
31.40 FI_FormsText Class Reference	448
31.40.1 Member Function Documentation	448
31.40.1.1 draw()	449

31.41 FI_Free Class Reference	449
31.41.1 Detailed Description	449
31.41.2 Constructor & Destructor Documentation	450
31.41.2.1 FI_Free()	450
31.41.3 Member Function Documentation	450
31.41.3.1 draw()	450
31.41.3.2 handle()	450
31.42 FI_GDI_Graphics_Driver Class Reference	451
31.42.1 Detailed Description	452
31.42.2 Member Function Documentation	452
31.42.2.1 class_name()	452
31.42.2.2 color() [1/2]	453
31.42.2.3 color() [2/2]	453
31.42.2.4 copy_offscreen()	453
31.42.2.5 descent()	453
31.42.2.6 draw() [1/5]	453
31.42.2.7 draw() [2/5]	453
31.42.2.8 draw() [3/5]	454
31.42.2.9 draw() [4/5]	454
31.42.2.10 draw() [5/5]	454
31.42.2.11 draw_image() [1/2]	454
31.42.2.12 draw_image() [2/2]	455
31.42.2.13 draw_image_mono() [1/2]	455
31.42.2.14 draw_image_mono() [2/2]	455
31.42.2.15 font()	455
31.42.2.16 height()	456
31.42.2.17 rtl_draw()	456
31.42.2.18 text_extents()	456
31.42.2.19 width() [1/2]	456
31.42.2.20 width() [2/2]	456
31.43 FI_GDI_Printer_Graphics_Driver Class Reference	456
31.43.1 Detailed Description	457
31.43.2 Member Function Documentation	457
31.43.2.1 class_name()	457
31.43.2.2 draw() [1/2]	457
31.43.2.3 draw() [2/2]	458
31.43.2.4 draw_scaled()	458
31.44 FI_GIF_Image Class Reference	458
31.44.1 Detailed Description	459
31.44.2 Constructor & Destructor Documentation	459
31.44.2.1 FI_GIF_Image()	459
31.45 FI_GI_Choice Class Reference	459

31.46 FI_Gl_Window Class Reference	459
31.46.1 Detailed Description	461
31.46.2 Constructor & Destructor Documentation	462
31.46.2.1 FI_Gl_Window() [1/2]	462
31.46.2.2 FI_Gl_Window() [2/2]	462
31.46.3 Member Function Documentation	462
31.46.3.1 as_gl_window()	462
31.46.3.2 can_do()	462
31.46.3.3 can_do_overlay()	463
31.46.3.4 context() [1/2]	463
31.46.3.5 context() [2/2]	463
31.46.3.6 context_valid()	463
31.46.3.7 draw()	463
31.46.3.8 flush()	463
31.46.3.9 handle()	464
31.46.3.10 hide()	464
31.46.3.11 make_current()	464
31.46.3.12 make_overlay_current()	464
31.46.3.13 mode() [1/3]	464
31.46.3.14 mode() [2/3]	464
31.46.3.15 mode() [3/3]	464
31.46.3.16 ortho()	465
31.46.3.17 pixel_h()	465
31.46.3.18 pixel_w()	466
31.46.3.19 pixels_per_unit()	466
31.46.3.20 redraw_overlay()	466
31.46.3.21 resize()	466
31.46.3.22 show()	466
31.46.3.23 swap_buffers()	467
31.46.3.24 valid()	467
31.47 FI_Glut_Bitmap_Font Struct Reference	467
31.47.1 Detailed Description	468
31.48 FI_Glut_StrokeChar Struct Reference	468
31.49 FI_Glut_StrokeFont Struct Reference	468
31.50 FI_Glut_StrokeStrip Struct Reference	468
31.51 FI_Glut_StrokeVertex Struct Reference	468
31.52 FI_Glut_Window Class Reference	469
31.52.1 Detailed Description	470
31.52.2 Member Function Documentation	470
31.52.2.1 draw()	470
31.52.2.2 draw_overlay()	470
31.52.2.3 handle()	470

31.53 FI_Graphics_Driver Class Reference	470
31.53.1 Detailed Description	477
31.53.2 Member Function Documentation	477
31.53.2.1 arc() [1/2]	477
31.53.2.2 arc() [2/2]	478
31.53.2.3 begin_complex_polygon()	478
31.53.2.4 begin_line()	478
31.53.2.5 begin_loop()	478
31.53.2.6 begin_points()	478
31.53.2.7 begin_polygon()	478
31.53.2.8 circle()	478
31.53.2.9 class_name()	479
31.53.2.10 clip_box()	479
31.53.2.11 color() [1/2]	479
31.53.2.12 color() [2/2]	479
31.53.2.13 copy_offscreen()	479
31.53.2.14 curve()	480
31.53.2.15 descent()	480
31.53.2.16 draw() [1/5]	480
31.53.2.17 draw() [2/5]	480
31.53.2.18 draw() [3/5]	480
31.53.2.19 draw() [4/5]	481
31.53.2.20 draw() [5/5]	481
31.53.2.21 draw_image() [1/2]	481
31.53.2.22 draw_image() [2/2]	481
31.53.2.23 draw_image_mono() [1/2]	482
31.53.2.24 draw_image_mono() [2/2]	482
31.53.2.25 draw_scaled()	482
31.53.2.26 end_complex_polygon()	482
31.53.2.27 end_line()	483
31.53.2.28 end_loop()	483
31.53.2.29 end_points()	483
31.53.2.30 end_polygon()	483
31.53.2.31 font()	483
31.53.2.32 gap()	483
31.53.2.33 height()	483
31.53.2.34 line() [1/2]	484
31.53.2.35 line() [2/2]	484
31.53.2.36 line_style()	484
31.53.2.37 loop() [1/2]	484
31.53.2.38 loop() [2/2]	484
31.53.2.39 not_clipped()	485

31.53.2.40 pie()	485
31.53.2.41 point()	485
31.53.2.42 polygon() [1/2]	485
31.53.2.43 polygon() [2/2]	485
31.53.2.44 pop_clip()	486
31.53.2.45 push_clip()	486
31.53.2.46 push_no_clip()	486
31.53.2.47 rect()	486
31.53.2.48 rectf()	486
31.53.2.49 rtl_draw()	486
31.53.2.50 text_extents()	487
31.53.2.51 transformed_vertex()	487
31.53.2.52 vertex()	487
31.53.2.53 width() [1/2]	487
31.53.2.54 width() [2/2]	487
31.53.2.55 xyline() [1/3]	488
31.53.2.56 xyline() [2/3]	488
31.53.2.57 xyline() [3/3]	488
31.53.2.58 yxline() [1/3]	488
31.53.2.59 yxline() [2/3]	488
31.53.2.60 yxline() [3/3]	488
31.53.3 Friends And Related Function Documentation	489
31.53.3.1 fl_arc [1/2]	489
31.53.3.2 fl_arc [2/2]	489
31.53.3.3 fl_begin_complex_polygon	490
31.53.3.4 fl_begin_points	490
31.53.3.5 fl_circle	490
31.53.3.6 fl_clip_box	490
31.53.3.7 fl_clip_region	491
31.53.3.8 fl_color [1/2]	491
31.53.3.9 fl_color [2/2]	491
31.53.3.10 fl_copy_offscreen	492
31.53.3.11 fl_curve	492
31.53.3.12 fl_draw	492
31.53.3.13 fl_draw_image [1/2]	493
31.53.3.14 fl_draw_image [2/2]	493
31.53.3.15 fl_draw_image_mono [1/2]	494
31.53.3.16 fl_draw_image_mono [2/2]	494
31.53.3.17 fl_font	495
31.53.3.18 fl_gap	495
31.53.3.19 fl_line_style	495
31.53.3.20 fl_mult_matrix	495

31.53.3.21 fl_not_clipped	496
31.53.3.22 fl_pie	496
31.53.3.23 fl_polygon [1/2]	496
31.53.3.24 fl_polygon [2/2]	497
31.53.3.25 fl_pop_clip	497
31.53.3.26 fl_push_clip	497
31.53.3.27 fl_push_matrix	497
31.53.3.28 fl_rect	497
31.53.3.29 fl_rotate	498
31.53.3.30 fl_scale [1/2]	498
31.53.3.31 fl_scale [2/2]	498
31.53.3.32 fl_transform_dx	498
31.53.3.33 fl_transform_dy	498
31.53.3.34 fl_transform_x	499
31.53.3.35 fl_transform_y	499
31.53.3.36 fl_transformed_vertex	499
31.53.3.37 fl_translate	499
31.53.3.38 fl_vertex	500
31.54 FI_Group Class Reference	500
31.54.1 Detailed Description	503
31.54.2 Constructor & Destructor Documentation	503
31.54.2.1 FI_Group()	503
31.54.2.2 ~FI_Group()	503
31.54.3 Member Function Documentation	503
31.54.3.1 array()	504
31.54.3.2 as_group()	504
31.54.3.3 begin()	504
31.54.3.4 child()	504
31.54.3.5 clear()	504
31.54.3.6 clip_children() [1/2]	504
31.54.3.7 clip_children() [2/2]	505
31.54.3.8 current() [1/2]	505
31.54.3.9 current() [2/2]	505
31.54.3.10 draw()	505
31.54.3.11 draw_child()	505
31.54.3.12 draw_children()	506
31.54.3.13 end()	506
31.54.3.14 find()	506
31.54.3.15 focus()	506
31.54.3.16 handle()	506
31.54.3.17 init_sizes()	507
31.54.3.18 insert() [1/2]	507

31.54.3.19 insert() [2/2]	507
31.54.3.20 remove() [1/3]	507
31.54.3.21 remove() [2/3]	507
31.54.3.22 remove() [3/3]	508
31.54.3.23 resizable()	508
31.54.3.24 resize()	509
31.54.3.25 sizes()	509
31.54.3.26 update_child()	509
31.55 FI_GTK_File_Chooser Class Reference	510
31.56 FI_Help_Block Struct Reference	510
31.57 FI_Help_Dialog Class Reference	510
31.57.1 Detailed Description	511
31.57.2 Member Function Documentation	511
31.57.2.1 load()	512
31.57.2.2 show()	512
31.57.2.3 textsize()	512
31.57.2.4 value() [1/2]	512
31.57.2.5 value() [2/2]	512
31.58 FI_Help_Font_Stack Struct Reference	512
31.59 FI_Help_Font_Style Struct Reference	513
31.59.1 Detailed Description	513
31.60 FI_Help_Link Struct Reference	513
31.60.1 Detailed Description	514
31.61 FI_Help_Target Struct Reference	514
31.61.1 Detailed Description	514
31.62 FI_Help_View Class Reference	514
31.62.1 Detailed Description	516
31.62.2 Constructor & Destructor Documentation	517
31.62.2.1 ~FI_Help_View()	517
31.62.3 Member Function Documentation	517
31.62.3.1 draw()	518
31.62.3.2 find()	518
31.62.3.3 handle()	518
31.62.3.4 leftline()	518
31.62.3.5 link()	518
31.62.3.6 load()	518
31.62.3.7 resize()	519
31.62.3.8 scrollbar_size() [1/2]	519
31.62.3.9 scrollbar_size() [2/2]	519
31.62.3.10 topline() [1/2]	519
31.62.3.11 topline() [2/2]	520
31.62.3.12 value()	520

31.63 FI_Hold_Browser Class Reference	520
31.63.1 Detailed Description	521
31.63.2 Constructor & Destructor Documentation	521
31.63.2.1 FI_Hold_Browser()	521
31.64 FI_Hor_Fill_Slider Class Reference	521
31.65 FI_Hor_Nice_Slider Class Reference	522
31.66 FI_Hor_Slider Class Reference	522
31.66.1 Detailed Description	522
31.67 FI_Hor_Value_Slider Class Reference	523
31.68 FI_Image Class Reference	523
31.68.1 Detailed Description	525
31.68.2 Constructor & Destructor Documentation	525
31.68.2.1 FI_Image()	525
31.68.3 Member Function Documentation	525
31.68.3.1 color_average()	525
31.68.3.2 copy() [1/2]	526
31.68.3.3 copy() [2/2]	526
31.68.3.4 count()	526
31.68.3.5 d()	526
31.68.3.6 data()	526
31.68.3.7 desaturate()	526
31.68.3.8 draw() [1/2]	526
31.68.3.9 draw() [2/2]	526
31.68.3.10 draw_empty()	527
31.68.3.11 fail()	527
31.68.3.12 inactive()	527
31.68.3.13 label() [1/2]	527
31.68.3.14 label() [2/2]	528
31.68.3.15 ld() [1/2]	528
31.68.3.16 ld() [2/2]	528
31.68.3.17 RGB_scaling()	528
31.68.3.18 uncache()	528
31.69 FI_Image_Surface Class Reference	528
31.69.1 Detailed Description	529
31.69.2 Constructor & Destructor Documentation	529
31.69.2.1 FI_Image_Surface()	529
31.69.3 Member Function Documentation	530
31.69.3.1 class_name()	530
31.69.3.2 draw()	530
31.69.3.3 draw_decorated_window()	530
31.69.3.4 highres_image()	531
31.69.3.5 image()	531

31.69.3.6 set_current()	531
31.70 FI_Input Class Reference	531
31.70.1 Detailed Description	532
31.70.2 Constructor & Destructor Documentation	533
31.70.2.1 FI_Input()	533
31.70.3 Member Function Documentation	533
31.70.3.1 draw()	533
31.70.3.2 handle()	534
31.71 FI_Input_ Class Reference	534
31.71.1 Detailed Description	537
31.71.2 Constructor & Destructor Documentation	537
31.71.2.1 FI_Input_()	537
31.71.2.2 ~FI_Input_()	538
31.71.3 Member Function Documentation	538
31.71.3.1 copy()	538
31.71.3.2 copy_cuts()	538
31.71.3.3 cursor_color() [1/2]	538
31.71.3.4 cursor_color() [2/2]	539
31.71.3.5 cut() [1/3]	539
31.71.3.6 cut() [2/3]	539
31.71.3.7 cut() [3/3]	539
31.71.3.8 drawtext()	540
31.71.3.9 handle_mouse()	540
31.71.3.10 handletext()	540
31.71.3.11 index()	540
31.71.3.12 input_type() [1/2]	541
31.71.3.13 input_type() [2/2]	541
31.71.3.14 insert()	541
31.71.3.15 line_end()	541
31.71.3.16 line_start()	542
31.71.3.17 mark() [1/2]	542
31.71.3.18 mark() [2/2]	542
31.71.3.19 maximum_size() [1/2]	543
31.71.3.20 maximum_size() [2/2]	543
31.71.3.21 position() [1/3]	543
31.71.3.22 position() [2/3]	543
31.71.3.23 position() [3/3]	543
31.71.3.24 readonly() [1/2]	544
31.71.3.25 readonly() [2/2]	544
31.71.3.26 replace()	544
31.71.3.27 resize()	545
31.71.3.28 shortcut() [1/2]	545

31.71.3.29 shortcut() [2/2]	546
31.71.3.30 size() [1/2]	546
31.71.3.31 size() [2/2]	546
31.71.3.32 static_value() [1/2]	546
31.71.3.33 static_value() [2/2]	547
31.71.3.34 tab_nav() [1/2]	547
31.71.3.35 tab_nav() [2/2]	547
31.71.3.36 textcolor() [1/2]	549
31.71.3.37 textcolor() [2/2]	549
31.71.3.38 textfont() [1/2]	549
31.71.3.39 textfont() [2/2]	549
31.71.3.40 textsize() [1/2]	550
31.71.3.41 textsize() [2/2]	550
31.71.3.42 undo()	550
31.71.3.43 up_down_position()	550
31.71.3.44 value() [1/3]	551
31.71.3.45 value() [2/3]	551
31.71.3.46 value() [3/3]	551
31.71.3.47 word_end()	552
31.71.3.48 word_start()	552
31.71.3.49 wrap() [1/2]	552
31.71.3.50 wrap() [2/2]	552
31.72 FI_Input_Choice Class Reference	553
31.72.1 Detailed Description	554
31.72.2 Constructor & Destructor Documentation	555
31.72.2.1 FI_Input_Choice()	555
31.72.3 Member Function Documentation	555
31.72.3.1 add()	555
31.72.3.2 input()	555
31.72.3.3 menubutton()	555
31.72.3.4 resize()	555
31.72.3.5 value() [1/2]	556
31.72.3.6 value() [2/2]	556
31.73 FI_Int_Input Class Reference	556
31.73.1 Detailed Description	556
31.73.2 Constructor & Destructor Documentation	557
31.73.2.1 FI_Int_Input()	557
31.74 FI_JPEG_Image Class Reference	557
31.74.1 Detailed Description	557
31.74.2 Constructor & Destructor Documentation	557
31.74.2.1 FI_JPEG_Image() [1/2]	558
31.74.2.2 FI_JPEG_Image() [2/2]	558

31.75 FI_Label Struct Reference	558
31.75.1 Detailed Description	559
31.75.2 Member Function Documentation	559
31.75.2.1 draw()	559
31.75.2.2 measure()	559
31.75.3 Member Data Documentation	559
31.75.3.1 type	560
31.76 FI_Light_Button Class Reference	560
31.76.1 Detailed Description	560
31.76.2 Constructor & Destructor Documentation	561
31.76.2.1 FI_Light_Button()	561
31.76.3 Member Function Documentation	561
31.76.3.1 draw()	561
31.76.3.2 handle()	561
31.77 FI_Line_Dial Class Reference	562
31.78 FI_Mac_App_Menu Class Reference	562
31.78.1 Detailed Description	563
31.78.2 Member Function Documentation	563
31.78.2.1 custom_application_menu_items()	563
31.78.3 Member Data Documentation	563
31.78.3.1 print	563
31.79 FI_Menu_ Class Reference	563
31.79.1 Detailed Description	566
31.79.2 Constructor & Destructor Documentation	566
31.79.2.1 FI_Menu_()	566
31.79.3 Member Function Documentation	566
31.79.3.1 add() [1/2]	566
31.79.3.2 add() [2/2]	567
31.79.3.3 clear()	569
31.79.3.4 clear_submenu()	569
31.79.3.5 copy()	570
31.79.3.6 down_box()	570
31.79.3.7 find_index() [1/3]	570
31.79.3.8 find_index() [2/3]	570
31.79.3.9 find_index() [3/3]	571
31.79.3.10 find_item() [1/2]	571
31.79.3.11 find_item() [2/2]	572
31.79.3.12 global()	572
31.79.3.13 insert()	572
31.79.3.14 item_pathname()	573
31.79.3.15 menu() [1/2]	574
31.79.3.16 menu() [2/2]	574

31.79.3.17 mode() [1/2]	574
31.79.3.18 mode() [2/2]	574
31.79.3.19 mvalue()	574
31.79.3.20 picked()	574
31.79.3.21 remove()	575
31.79.3.22 replace()	575
31.79.3.23 size()	575
31.79.3.24 test_shortcut()	575
31.79.3.25 text() [1/2]	575
31.79.3.26 text() [2/2]	576
31.79.3.27 textcolor()	576
31.79.3.28 textfont() [1/2]	576
31.79.3.29 textfont() [2/2]	576
31.79.3.30 textsize() [1/2]	576
31.79.3.31 textsize() [2/2]	576
31.79.3.32 value() [1/3]	576
31.79.3.33 value() [2/3]	576
31.79.3.34 value() [3/3]	577
31.80 FI_Menu_Bar Class Reference	577
31.80.1 Detailed Description	577
31.80.2 Constructor & Destructor Documentation	578
31.80.2.1 FI_Menu_Bar()	578
31.80.3 Member Function Documentation	578
31.80.3.1 draw()	578
31.80.3.2 handle()	578
31.81 FI_Menu_Button Class Reference	579
31.81.1 Detailed Description	580
31.81.2 Member Enumeration Documentation	580
31.81.2.1 popup_buttons	580
31.81.3 Constructor & Destructor Documentation	581
31.81.3.1 FI_Menu_Button()	581
31.81.4 Member Function Documentation	581
31.81.4.1 draw()	581
31.81.4.2 handle()	581
31.81.4.3 popup()	582
31.82 FI_Menu_Item Struct Reference	582
31.82.1 Detailed Description	585
31.82.2 Member Function Documentation	586
31.82.2.1 add()	586
31.82.2.2 argument() [1/2]	587
31.82.2.3 argument() [2/2]	587
31.82.2.4 callback() [1/5]	587

31.82.2.5 callback() [2/5]	587
31.82.2.6 callback() [3/5]	587
31.82.2.7 callback() [4/5]	587
31.82.2.8 callback() [5/5]	588
31.82.2.9 check()	588
31.82.2.10 checkbox()	588
31.82.2.11 checked()	588
31.82.2.12 deactivate()	588
31.82.2.13 do_callback() [1/3]	588
31.82.2.14 do_callback() [2/3]	588
31.82.2.15 do_callback() [3/3]	589
31.82.2.16 find_shortcut()	589
31.82.2.17 insert()	589
31.82.2.18 label()	590
31.82.2.19 labelcolor() [1/2]	590
31.82.2.20 labelcolor() [2/2]	590
31.82.2.21 labelfont() [1/2]	590
31.82.2.22 labelfont() [2/2]	590
31.82.2.23 labeltype() [1/2]	590
31.82.2.24 labeltype() [2/2]	590
31.82.2.25 measure()	591
31.82.2.26 next() [1/2]	591
31.82.2.27 next() [2/2]	591
31.82.2.28 popup()	591
31.82.2.29 pulldown()	591
31.82.2.30 radio()	592
31.82.2.31 set()	592
31.82.2.32 setonly()	592
31.82.2.33 shortcut()	592
31.82.2.34 size()	592
31.82.2.35 submenu()	592
31.82.2.36 test_shortcut()	593
31.82.2.37 uncheck()	593
31.82.2.38 value()	593
31.83 FI_Menu_Window Class Reference	593
31.83.1 Detailed Description	594
31.83.2 Member Function Documentation	594
31.83.2.1 clear_overlay()	594
31.83.2.2 flush()	594
31.83.2.3 hide()	594
31.83.2.4 set_overlay()	594
31.83.2.5 show()	595

31.84 FI_Multi_Browser Class Reference	595
31.84.1 Detailed Description	595
31.84.2 Constructor & Destructor Documentation	596
31.84.2.1 FI_Multi_Browser()	596
31.85 FI_Multi_Label Struct Reference	596
31.85.1 Detailed Description	596
31.85.2 Member Data Documentation	597
31.85.2.1 labela	597
31.85.2.2 labelb	597
31.85.2.3 typea	597
31.85.2.4 typeb	597
31.86 FI_Multiline_Input Class Reference	597
31.86.1 Detailed Description	598
31.86.2 Constructor & Destructor Documentation	598
31.86.2.1 FI_Multiline_Input()	598
31.87 FI_Multiline_Output Class Reference	598
31.87.1 Detailed Description	599
31.87.2 Constructor & Destructor Documentation	599
31.87.2.1 FI_Multiline_Output()	599
31.88 FI_Native_File_Chooser Class Reference	599
31.88.1 Detailed Description	601
31.88.2 Member Enumeration Documentation	602
31.88.2.1 Option	602
31.88.2.2 Type	602
31.88.3 Constructor & Destructor Documentation	603
31.88.3.1 FI_Native_File_Chooser()	603
31.88.3.2 ~FI_Native_File_Chooser()	603
31.88.4 Member Function Documentation	603
31.88.4.1 count()	603
31.88.4.2 directory()	603
31.88.4.3 errmsg()	603
31.88.4.4 filename() [1/2]	603
31.88.4.5 filename() [2/2]	604
31.88.4.6 filter() [1/2]	604
31.88.4.7 filter() [2/2]	604
31.88.4.8 filter_value() [1/2]	604
31.88.4.9 filter_value() [2/2]	604
31.88.4.10 options()	604
31.88.4.11 preset_file()	605
31.88.4.12 show()	605
31.88.4.13 title() [1/2]	605
31.88.4.14 title() [2/2]	605

31.89 FI_Nice_Slider Class Reference	605
31.90 FI_Output Class Reference	606
31.90.1 Detailed Description	606
31.90.2 Constructor & Destructor Documentation	607
31.90.2.1 FI_Output()	607
31.91 FI_Overlay_Window Class Reference	607
31.91.1 Detailed Description	608
31.91.2 Constructor & Destructor Documentation	608
31.91.2.1 FI_Overlay_Window()	609
31.91.3 Member Function Documentation	609
31.91.3.1 draw_overlay()	609
31.91.3.2 flush()	609
31.91.3.3 hide()	609
31.91.3.4 redraw_overlay()	609
31.91.3.5 resize()	609
31.91.3.6 show()	610
31.92 FI_Pack Class Reference	610
31.92.1 Detailed Description	611
31.92.2 Constructor & Destructor Documentation	611
31.92.2.1 FI_Pack()	611
31.92.3 Member Function Documentation	611
31.92.3.1 draw()	611
31.93 FI_Paged_Device Class Reference	612
31.93.1 Detailed Description	613
31.93.2 Member Enumeration Documentation	614
31.93.2.1 Page_Format	614
31.93.2.2 Page_Layout	614
31.93.3 Member Function Documentation	614
31.93.3.1 class_name()	614
31.93.3.2 end_job()	614
31.93.3.3 end_page()	614
31.93.3.4 margins()	615
31.93.3.5 origin() [1/2]	615
31.93.3.6 origin() [2/2]	615
31.93.3.7 print_widget()	616
31.93.3.8 print_window()	616
31.93.3.9 print_window_part()	616
31.93.3.10 printable_rect()	617
31.93.3.11 rotate()	617
31.93.3.12 scale()	617
31.93.3.13 start_job()	617
31.93.3.14 start_page()	618

31.93.3.15 translate()	618
31.93.3.16 untranslate()	618
31.94 FI_Pixmap Class Reference	618
31.94.1 Detailed Description	620
31.94.2 Constructor & Destructor Documentation	620
31.94.2.1 FI_Pixmap() [1/4]	620
31.94.2.2 FI_Pixmap() [2/4]	620
31.94.2.3 FI_Pixmap() [3/4]	620
31.94.2.4 FI_Pixmap() [4/4]	620
31.94.3 Member Function Documentation	620
31.94.3.1 color_average()	620
31.94.3.2 copy()	621
31.94.3.3 desaturate()	621
31.94.3.4 draw()	621
31.94.3.5 label() [1/2]	621
31.94.3.6 label() [2/2]	621
31.94.3.7 uncache()	621
31.95 FI_Plugin Class Reference	622
31.95.1 Detailed Description	622
31.95.2 Constructor & Destructor Documentation	622
31.95.2.1 FI_Plugin()	622
31.96 FI_Plugin_Manager Class Reference	623
31.96.1 Detailed Description	623
31.96.2 Constructor & Destructor Documentation	624
31.96.2.1 ~FI_Plugin_Manager()	624
31.96.3 Member Function Documentation	624
31.96.3.1 addPlugin()	624
31.96.3.2 load()	624
31.96.3.3 removePlugin()	624
31.97 FI_PNG_Image Class Reference	624
31.97.1 Detailed Description	625
31.97.2 Constructor & Destructor Documentation	625
31.97.2.1 FI_PNG_Image() [1/2]	625
31.97.2.2 FI_PNG_Image() [2/2]	625
31.98 FI_PNM_Image Class Reference	626
31.98.1 Detailed Description	626
31.98.2 Constructor & Destructor Documentation	626
31.98.2.1 FI_PNM_Image()	626
31.99 FI_Positioner Class Reference	626
31.99.1 Detailed Description	628
31.99.2 Constructor & Destructor Documentation	628
31.99.2.1 FI_Positioner()	628

31.99.3 Member Function Documentation	628
31.99.3.1 draw()	628
31.99.3.2 handle()	628
31.100 FI_PostScript_File_Device Class Reference	629
31.100.1 Detailed Description	630
31.100.2 Member Function Documentation	630
31.100.2.1 class_name()	630
31.100.2.2 end_job()	631
31.100.2.3 end_page()	631
31.100.2.4 margins()	631
31.100.2.5 origin() [1/2]	631
31.100.2.6 origin() [2/2]	631
31.100.2.7 printable_rect()	632
31.100.2.8 rotate()	632
31.100.2.9 scale()	632
31.100.2.10 start_job() [1/3]	633
31.100.2.11 start_job() [2/3]	633
31.100.2.12 start_job() [3/3]	633
31.100.2.13 start_page()	634
31.100.2.14 translate()	634
31.100.2.15 untranslate()	634
31.101 FI_PostScript_Graphics_Driver Class Reference	634
31.101.1 Detailed Description	637
31.101.2 Member Function Documentation	638
31.101.2.1 arc() [1/2]	638
31.101.2.2 arc() [2/2]	638
31.101.2.3 begin_complex_polygon()	638
31.101.2.4 begin_line()	638
31.101.2.5 begin_loop()	638
31.101.2.6 begin_points()	639
31.101.2.7 begin_polygon()	639
31.101.2.8 circle()	639
31.101.2.9 class_name()	639
31.101.2.10 clip_box()	639
31.101.2.11 clocale_printf()	639
31.101.2.12 color() [1/2]	640
31.101.2.13 color() [2/2]	640
31.101.2.14 curve()	640
31.101.2.15 descent()	640
31.101.2.16 draw() [1/5]	640
31.101.2.17 draw() [2/5]	640
31.101.2.18 draw() [3/5]	641

31.101.2.19 draw() [4/5]	641
31.101.2.20 draw() [5/5]	641
31.101.2.21 draw_image() [1/2]	641
31.101.2.22 draw_image() [2/2]	642
31.101.2.23 draw_image_mono() [1/2]	642
31.101.2.24 draw_image_mono() [2/2]	642
31.101.2.25 draw_scaled()	642
31.101.2.26 end_complex_polygon()	643
31.101.2.27 end_line()	643
31.101.2.28 end_loop()	643
31.101.2.29 end_points()	643
31.101.2.30 end_polygon()	643
31.101.2.31 font()	643
31.101.2.32 gap()	643
31.101.2.33 height()	644
31.101.2.34 line() [1/2]	644
31.101.2.35 line() [2/2]	644
31.101.2.36 line_style()	644
31.101.2.37 loop() [1/2]	644
31.101.2.38 loop() [2/2]	644
31.101.2.39 not_clipped()	645
31.101.2.40 pie()	645
31.101.2.41 point()	645
31.101.2.42 polygon() [1/2]	645
31.101.2.43 polygon() [2/2]	645
31.101.2.44 pop_clip()	646
31.101.2.45 push_clip()	646
31.101.2.46 push_no_clip()	646
31.101.2.47 rect()	646
31.101.2.48 rectf()	646
31.101.2.49 rtl_draw()	647
31.101.2.50 text_extents()	647
31.101.2.51 transformed_vertex()	647
31.101.2.52 vertex()	647
31.101.2.53 width() [1/2]	647
31.101.2.54 width() [2/2]	647
31.101.2.55 xyline() [1/3]	648
31.101.2.56 xyline() [2/3]	648
31.101.2.57 xyline() [3/3]	648
31.101.2.58 yxline() [1/3]	648
31.101.2.59 yxline() [2/3]	648
31.101.2.60 yxline() [3/3]	648

31.102 FI_PostScript_Printer Class Reference	649
31.102.1 Detailed Description	649
31.102.2 Member Function Documentation	650
31.102.2.1 class_name()	650
31.102.2.2 start_job()	650
31.103 FI_Preferences Class Reference	650
31.103.1 Detailed Description	653
31.103.2 Member Typedef Documentation	653
31.103.2.1 ID	653
31.103.3 Member Enumeration Documentation	653
31.103.3.1 Root	653
31.103.4 Constructor & Destructor Documentation	653
31.103.4.1 FI_Preferences() [1/7]	653
31.103.4.2 FI_Preferences() [2/7]	654
31.103.4.3 FI_Preferences() [3/7]	654
31.103.4.4 FI_Preferences() [4/7]	654
31.103.4.5 FI_Preferences() [5/7]	655
31.103.4.6 FI_Preferences() [6/7]	655
31.103.4.7 FI_Preferences() [7/7]	655
31.103.4.8 ~FI_Preferences()	655
31.103.5 Member Function Documentation	656
31.103.5.1 deleteEntry()	656
31.103.5.2 deleteGroup()	656
31.103.5.3 entries()	656
31.103.5.4 entry()	656
31.103.5.5 entryExists()	657
31.103.5.6 flush()	657
31.103.5.7 get() [1/7]	657
31.103.5.8 get() [2/7]	657
31.103.5.9 get() [3/7]	658
31.103.5.10 get() [4/7]	658
31.103.5.11 get() [5/7]	659
31.103.5.12 get() [6/7]	659
31.103.5.13 get() [7/7]	659
31.103.5.14 getUserdataPath()	660
31.103.5.15 group()	660
31.103.5.16 groupExists()	661
31.103.5.17 groups()	661
31.103.5.18 newUUID()	661
31.103.5.19 set() [1/7]	661
31.103.5.20 set() [2/7]	662
31.103.5.21 set() [3/7]	662

31.103.5.22 set() [4/7]	662
31.103.5.23 set() [5/7]	663
31.103.5.24 set() [6/7]	663
31.103.5.25 set() [7/7]	663
31.103.5.26 size()	664
31.104 FI_Printer Class Reference	664
31.104.1 Detailed Description	666
31.104.2 Member Function Documentation	667
31.104.2.1 class_name()	667
31.104.2.2 end_job()	667
31.104.2.3 end_page()	667
31.104.2.4 margins()	667
31.104.2.5 origin() [1/2]	668
31.104.2.6 origin() [2/2]	668
31.104.2.7 print_widget()	668
31.104.2.8 print_window_part()	669
31.104.2.9 printable_rect()	669
31.104.2.10 rotate()	669
31.104.2.11 scale()	670
31.104.2.12 set_current()	670
31.104.2.13 start_job()	670
31.104.2.14 start_page()	671
31.104.2.15 translate()	671
31.104.2.16 untranslate()	671
31.105 FI_Progress Class Reference	671
31.105.1 Detailed Description	672
31.105.2 Constructor & Destructor Documentation	672
31.105.2.1 FI_Progress()	672
31.105.3 Member Function Documentation	672
31.105.3.1 draw()	672
31.105.3.2 maximum() [1/2]	672
31.105.3.3 maximum() [2/2]	673
31.105.3.4 minimum() [1/2]	673
31.105.3.5 minimum() [2/2]	673
31.105.3.6 value() [1/2]	673
31.105.3.7 value() [2/2]	673
31.106 FI_Quartz_Graphics_Driver Class Reference	673
31.106.1 Detailed Description	675
31.106.2 Member Function Documentation	675
31.106.2.1 class_name()	675
31.106.2.2 color() [1/2]	675
31.106.2.3 color() [2/2]	675

31.106.2.4 descent()	675
31.106.2.5 draw() [1/5]	675
31.106.2.6 draw() [2/5]	676
31.106.2.7 draw() [3/5]	676
31.106.2.8 draw() [4/5]	676
31.106.2.9 draw() [5/5]	676
31.106.2.10 draw_image() [1/2]	677
31.106.2.11 draw_image() [2/2]	677
31.106.2.12 draw_image_mono() [1/2]	677
31.106.2.13 draw_image_mono() [2/2]	677
31.106.2.14 draw_scaled()	677
31.106.2.15 font()	678
31.106.2.16 height()	678
31.106.2.17 rtl_draw()	678
31.106.2.18 text_extents()	678
31.106.2.19 width() [1/2]	678
31.106.2.20 width() [2/2]	679
31.107 FI_Radio_Button Class Reference	679
31.107.1 Constructor & Destructor Documentation	679
31.107.1.1 FI_Radio_Button()	679
31.108 FI_Radio_Light_Button Class Reference	680
31.109 FI_Radio_Round_Button Class Reference	680
31.109.1 Constructor & Destructor Documentation	680
31.109.1.1 FI_Radio_Round_Button()	681
31.110 FI_Scroll::FI_Region_LRTB Struct Reference	681
31.110.1 Detailed Description	681
31.111 FI_Scroll::FI_Region_XYWH Struct Reference	681
31.111.1 Detailed Description	682
31.112 FI_Repeat_Button Class Reference	682
31.112.1 Detailed Description	682
31.112.2 Constructor & Destructor Documentation	682
31.112.2.1 FI_Repeat_Button()	682
31.112.3 Member Function Documentation	682
31.112.3.1 handle()	683
31.113 FI_Return_Button Class Reference	683
31.113.1 Detailed Description	684
31.113.2 Constructor & Destructor Documentation	684
31.113.2.1 FI_Return_Button()	684
31.113.3 Member Function Documentation	684
31.113.3.1 draw()	684
31.113.3.2 handle()	684
31.114 FI_RGB_Image Class Reference	685

31.114.1 Detailed Description	686
31.114.2 Constructor & Destructor Documentation	686
31.114.2.1 FI_RGB_Image() [1/2]	686
31.114.2.2 FI_RGB_Image() [2/2]	687
31.114.3 Member Function Documentation	687
31.114.3.1 color_average()	687
31.114.3.2 copy()	687
31.114.3.3 desaturate()	688
31.114.3.4 draw()	688
31.114.3.5 label() [1/2]	688
31.114.3.6 label() [2/2]	688
31.114.3.7 max_size() [1/2]	688
31.114.3.8 max_size() [2/2]	688
31.114.3.9 uncache()	689
31.115 FI_Roller Class Reference	689
31.115.1 Detailed Description	689
31.115.2 Constructor & Destructor Documentation	690
31.115.2.1 FI_Roller()	690
31.115.3 Member Function Documentation	690
31.115.3.1 draw()	690
31.115.3.2 handle()	690
31.116 FI_Round_Button Class Reference	691
31.116.1 Detailed Description	691
31.116.2 Constructor & Destructor Documentation	691
31.116.2.1 FI_Round_Button()	691
31.117 FI_Round_Clock Class Reference	692
31.117.1 Detailed Description	692
31.118 FI_Scroll Class Reference	692
31.118.1 Detailed Description	694
31.118.2 Constructor & Destructor Documentation	695
31.118.2.1 FI_Scroll()	695
31.118.3 Member Function Documentation	695
31.118.3.1 bbox()	695
31.118.3.2 draw()	695
31.118.3.3 handle()	695
31.118.3.4 recalc_scrollbars()	696
31.118.3.5 resize()	696
31.118.3.6 scroll_to()	697
31.118.3.7 scrollbar_size() [1/2]	697
31.118.3.8 scrollbar_size() [2/2]	697
31.118.3.9 xposition()	698
31.118.3.10 yposition()	698

31.119 FI_Scrollbar Class Reference	698
31.119.1 Detailed Description	699
31.119.2 Constructor & Destructor Documentation	699
31.119.2.1 FI_Scrollbar()	699
31.119.3 Member Function Documentation	699
31.119.3.1 draw()	699
31.119.3.2 handle()	700
31.119.3.3 linesize()	700
31.119.3.4 value() [1/3]	700
31.119.3.5 value() [2/3]	700
31.119.3.6 value() [3/3]	701
31.120 FI_Scroll::FI_Scrollbar_Data Struct Reference	701
31.120.1 Detailed Description	701
31.121 FI_Secret_Input Class Reference	702
31.121.1 Detailed Description	702
31.121.2 Constructor & Destructor Documentation	702
31.121.2.1 FI_Secret_Input()	702
31.121.3 Member Function Documentation	702
31.121.3.1 handle()	702
31.122 FI_Select_Browser Class Reference	703
31.122.1 Detailed Description	703
31.122.2 Constructor & Destructor Documentation	704
31.122.2.1 FI_Select_Browser()	704
31.123 FI_Shared_Image Class Reference	704
31.123.1 Detailed Description	706
31.123.2 Constructor & Destructor Documentation	706
31.123.2.1 FI_Shared_Image() [1/2]	706
31.123.2.2 FI_Shared_Image() [2/2]	706
31.123.2.3 ~FI_Shared_Image()	706
31.123.3 Member Function Documentation	706
31.123.3.1 add()	706
31.123.3.2 color_average()	707
31.123.3.3 compare()	707
31.123.3.4 copy()	707
31.123.3.5 desaturate()	708
31.123.3.6 draw()	708
31.123.3.7 find()	708
31.123.3.8 get() [1/2]	708
31.123.3.9 get() [2/2]	709
31.123.3.10 original()	709
31.123.3.11 refcount()	710
31.123.3.12 release()	710

31.123.3.13 scale()	710
31.123.3.14 scaling_algorithm()	710
31.123.3.15 uncache()	711
31.124 FI_Simple_Counter Class Reference	711
31.124.1 Detailed Description	711
31.125 FI_Single_Window Class Reference	711
31.125.1 Detailed Description	712
31.125.2 Member Function Documentation	712
31.125.2.1 flush()	712
31.125.2.2 show()	712
31.126 FI_Slider Class Reference	713
31.126.1 Detailed Description	714
31.126.2 Constructor & Destructor Documentation	714
31.126.2.1 FI_Slider()	714
31.126.3 Member Function Documentation	714
31.126.3.1 bounds()	715
31.126.3.2 draw()	715
31.126.3.3 handle()	715
31.126.3.4 scrollvalue()	715
31.126.3.5 slider_size()	716
31.127 FI_Spinner Class Reference	716
31.127.1 Detailed Description	718
31.127.2 Constructor & Destructor Documentation	718
31.127.2.1 FI_Spinner()	718
31.127.3 Member Function Documentation	718
31.127.3.1 handle()	718
31.127.3.2 maximum()	719
31.127.3.3 minimum()	719
31.127.3.4 resize()	719
31.127.3.5 step()	719
31.127.3.6 type() [1/2]	719
31.127.3.7 type() [2/2]	720
31.127.3.8 value()	720
31.128 FI_Surface_Device Class Reference	720
31.128.1 Detailed Description	721
31.128.2 Member Function Documentation	721
31.128.2.1 class_name()	721
31.128.2.2 set_current()	721
31.128.2.3 surface()	722
31.129 FI_Sys_Menu_Bar Class Reference	722
31.129.1 Detailed Description	723
31.129.2 Constructor & Destructor Documentation	723

31.129.2.1 FI_Sys_Menu_Bar()	723
31.129.3 Member Function Documentation	723
31.129.3.1 add() [1/3]	724
31.129.3.2 add() [2/3]	724
31.129.3.3 add() [3/3]	724
31.129.3.4 clear()	725
31.129.3.5 clear_submenu()	725
31.129.3.6 draw()	725
31.129.3.7 insert() [1/2]	725
31.129.3.8 insert() [2/2]	725
31.129.3.9 menu()	726
31.129.3.10 mode()	726
31.129.3.11 remove()	726
31.129.3.12 replace()	726
31.130 FI_System_Printer Class Reference	727
31.130.1 Detailed Description	728
31.130.2 Member Function Documentation	728
31.130.2.1 class_name()	728
31.130.2.2 end_job()	728
31.130.2.3 end_page()	728
31.130.2.4 margins()	729
31.130.2.5 origin() [1/2]	729
31.130.2.6 origin() [2/2]	729
31.130.2.7 printable_rect()	729
31.130.2.8 rotate()	730
31.130.2.9 scale()	730
31.130.2.10 start_job()	730
31.130.2.11 start_page()	731
31.130.2.12 translate()	731
31.130.2.13 untranslate()	731
31.131 FI_Table Class Reference	731
31.131.1 Detailed Description	736
31.131.2 Member Enumeration Documentation	738
31.131.2.1 TableContext	738
31.131.3 Constructor & Destructor Documentation	738
31.131.3.1 FI_Table()	739
31.131.3.2 ~FI_Table()	739
31.131.4 Member Function Documentation	739
31.131.4.1 callback()	739
31.131.4.2 callback_col()	740
31.131.4.3 callback_context()	740
31.131.4.4 callback_row()	740

31.131.4.5	child()	740
31.131.4.6	children()	740
31.131.4.7	clear()	740
31.131.4.8	col_header()	741
31.131.4.9	col_resize()	741
31.131.4.10	col_resize_min()	741
31.131.4.11	col_width()	741
31.131.4.12	col_width_all()	741
31.131.4.13	draw()	741
31.131.4.14	draw_cell()	742
31.131.4.15	get_selection()	743
31.131.4.16	handle()	743
31.131.4.17	is_interactive_resize()	744
31.131.4.18	is_selected()	744
31.131.4.19	resize()	744
31.131.4.20	row_header()	744
31.131.4.21	row_height()	744
31.131.4.22	row_height_all()	745
31.131.4.23	row_resize()	745
31.131.4.24	row_resize_min()	745
31.131.4.25	rows()	745
31.131.4.26	scrollbar_size() [1/2]	745
31.131.4.27	scrollbar_size() [2/2]	745
31.131.4.28	set_selection()	746
31.131.4.29	tab_cell_nav() [1/2]	746
31.131.4.30	tab_cell_nav() [2/2]	746
31.131.4.31	table_box()	747
31.131.4.32	top_row() [1/2]	747
31.131.4.33	top_row() [2/2]	747
31.131.4.34	visible_cells()	747
31.131.4.35	when()	747
31.132	FI_Table_Row Class Reference	748
31.132.1	Detailed Description	749
31.132.2	Constructor & Destructor Documentation	749
31.132.2.1	FI_Table_Row()	749
31.132.2.2	~FI_Table_Row()	749
31.132.3	Member Function Documentation	749
31.132.3.1	clear()	749
31.132.3.2	handle()	750
31.132.3.3	row_selected()	750
31.132.3.4	rows()	750
31.132.3.5	select_all_rows()	750

31.132.3.6	select_row()	750
31.132.3.7	type()	751
31.133	FI_Tabs Class Reference	751
31.133.1	Detailed Description	752
31.133.2	Constructor & Destructor Documentation	754
31.133.2.1	FI_Tabs()	754
31.133.3	Member Function Documentation	755
31.133.3.1	client_area()	755
31.133.3.2	draw()	755
31.133.3.3	handle()	755
31.133.3.4	push() [1/2]	756
31.133.3.5	push() [2/2]	756
31.133.3.6	value() [1/2]	756
31.133.3.7	value() [2/2]	757
31.133.3.8	which()	757
31.134	FI_Text_Buffer Class Reference	757
31.134.1	Detailed Description	762
31.134.2	Constructor & Destructor Documentation	762
31.134.2.1	FI_Text_Buffer()	762
31.134.3	Member Function Documentation	762
31.134.3.1	add_modify_callback()	762
31.134.3.2	address() [1/2]	762
31.134.3.3	address() [2/2]	763
31.134.3.4	append()	763
31.134.3.5	appendfile()	763
31.134.3.6	byte_at()	763
31.134.3.7	char_at()	763
31.134.3.8	copy()	764
31.134.3.9	count_displayed_characters()	764
31.134.3.10	count_lines()	764
31.134.3.11	findchar_backward()	764
31.134.3.12	findchar_forward()	765
31.134.3.13	highlight()	765
31.134.3.14	highlight_text()	765
31.134.3.15	insert()	765
31.134.3.16	insert_()	766
31.134.3.17	insertfile()	766
31.134.3.18	length()	766
31.134.3.19	line_end()	766
31.134.3.20	line_start()	767
31.134.3.21	line_text()	767
31.134.3.22	loadfile()	767

31.134.3.23	next_char()	767
31.134.3.24	outputfile()	768
31.134.3.25	prev_char()	768
31.134.3.26	remove()	768
31.134.3.27	remove_()	768
31.134.3.28	replace()	769
31.134.3.29	rewind_lines()	769
31.134.3.30	savefile()	769
31.134.3.31	search_backward()	769
31.134.3.32	search_forward()	770
31.134.3.33	secondary_selection_text()	770
31.134.3.34	selection_text()	770
31.134.3.35	skip_displayed_characters()	770
31.134.3.36	tab_distance()	771
31.134.3.37	text() [1/2]	771
31.134.3.38	text() [2/2]	771
31.134.3.39	text_range()	771
31.134.3.40	word_end()	772
31.134.3.41	word_start()	772
31.134.4	Member Data Documentation	772
31.134.4.1	file_encoding_warning_message	772
31.134.4.2	mTabDist	772
31.134.4.3	transcoding_warning_action	772
31.135	FI_Text_Display Class Reference	773
31.135.1	Detailed Description	779
31.135.2	Member Enumeration Documentation	780
31.135.2.1	anonymous enum	781
31.135.2.2	anonymous enum	781
31.135.3	Constructor & Destructor Documentation	781
31.135.3.1	FI_Text_Display()	781
31.135.3.2	~FI_Text_Display()	781
31.135.4	Member Function Documentation	782
31.135.4.1	absolute_top_line_number()	782
31.135.4.2	buffer() [1/3]	782
31.135.4.3	buffer() [2/3]	782
31.135.4.4	buffer() [3/3]	782
31.135.4.5	buffer_modified_cb()	782
31.135.4.6	buffer_predelete_cb()	783
31.135.4.7	calc_last_char()	783
31.135.4.8	calc_line_starts()	783
31.135.4.9	clear_rect()	784
31.135.4.10	col_to_x()	784

31.135.4.11	<code>count_lines()</code>	784
31.135.4.12	<code>cursor_color()</code> [1/2]	784
31.135.4.13	<code>cursor_color()</code> [2/2]	785
31.135.4.14	<code>cursor_style()</code>	785
31.135.4.15	<code>display_insert()</code>	785
31.135.4.16	<code>draw()</code>	785
31.135.4.17	<code>draw_cursor()</code>	786
31.135.4.18	<code>draw_line_numbers()</code>	786
31.135.4.19	<code>draw_range()</code>	786
31.135.4.20	<code>draw_string()</code>	786
31.135.4.21	<code>draw_text()</code>	787
31.135.4.22	<code>draw_vline()</code>	787
31.135.4.23	<code>empty_vlines()</code>	787
31.135.4.24	<code>extend_range_for_styles()</code>	788
31.135.4.25	<code>find_line_end()</code>	788
31.135.4.26	<code>find_wrap_range()</code>	788
31.135.4.27	<code>find_x()</code>	789
31.135.4.28	<code>get_absolute_top_line_number()</code>	789
31.135.4.29	<code>handle()</code>	789
31.135.4.30	<code>handle_vline()</code>	790
31.135.4.31	<code>highlight_data()</code>	790
31.135.4.32	<code>in_selection()</code>	791
31.135.4.33	<code>insert()</code>	791
31.135.4.34	<code>insert_position()</code> [1/2]	792
31.135.4.35	<code>insert_position()</code> [2/2]	792
31.135.4.36	<code>line_end()</code>	792
31.135.4.37	<code>line_start()</code>	792
31.135.4.38	<code>linenumber_align()</code>	793
31.135.4.39	<code>linenumber_bgcolor()</code>	793
31.135.4.40	<code>linenumber_fgcolor()</code>	793
31.135.4.41	<code>linenumber_font()</code>	793
31.135.4.42	<code>linenumber_format()</code>	794
31.135.4.43	<code>linenumber_size()</code>	794
31.135.4.44	<code>linenumber_width()</code>	794
31.135.4.45	<code>longest_vline()</code>	794
31.135.4.46	<code>maintain_absolute_top_line_number()</code>	794
31.135.4.47	<code>maintaining_absolute_top_line_number()</code>	795
31.135.4.48	<code>measure_deleted_lines()</code>	795
31.135.4.49	<code>measure_proportional_character()</code>	795
31.135.4.50	<code>measure_vline()</code>	796
31.135.4.51	<code>move_down()</code>	796
31.135.4.52	<code>move_left()</code>	796

31.135.4.53	<code>move_right()</code>	796
31.135.4.54	<code>move_up()</code>	796
31.135.4.55	<code>offset_line_starts()</code>	797
31.135.4.56	<code>overstrike()</code>	797
31.135.4.57	<code>position_style()</code>	797
31.135.4.58	<code>position_to_line()</code>	798
31.135.4.59	<code>position_to_linecol()</code>	798
31.135.4.60	<code>position_to_xy()</code>	798
31.135.4.61	<code>redisplay_range()</code>	799
31.135.4.62	<code>reset_absolute_top_line_number()</code>	799
31.135.4.63	<code>resize()</code>	799
31.135.4.64	<code>rewind_lines()</code>	799
31.135.4.65	<code>scroll()</code>	800
31.135.4.66	<code>scroll_()</code>	800
31.135.4.67	<code>scroll_timer_cb()</code>	800
31.135.4.68	<code>scrollbar_align()</code> [1/2]	800
31.135.4.69	<code>scrollbar_align()</code> [2/2]	801
31.135.4.70	<code>scrollbar_width()</code> [1/2]	801
31.135.4.71	<code>scrollbar_width()</code> [2/2]	801
31.135.4.72	<code>shortcut()</code> [1/2]	801
31.135.4.73	<code>shortcut()</code> [2/2]	801
31.135.4.74	<code>show_cursor()</code>	802
31.135.4.75	<code>show_insert_position()</code>	802
31.135.4.76	<code>skip_lines()</code>	802
31.135.4.77	<code>string_width()</code>	802
31.135.4.78	<code>textcolor()</code> [1/2]	803
31.135.4.79	<code>textcolor()</code> [2/2]	803
31.135.4.80	<code>textfont()</code> [1/2]	803
31.135.4.81	<code>textfont()</code> [2/2]	803
31.135.4.82	<code>textsize()</code> [1/2]	804
31.135.4.83	<code>textsize()</code> [2/2]	804
31.135.4.84	<code>update_h_scrollbar()</code>	804
31.135.4.85	<code>update_line_starts()</code>	804
31.135.4.86	<code>update_v_scrollbar()</code>	804
31.135.4.87	<code>vline_length()</code>	805
31.135.4.88	<code>word_end()</code>	805
31.135.4.89	<code>word_start()</code>	805
31.135.4.90	<code>wrap_mode()</code>	805
31.135.4.91	<code>wrap_uses_character()</code>	806
31.135.4.92	<code>wrapped_column()</code>	806
31.135.4.93	<code>wrapped_line_counter()</code>	807
31.135.4.94	<code>wrapped_row()</code>	807

31.135.4.95 x_to_col()	808
31.135.4.96 xy_to_position()	808
31.135.4.97 xy_to_rowcol()	808
31.136 FI_Text_Editor Class Reference	809
31.136.1 Detailed Description	811
31.136.2 Member Function Documentation	811
31.136.2.1 add_key_binding()	812
31.136.2.2 handle()	812
31.136.2.3 insert_mode() [1/2]	812
31.136.2.4 insert_mode() [2/2]	812
31.136.2.5 kf_backspace()	812
31.136.2.6 kf_c_s_move()	812
31.136.2.7 kf_copy()	812
31.136.2.8 kf_ctrl_move()	813
31.136.2.9 kf_cut()	813
31.136.2.10 kf_default()	813
31.136.2.11 kf_delete()	813
31.136.2.12 kf_down()	813
31.136.2.13 kf_end()	813
31.136.2.14 kf_enter()	813
31.136.2.15 kf_home()	814
31.136.2.16 kf_ignore()	814
31.136.2.17 kf_insert()	814
31.136.2.18 kf_left()	814
31.136.2.19 kf_m_s_move()	814
31.136.2.20 kf_meta_move()	814
31.136.2.21 kf_move()	815
31.136.2.22 kf_page_down()	815
31.136.2.23 kf_page_up()	815
31.136.2.24 kf_paste()	815
31.136.2.25 kf_right()	815
31.136.2.26 kf_select_all()	815
31.136.2.27 kf_shift_move()	815
31.136.2.28 kf_undo()	816
31.136.2.29 kf_up()	816
31.136.2.30 remove_key_binding()	816
31.136.2.31 tab_nav() [1/2]	816
31.136.2.32 tab_nav() [2/2]	817
31.136.3 Member Data Documentation	817
31.136.3.1 global_key_bindings	817
31.137 FI_Text_Selection Class Reference	817
31.137.1 Detailed Description	818

31.137.2 Member Function Documentation	818
31.137.2.1 end()	818
31.137.2.2 position()	818
31.137.2.3 selected() [1/2]	819
31.137.2.4 selected() [2/2]	819
31.137.2.5 set()	819
31.137.2.6 start()	819
31.137.2.7 update()	819
31.138 FI_Tile Class Reference	821
31.138.1 Detailed Description	821
31.138.2 Constructor & Destructor Documentation	822
31.138.2.1 FI_Tile()	822
31.138.3 Member Function Documentation	823
31.138.3.1 handle()	823
31.138.3.2 position()	823
31.138.3.3 resize()	823
31.139 FI_Tiled_Image Class Reference	824
31.139.1 Detailed Description	824
31.139.2 Constructor & Destructor Documentation	824
31.139.2.1 FI_Tiled_Image()	824
31.139.3 Member Function Documentation	825
31.139.3.1 color_average()	825
31.139.3.2 copy()	825
31.139.3.3 desaturate()	825
31.139.3.4 draw()	826
31.140 FI_Timer Class Reference	826
31.140.1 Detailed Description	827
31.140.2 Constructor & Destructor Documentation	827
31.140.2.1 FI_Timer()	827
31.140.3 Member Function Documentation	827
31.140.3.1 direction() [1/2]	827
31.140.3.2 direction() [2/2]	827
31.140.3.3 draw()	828
31.140.3.4 handle()	828
31.140.3.5 suspended()	828
31.141 FI_Toggle_Button Class Reference	828
31.141.1 Detailed Description	829
31.141.2 Constructor & Destructor Documentation	829
31.141.2.1 FI_Toggle_Button()	829
31.142 FI_Tooltip Class Reference	829
31.142.1 Detailed Description	831
31.142.2 Member Function Documentation	831

31.142.2.1 color() [1/2]	831
31.142.2.2 color() [2/2]	831
31.142.2.3 current()	831
31.142.2.4 delay() [1/2]	831
31.142.2.5 delay() [2/2]	832
31.142.2.6 disable()	832
31.142.2.7 enable()	832
31.142.2.8 enabled()	832
31.142.2.9 enter_area()	832
31.142.2.10 font() [1/2]	832
31.142.2.11 font() [2/2]	832
31.142.2.12 hoverdelay() [1/2]	833
31.142.2.13 hoverdelay() [2/2]	833
31.142.2.14 margin_height() [1/2]	833
31.142.2.15 margin_height() [2/2]	833
31.142.2.16 margin_width() [1/2]	833
31.142.2.17 margin_width() [2/2]	833
31.142.2.18 size() [1/2]	833
31.142.2.19 size() [2/2]	833
31.142.2.20 textcolor() [1/2]	833
31.142.2.21 textcolor() [2/2]	834
31.142.2.22 wrap_width() [1/2]	834
31.142.2.23 wrap_width() [2/2]	834
31.143 FI_Tree Class Reference	834
31.143.1 Detailed Description	840
31.143.2 Member Function Documentation	844
31.143.2.1 add() [1/2]	844
31.143.2.2 add() [2/2]	845
31.143.2.3 calc_dimensions()	845
31.143.2.4 calc_tree()	845
31.143.2.5 callback_item() [1/2]	846
31.143.2.6 callback_item() [2/2]	846
31.143.2.7 callback_reason() [1/2]	846
31.143.2.8 callback_reason() [2/2]	846
31.143.2.9 clear()	846
31.143.2.10 clear_children()	846
31.143.2.11 close() [1/2]	846
31.143.2.12 close() [2/2]	847
31.143.2.13 closeicon() [1/2]	848
31.143.2.14 closeicon() [2/2]	848
31.143.2.15 connectorstyle()	848
31.143.2.16 deselect() [1/2]	848

31.143.2.17 <code>deselect()</code> [2/2]	848
31.143.2.18 <code>deselect_all()</code>	849
31.143.2.19 <code>display()</code>	849
31.143.2.20 <code>displayed()</code>	850
31.143.2.21 <code>draw()</code>	850
31.143.2.22 <code>extend_selection()</code>	850
31.143.2.23 <code>extend_selection_dir()</code>	851
31.143.2.24 <code>find_clicked()</code>	851
31.143.2.25 <code>find_item()</code>	852
31.143.2.26 <code>first()</code>	852
31.143.2.27 <code>first_selected_item()</code>	852
31.143.2.28 <code>first_visible()</code>	853
31.143.2.29 <code>first_visible_item()</code>	853
31.143.2.30 <code>get_selected_items()</code>	853
31.143.2.31 <code>handle()</code>	854
31.143.2.32 <code>hposition()</code> [1/2]	854
31.143.2.33 <code>hposition()</code> [2/2]	854
31.143.2.34 <code>insert()</code>	854
31.143.2.35 <code>insert_above()</code>	855
31.143.2.36 <code>is_close()</code> [1/2]	855
31.143.2.37 <code>is_close()</code> [2/2]	857
31.143.2.38 <code>is_hscroll_visible()</code>	857
31.143.2.39 <code>is_open()</code> [1/2]	857
31.143.2.40 <code>is_open()</code> [2/2]	858
31.143.2.41 <code>is_scrollbar()</code>	858
31.143.2.42 <code>is_selected()</code> [1/2]	858
31.143.2.43 <code>is_selected()</code> [2/2]	859
31.143.2.44 <code>is_vscroll_visible()</code>	859
31.143.2.45 <code>item_clicked()</code> [1/2]	859
31.143.2.46 <code>item_clicked()</code> [2/2]	859
31.143.2.47 <code>item_draw_mode()</code> [1/3]	860
31.143.2.48 <code>item_draw_mode()</code> [2/3]	860
31.143.2.49 <code>item_draw_mode()</code> [3/3]	860
31.143.2.50 <code>item_labelbgcolor()</code> [1/2]	860
31.143.2.51 <code>item_labelbgcolor()</code> [2/2]	860
31.143.2.52 <code>item_labelfgcolor()</code>	860
31.143.2.53 <code>item_labelfont()</code>	861
31.143.2.54 <code>item_labelsize()</code>	861
31.143.2.55 <code>item_pathname()</code>	861
31.143.2.56 <code>item_reselect_mode()</code> [1/2]	861
31.143.2.57 <code>item_reselect_mode()</code> [2/2]	861
31.143.2.58 <code>last()</code>	862

31.143.2.59 last_selected_item()	862
31.143.2.60 last_visible()	862
31.143.2.61 last_visible_item()	862
31.143.2.62 load()	863
31.143.2.63 next()	863
31.143.2.64 next_item()	863
31.143.2.65 next_selected_item()	864
31.143.2.66 next_visible_item()	865
31.143.2.67 open() [1/2]	865
31.143.2.68 open() [2/2]	866
31.143.2.69 open_toggle()	866
31.143.2.70 openicon() [1/2]	867
31.143.2.71 openicon() [2/2]	867
31.143.2.72 prev()	867
31.143.2.73 recalc_tree()	868
31.143.2.74 remove()	868
31.143.2.75 resize()	868
31.143.2.76 root()	868
31.143.2.77 root_label()	869
31.143.2.78 scrollbar_size() [1/2]	869
31.143.2.79 scrollbar_size() [2/2]	869
31.143.2.80 select() [1/2]	869
31.143.2.81 select() [2/2]	870
31.143.2.82 select_all()	870
31.143.2.83 select_only()	871
31.143.2.84 select_toggle()	871
31.143.2.85 selectbox() [1/2]	872
31.143.2.86 selectbox() [2/2]	872
31.143.2.87 selectmode() [1/2]	872
31.143.2.88 selectmode() [2/2]	872
31.143.2.89 set_item_focus()	872
31.143.2.90 show_item() [1/2]	872
31.143.2.91 show_item() [2/2]	873
31.143.2.92 show_item_bottom()	873
31.143.2.93 show_item_middle()	873
31.143.2.94 show_item_top()	873
31.143.2.95 show_self()	874
31.143.2.96 showcollapse() [1/2]	874
31.143.2.97 showcollapse() [2/2]	874
31.143.2.98 showroot()	874
31.143.2.99 sortorder()	875
31.143.2.100 usericon() [1/2]	875

31.143.2.101 usericon() [2/2]	875
31.143.2.102 vposition() [1/2]	875
31.143.2.103 vposition() [2/2]	875
31.144 FI_Tree_Item Class Reference	876
31.144.1 Detailed Description	880
31.144.2 Constructor & Destructor Documentation	881
31.144.2.1 FI_Tree_Item() [1/2]	881
31.144.2.2 FI_Tree_Item() [2/2]	881
31.144.3 Member Function Documentation	881
31.144.3.1 activate()	881
31.144.3.2 add() [1/4]	881
31.144.3.3 add() [2/4]	882
31.144.3.4 add() [3/4]	882
31.144.3.5 add() [4/4]	882
31.144.3.6 calc_item_height()	883
31.144.3.7 child()	883
31.144.3.8 deactivate()	883
31.144.3.9 deparent()	883
31.144.3.10 depth()	883
31.144.3.11 deselect_all()	883
31.144.3.12 draw()	884
31.144.3.13 draw_horizontal_connector()	884
31.144.3.14 draw_item_content()	884
31.144.3.15 draw_vertical_connector()	885
31.144.3.16 drawbgcolor()	886
31.144.3.17 drawfgcolor()	886
31.144.3.18 find_child() [1/2]	886
31.144.3.19 find_child() [2/2]	886
31.144.3.20 find_child_item() [1/2]	887
31.144.3.21 find_child_item() [2/2]	887
31.144.3.22 find_clicked()	887
31.144.3.23 find_item()	887
31.144.3.24 hide_widgets()	888
31.144.3.25 insert()	888
31.144.3.26 insert_above()	888
31.144.3.27 label()	888
31.144.3.28 label_h()	888
31.144.3.29 label_w()	889
31.144.3.30 label_x()	889
31.144.3.31 label_y()	889
31.144.3.32 labelbgcolor() [1/2]	889
31.144.3.33 labelbgcolor() [2/2]	889

31.144.3.34	move() [1/2]	889
31.144.3.35	move() [2/2]	890
31.144.3.36	move_above()	890
31.144.3.37	move_below()	890
31.144.3.38	move_into()	891
31.144.3.39	next()	891
31.144.3.40	next_displayed()	891
31.144.3.41	next_sibling()	891
31.144.3.42	next_visible()	891
31.144.3.43	parent()	892
31.144.3.44	prefs()	892
31.144.3.45	prev()	892
31.144.3.46	prev_displayed()	892
31.144.3.47	prev_sibling()	892
31.144.3.48	prev_visible()	892
31.144.3.49	recalc_tree()	893
31.144.3.50	remove_child() [1/2]	893
31.144.3.51	remove_child() [2/2]	893
31.144.3.52	reparent()	893
31.144.3.53	replace()	894
31.144.3.54	replace_child()	894
31.144.3.55	select()	894
31.144.3.56	select_all()	895
31.144.3.57	show_self()	895
31.144.3.58	show_widgets()	895
31.144.3.59	swap_children() [1/2]	895
31.144.3.60	swap_children() [2/2]	895
31.144.3.61	tree() [1/2]	895
31.144.3.62	tree() [2/2]	896
31.144.3.63	update_prev_next()	896
31.144.3.64	userdeicon() [1/2]	896
31.144.3.65	userdeicon() [2/2]	896
31.144.3.66	usericon()	897
31.144.3.67	visible_r()	897
31.145	FI_Tree_Item_Array Class Reference	897
31.145.1	Detailed Description	898
31.145.2	Constructor & Destructor Documentation	898
31.145.2.1	FI_Tree_Item_Array()	898
31.145.3	Member Function Documentation	898
31.145.3.1	add()	898
31.145.3.2	clear()	898
31.145.3.3	deparent()	899

31.145.3.4	insert()	899
31.145.3.5	manage_item_destroy()	899
31.145.3.6	move()	899
31.145.3.7	remove() [1/2]	899
31.145.3.8	remove() [2/2]	899
31.145.3.9	reparent()	900
31.145.3.10	replace()	900
31.146	FI_Tree_Prefs Class Reference	900
31.146.1	Detailed Description	903
31.146.2	Member Function Documentation	903
31.146.2.1	closedeicon()	903
31.146.2.2	closeicon()	903
31.146.2.3	item_draw_mode()	903
31.146.2.4	item_labelbgcolor() [1/2]	903
31.146.2.5	item_labelbgcolor() [2/2]	904
31.146.2.6	marginbottom()	904
31.146.2.7	opendeicon()	904
31.146.2.8	openicon() [1/2]	904
31.146.2.9	openicon() [2/2]	904
31.146.2.10	selectmode()	904
31.146.2.11	showcollapse()	904
31.146.2.12	showroot()	905
31.146.2.13	sortorder()	905
31.146.2.14	userdeicon()	905
31.147	FI_Valuator Class Reference	905
31.147.1	Detailed Description	907
31.147.2	Constructor & Destructor Documentation	908
31.147.2.1	FI_Valuator()	908
31.147.3	Member Function Documentation	908
31.147.3.1	format()	908
31.147.3.2	increment()	908
31.147.3.3	maximum() [1/2]	908
31.147.3.4	maximum() [2/2]	908
31.147.3.5	minimum() [1/2]	908
31.147.3.6	minimum() [2/2]	909
31.147.3.7	precision()	909
31.147.3.8	range()	909
31.147.3.9	round()	909
31.147.3.10	step()	909
31.147.3.11	value() [1/2]	909
31.147.3.12	value() [2/2]	910
31.147.3.13	value_damage()	910

31.148 FI_Value_Input Class Reference	910
31.148.1 Detailed Description	911
31.148.2 Constructor & Destructor Documentation	911
31.148.2.1 FI_Value_Input()	911
31.148.3 Member Function Documentation	912
31.148.3.1 cursor_color() [1/2]	912
31.148.3.2 cursor_color() [2/2]	912
31.148.3.3 draw()	912
31.148.3.4 handle()	912
31.148.3.5 resize()	912
31.148.3.6 shortcut() [1/2]	913
31.148.3.7 shortcut() [2/2]	913
31.148.3.8 soft()	913
31.148.3.9 textcolor()	913
31.148.3.10 textfont() [1/2]	914
31.148.3.11 textfont() [2/2]	914
31.148.3.12 textsize() [1/2]	914
31.148.3.13 textsize() [2/2]	914
31.149 FI_Value_Output Class Reference	914
31.149.1 Detailed Description	915
31.149.2 Constructor & Destructor Documentation	915
31.149.2.1 FI_Value_Output()	915
31.149.3 Member Function Documentation	915
31.149.3.1 draw()	916
31.149.3.2 handle()	916
31.149.3.3 soft() [1/2]	916
31.149.3.4 soft() [2/2]	916
31.149.3.5 textcolor() [1/2]	916
31.149.3.6 textcolor() [2/2]	917
31.149.3.7 textfont() [1/2]	917
31.149.3.8 textfont() [2/2]	917
31.149.3.9 textsize()	917
31.150 FI_Value_Slider Class Reference	917
31.150.1 Detailed Description	918
31.150.2 Constructor & Destructor Documentation	918
31.150.2.1 FI_Value_Slider()	918
31.150.3 Member Function Documentation	919
31.150.3.1 draw()	919
31.150.3.2 handle()	919
31.150.3.3 textcolor() [1/2]	919
31.150.3.4 textcolor() [2/2]	919
31.150.3.5 textfont() [1/2]	920

31.150.3.6 <code>textfont()</code> [2/2]	920
31.150.3.7 <code>textsize()</code> [1/2]	920
31.150.3.8 <code>textsize()</code> [2/2]	920
31.151 <code>Fl_Widget</code> Class Reference	920
31.151.1 Detailed Description	927
31.151.2 Member Enumeration Documentation	927
31.151.2.1 anonymous enum	927
31.151.3 Constructor & Destructor Documentation	927
31.151.3.1 <code>Fl_Widget()</code>	928
31.151.3.2 <code>~Fl_Widget()</code>	928
31.151.4 Member Function Documentation	928
31.151.4.1 <code>activate()</code>	928
31.151.4.2 <code>active()</code>	928
31.151.4.3 <code>active_r()</code>	929
31.151.4.4 <code>align()</code> [1/2]	929
31.151.4.5 <code>align()</code> [2/2]	929
31.151.4.6 <code>argument()</code> [1/2]	929
31.151.4.7 <code>argument()</code> [2/2]	930
31.151.4.8 <code>as_gl_window()</code>	930
31.151.4.9 <code>as_group()</code>	930
31.151.4.10 <code>as_window()</code>	931
31.151.4.11 <code>box()</code> [1/2]	931
31.151.4.12 <code>box()</code> [2/2]	931
31.151.4.13 <code>callback()</code> [1/5]	931
31.151.4.14 <code>callback()</code> [2/5]	932
31.151.4.15 <code>callback()</code> [3/5]	932
31.151.4.16 <code>callback()</code> [4/5]	932
31.151.4.17 <code>callback()</code> [5/5]	932
31.151.4.18 <code>changed()</code>	933
31.151.4.19 <code>clear_active()</code>	933
31.151.4.20 <code>clear_changed()</code>	933
31.151.4.21 <code>clear_damage()</code>	933
31.151.4.22 <code>clear_output()</code>	934
31.151.4.23 <code>clear_visible()</code>	934
31.151.4.24 <code>clear_visible_focus()</code>	934
31.151.4.25 <code>color()</code> [1/3]	934
31.151.4.26 <code>color()</code> [2/3]	934
31.151.4.27 <code>color()</code> [3/3]	935
31.151.4.28 <code>color2()</code> [1/2]	935
31.151.4.29 <code>color2()</code> [2/2]	935
31.151.4.30 <code>contains()</code>	935
31.151.4.31 <code>copy_label()</code>	936

31.151.4.32 copy_tooltip()	936
31.151.4.33 damage() [1/3]	936
31.151.4.34 damage() [2/3]	937
31.151.4.35 damage() [3/3]	937
31.151.4.36 deactivate()	937
31.151.4.37 default_callback()	937
31.151.4.38 deimage() [1/3]	938
31.151.4.39 deimage() [2/3]	938
31.151.4.40 deimage() [3/3]	938
31.151.4.41 do_callback() [1/3]	939
31.151.4.42 do_callback() [2/3]	939
31.151.4.43 do_callback() [3/3]	939
31.151.4.44 draw()	939
31.151.4.45 draw_label() [1/3]	940
31.151.4.46 draw_label() [2/3]	940
31.151.4.47 draw_label() [3/3]	940
31.151.4.48 h() [1/2]	940
31.151.4.49 h() [2/2]	940
31.151.4.50 handle()	940
31.151.4.51 hide()	941
31.151.4.52 image() [1/3]	941
31.151.4.53 image() [2/3]	941
31.151.4.54 image() [3/3]	942
31.151.4.55 inside()	942
31.151.4.56 is_label_copied()	942
31.151.4.57 label() [1/3]	942
31.151.4.58 label() [2/3]	943
31.151.4.59 label() [3/3]	943
31.151.4.60 label_shortcut()	943
31.151.4.61 labelcolor() [1/2]	943
31.151.4.62 labelcolor() [2/2]	944
31.151.4.63 labelfont() [1/2]	944
31.151.4.64 labelfont() [2/2]	944
31.151.4.65 labelsizes() [1/2]	944
31.151.4.66 labelsizes() [2/2]	945
31.151.4.67 labeltype() [1/2]	945
31.151.4.68 labeltype() [2/2]	945
31.151.4.69 measure_label()	945
31.151.4.70 output()	946
31.151.4.71 parent() [1/2]	946
31.151.4.72 parent() [2/2]	946
31.151.4.73 position()	946

31.151.4.74	redraw()	947
31.151.4.75	redraw_label()	947
31.151.4.76	resize()	947
31.151.4.77	selection_color() [1/2]	947
31.151.4.78	selection_color() [2/2]	948
31.151.4.79	set_active()	948
31.151.4.80	set_changed()	948
31.151.4.81	set_output()	948
31.151.4.82	set_visible()	948
31.151.4.83	set_visible_focus()	949
31.151.4.84	show()	949
31.151.4.85	size()	949
31.151.4.86	take_focus()	949
31.151.4.87	takeevents()	950
31.151.4.88	test_shortcut() [1/2]	950
31.151.4.89	test_shortcut() [2/2]	950
31.151.4.90	tooltip() [1/2]	950
31.151.4.91	tooltip() [2/2]	951
31.151.4.92	top_window()	951
31.151.4.93	top_window_offset()	951
31.151.4.94	type() [1/2]	952
31.151.4.95	type() [2/2]	952
31.151.4.96	user_data() [1/2]	952
31.151.4.97	user_data() [2/2]	952
31.151.4.98	visible()	952
31.151.4.99	visible_focus() [1/2]	953
31.151.4.100	visible_focus() [2/2]	953
31.151.4.101	visible_r()	953
31.151.4.102	w() [1/2]	953
31.151.4.103	w() [2/2]	954
31.151.4.104	when() [1/2]	954
31.151.4.105	when() [2/2]	954
31.151.4.106	window()	954
31.151.4.107	x() [1/2]	955
31.151.4.108	x() [2/2]	955
31.151.4.109	y() [1/2]	955
31.151.4.110	y() [2/2]	955
31.152	Fl_Widget_Tracker Class Reference	956
31.152.1	Detailed Description	956
31.152.2	Member Function Documentation	956
31.152.2.1	deleted()	956
31.152.2.2	exists()	956

31.152.2.3 widget()	957
31.153 Fl_Window Class Reference	957
31.153.1 Detailed Description	961
31.153.2 Constructor & Destructor Documentation	961
31.153.2.1 Fl_Window() [1/2]	961
31.153.2.2 Fl_Window() [2/2]	961
31.153.2.3 ~Fl_Window()	961
31.153.3 Member Function Documentation	962
31.153.3.1 as_window()	962
31.153.3.2 border()	962
31.153.3.3 clear_border()	962
31.153.3.4 clear_modal_states()	962
31.153.3.5 current()	963
31.153.3.6 cursor() [1/3]	963
31.153.3.7 cursor() [2/3]	963
31.153.3.8 cursor() [3/3]	963
31.153.3.9 decorated_h()	964
31.153.3.10 decorated_w()	964
31.153.3.11 default_cursor() [1/2]	964
31.153.3.12 default_cursor() [2/2]	964
31.153.3.13 default_icon()	964
31.153.3.14 default_icons()	964
31.153.3.15 default_xclass() [1/2]	965
31.153.3.16 default_xclass() [2/2]	965
31.153.3.17 draw()	965
31.153.3.18 flush()	966
31.153.3.19 force_position() [1/2]	966
31.153.3.20 force_position() [2/2]	966
31.153.3.21 free_icons()	966
31.153.3.22 free_position()	966
31.153.3.23 fullscreen()	967
31.153.3.24 fullscreen_screens()	967
31.153.3.25 handle()	967
31.153.3.26 hide()	968
31.153.3.27 hotspot()	968
31.153.3.28 icon() [1/3]	968
31.153.3.29 icon() [2/3]	968
31.153.3.30 icon() [3/3]	969
31.153.3.31 iconize()	969
31.153.3.32 icons()	969
31.153.3.33 make_current()	969
31.153.3.34 modal()	970

31.153.3.35	resize()	970
31.153.3.36	set_menu_window()	970
31.153.3.37	set_modal()	970
31.153.3.38	set_non_modal()	970
31.153.3.39	set_tooltip_window()	971
31.153.3.40	shape() [1/2]	971
31.153.3.41	shape() [2/2]	971
31.153.3.42	show() [1/2]	972
31.153.3.43	show() [2/2]	972
31.153.3.44	shown()	973
31.153.3.45	size_range()	973
31.153.3.46	wait_for_expose()	973
31.153.3.47	xclass() [1/2]	974
31.153.3.48	xclass() [2/2]	974
31.153.4	Member Data Documentation	974
31.153.4.1	current_	975
31.154	FI_Wizard Class Reference	975
31.154.1	Detailed Description	975
31.154.2	Constructor & Destructor Documentation	976
31.154.2.1	FI_Wizard()	976
31.154.3	Member Function Documentation	976
31.154.3.1	next()	976
31.155	FI_XBM_Image Class Reference	976
31.155.1	Detailed Description	976
31.155.2	Constructor & Destructor Documentation	976
31.155.2.1	FI_XBM_Image()	977
31.156	FI_XColor Struct Reference	977
31.157	FI_Xlib_Graphics_Driver Class Reference	977
31.157.1	Detailed Description	978
31.157.2	Member Function Documentation	978
31.157.2.1	class_name()	978
31.157.2.2	color() [1/2]	979
31.157.2.3	color() [2/2]	979
31.157.2.4	copy_offscreen()	979
31.157.2.5	descent()	979
31.157.2.6	draw() [1/5]	979
31.157.2.7	draw() [2/5]	979
31.157.2.8	draw() [3/5]	980
31.157.2.9	draw() [4/5]	980
31.157.2.10	draw() [5/5]	980
31.157.2.11	draw_image() [1/2]	980
31.157.2.12	draw_image() [2/2]	981

31.157.2.13 draw_image_mono() [1/2]	981
31.157.2.14 draw_image_mono() [2/2]	981
31.157.2.15 font()	981
31.157.2.16 height()	982
31.157.2.17 rtl_draw()	982
31.157.2.18 text_extents()	982
31.157.2.19 width() [1/2]	982
31.157.2.20 width() [2/2]	982
31.158 FI_XPM_Image Class Reference	983
31.158.1 Detailed Description	983
31.158.2 Constructor & Destructor Documentation	983
31.158.2.1 FI_XPM_Image()	983
31.159 FI_Text_Editor::Key_Binding Struct Reference	983
31.159.1 Detailed Description	984
31.160 FI_Graphics_Driver::matrix Struct Reference	984
31.160.1 Detailed Description	984
31.161 FI_Preferences::Name Class Reference	984
31.161.1 Detailed Description	984
31.161.2 Constructor & Destructor Documentation	984
31.161.2.1 Name() [1/2]	985
31.161.2.2 Name() [2/2]	985
31.162 FI_Preferences::Node Class Reference	985
31.163 FI_Paged_Device::page_format Struct Reference	986
31.163.1 Detailed Description	986
31.164 FI_Preferences::RootNode Class Reference	986
31.165 FI_Scroll::ScrollInfo Struct Reference	986
31.165.1 Detailed Description	987
31.166 FI_Window::shape_data_type Struct Reference	987
31.166.1 Detailed Description	987
31.167 FI_Text_Display::Style_Table_Entry Struct Reference	988
31.167.1 Detailed Description	988
32 File Documentation	989
32.1 abi-version.h	989
32.2 dirent.h	989
32.3 Enumerations.H File Reference	989
32.3.1 Detailed Description	999
32.3.2 Macro Definition Documentation	999
32.3.2.1 FL_ABI_VERSION	999
32.3.2.2 FL_API_VERSION	1000
32.3.2.3 FL_MAJOR_VERSION	1000
32.3.2.4 FL_MINOR_VERSION	1000

32.3.2.5 FL_PATCH_VERSION	1000
32.3.2.6 FL_VERSION	1000
32.3.3 Typedef Documentation	1001
32.3.3.1 FI_Fontsize	1001
32.3.4 Enumeration Type Documentation	1001
32.3.4.1 anonymous enum	1001
32.3.4.2 FI_Boxtype	1001
32.3.4.3 FI_Cursor	1003
32.3.4.4 FI_Damage	1003
32.3.4.5 FI_Event	1004
32.3.4.6 FI_Labeltype	1006
32.3.4.7 FI_When	1007
32.3.5 Function Documentation	1007
32.3.5.1 fl_box()	1007
32.3.5.2 fl_color_cube()	1008
32.3.5.3 fl_down()	1008
32.3.5.4 fl_frame()	1008
32.3.5.5 fl_gray_ramp()	1008
32.3.6 Variable Documentation	1008
32.3.6.1 FL_ALIGN_LEFT	1008
32.3.6.2 FL_ALIGN_TOP	1008
32.3.6.3 FL_NORMAL_SIZE	1008
32.4 Enumerations.H	1009
32.5 filename.H File Reference	1016
32.5.1 Detailed Description	1017
32.6 filename.H	1017
32.7 FI.H File Reference	1019
32.7.1 Detailed Description	1020
32.8 FI.H	1020
32.9 FI_Adjuster.H	1026
32.10 fl_ask.H File Reference	1026
32.10.1 Detailed Description	1028
32.10.2 Enumeration Type Documentation	1028
32.10.2.1 FI_Beep	1028
32.11 fl_ask.H	1028
32.12 FI_Bitmap.H	1029
32.13 FI_BMP_Image.H	1030
32.14 FI_Box.H	1030
32.15 FI_Browser.H	1031
32.16 FI_Browser_.H	1033
32.17 FI_Button.H	1035
32.18 FI_Cairo.H	1036

32.19 FI_Cairo_Window.H	1037
32.20 FI_Chart.H	1037
32.21 FI_Check_Browser.H	1038
32.22 FI_Check_Button.H	1040
32.23 FI_Choice.H	1040
32.24 FI_Clock.H	1041
32.25 FI_Color_Chooser.H File Reference	1042
32.25.1 Detailed Description	1042
32.26 FI_Color_Chooser.H	1042
32.27 FI_Copy_Surface.H	1043
32.28 FI_Counter.H	1045
32.29 FI_Device.H File Reference	1046
32.29.1 Detailed Description	1047
32.29.2 Typedef Documentation	1047
32.29.2.1 FI_Draw_Image_Cb	1047
32.30 FI_Device.H	1047
32.31 FI_Dial.H	1052
32.32 FI_Double_Window.H	1053
32.33 fl_draw.H File Reference	1053
32.33.1 Detailed Description	1058
32.34 fl_draw.H	1059
32.35 FI_Export.H	1062
32.36 FI_File_Browser.H	1062
32.37 FI_File_Chooser.H	1063
32.38 FI_File_Icon.H	1065
32.39 FI_File_Input.H	1067
32.40 FI_Fill_Dial.H	1068
32.41 FI_Fill_Slider.H	1068
32.42 FI_Float_Input.H	1069
32.43 FI_FormsBitmap.H	1069
32.44 FI_FormsPixmap.H	1070
32.45 FI_Free.H	1070
32.46 FI_GIF_Image.H	1071
32.47 FI_Gl_Window.H	1071
32.48 FI_Group.H	1073
32.49 FI_Help_Dialog.H	1074
32.50 FI_Help_View.H	1075
32.51 FI_Hold_Browser.H	1078
32.52 FI_Hor_Fill_Slider.H	1079
32.53 FI_Hor_Nice_Slider.H	1079
32.54 FI_Hor_Slider.H	1080
32.55 FI_Hor_Value_Slider.H	1080

32.56 FI_Image.H File Reference	1081
32.56.1 Detailed Description	1081
32.56.2 Enumeration Type Documentation	1081
32.56.2.1 FI_RGB_Scaling	1081
32.57 FI_Image.H	1081
32.58 FI_Image_Surface.H	1083
32.59 FI_Input.H	1084
32.60 FI_Input_.H	1085
32.61 FI_Input_Choice.H	1088
32.62 FI_Int_Input.H	1090
32.63 FI_JPEG_Image.H	1090
32.64 FI_Light_Button.H	1091
32.65 FI_Line_Dial.H	1091
32.66 FI_Menu.H	1092
32.67 FI_Menu_.H	1092
32.68 FI_Menu_Bar.H	1093
32.69 FI_Menu_Button.H	1094
32.70 FI_Menu_Item.H File Reference	1094
32.70.1 Enumeration Type Documentation	1095
32.70.1.1 anonymous enum	1095
32.71 FI_Menu_Item.H	1095
32.72 FI_Menu_Window.H	1098
32.73 fl_message.H	1098
32.74 FI_Multi_Browser.H	1099
32.75 FI_Multi_Label.H	1099
32.76 FI_Multiline_Input.H	1100
32.77 FI_Multiline_Output.H	1100
32.78 FI_Native_File_Chooser.H File Reference	1101
32.78.1 Detailed Description	1101
32.79 FI_Native_File_Chooser.H	1101
32.80 FI_Nice_Slider.H	1104
32.81 FI_Object.H	1105
32.82 FI_Output.H	1105
32.83 FI_Overlay_Window.H	1106
32.84 FI_Pack.H	1106
32.85 FI_Paged_Device.H File Reference	1107
32.85.1 Detailed Description	1107
32.86 FI_Paged_Device.H	1107
32.87 FI_Pixmap.H	1109
32.88 FI_Plugin.H	1110
32.89 FI_PNG_Image.H	1111
32.90 FI_PNM_Image.H	1111

32.91 FI_Positioner.H	1112
32.92 FI_PostScript.H File Reference	1112
32.92.1 Detailed Description	1113
32.93 FI_PostScript.H	1113
32.94 FI_Preferences.H	1116
32.95 FI_Printer.H File Reference	1118
32.95.1 Detailed Description	1118
32.96 FI_Printer.H	1119
32.97 FI_Progress.H	1120
32.98 FI_Radio_Button.H	1121
32.99 FI_Radio_Light_Button.H	1122
32.100 FI_Radio_Round_Button.H	1122
32.101 FI_Repeat_Button.H	1123
32.102 FI_Return_Button.H	1123
32.103 FI_RGB_Image.H	1124
32.104 FI_Roller.H	1124
32.105 FI_Round_Button.H	1124
32.106 FI_Round_Clock.H	1125
32.107 FI_Scroll.H	1125
32.108 FI_Scrollbar.H	1127
32.109 FI_Secret_Input.H	1127
32.110 FI_Select_Browser.H	1128
32.111 FI_Shared_Image.H File Reference	1128
32.111.1 Detailed Description	1129
32.111.2 Function Documentation	1129
32.111.2.1 fl_register_images()	1129
32.112 FI_Shared_Image.H	1129
32.113 fl_show_colormap.H File Reference	1130
32.113.1 Detailed Description	1131
32.114 fl_show_colormap.H	1131
32.115 fl_show_input.H	1131
32.116 FI_Simple_Counter.H	1131
32.117 FI_Single_Window.H	1132
32.118 FI_Slider.H	1132
32.119 FI_Spinner.H	1133
32.120 FI_Sys_Menu_Bar.H	1136
32.121 FI_Table.H	1137
32.122 FI_Table_Row.H	1143
32.123 FI_Tabs.H	1145
32.124 FI_Text_Buffer.H	1146
32.125 FI_Text_Display.H	1149
32.126 FI_Text_Editor.H	1153

32.127 FI_Tile.H	1155
32.128 FI_Tiled_Image.H	1155
32.129 FI_Timer.H	1156
32.130 FI_Toggle_Button.H	1157
32.131 FI_Toggle_Light_Button.H	1157
32.132 FI_Toggle_Round_Button.H	1158
32.133 FI_Tooltip.H	1158
32.134 FI_Tree.H File Reference	1159
32.134.1 Detailed Description	1160
32.134.2 Enumeration Type Documentation	1160
32.134.2.1 FI_Tree_Reason	1160
32.135 FI_Tree.H	1160
32.136 FI_Tree_Item.H File Reference	1164
32.136.1 Detailed Description	1164
32.137 FI_Tree_Item.H	1164
32.138 FI_Tree_Item_Array.H File Reference	1168
32.138.1 Detailed Description	1169
32.139 FI_Tree_Item_Array.H	1169
32.140 FI_Tree_Prefs.H File Reference	1170
32.140.1 Detailed Description	1170
32.140.2 Enumeration Type Documentation	1171
32.140.2.1 FI_Tree_Connector	1171
32.140.2.2 FI_Tree_Item_Draw_Mode	1171
32.140.2.3 FI_Tree_Item_Reselect_Mode	1171
32.140.2.4 FI_Tree_Select	1171
32.140.2.5 FI_Tree_Sort	1172
32.141 FI_Tree_Prefs.H	1172
32.142 fl_types.h File Reference	1176
32.142.1 Detailed Description	1176
32.142.2 Typedef Documentation	1176
32.142.2.1 FI_CString	1176
32.142.2.2 FI_String	1176
32.143 fl_types.h	1176
32.144 fl_utf8.h File Reference	1177
32.144.1 Detailed Description	1179
32.145 fl_utf8.h	1179
32.146 FI_Valuator.H	1182
32.147 FI_Value_Input.H	1183
32.148 FI_Value_Output.H	1183
32.149 FI_Value_Slider.H	1184
32.150 FI_Widget.H File Reference	1185
32.150.1 Detailed Description	1185

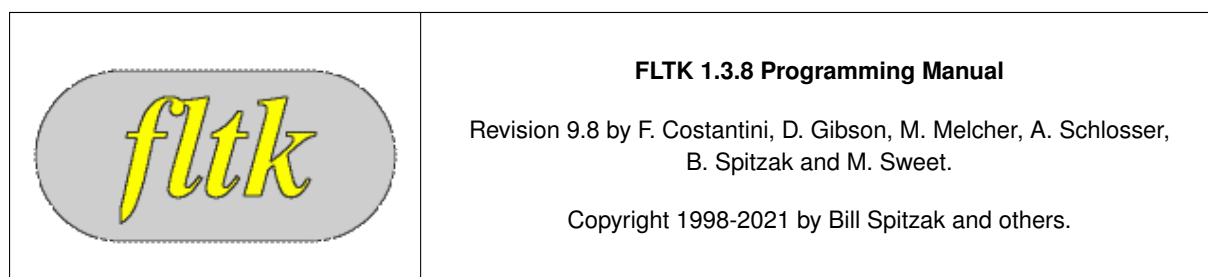
32.150.2 Macro Definition Documentation	1185
32.150.2.1 FL_RESERVED_TYPE	1185
32.150.3 Typedef Documentation	1185
32.150.3.1 fl_intptr_t	1186
32.151 FI_Widget.H	1186
32.152 FI_Window.H File Reference	1190
32.152.1 Detailed Description	1190
32.153 FI_Window.H	1190
32.154 FI_Wizard.H	1193
32.155 FI_XBM_Image.H	1194
32.156 FI_XPM_Image.H	1195
32.157 forms.H	1195
32.158 gl.h File Reference	1205
32.158.1 Detailed Description	1206
32.158.2 Function Documentation	1206
32.158.2.1 gl_color()	1206
32.158.2.2 gl_draw() [1/7]	1206
32.158.2.3 gl_draw() [2/7]	1206
32.158.2.4 gl_draw() [3/7]	1207
32.158.2.5 gl_draw() [4/7]	1207
32.158.2.6 gl_draw() [5/7]	1207
32.158.2.7 gl_draw() [6/7]	1207
32.158.2.8 gl_draw() [7/7]	1207
32.158.2.9 gl_rect()	1208
32.158.2.10 gl_rectf()	1208
32.159 gl.h	1208
32.160 gl2opengl.h	1209
32.161 gl_draw.H	1209
32.162 glu.h	1210
32.163 glut.H	1211
32.164 mac.H File Reference	1216
32.164.1 Detailed Description	1217
32.165 mac.H	1217
32.166 math.h	1220
32.167 names.h	1221
32.168 platform.H	1222
32.169 win32.H	1222
32.170 x.H	1224
32.171 cgdebug.h	1226
32.172 fastarrow.h	1229
32.173 fl_arc.cxx File Reference	1229
32.173.1 Detailed Description	1229

32.174 fl_arci.cxx File Reference	1229
32.174.1 Detailed Description	1229
32.175 fl_ask.cxx File Reference	1229
32.175.1 Detailed Description	1231
32.176 fl_boxtype.cxx File Reference	1231
32.176.1 Detailed Description	1232
32.176.2 Function Documentation	1232
32.176.2.1 fl_internal_boxtype()	1232
32.176.2.2 fl_rectbound()	1232
32.177 fl_cmap.h	1232
32.178 fl_color.cxx File Reference	1236
32.178.1 Detailed Description	1236
32.179 FI_compose.cxx File Reference	1237
32.179.1 Detailed Description	1237
32.180 fl_curve.cxx File Reference	1237
32.180.1 Detailed Description	1237
32.181 FI_Double_Window.cxx File Reference	1237
32.181.1 Detailed Description	1238
32.182 FI_Font.H	1238
32.183 FI_GI_Choice.H	1239
32.184 fl_line_style.cxx File Reference	1240
32.184.1 Detailed Description	1241
32.185 FI_Paged_Device.cxx File Reference	1241
32.185.1 Detailed Description	1241
32.186 fl_rect.cxx File Reference	1241
32.186.1 Detailed Description	1241
32.187 fl_vertex.cxx File Reference	1241
32.187.1 Detailed Description	1241
32.188 FI_XColor.H	1242
32.189 flstring.h	1242
32.190 freeglut_teapot_data.h	1243
32.191 mediumarrow.h	1245
32.192 print_panel.h	1245
32.193 slowarrow.h	1246
32.194 Xutf8.h	1246
32.195 case.h	1248
32.196 dingbats_.h	1268
32.197 spacing.h	1275
32.198 symbol_.h	1298
32.199 armSCII_8.h	1311
32.200 ascii.h	1312
32.201 big5.h	1312

32.202 big5_emacs.h	1360
32.203 cp1133.h	1362
32.204 cp1251.h	1363
32.205 cp1255.h	1364
32.206 cp1256.h	1366
32.207 cp936ext.h	1368
32.208 gb2312.h	1439
32.209 georgian_academy.h	1469
32.210 georgian_ps.h	1470
32.211 iso8859_1.h	1471
32.212 iso8859_10.h	1472
32.213 iso8859_11.h	1473
32.214 iso8859_13.h	1474
32.215 iso8859_14.h	1475
32.216 iso8859_15.h	1476
32.217 iso8859_16.h	1477
32.218 iso8859_2.h	1478
32.219 iso8859_3.h	1479
32.220 iso8859_4.h	1481
32.221 iso8859_5.h	1482
32.222 iso8859_6.h	1483
32.223 iso8859_7.h	1484
32.224 iso8859_8.h	1485
32.225 iso8859_9.h	1486
32.226 iso8859_9e.h	1487
32.227 jisx0201.h	1488
32.228 jisx0208.h	1489
32.229 jisx0212.h	1516
32.230 koi8_c.h	1541
32.231 koi8_r.h	1543
32.232 koi8_u.h	1544
32.233 ksc5601.h	1546
32.234 mulelao.h	1581
32.235 tatar_cyr.h	1582
32.236 tcvn.h	1583
32.237 tis620.h	1585
32.238 ucs2be.h	1586
32.239 utf8.h	1586
32.240 viscii.h	1587
32.241 Ximint.h	1589
32.242 Xlibint.h	1589

Chapter 1

FLTK Programming Manual



This software and manual are provided under the terms of the GNU Library General Public License. Permission is granted to reproduce this manual or any portion for any purpose, provided this copyright and permission notice are preserved.

<p>Preface Introduction to FLTK FLTK Basics Common Widgets and Attributes</p> <ul style="list-style-type: none">• Colors• Box Types• Labels and Label Types• Drawing Images <p>Designing a Simple Text Editor Drawing Things in FLTK Handling Events</p> <ul style="list-style-type: none">• Fl::event_*() methods• Event Propagation <p>Adding and Extending Widgets Using OpenGL Programming with FLUID</p> <ul style="list-style-type: none">• GUI Attributes• Selecting and Moving Widgets• Image Labels	<p>Advanced FLTK Unicode and UTF-8 Support</p> <p>Appendices:</p> <ul style="list-style-type: none">• FLTK Enumerations• GLUT Compatibility<ul style="list-style-type: none">– class Fl_Glut_Window• Forms Compatibility• Operating System Issues• Migrating Code from FLTK 1.0 to 1.1• Migrating Code from FLTK 1.1 to 1.3• Developer Information• Software License• Example Source Code• FAQ (Frequently Asked Questions)
--	---

Chapter 2

Preface

This manual describes the Fast Light Tool Kit ("FLTK") version 1.3.8, a C++ Graphical User Interface ("GUI") toolkit for UNIX, Microsoft Windows and Apple OS X.

Each of the chapters in this manual is designed as a tutorial for using FLTK, while the appendices provide a convenient reference for all FLTK widgets, functions, and operating system interfaces.

This manual may be printed, modified, and/or used under the terms of the FLTK license provided in [Software License](#).

2.1 Organization

This manual is organized into the following chapters and appendices:

- [Introduction to FLTK](#)
- [FLTK Basics](#)
- [Common Widgets and Attributes](#)
- [Designing a Simple Text Editor](#)
- [Drawing Things in FLTK](#)
- [Handling Events](#)
- [Adding and Extending Widgets](#)
- [Using OpenGL](#)
- [Programming with FLUID](#)
- [Advanced FLTK](#)
- [Unicode and UTF-8 Support](#)
- [FLTK Enumerations](#)
- [GLUT Compatibility](#)
- [Forms Compatibility](#)
- [Operating System Issues](#)
- [Migrating Code from FLTK 1.0 to 1.1](#)
- [Migrating Code from FLTK 1.1 to 1.3](#)
- [Developer Information](#)
- [Software License](#)
- [Example Source Code](#)

2.2 Conventions

This manual was generated using Doxygen (see <http://www.doxygen.org/>) to process the source code itself, special comments in the code, and additional documentation files. In general, Doxygen recognizes and denotes the following entities as shown:

- classes, such as `Fl_Widget`,
- methods, such as `Fl_Widget::callback(Fl_Callback* cb, void* p)`,
- functions, such as `fl_draw(const char *str, int x, int y)`,
- internal links, such as [Conventions](#),
- external links, such as <http://www.stack.nl/~dimitri/doxygen/>

Other code samples and commands are shown in regular courier type.

2.3 Abbreviations

The following abbreviations are used in this manual:

X11

The X Window System version 11.

Xlib

The X Window System interface library.

MS Windows, WIN32

The Microsoft Windows Application Programmer's Interface for Windows 2000, Windows XP, Windows Vista, and Windows 7. FLTK uses the preprocessor definition `WIN32` for the 32 bit and 64 bit MS Windows API.

OS X, APPLE

The Apple desktop operating system OS X 10.0 and later. MacOS 8 and 9 support was dropped after FLTK 1.0.10. FLTK uses the preprocessor definition `APPLE` for OS X.

2.4 Copyrights and Trademarks

FLTK is Copyright 1998-2021 by Bill Spitzak and others. Use and distribution of FLTK is governed by the GNU Library General Public License with 4 exceptions, located in [Software License](#).

UNIX is a registered trademark of the X Open Group, Inc. Microsoft and Windows are registered trademarks of Microsoft Corporation. OpenGL is a registered trademark of Silicon Graphics, Inc. Apple, Macintosh, MacOS, and Mac OS X are registered trademarks of Apple Computer, Inc.

Chapter 3

Introduction to FLTK

The Fast Light Tool Kit ("FLTK", pronounced "fulltick") is a cross-platform C++ GUI toolkit for UNIX®/Linux® (X11), Microsoft® Windows®, and Apple® OS X®.

FLTK provides modern GUI functionality without the bloat and supports 3D graphics via OpenGL® and its built-in GLUT emulation. It was originally developed by Mr. Bill Spitzak and is currently maintained by a small group of developers across the world with a central repository in the US.

3.1 History of FLTK

It has always been Bill's belief that the GUI API of all modern systems is much too high level. Toolkits (even FLTK) are *not* what should be provided and documented as part of an operating system. The system only has to provide arbitrary shaped but featureless windows, a powerful set of graphics drawing calls, and a simple *unalterable* method of delivering events to the owners of the windows. NeXT (if you ignored NextStep) provided this, but they chose to hide it and tried to push their own baroque toolkit instead.

Many of the ideas in FLTK were developed on a NeXT (but *not* using NextStep) in 1987 in a C toolkit Bill called "views". Here he came up with passing events downward in the tree and having the handle routine return a value indicating whether it used the event, and the table-driven menus. In general he was trying to prove that complex UI ideas could be entirely implemented in a user space toolkit, with no knowledge or support by the system.

After going to film school for a few years, Bill worked at Sun Microsystems on the (doomed) NeWS project. Here he found an even better and cleaner windowing system, and he reimplemented "views" atop that. NeWS did have an unnecessarily complex method of delivering events which hurt it. But the designers did admit that perhaps the user could write just as good of a button as they could, and officially exposed the lower level interface.

With the death of NeWS Bill realized that he would have to live with X. The biggest problem with X is the "window manager", which means that the toolkit can no longer control the window borders or drag the window around.

At Digital Domain Bill discovered another toolkit, "Forms". Forms was similar to his work, but provided many more widgets, since it was used in many real applications, rather than as theoretical work. He decided to use Forms, except he integrated his table-driven menus into it. Several very large programs were created using this version of Forms.

The need to switch to OpenGL and GLX, portability, and a desire to use C++ subclassing required a rewrite of Forms. This produced the first version of FLTK. The conversion to C++ required so many changes it made it impossible to recompile any Forms objects. Since it was incompatible anyway, Bill decided to incorporate his older ideas as much as possible by simplifying the lower level interface and the event passing mechanism.

Bill received permission to release it for free on the Internet, with the GNU general public license. Response from Internet users indicated that the Linux market dwarfed the SGI and high-speed GL market, so he rewrote it to use X for all drawing, greatly speeding it up on these machines. That is the version you have now.

Digital Domain has since withdrawn support for FLTK. While Bill is no longer able to actively develop it, he still contributes to FLTK in his free time and is a part of the FLTK development team.

3.2 Features

FLTK was designed to be statically linked. This was done by splitting it into many small objects and designing it so that functions that are not used do not have pointers to them in the parts that are used, and thus do not get linked in. This allows you to make an easy-to-install program or to modify FLTK to the exact requirements of your application without worrying about bloat. FLTK works fine as a shared library, though, and is now included with several Linux distributions.

Here are some of the core features unique to FLTK:

- `sizeof(Fl_Widget) == 64` to `92`.
- The "core" (the "hello" program compiled & linked with a static FLTK library using gcc on a 486 and then stripped) is 114K.
- The FLUID program (which includes every widget) is 538k.
- Written directly atop core libraries (Xlib, WIN32 or Cocoa) for maximum speed, and carefully optimized for code size and performance.
- Precise low-level compatibility between the X11, WIN32 and MacOS versions - only about 10% of the code is different.
- Interactive user interface builder program. Output is human-readable and editable C++ source code.
- Support for overlay hardware, with emulation if none is available.
- Very small & fast portable 2-D drawing library to hide Xlib, WIN32, or QuickDraw.
- OpenGL/Mesa drawing area widget.
- Support for OpenGL overlay hardware on both X11 and WIN32, with emulation if none is available.
- Text widgets with cut & paste, undo, and support for Unicode text and international input methods.
- Compatibility header file for the GLUT library.
- Compatibility header file for the XForms library.

3.3 Licensing

FLTK comes with complete free source code. FLTK is available under the terms of the [GNU Library General Public License](#) with exceptions that allow for static linking. Contrary to popular belief, it can be used in commercial software - even Bill Gates could use it!

3.4 What Does "FLTK" Mean?

FLTK was originally designed to be compatible with the Forms Library written for SGI machines. In that library all the functions and structures started with "fl_". This naming was extended to all new methods and widgets in the C++ library, and this prefix was taken as the name of the library. It is almost impossible to search for "FL" on the Internet, due to the fact that it is also the abbreviation for Florida. After much debating and searching for a new name for the toolkit, which was already in use by several people, Bill came up with "FLTK", including a bogus excuse that it stands for "The Fast Light Toolkit".

3.5 Building and Installing FLTK Under UNIX and Apple OS X

In most cases you can just type "make". This will run configure with the default of no options and then compile everything.

For OS X, Xcode 3 project files can be found in the 'ide' directory.

FLTK uses GNU autoconf to configure itself for your UNIX platform. The main things that the configure script will look for are the X11 and OpenGL (or Mesa) header and library files. If these cannot be found in the standard include/library locations you'll need to define the CFLAGS, CXXFLAGS, and LDFLAGS environment variables. For the Bourne and Korn shells you'd use:

```
CFLAGS=-Iincludedir; export CFLAGS
CXXFLAGS=-Iincludedir; export CXXFLAGS
LDFLAGS=-Llibdir; export LDFLAGS
```

For C shell and tcsh, use:

```
setenv CFLAGS "-Iincludedir"
setenv CXXFLAGS "-Iincludedir"
setenv LDFLAGS "-Llibdir"
```

By default configure will look for a C++ compiler named CC, c++, g++, or gcc in that order. To use another compiler you need to set the CXX environment variable:

```
CXX=x1C; export CXX
setenv CXX "x1C"
```

The CC environment variable can also be used to override the default C compiler (cc or gcc), which is used for a few FLTK source files.

You can run configure yourself to get the exact setup you need. Type "./configure <options>", where options are:

-enable-cygwin

Enable the Cygwin libraries under WIN32

-enable-debug

Enable debugging code & symbols

-disable-gl

Disable OpenGL support

-enable-shared

Enable generation of shared libraries

-enable-threads

Enable multithreading support

-enable-xdbe

Enable the X double-buffer extension

-enable-xft

Enable the Xft library for anti-aliased fonts under X11

-enable-x11

When targeting cygwin, build with X11 GUI instead of windows GDI

-enable-cp936

Under X11, enable use of the GB2312 locale

-bindir=/path

Set the location for executables [default = \$prefix/bin]

-datadir=/path

Set the location for data files. [default = \$prefix/share]

-libdir=/path

Set the location for libraries [default = \$prefix/lib]

-includedir=/path

Set the location for include files. [default = \$prefix/include]

-mandir=/path

Set the location for man pages. [default = \$prefix/man]

-prefix=/dir

Set the directory prefix for files [default = /usr/local]

When the configure script is done you can just run the "make" command. This will build the library, FLUID tool, and all of the test programs.

To install the library, become root and type "make install". This will copy the "fluid" executable to "bindir", the header files to "includedir", and the library files to "libdir".

3.6 Building FLTK Under Microsoft Windows

NOTE: This documentation section is currently under review. More up-to-date information for this release may be available in the file "README.MSWindows.txt" and you should read that file to determine if there are changes that may be applicable to your build environment.

FLTK 1.3 is officially supported on Windows (2000,) 2003, XP, and later. Older Windows versions prior to Windows 2000 are not officially supported, but may still work. The main reason is that the OS version needs to support UTF-8. FLTK 1.3 is known to work on recent versions of Windows such as Windows 7, Windows 8/8.1 and Windows 10 and has been reported to work in both 32-bit and 64-bit versions of these.

FLTK currently supports the following development environments on the Windows platform:

CAUTION: Libraries built by any one of these build environments can not be mixed with object files from any of the other environments! (They use incompatible C++ conventions internally.)

Free Microsoft Visual C++ 2008 Express and Visual C++ 2010 Express or later versions using the supplied workspace and project files. Older versions, and the commercial versions, can be used as well, if they can open the project files. Be sure to get your service packs!

The project files can be found in the "ide/" directory. Please read "ide/README.IDE" for more info about this.

3.6.1 GNU toolsets (Cygwin or MinGW) hosted on Windows

If using Cygwin with the Cygwin shell, or MinGW with the Msys shell, these build environments behave very much like a Unix or OS X build and the notes above in the section on *Building and Installing FLTK Under UNIX and Apple OS X* apply, in particular the descriptions of using the "configure" script and its related options.

In general for a build using these tools, e.g. for the Msys shell with MinGW, it should suffice to "cd" into the directory where you have extracted the fltk tarball and type:

```
./configure
make
```

This will build the fltk libraries and they can then be utilised directly from the build location. NOTE: this may be simpler than "installing" them in many cases as different tool chains on Windows have different ideas about where the files should be "installed" to.

For example, if you "install" the libraries using Msys/MinGW with the following command:

```
make install
```

Then Msys will "install" the libraries to where it thinks the path "/usr/local/" leads to. If you only ever build code from within the Msys environment this works well, but the actual "Windows path" these files are located in will be something like "C:\msys\1.0\local\lib", depending on where your Msys installation is rooted, which may not be useful to other tools.

If you want to install your built fltk libraries in a non-standard location you may do:

```
sh configure --prefix=C:/FLTK
make
```

Where the value passed to "prefix" is the path at which you would like fltk to be installed.

A subsequent invocation of "make install" will then place the fltk libraries and header files into that path.

The other options to "configure" may also be used to tailor the build to suit your environment.

3.6.2 Using the Visual C++ DLL Library

The "fltkdll.dsp" project file builds a DLL-version of the FLTK library. Because of name mangling differences between PC compilers (even between different versions of Visual C++!) you can only use the DLL that is generated with the same version compiler that you built it with.

When compiling an application or DLL that uses the FLTK DLL, you will need to define the `FL_DLL` preprocessor symbol to get the correct linkage commands embedded within the FLTK header files.

3.7 Internet Resources

FLTK is available on the 'net in a bunch of locations:

WWW

<http://www.fltk.org/>
<http://www.fltk.org/str.php> [for reporting bugs]
<https://www.fltk.org/software.php> [source code]
<http://www.fltk.org/newsgroups.php> [newsgroup/forums]

NNTP Newsgroups

<https://groups.google.com/forum/#!forum/fltkgeneral> [Google Groups interface]
<news://fltk.org:1024/> [NNTP interface]
<http://fltk.org/newsgroups.php> [web interface]

3.8 Reporting Bugs

To report a bug in FLTK, or for feature requests, please use the form at <http://www.fltk.org/str.php>, and click on "Submit Bug or Feature Request".

You'll be prompted for the FLTK version, operating system & version, and compiler that you are using. We will be unable to provide any kind of help without that basic information.

For general support and questions, please use the fltk.general newsgroup (see above, "NNTP Newsgroups") or the web interface to the newsgroups at <http://fltk.org/newsgroups.php>.

Chapter 4

FLTK Basics

This chapter teaches you the basics of compiling programs that use FLTK.

4.1 Writing Your First FLTK Program

All programs must include the file `<FL/Fl.H>`. In addition the program must include a header file for each FLTK class it uses. Listing 1 shows a simple "Hello, World!" program that uses FLTK to display the window.

Listing 1 - "hello.cxx"

```
#include <FL/Fl.H>
#include <FL/Fl_Window.H>
#include <FL/Fl_Box.H>
int main(int argc, char **argv) {
    Fl_Window *window = new Fl_Window(340,180);
    Fl_Box *box = new Fl_Box(20,40,300,100,"Hello, World!");
    box->box(FL_UP_BOX);
    box->labelfont(FL_BOLD+FL_ITALIC);
    box->labelsize(36);
    box->labeltype(FL_SHADOW_LABEL);
    window->end();
    window->show(argc, argv);
    return Fl::run();
}
```

After including the required header files, the program then creates a window. All following widgets will automatically be children of this window.

```
Fl_Window *window = new Fl_Window(340,180);
```

Then we create a box with the "Hello, World!" string in it. FLTK automatically adds the new box to window, the current grouping widget.

```
Fl_Box *box = new Fl_Box(20,40,300,100,"Hello, World!");
```

Next, we set the type of box and the font, size, and style of the label:

```
box->box(FL_UP_BOX);
box->labelfont(FL_BOLD+FL_ITALIC);
box->labelsize(36);
box->labeltype(FL_SHADOW_LABEL);
```

We tell FLTK that we will not add any more widgets to window.

```
window->end();
```

Finally, we show the window and enter the FLTK event loop:

```
window->show(argc, argv);
return Fl::run();
```

The resulting program will display the window in Figure 4.1. You can quit the program by closing the window or pressing the `ESC` key.



Figure 4.1 The Hello, World! Window

4.1.1 Creating the Widgets

The widgets are created using the C++ `new` operator. For most widgets the arguments to the constructor are:
`Fl_Widget(x, y, width, height, label)`

The `x` and `y` parameters determine where the widget or window is placed on the screen. In FLTK the top left corner of the window or screen is the origin (i.e. `x = 0`, `y = 0`) and the units are in pixels.

The `width` and `height` parameters determine the size of the widget or window in pixels. The maximum widget size is typically governed by the underlying window system or hardware.

`label` is a pointer to a character string to label the widget with or `NULL`. If not specified the label defaults to `NULL`. The label string must be in static storage such as a string constant because FLTK does not make a copy of it - it just uses the pointer.

4.1.2 Creating Widget hierarchies

Widgets are commonly ordered into functional groups, which in turn may be grouped again, creating a hierarchy of widgets. FLTK makes it easy to fill groups by automatically adding all widgets that are created between a `myGroup->begin()` and `myGroup->end()`. In this example, `myGroup` would be the *current* group.

Newly created groups and their derived widgets implicitly call `begin()` in the constructor, effectively adding all subsequently created widgets to itself until `end()` is called.

Setting the current group to `NULL` will stop automatic hierarchies. New widgets can now be added manually using `Fl_Group::add(...)` and `Fl_Group::insert(...)`.

4.1.3 Get/Set Methods

`box->box(FL_UP_BOX)` sets the type of box the `Fl_Box` draws, changing it from the default of `FL_NO_BOX`, which means that no box is drawn. In our "Hello, World!" example we use `FL_UP_BOX`, which means that a raised button border will be drawn around the widget. More details are available in the [Box Types](#) section.

You could examine the `boxtype` in by doing `box->box()`. FLTK uses method name overloading to make short names for get/set methods. A "set" method is always of the form "void name(type)", and a "get" method is always of the form "type name() const".

4.1.4 Redrawing After Changing Attributes

Almost all of the set/get pairs are very fast, short inline functions and thus very efficient. However, *the "set" methods do not call `redraw()`* - you have to call it yourself. This greatly reduces code size and execution time. The only common exceptions are `value()` which calls `redraw()` and `label()` which calls `redraw_label()` if necessary.

4.1.5 Labels

All widgets support labels. In the case of window widgets, the label is used for the label in the title bar. Our example program calls the `labelfont()`, `labelsize()`, and `labeltype()` methods.

The `labelfont()` method sets the typeface and style that is used for the label, which for this example we are using `FL_BOLD` and `FL_ITALIC`. You can also specify typefaces directly.

The `labelsize()` method sets the height of the font in pixels.

The `labeltype()` method sets the type of label. FLTK supports normal, embossed, and shadowed labels internally, and more types can be added as desired.

A complete list of all label options can be found in the section on [Labels and Label Types](#).

4.1.6 Showing the Window

The `show()` method shows the widget or window. For windows you can also provide the command-line arguments to allow users to customize the appearance, size, and position of your windows.

4.1.7 The Main Event Loop

All FLTK applications (and most GUI applications in general) are based on a simple event processing model. User actions such as mouse movement, button clicks, and keyboard activity generate events that are sent to an application. The application may then ignore the events or respond to the user, typically by redrawing a button in the "down" position, adding the text to an input field, and so forth.

FLTK also supports idle, timer, and file pseudo-events that cause a function to be called when they occur. Idle functions are called when no user input is present and no timers or files need to be handled - in short, when the application is not doing anything. Idle callbacks are often used to update a 3D display or do other background processing.

Timer functions are called after a specific amount of time has expired. They can be used to pop up a progress dialog after a certain amount of time or do other things that need to happen at more-or-less regular intervals. FLTK timers are not 100% accurate, so they should not be used to measure time intervals, for example.

File functions are called when data is ready to read or write, or when an error condition occurs on a file. They are most often used to monitor network connections (sockets) for data-driven displays.

FLTK applications must periodically check (`Fl::check()`) or wait (`Fl::wait()`) for events or use the `Fl::run()` method to enter a standard event processing loop. Calling `Fl::run()` is equivalent to the following code:

```
while (Fl::wait());
```

`Fl::run()` does not return until all of the windows under FLTK control are closed by the user or your program.

4.2 Compiling Programs with Standard Compilers

Under UNIX (and under Microsoft Windows when using the GNU development tools) you will probably need to tell the compiler where to find the header files. This is usually done using the `-I` option:

```
CC -I/usr/local/include ...
gcc -I/usr/local/include ...
```

The `fltk-config` script included with FLTK can be used to get the options that are required by your compiler:

```
CC `fltk-config --cxxflags` ...
```

Similarly, when linking your application you will need to tell the compiler to use the FLTK library:

```
CC ... -L/usr/local/lib -lfltk -lXext -lX11 -lm
gcc ... -L/usr/local/lib -lfltk -lXext -lX11 -lm
```

Aside from the "fltk" library, there is also a "fltk_forms" library for the XForms compatibility classes, "fltk_gl" for the OpenGL and GLUT classes, and "fltk_images" for the image file classes, [Fl_Help_Dialog](#) widget, and system icon support.

Note

The libraries are named "fltk.lib", "fltkgl.lib", "fltkforms.lib", and "fltkimages.lib", respectively under Windows.

As before, the `fltk-config` script included with FLTK can be used to get the options that are required by your linker:

```
CC ... `fltk-config --ldflags`
```

The forms, GL, and images libraries are included with the "--use-foo" options, as follows:

```
CC ... `fltk-config --use-forms --ldflags`
CC ... `fltk-config --use-gl --ldflags`
CC ... `fltk-config --use-images --ldflags`
CC ... `fltk-config --use-forms --use-gl --use-images --ldflags`
```

Finally, you can use the `fltk-config` script to compile a single source file as a FLTK program:

```
fltk-config --compile filename.cpp
fltk-config --use-forms --compile filename.cpp
fltk-config --use-gl --compile filename.cpp
fltk-config --use-images --compile filename.cpp
fltk-config --use-forms --use-gl --use-images --compile filename.cpp
```

Any of these will create an executable named `filename`.

4.3 Compiling Programs with Makefiles

The previous section described how to use `fltk-config` to build a program consisting of a single source file from the command line, and this is very convenient for small test programs. But `fltk-config` can also be used to set the compiler and linker options as variables within a Makefile that can be used to build programs out of multiple source files:

```
CXX      = $(shell fltk-config --cxx)
DEBUG    = -g
CXXFLAGS = $(shell fltk-config --use-gl --use-images --cxxflags ) -I.
LDLAGS   = $(shell fltk-config --use-gl --use-images --ldflags )
LDSTATIC = $(shell fltk-config --use-gl --use-images --ldstaticflags )
LINK     = $(CXX)
TARGET  = cube
OBJS    = CubeMain.o CubeView.o CubeViewUI.o
SRCS    = CubeMain.cxx CubeView.cxx CubeViewUI.cxx
.SUFFIXES: .o .cxx
%.o:    %.cxx
        $(CXX) $(CXXFLAGS) $(DEBUG) -c $<
all:    $(TARGET)
        $(LINK) -o $(TARGET) $(OBJS) $(LDSTATIC)
$(TARGET): $(OBJS)
CubeMain.o: CubeMain.cxx CubeViewUI.h
CubeView.o: CubeView.cxx CubeView.h CubeViewUI.h
CubeViewUI.o: CubeViewUI.cxx CubeView.h
clean:  $(TARGET) $(OBJS)
        rm -f *.o 2> /dev/null
        rm -f $(TARGET) 2> /dev/null
```


4.4 Compiling Programs with Microsoft Visual C++

In Visual C++ you will need to tell the compiler where to find the FLTK header files. This can be done by selecting "Settings" from the "Project" menu and then changing the "Preprocessor" settings under the "C/C++" tab. You will also need to add the FLTK (`FLTK.LIB` or `FLTKD.LIB`) and the Windows Common Controls (`COMCTL32.LIB`) libraries to the "Link" settings. You must also define `WIN32`.

More information can be found in `README.MSWindows.txt`.

You can build your Microsoft Windows applications as Console or Desktop applications. If you want to use the standard `C main()` function as the entry point, FLTK includes a `WinMain()` function that will call your `main()` function for you.

4.5 Naming

All public symbols in FLTK start with the characters 'F' and 'L':

- Functions are either `Fl::foo()` or `fl_foo()`.
- Class and type names are capitalized: `Fl_Foo`.
- [Constants and enumerations](#) are uppercase: `FL_FOO`.
- All header files start with `<FL/...>`.

4.6 Header Files

The proper way to include FLTK header files is:

```
#include <FL/Fl_xyz.H>
```

Note

Case *is* significant on many operating systems, and the C standard uses the forward slash (/) to separate directories. *Do not use any of the following include lines:*

```
#include <FL\Fl_xyz.H>  
#include <fl/fl_xyz.h>  
#include <Fl/fl_xyz.h>
```


Chapter 5

Common Widgets and Attributes

This chapter describes many of the widgets that are provided with FLTK and covers how to query and set the standard attributes.

5.1 Buttons

FLTK provides many types of buttons:

- [Fl_Button](#) - A standard push button.
- [Fl_Check_Button](#) - A button with a check box.
- [Fl_Light_Button](#) - A push button with a light.
- [Fl_Repeat_Button](#) - A push button that repeats when held.
- [Fl_Return_Button](#) - A push button that is activated by the `Enter` key.
- [Fl_Round_Button](#) - A button with a radio circle.

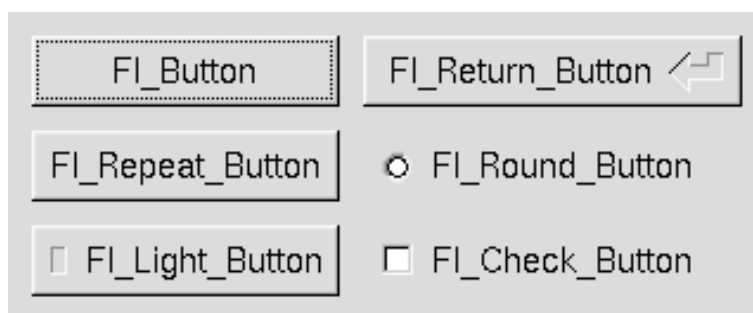


Figure 5.1 FLTK Button Widgets

All of these buttons just need the corresponding `<FL/Fl_xyz_Button.H>` header file. The constructor takes the bounding box of the button and optionally a label string:

```
Fl_Button *button = new Fl_Button(x, y, width, height, "label");
Fl_Light_Button *lbutton = new Fl_Light_Button(x, y, width, height);
Fl_Round_Button *rbutton = new Fl_Round_Button(x, y, width, height, "label");
```

Each button has an associated `type()` which allows it to behave as a push button, toggle button, or radio button:

```
button->type(FL_NORMAL_BUTTON);
lbutton->type(FL_TOGGLE_BUTTON);
rbutton->type(FL_RADIO_BUTTON);
```

For toggle and radio buttons, the `value()` method returns the current button state (0 = off, 1 = on). The `set()` and `clear()` methods can be used on toggle buttons to turn a toggle button on or off, respectively. Radio buttons can be turned on with the `setonly()` method; this will also turn off other radio buttons in the same group.

5.2 Text

FLTK provides several text widgets for displaying and receiving text:

- [Fl_Input](#) - A one-line text input field.
- [Fl_Output](#) - A one-line text output field.
- [Fl_Multiline_Input](#) - A multi-line text input field.
- [Fl_Multiline_Output](#) - A multi-line text output field.
- [Fl_Text_Display](#) - A multi-line text display widget.
- [Fl_Text_Editor](#) - A multi-line text editing widget.
- [Fl_Help_View](#) - A HTML text display widget.

The [Fl_Output](#) and [Fl_Multiline_Output](#) widgets allow the user to copy text from the output field but not change it.

The `value()` method is used to get or set the string that is displayed:

```
Fl_Input *input = new Fl_Input(x, y, width, height, "label");
input->value("Now is the time for all good men...");
```

The string is copied to the widget's own storage when you set the `value()` of the widget.

The [Fl_Text_Display](#) and [Fl_Text_Editor](#) widgets use an associated [Fl_Text_Buffer](#) class for the value, instead of a simple string.

5.3 Valuers

Unlike text widgets, valuers keep track of numbers instead of strings. FLTK provides the following valuers:

- [Fl_Counter](#) - A widget with arrow buttons that shows the current value.
- [Fl_Dial](#) - A round knob.
- [Fl_Roller](#) - An SGI-like dolly widget.
- [Fl_Scrollbar](#) - A standard scrollbar widget.
- [Fl_Slider](#) - A scrollbar with a knob.
- [Fl_Value_Slider](#) - A slider that shows the current value.

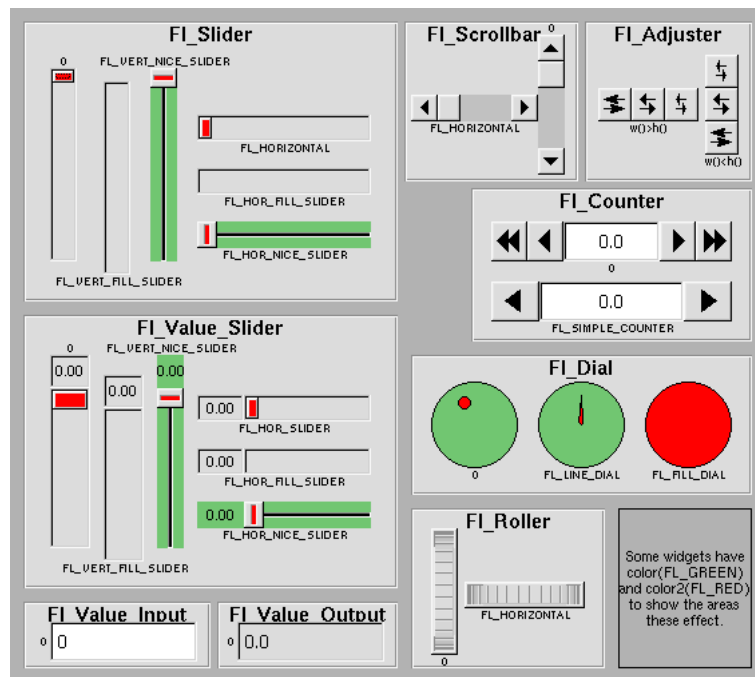


Figure 5.2 FLTK valuator widgets

The `value()` method gets and sets the current value of the widget. The `minimum()` and `maximum()` methods set the range of values that are reported by the widget.

5.4 Groups

The `FL_Group` widget class is used as a general purpose "container" widget. Besides grouping radio buttons, the groups are used to encapsulate windows, tabs, and scrolled windows. The following group classes are available with FLTK:

- `FL_Double_Window` - A double-buffered window on the screen.
- `FL_Gl_Window` - An OpenGL window on the screen.
- `FL_Group` - The base container class; can be used to group any widgets together.
- `FL_Pack` - A collection of widgets that are packed into the group area.
- `FL_Scroll` - A scrolled window area.
- `FL_Tabs` - Displays child widgets as tabs.
- `FL_Tile` - A tiled window area.
- `FL_Window` - A window on the screen.
- `FL_Wizard` - Displays one group of widgets at a time.

5.5 Setting the Size and Position of Widgets

The size and position of widgets is usually set when you create them. You can access them with the `x()`, `y()`, `w()`, and `h()` methods.

You can change the size and position by using the `position()`, `resize()`, and `size()` methods:

```
button->position(x, y);
group->resize(x, y, width, height);
window->size(width, height);
```

If you change a widget's size or position after it is displayed you will have to call `redraw()` on the widget's parent.

5.6 Colors

FLTK stores the colors of widgets as an 32-bit unsigned number that is either an index into a color palette of 256 colors or a 24-bit RGB color. The color palette is *not* the X or MS Windows colormap, but instead is an internal table with fixed contents.

See the [Colors](#) section of [Drawing Things in FLTK](#) for implementation details.

There are symbols for naming some of the more common colors:

- `FL_BLACK`
- `FL_RED`
- `FL_GREEN`
- `FL_YELLOW`
- `FL_BLUE`
- `FL_MAGENTA`
- `FL_CYAN`
- `FL_WHITE`
- `FL_WHITE`

Other symbols are used as the default colors for all FLTK widgets.

- `FL_FOREGROUND_COLOR`
- `FL_BACKGROUND_COLOR`
- `FL_INACTIVE_COLOR`
- `FL_SELECTION_COLOR`

The full list of named color values can be found in [FLTK Enumerations](#).

A color value can be created from its RGB components by using the `fl_rgb_color()` function, and decomposed again with `Fl::get_color()`:

```
Fl_Color c = fl_rgb_color(85, 170, 255); // RGB to Fl_Color
Fl::get_color(c, r, g, b); // Fl_Color to RGB
```

The widget color is set using the `color()` method:

```
button->color(FL_RED); // set color using named value
```

Similarly, the label color is set using the `labelcolor()` method:

```
button->labelcolor(FL_WHITE);
```

The `Fl_Color` encoding maps to a 32-bit unsigned integer representing RGBI, so it is also possible to specify a color using a hex constant as a color map index:

```
button->color(0x000000ff); // colormap index #255 (FL_WHITE)
```

or specify a color using a hex constant for the RGB components:

```
button->color(0xff000000); // RGB: red
button->color(0x00ff0000); // RGB: green
button->color(0x0000ff00); // RGB: blue
button->color(0xffffffff); // RGB: white
```

Note

If TrueColor is not available, any RGB colors will be set to the nearest entry in the colormap.

5.7 Box Types

The type `Fl_Boxtype` stored and returned in `Fl_Widget::box()` is an enumeration defined in [Enumerations.H](#).

Figure 3-3 shows the standard box types included with FLTK.

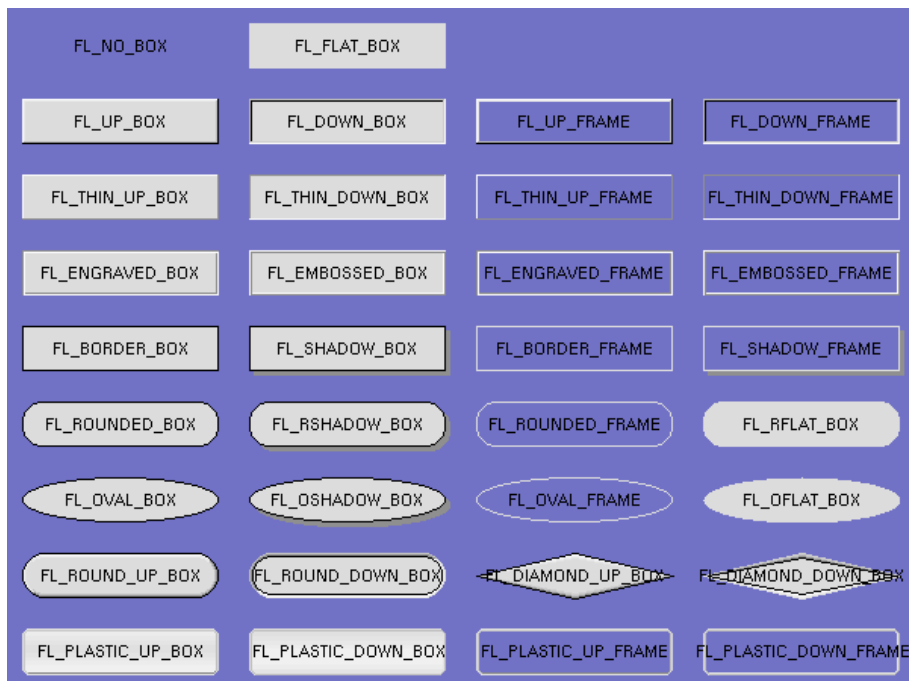


Figure 5.3 FLTK box types

`FL_NO_BOX` means nothing is drawn at all, so whatever is already on the screen remains. The `FL_..._FRAME` types only draw their edges, leaving the interior unchanged. The blue color in Figure 3-3 is the area that is not drawn by the frame types.

5.7.1 Making Your Own Boxtypes

You can define your own boxtypes by making a small function that draws the box and adding it to the table of boxtypes.

The Drawing Function

The drawing function is passed the bounding box and background color for the widget:

```
void xyz_draw(int x, int y, int w, int h, Fl_Color c) {
    ...
}
```

A simple drawing function might fill a rectangle with the given color and then draw a black outline:

```
void xyz_draw(int x, int y, int w, int h, Fl_Color c) {
    fl_color(c);
    fl_rectf(x, y, w, h);
    fl_color(FL_BLACK);
    fl_rect(x, y, w, h);
}
```

Fl_Boxtype fl_down(Fl_Boxtype b)

[fl_down\(\)](#) returns the "pressed" or "down" version of a box. If no "down" version of a given box exists, the behavior of this function is undefined and some random box or frame is returned. See [Drawing Functions](#) for more details.

Fl_Boxtype fl_frame(Fl_Boxtype b)

[fl_frame\(\)](#) returns the unfilled, frame-only version of a box. If no frame version of a given box exists, the behavior of this function is undefined and some random box or frame is returned. See [Drawing Functions](#) for more details.

Fl_Boxtype fl_box(Fl_Boxtype b)

[fl_box\(\)](#) returns the filled version of a frame. If no filled version of a given frame exists, the behavior of this function is undefined and some random box or frame is returned. See [Drawing Functions](#) for more details.

Adding Your Box Type

The [Fl::set_boxtype\(\)](#) method adds or replaces the specified box type:

```
#define XYZ_BOX FL_FREE_BOXTYPE
Fl::set_boxtype(XYZ_BOX, xyz_draw, 1, 1, 2, 2);
```

The last 4 arguments to [Fl::set_boxtype\(\)](#) are the offsets for the `x`, `y`, `width`, and `height` values that should be subtracted when drawing the label inside the box.

A complete box design contains four box types in this order: a filled, neutral box (`UP_BOX`), a filled, depressed box (`DOWN_BOX`), and the same as outlines only (`UP_FRAME` and `DOWN_FRAME`). The function [fl_down\(Fl_Boxtype\)](#) expects the neutral design on a boxtype with a numerical value evenly dividable by two. [fl_frame\(Fl_Boxtype\)](#) expects the `UP_BOX` design at a value dividable by four.

5.8 Labels and Label Types

The `label()`, `align()`, `labelfont()`, `labelsize()`, `labeltype()`, `image()`, and `deimage()` methods control the labeling of widgets.

`label()`

The `label()` method sets the string that is displayed for the label. Symbols can be included with the label string by escaping them using the "@" symbol - "@@" displays a single at sign. Figure 3-4 shows the available symbols.



Figure 5.4 FLTK label symbols

The @ sign may also be followed by the following optional "formatting" characters, in this order:

- '#' forces square scaling, rather than distortion to the widget's shape.
- '+[1-9] or -[1-9] tweaks the scaling a little bigger or smaller.
- '\$' flips the symbol horizontally, '%' flips it vertically.
- '[0-9] - rotates by a multiple of 45 degrees. '5' and '6' do no rotation while the others point in the direction of that key on a numeric keypad. '0', followed by four more digits rotates the symbol by that amount in degrees.

Thus, to show a very large arrow pointing downward you would use the label string "@+92->".

align()

The `align()` method positions the label. The following constants are defined and may be OR'd together as needed:

- `FL_ALIGN_CENTER` - center the label in the widget.
- `FL_ALIGN_TOP` - align the label at the top of the widget.
- `FL_ALIGN_BOTTOM` - align the label at the bottom of the widget.
- `FL_ALIGN_LEFT` - align the label to the left of the widget.
- `FL_ALIGN_RIGHT` - align the label to the right of the widget.
- `FL_ALIGN_LEFT_TOP` - The label appears to the left of the widget, aligned at the top. Outside labels only.
- `FL_ALIGN_RIGHT_TOP` - The label appears to the right of the widget, aligned at the top. Outside labels only.
- `FL_ALIGN_LEFT_BOTTOM` - The label appears to the left of the widget, aligned at the bottom. Outside labels only.
- `FL_ALIGN_RIGHT_BOTTOM` - The label appears to the right of the widget, aligned at the bottom. Outside labels only.
- `FL_ALIGN_INSIDE` - align the label inside the widget.
- `FL_ALIGN_CLIP` - clip the label to the widget's bounding box.
- `FL_ALIGN_WRAP` - wrap the label text as needed.
- `FL_ALIGN_TEXT_OVER_IMAGE` - show the label text over the image.
- `FL_ALIGN_IMAGE_OVER_TEXT` - show the label image over the text (default).
- `FL_ALIGN_IMAGE_NEXT_TO_TEXT` - The image will appear to the left of the text.
- `FL_ALIGN_TEXT_NEXT_TO_IMAGE` - The image will appear to the right of the text.
- `FL_ALIGN_IMAGE_BACKDROP` - The image will be used as a background for the widget.

labeltype()

The `labeltype()` method sets the type of the label. The following standard label types are included:

- `FL_NORMAL_LABEL` - draws the text.
- `FL_NO_LABEL` - does nothing.
- `FL_SHADOW_LABEL` - draws a drop shadow under the text.
- `FL_ENGRAVED_LABEL` - draws edges as though the text is engraved.
- `FL_EMBOSSED_LABEL` - draws edges as though the text is raised.
- `FL_ICON_LABEL` - draws the icon associated with the text.

image() and deimage()

The `image()` and `deimage()` methods set an image that will be displayed with the widget. The `deimage()` method sets the image that is shown when the widget is inactive, while the `image()` method sets the image that is shown when the widget is active.

To make an image you use a subclass of `Fl_Image`.

Making Your Own Label Types

Label types are actually indexes into a table of functions that draw them. The primary purpose of this is to use this to draw the labels in ways inaccessible through the `fl_font()` mechanism (e.g. `FL_ENGRAVED_LABEL`) or with program-generated letters or symbology.

Label Type Functions

To setup your own label type you will need to write two functions: one to draw and one to measure the label. The draw function is called with a pointer to a `Fl_Label` structure containing the label information, the bounding box for the label, and the label alignment:

```
void xyz_draw(const Fl_Label *label, int x, int y, int w, int h, Fl_Align align) {
    ...
}
```

The label should be drawn *inside* this bounding box, even if `FL_ALIGN_INSIDE` is not enabled. The function is not called if the label value is `NULL`.

The measure function is called with a pointer to a `Fl_Label` structure and references to the width and height:

```
void xyz_measure(const Fl_Label *label, int &w, int &h) {
    ...
}
```

The function should measure the size of the label and set `w` and `h` to the size it will occupy.

Adding Your Label Type

The `Fl::set_labeltype()` method creates a label type using your draw and measure functions:

```
#define XYZ_LABEL FL_FREE_LABELTYPE
Fl::set_labeltype(XYZ_LABEL, xyz_draw, xyz_measure);
```

The label type number `n` can be any integer value starting at the constant `FL_FREE_LABELTYPE`. Once you have added the label type you can use the `labeltype()` method to select your label type.

The `Fl::set_labeltype()` method can also be used to overload an existing label type such as `FL_NORMAL_LABEL`.

Making your own symbols

It is also possible to define your own drawings and add them to the symbol list, so they can be rendered as part of any label.

To create a new symbol, you implement a drawing function `void drawit(Fl_Color c)` which typically uses the functions described in [Drawing Complex Shapes](#) to generate a vector shape inside a two-by-two units sized box around the origin. This function is then linked into the symbols table using `fl_add_symbol()`:

```
int fl_add_symbol(const char *name, void (*drawit)(Fl_Color), int scalable)
```

`name` is the name of the symbol without the "@"; `scalable` must be set to 1 if the symbol is generated using scalable vector drawing functions.

```
int fl_draw_symbol(const char *name, int x, int y, int w, int h, Fl_Color col)
```

This function draws a named symbol fitting the given rectangle.

5.9 Callbacks

Callbacks are functions that are called when the value of a widget changes. A callback function is sent a `Fl_Widget` pointer of the widget that changed and a pointer to data that you provide:

```
void xyz_callback(Fl_Widget *w, void *data) {
    ...
}
```

The `callback()` method sets the callback function for a widget. You can optionally pass a pointer to some data needed for the callback:

```
int xyz_data;
button->callback(xyz_callback, &xyz_data);
```

Normally callbacks are performed only when the value of the widget changes. You can change this using the `Fl_Widget::when()` method:

```
button->when(FL_WHEN_NEVER);
button->when(FL_WHEN_CHANGED);
button->when(FL_WHEN_RELEASE);
button->when(FL_WHEN_RELEASE_ALWAYS);
button->when(FL_WHEN_ENTER_KEY);
button->when(FL_WHEN_ENTER_KEY_ALWAYS);
button->when(FL_WHEN_CHANGED | FL_WHEN_NOT_CHANGED);
```

Note:

You cannot delete a widget inside a callback, as the widget may still be accessed by FLTK after your callback is completed. Instead, use the `Fl::delete_widget()` method to mark your widget for deletion when it is safe to do so.

Hint:

Many programmers new to FLTK or C++ try to use a non-static class method instead of a static class method or function for their callback. Since callbacks are done outside a C++ class, the `this` pointer is not initialized for class methods.

To work around this problem, define a static method in your class that accepts a pointer to the class, and then have the static method call the class method(s) as needed. The data pointer you provide to the `callback()` method of the widget can be a pointer to the instance of your class.

```
class Foo {
    void my_callback(Fl_Widget *w);
    static void my_static_callback(Fl_Widget *w, void *f) { ((Foo *)f)->my_callback(w); }
    ...
}
...
w->callback(my_static_callback, (void *)this);
```

5.10 Shortcuts

Shortcuts are key sequences that activate widgets such as buttons or menu items. The `shortcut()` method sets the shortcut for a widget:

```
button->shortcut(FL_Enter);
button->shortcut(FL_SHIFT + 'b');
button->shortcut(FL_CTRL + 'b');
button->shortcut(FL_ALT + 'b');
button->shortcut(FL_CTRL + FL_ALT + 'b');
button->shortcut(0); // no shortcut
```

The shortcut value is the key event value - the ASCII value or one of the special keys described in `Fl::event_key() Values` combined with any modifiers like `Shift`, `Alt`, and `Control`.

Chapter 6

Designing a Simple Text Editor

This chapter takes you through the design of a simple FLTK-based text editor.

6.1 Determining the Goals of the Text Editor

Since this will be the first big project you'll be doing with FLTK, let's define what we want our text editor to do:

1. Provide a menubar/menus for all functions.
2. Edit a single text file, possibly with multiple views.
3. Load from a file.
4. Save to a file.
5. Cut/copy/delete/paste functions.
6. Search and replace functions.
7. Keep track of when the file has been changed.

6.2 Designing the Main Window

Now that we've outlined the goals for our editor, we can begin with the design of our GUI. Obviously the first thing that we need is a window, which we'll place inside a class called `EditorWindow`:

```
class EditorWindow : public Fl_Double_Window {
public:
    EditorWindow(int w, int h, const char* t);
    ~EditorWindow();
    Fl_Window      *replace_dlg;
    Fl_Input       *replace_find;
    Fl_Input       *replace_with;
    Fl_Button      *replace_all;
    Fl_Return_Button *replace_next;
    Fl_Button      *replace_cancel;
    Fl_Text_Editor *editor;
    char           search[256];
};
```

6.3 Variables

Our text editor will need some global variables to keep track of things:

```
int          changed = 0;
char        filename[256] = "";
Fl_Text_Buffer *textbuf;
```

The `textbuf` variable is the text editor buffer for our window class described previously. We'll cover the other variables as we build the application.

6.4 Menubars and Menus

The first goal requires us to use a menubar and menus that define each function the editor needs to perform. The `Fl_Menu_Item` structure is used to define the menus and items in a menubar:

```
Fl_Menu_Item menuitems[] = {
  { "&File", 0, 0, 0, FL_SUBMENU },
  { "&New File", 0, (Fl_Callback *)new_cb },
  { "&Open File...", FL_COMMAND + 'o', (Fl_Callback *)open_cb },
  { "&Insert File...", FL_COMMAND + 'i', (Fl_Callback *)insert_cb, 0, FL_MENU_DIVIDER },
  { "&Save File", FL_COMMAND + 's', (Fl_Callback *)save_cb },
  { "Save File &As...", FL_COMMAND + FL_SHIFT + 's', (Fl_Callback *)saveas_cb, 0, FL_MENU_DIVIDER },
  { "New &View", FL_ALT + 'v', (Fl_Callback *)view_cb, 0 },
  { "&Close View", FL_COMMAND + 'w', (Fl_Callback *)close_cb, 0, FL_MENU_DIVIDER },
  { "E&xit", FL_COMMAND + 'q', (Fl_Callback *)quit_cb, 0 },
  { 0 },
  { "&Edit", 0, 0, 0, FL_SUBMENU },
  { "&Undo", FL_COMMAND + 'z', (Fl_Callback *)undo_cb, 0, FL_MENU_DIVIDER },
  { "Cu&t", FL_COMMAND + 'x', (Fl_Callback *)cut_cb },
  { "&Copy", FL_COMMAND + 'c', (Fl_Callback *)copy_cb },
  { "&Paste", FL_COMMAND + 'v', (Fl_Callback *)paste_cb },
  { "&Delete", 0, (Fl_Callback *)delete_cb },
  { 0 },
  { "&Search", 0, 0, 0, FL_SUBMENU },
  { "&Find...", FL_COMMAND + 'f', (Fl_Callback *)find_cb },
  { "F&ind Again", FL_COMMAND + 'g', find2_cb },
  { "&Replace...", FL_COMMAND + 'r', replace_cb },
  { "Re&place Again", FL_COMMAND + 't', replace2_cb },
  { 0 },
  { 0 }
};
```

Once we have the menus defined we can create the `Fl_Menu_Bar` widget and assign the menus to it with:

```
Fl_Menu_Bar *m = new Fl_Menu_Bar(0, 0, 640, 30);
m->copy(menuitems);
```

We'll define the callback functions later.

6.5 Editing the Text

To keep things simple our text editor will use the `Fl_Text_Editor` widget to edit the text:

```
w->editor = new Fl_Text_Editor(0, 30, 640, 370);
w->editor->buffer(textbuf);
```

So that we can keep track of changes to the file, we also want to add a "modify" callback:

```
textbuf->add_modify_callback(changed_cb, w);
textbuf->call_modify_callbacks();
```

Finally, we want to use a mono-spaced font like `FL_COURIER`:

```
w->editor->textfont(FL_COURIER);
```

6.6 The Replace Dialog

We can use the FLTK convenience functions for many of the editor's dialogs, however the replace dialog needs its own custom window. To keep things simple we will have a "find" string, a "replace" string, and "replace all", "replace next", and "cancel" buttons. The strings are just `Fl_Input` widgets, the "replace all" and "cancel" buttons are `Fl_Button` widgets, and the "replace next" button is a `Fl_Return_Button` widget:

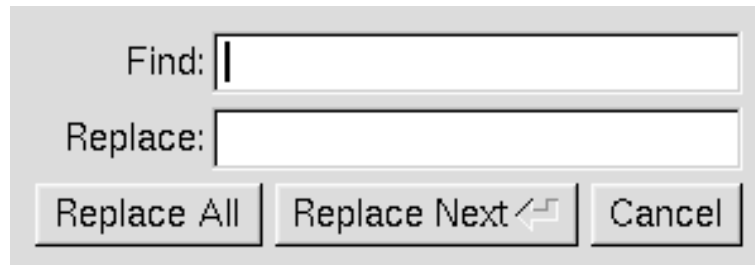


Figure 6.1 The search and replace dialog

```
Fl_Window *replace_dlg = new Fl_Window(300, 105, "Replace");
Fl_Input *replace_find = new Fl_Input(70, 10, 200, 25, "Find:");
Fl_Input *replace_with = new Fl_Input(70, 40, 200, 25, "Replace:");
Fl_Button *replace_all = new Fl_Button(10, 70, 90, 25, "Replace All");
Fl_Button *replace_next = new Fl_Button(105, 70, 120, 25, "Replace Next");
Fl_Button *replace_cancel = new Fl_Button(230, 70, 60, 25, "Cancel");
```

6.7 Callbacks

Now that we've defined the GUI components of our editor, we need to define our callback functions.

6.7.1 changed_cb()

This function will be called whenever the user changes any text in the `editor` widget:

```
void changed_cb(int, int nInserted, int nDeleted, int, const char*, void* v) {
    if ((nInserted || nDeleted) && !loading) changed = 1;
    EditorWindow *w = (EditorWindow *)v;
    set_title(w);
    if (loading) w->editor->show_insert_position();
}
```

The `set_title()` function is one that we will write to set the changed status on the current file. We're doing it this way because we want to show the changed status in the window's title bar.

6.7.2 copy_cb()

This callback function will call `Fl_Text_Editor::kf_copy()` to copy the currently selected text to the clipboard:

```
void copy_cb(Fl_Widget*, void* v) {
    EditorWindow* e = (EditorWindow*)v;
    Fl_Text_Editor::kf_copy(0, e->editor);
}
```

6.7.3 cut_cb()

This callback function will call `Fl_Text_Editor::kf_cut()` to cut the currently selected text to the clipboard:

```
void cut_cb(Fl_Widget*, void* v) {
    EditorWindow* e = (EditorWindow*)v;
    Fl_Text_Editor::kf_cut(0, e->editor);
}
```

6.7.4 delete_cb()

This callback function will call `FL_Text_Buffer::remove_selection()` to delete the currently selected text to the clipboard:

```
void delete_cb(FL_Widget*, void* v) {
    textbuf->remove_selection();
}
```

6.7.5 find_cb()

This callback function asks for a search string using the `fl_input()` convenience function and then calls the `find2←_cb()` function to find the string:

```
void find_cb(FL_Widget* w, void* v) {
    EditorWindow* e = (EditorWindow*)v;
    const char *val;
    val = fl_input("Search String:", e->search);
    if (val != NULL) {
        // User entered a string - go find it!
        strcpy(e->search, val);
        find2_cb(w, v);
    }
}
```

6.7.6 find2_cb()

This function will find the next occurrence of the search string. If the search string is blank then we want to pop up the search dialog:

```
void find2_cb(FL_Widget* w, void* v) {
    EditorWindow* e = (EditorWindow*)v;
    if (e->search[0] == '\0') {
        // Search string is blank; get a new one...
        find_cb(w, v);
        return;
    }
    int pos = e->editor->insert_position();
    int found = textbuf->search_forward(pos, e->search, &pos);
    if (found) {
        // Found a match; select and update the position...
        textbuf->select(pos, pos+strlen(e->search));
        e->editor->insert_position(pos+strlen(e->search));
        e->editor->show_insert_position();
    }
    else fl_alert("No occurrences of \'%s\' found!", e->search);
}
```

If the search string cannot be found we use the `fl_alert()` convenience function to display a message to that effect.

6.7.7 new_cb()

This callback function will clear the editor widget and current filename. It also calls the `check_save()` function to give the user the opportunity to save the current file first as needed:

```
void new_cb(FL_Widget*, void*) {
    if (!check_save()) return;
    filename[0] = '\0';
    textbuf->select(0, textbuf->length());
    textbuf->remove_selection();
    changed = 0;
    textbuf->call_modify_callbacks();
}
```


6.7.8 open_cb()

This callback function will ask the user for a filename and then load the specified file into the input widget and current filename. It also calls the `check_save()` function to give the user the opportunity to save the current file first as needed:

```
void open_cb(Fl_Widget*, void*) {
    if (!check_save()) return;
    char *newfile = fl_file_chooser("Open File?", "*", filename);
    if (newfile != NULL) load_file(newfile, -1);
}
```

We call the `load_file()` function to actually load the file.

6.7.9 paste_cb()

This callback function will call `Fl_Text_Editor::kf_paste()` to paste the clipboard at the current position:

```
void paste_cb(Fl_Widget*, void* v) {
    EditorWindow* e = (EditorWindow*)v;
    Fl_Text_Editor::kf_paste(0, e->editor);
}
```

6.7.10 quit_cb()

The quit callback will first see if the current file has been modified, and if so give the user a chance to save it. It then exits from the program:

```
void quit_cb(Fl_Widget*, void*) {
    if (changed && !check_save())
        return;
    exit(0);
}
```

6.7.11 replace_cb()

The replace callback just shows the replace dialog:

```
void replace_cb(Fl_Widget*, void* v) {
    EditorWindow* e = (EditorWindow*)v;
    e->replace_dlg->show();
}
```

6.7.12 replace2_cb()

This callback will replace the next occurrence of the replacement string. If nothing has been entered for the replacement string, then the replace dialog is displayed instead:

```
void replace2_cb(Fl_Widget*, void* v) {
    EditorWindow* e = (EditorWindow*)v;
    const char *find = e->replace_find->value();
    const char *replace = e->replace_with->value();
    if (find[0] == '\0') {
        // Search string is blank; get a new one...
        e->replace_dlg->show();
        return;
    }
    e->replace_dlg->hide();
    int pos = e->editor->insert_position();
    int found = textbuf->search_forward(pos, find, &pos);
    if (found) {
        // Found a match; update the position and replace text...
        textbuf->select(pos, pos+strlen(find));
        textbuf->remove_selection();
        textbuf->insert(pos, replace);
        textbuf->select(pos, pos+strlen(replace));
        e->editor->insert_position(pos+strlen(replace));
        e->editor->show_insert_position();
    }
    else fl_alert("No occurrences of \'%s\' found!", find);
}
```

6.7.13 replall_cb()

This callback will replace all occurrences of the search string in the file:

```
void replall_cb(Fl_Widget*, void* v) {
    EditorWindow* e = (EditorWindow*)v;
    const char *find = e->replace_find->value();
    const char *replace = e->replace_with->value();
    find = e->replace_find->value();
    if (find[0] == '\\0') {
        // Search string is blank; get a new one...
        e->replace_dlg->show();
        return;
    }
    e->replace_dlg->hide();
    e->editor->insert_position(0);
    int times = 0;
    // Loop through the whole string
    for (int found = 1; found;) {
        int pos = e->editor->insert_position();
        found = textbuf->search_forward(pos, find, &pos);
        if (found) {
            // Found a match; update the position and replace text...
            textbuf->select(pos, pos+strlen(find));
            textbuf->remove_selection();
            textbuf->insert(pos, replace);
            e->editor->insert_position(pos+strlen(replace));
            e->editor->show_insert_position();
            times++;
        }
    }
    if (times) fl_message("Replaced %d occurrences.", times);
    else fl_alert("No occurrences of '\\%s\\' found!", find);
}
```

6.7.14 replcan_cb()

This callback just hides the replace dialog:

```
void replcan_cb(Fl_Widget*, void* v) {
    EditorWindow* e = (EditorWindow*)v;
    e->replace_dlg->hide();
}
```

6.7.15 save_cb()

This callback saves the current file. If the current filename is blank it calls the "save as" callback:

```
void save_cb(void) {
    if (filename[0] == '\\0') {
        // No filename - get one!
        saveas_cb();
        return;
    }
    else save_file(filename);
}
```

The `save_file()` function saves the current file to the specified filename.

6.7.16 saveas_cb()

This callback asks the user for a filename and saves the current file:

```
void saveas_cb(void) {
    char *newfile;
    newfile = fl_file_chooser("Save File As?", "*", filename);
    if (newfile != NULL) save_file(newfile);
}
```

The `save_file()` function saves the current file to the specified filename.

6.8 Other Functions

Now that we've defined the callback functions, we need our support functions to make it all work:

6.8.1 check_save()

This function checks to see if the current file needs to be saved. If so, it asks the user if they want to save it:

```
int check_save(void) {
    if (!changed) return 1;
    int r = fl_choice("The current file has not been saved.\n"
                    "Would you like to save it now?",
                    "Cancel", "Save", "Discard");

    if (r == 1) {
        save_cb(); // Save the file...
        return !changed;
    }
    return (r == 2) ? 1 : 0;
}
```

6.8.2 load_file()

This function loads the specified file into the `textbuf` variable:

```
int loading = 0;
void load_file(char *newfile, int ipos) {
    loading = 1;
    int insert = (ipos != -1);
    changed = insert;
    if (!insert) strcpy(filename, "");
    int r;
    if (!insert) r = textbuf->loadfile(newfile);
    else r = textbuf->insertfile(newfile, ipos);
    if (r)
        fl_alert("Error reading from file '%s':\n%s.", newfile, strerror(errno));
    else
        if (!insert) strcpy(filename, newfile);
    loading = 0;
    textbuf->call_modify_callbacks();
}
```

When loading the file we use the `Fl_Text_Buffer::loadfile()` method to "replace" the text in the buffer, or the `Fl_Text_Buffer::insertfile()` method to insert text in the buffer from the named file.

6.8.3 save_file()

This function saves the current buffer to the specified file:

```
void save_file(char *newfile) {
    if (textbuf->savefile(newfile))
        fl_alert("Error writing to file '%s':\n%s.", newfile, strerror(errno));
    else
        strcpy(filename, newfile);
    changed = 0;
    textbuf->call_modify_callbacks();
}
```

6.8.4 set_title()

This function checks the `changed` variable and updates the window label accordingly:

```
void set_title(Fl_Window* w) {
    if (filename[0] == '\0') strcpy(title, "Untitled");
    else {
        char *slash;
        slash = strrchr(filename, '/');
#ifdef WIN32
        if (slash == NULL) slash = strrchr(filename, '\\');
#endif
        if (slash != NULL) strcpy(title, slash + 1);
        else strcpy(title, filename);
    }
    if (changed) strcat(title, " (modified)");
    w->label(title);
}
```

6.9 The main() Function

Once we've created all of the support functions, the only thing left is to tie them all together with the `main()` function. The `main()` function creates a new text buffer, creates a new view (window) for the text, shows the window, loads the file on the command-line (if any), and then enters the FLTK event loop:

```
int main(int argc, char **argv) {
    textbuf = new Fl_Text_Buffer;
    Fl_Window* window = new_view();
    window->show(1, argv);
    if (argc > 1) load_file(argv[1], -1);
    return Fl::run();
}
```

6.10 Compiling the Editor

The complete source for our text editor can be found in the `test/editor.cxx` source file. Both the Makefile and Visual C++ workspace include the necessary rules to build the editor. You can also compile it using a standard compiler with:

```
CC -o editor editor.cxx -lfltk -lXext -lX11 -lm
```

or by using the `fltk-config` script with:

```
fltk-config --compile editor.cxx
```

As noted in [Compiling Programs with Standard Compilers](#), you may need to include compiler and linker options to tell them where to find the FLTK library. Also, the `CC` command may also be called `gcc` or `c++` on your system.

Congratulations, you've just built your own text editor!

6.11 The Final Product

The final editor window should look like the image in Figure 4-2.

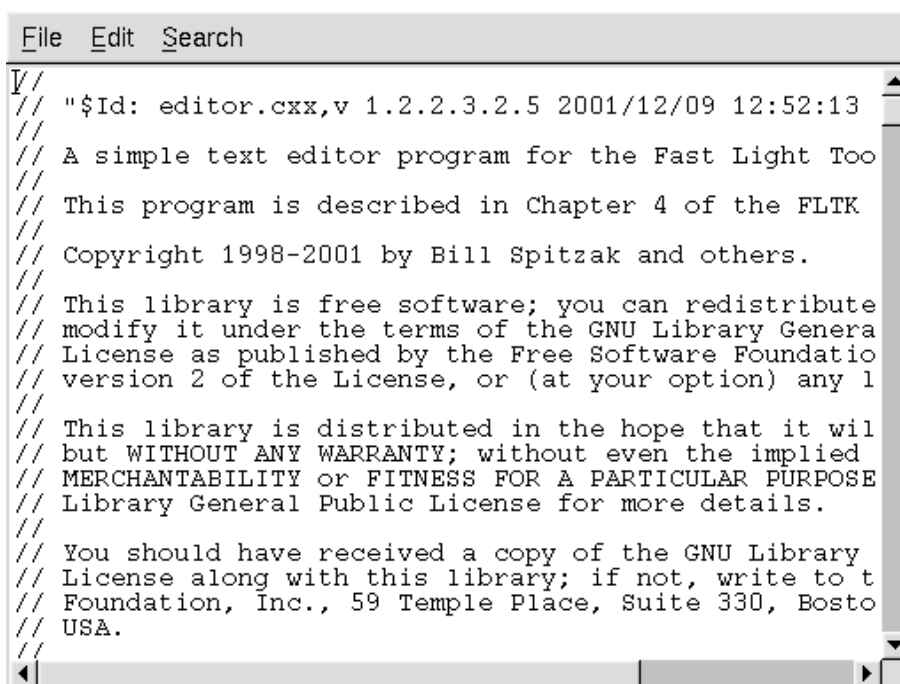


Figure 6.2 The completed editor window

6.12 Advanced Features

Now that we've implemented the basic functionality, it is time to show off some of the advanced features of the `Fl_Text_Editor` widget.

6.12.1 Syntax Highlighting

The `Fl_Text_Editor` widget supports highlighting of text with different fonts, colors, and sizes. The implementation is based on the excellent `NEdit` text editor core, from <http://www.nedit.org/>, which uses a parallel "style" buffer which tracks the font, color, and size of the text that is drawn.

Styles are defined using the `Fl_Text_Display::Style_Table_Entry` structure defined in `<FL/Fl_Text_Display.H>`←

```
:
struct Style_Table_Entry {
    Fl_Color color;
    Fl_Font font;
    int size;
    unsigned attr;
};
```

The `color` member sets the color for the text, the `font` member sets the FLTK font index to use, and the `size` member sets the pixel size of the text. The `attr` member is currently not used.

For our text editor we'll define 7 styles for plain code, comments, keywords, and preprocessor directives:

```
Fl_Text_Display::Style_Table_Entry styletable[] = { // Style table
    { FL_BLACK, FL_COURIER, FL_NORMAL_SIZE }, // A - Plain
    { FL_DARK_GREEN, FL_COURIER_ITALIC, FL_NORMAL_SIZE }, // B - Line comments
    { FL_DARK_GREEN, FL_COURIER_ITALIC, FL_NORMAL_SIZE }, // C - Block comments
    { FL_BLUE, FL_COURIER, FL_NORMAL_SIZE }, // D - Strings
    { FL_DARK_RED, FL_COURIER, FL_NORMAL_SIZE }, // E - Directives
    { FL_DARK_RED, FL_COURIER_BOLD, FL_NORMAL_SIZE }, // F - Types
    { FL_BLUE, FL_COURIER_BOLD, FL_NORMAL_SIZE } // G - Keywords
};
```

You'll notice that the comments show a letter next to each style - each style in the style buffer is referenced using a character starting with the letter 'A'.

You call the `highlight_data()` method to associate the style data and buffer with the text editor widget:

```
Fl_Text_Buffer *stylebuf;
w->editor->highlight_data(stylebuf, styletable,
    sizeof(styletable) / sizeof(styletable[0]),
    'A', style_unfinished_cb, 0);
```

Finally, you need to add a callback to the main text buffer so that changes to the text buffer are mirrored in the style buffer:

```
textbuf->add_modify_callback(style_update, w->editor);
```

The `style_update()` function, like the `change_cb()` function described earlier, is called whenever text is added or removed from the text buffer. It mirrors the changes in the style buffer and then updates the style data as necessary:

```
//
// 'style_update()' - Update the style buffer...
//
void
style_update(int pos, // I - Position of update
             int nInserted, // I - Number of inserted chars
             int nDeleted, // I - Number of deleted chars
             int nRestyled, // I - Number of restyled chars
             const char *deletedText, // I - Text that was deleted
             void *cbArg) { // I - Callback data
    int start, // Start of text
        end; // End of text
    char last, // Last style on line
        *style, // Style data
        *text; // Text data
    // If this is just a selection change, just unselect the style buffer...
    if (nInserted == 0 && nDeleted == 0) {
        stylebuf->unselect();
        return;
    }
```

```

}
// Track changes in the text buffer...
if (nInserted > 0) {
    // Insert characters into the style buffer...
    style = new char[nInserted + 1];
    memset(style, 'A', nInserted);
    style[nInserted] = '\0';
    stylebuf->replace(pos, pos + nDeleted, style);
    delete[] style;
} else {
    // Just delete characters in the style buffer...
    stylebuf->remove(pos, pos + nDeleted);
}
// Select the area that was just updated to avoid unnecessary
// callbacks...
stylebuf->select(pos, pos + nInserted - nDeleted);
// Re-parse the changed region; we do this by parsing from the
// beginning of the line of the changed region to the end of
// the line of the changed region... Then we check the last
// style character and keep updating if we have a multi-line
// comment character...
start = textbuf->line_start(pos);
end = textbuf->line_end(pos + nInserted - nDeleted);
text = textbuf->text_range(start, end);
style = stylebuf->text_range(start, end);
last = style[end - start - 1];
style_parse(text, style, end - start);
stylebuf->replace(start, end, style);
((Fl_Text_Editor *)cbArg)->redisplay_range(start, end);
if (last != style[end - start - 1]) {
    // The last character on the line changed styles, so reparse the
    // remainder of the buffer...
    free(text);
    free(style);
    end = textbuf->length();
    text = textbuf->text_range(start, end);
    style = stylebuf->text_range(start, end);
    style_parse(text, style, end - start);
    stylebuf->replace(start, end, style);
    ((Fl_Text_Editor *)cbArg)->redisplay_range(start, end);
}
free(text);
free(style);
}

```

The `style_parse()` function scans a copy of the text in the buffer and generates the necessary style characters for display. It assumes that parsing begins at the start of a line:

```

//
// 'style_parse()' - Parse text and produce style data.
//
void
style_parse(const char *text,
            char *style,
            int length) {
    char current;
    int col;
    int last;
    char buf[255],
        *bufptr;
    const char *temp;
    for (current = *style, col = 0, last = 0; length > 0; length --, text ++ ) {
        if (current == 'A') {
            // Check for directives, comments, strings, and keywords...
            if (col == 0 && *text == '#') {
                // Set style to directive
                current = 'E';
            } else if (strncmp(text, "//", 2) == 0) {
                current = 'B';
            } else if (strncmp(text, "/*", 2) == 0) {
                current = 'C';
            } else if (strncmp(text, "\\\"", 2) == 0) {
                // Quoted quote...
                *style++ = current;
                *style++ = current;
                text ++;
                length --;
                col += 2;
                continue;
            } else if (*text == '\\') {
                current = 'D';
            } else if (!last && islower(*text)) {
                // Might be a keyword...
                for (temp = text, bufptr = buf;
                     islower(*temp) && bufptr < (buf + sizeof(buf) - 1);
                     *bufptr++ = *temp++);
                if (!islower(*temp)) {

```

```

*bufptr = '\0';
bufptr = buf;
if (bsearch(&bufptr, code_types,
           sizeof(code_types) / sizeof(code_types[0]),
           sizeof(code_types[0]), compare_keywords)) {
    while (text < temp) {
        *style++ = 'F';
        text ++;
        length --;
        col ++;
    }
    text --;
    length ++;
    last = 1;
    continue;
} else if (bsearch(&bufptr, code_keywords,
                 sizeof(code_keywords) / sizeof(code_keywords[0]),
                 sizeof(code_keywords[0]), compare_keywords)) {
    while (text < temp) {
        *style++ = 'G';
        text ++;
        length --;
        col ++;
    }
    text --;
    length ++;
    last = 1;
    continue;
}
}
} else if (current == 'C' && strcmp(text, "*/", 2) == 0) {
    // Close a C comment...
    *style++ = current;
    *style++ = current;
    text ++;
    length --;
    current = 'A';
    col += 2;
    continue;
} else if (current == 'D') {
    // Continuing in string...
    if (strcmp(text, "\\\"", 2) == 0) {
        // Quoted end quote...
        *style++ = current;
        *style++ = current;
        text ++;
        length --;
        col += 2;
        continue;
    } else if (*text == '\\') {
        // End quote...
        *style++ = current;
        col ++;
        current = 'A';
        continue;
    }
}
// Copy style info...
if (current == 'A' && (*text == '{' || *text == '}')) *style++ = 'G';
else *style++ = current;
col ++;
last = isalnum(*text) || *text == '.';
if (*text == '\n') {
    // Reset column and possibly reset the style
    col = 0;
    if (current == 'B' || current == 'E') current = 'A';
}
}
}
}

```


Chapter 7

Drawing Things in FLTK

This chapter covers the drawing functions that are provided with FLTK.

7.1 When Can You Draw Things in FLTK?

There are only certain places you can execute FLTK code that draws to the computer's display. Calling these functions at other places will result in undefined behavior!

- The most common place is inside the virtual `Fl_Widget::draw()` method. To write code here, you must subclass one of the existing `Fl_Widget` classes and implement your own version of `draw()`.
- You can also create custom `boxtypes` and `labeltypes`. These involve writing small procedures that can be called by existing `Fl_Widget::draw()` methods. These "types" are identified by an 8-bit index that is stored in the widget's `box()`, `labeltype()`, and possibly other properties.
- You can call `Fl_Window::make_current()` to do incremental update of a widget. Use `Fl_Widget::window()` to find the window.

In contrast, code that draws to other drawing surfaces than the display (i.e., instances of derived classes of the `Fl_Surface_Device` class, except `Fl_Display_Device`, such as `Fl_Printer` and `Fl_Copy_Surface`) can be executed at any time as follows:

1. Memorize what is the current drawing surface calling `Fl_Surface_Device::surface()`, and make your surface the new current drawing surface calling the surface's `set_current()` function;
2. Make a series of calls to any of the drawing functions described below; these will operate on the new current drawing surface;
3. Set the current drawing surface back to its previous state calling the previous surface's `set_current()`.

7.1.1 What Drawing Unit do FLTK drawing functions use?

When drawing to the display or to instances of `Fl_Copy_Surface` and `Fl_Image_Surface`, the unit of drawing functions corresponds generally to one pixel. The so-called 'retina' displays of some recent Apple computers are an exception to this rule: one drawing unit corresponds to the width or the height of 2 display pixels on a retina display.

When drawing to surfaces that are instances of `Fl_Paged_Device` derived classes (i.e., `Fl_Printer` or `Fl_PostScript_File_Device`), the drawing unit is initially one point, that is, 1/72 of an inch. But this unit is changed after calls to `Fl_Paged_Device::scale()`.

7.2 Drawing Functions

To use the drawing functions you must first include the `<FL/fl_draw.H>` header file. FLTK provides the following types of drawing functions:

- [Boxes](#)
- [Clipping](#)
- [Colors](#)
- [Line Dashes and Thickness](#)
- [Drawing Fast Shapes](#)
- [Drawing Complex Shapes](#)
- [Drawing Text](#)
- [Fonts](#)
- [Character Encoding](#)
- [Drawing Overlays](#)
- [Drawing Images](#)
- [Direct Image Drawing](#)
- [Direct Image Reading](#)
- [Image Classes](#)
- [Offscreen Drawing](#)

7.2.1 Boxes

FLTK provides three functions that can be used to draw boxes for buttons and other UI controls. Each function uses the supplied upper-lefthand corner and width and height to determine where to draw the box.

```
void fl_draw_box(FL_Boxtype b, int x, int y, int w, int h, FL_Color c)
```

The `fl_draw_box()` function draws a standard boxtype `b` in the specified color `c`.

```
void fl_frame(const char *s, int x, int y, int w, int h)
void fl_frame2(const char *s, int x, int y, int w, int h)
```

The `fl_frame()` and `fl_frame2()` functions draw a series of line segments around the given box. The string `s` must contain groups of 4 letters which specify one of 24 standard grayscale values, where 'A' is black and 'X' is white. The results of calling these functions with a string that is not a multiple of 4 characters in length are undefined.

The only difference between `fl_frame()` and `fl_frame2()` is the order of the line segments:

- For `fl_frame()` the order of each set of 4 characters is: top, left, bottom, right.
- For `fl_frame2()` the order of each set of 4 characters is: bottom, right, top, left.

Note that `fl_frame(FL_Boxtype b)` is described in the [Box Types](#) section.

7.2.2 Clipping

You can limit all your drawing to a rectangular region by calling `fl_push_clip()`, and put the drawings back by using `fl_pop_clip()`. This rectangle is measured in pixels and is unaffected by the current transformation matrix.

In addition, the system may provide clipping when updating windows which may be more complex than a simple rectangle.

```
void fl_push_clip(int x, int y, int w, int h)
void fl_clip(int x, int y, int w, int h)
```

Intersect the current clip region with a rectangle and push this new region onto the stack.

The `fl_clip()` version is deprecated and will be removed from future releases.

```
void fl_push_no_clip()
```

Pushes an empty clip region on the stack so nothing will be clipped.

```
void fl_pop_clip()
```

Restore the previous clip region.

Note: You must call `fl_pop_clip()` once for every time you call `fl_push_clip()`. If you return to FLTK with the clip stack not empty unpredictable results occur.

```
int fl_not_clipped(int x, int y, int w, int h)
```

Returns non-zero if any of the rectangle intersects the current clip region. If this returns 0 you don't have to draw the object.

Note: Under X this returns 2 if the rectangle is partially clipped, and 1 if it is entirely inside the clip region.

```
int fl_clip_box(int x, int y, int w, int h, int &X, int &Y, int &W, int &H)
```

Intersect the rectangle `x, y, w, h` with the current clip region and returns the bounding box of the result in `X, Y, W, H`. Returns non-zero if the resulting rectangle is different than the original. This can be used to limit the necessary drawing to a rectangle. `W` and `H` are set to zero if the rectangle is completely outside the region.

```
void fl_clip_region(FL_Region r)
FL_Region fl_clip_region()
```

Replace the top of the clip stack with a clipping region of any shape. `FL_Region` is an operating system specific type. The second form returns the current clipping region.

7.3 Colors

FLTK manages colors as 32-bit unsigned integers, encoded as RGBA. When the "RGB" bytes are non-zero, the value is treated as RGB. If these bytes are zero, the "A" byte will be used as an index into the colormap. Colors with both "RGB" set and an "A" >0 are reserved for special use.

Values from 0 to 255, i.e. the "A" index value, represent colors from the FLTK 1.3.x standard colormap and are allocated as needed on screens without TrueColor support. The `Fl_Color` enumeration type defines the standard colors and color cube for the first 256 colors. All of these are named with symbols in `<FL/Enumerations.H>`. Example:

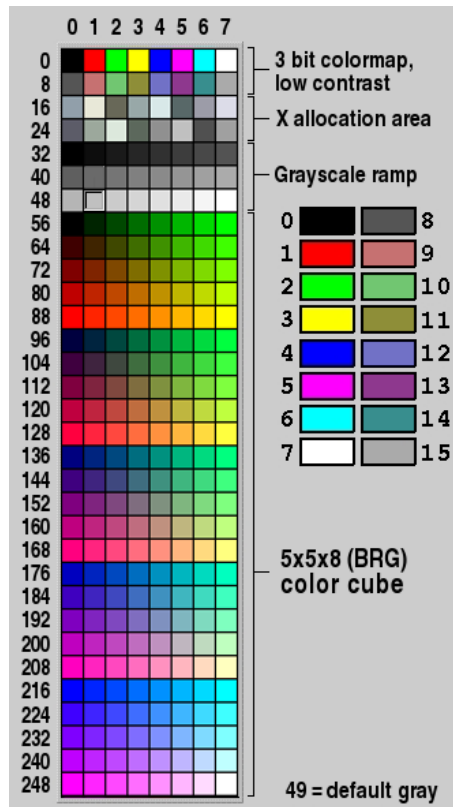


Figure 7.1 FLTK default colormap (`Fl_Color 0x00 - 0xff`)

Color values greater than 255 are treated as 24-bit RGB values. These are mapped to the closest color supported by the screen, either from one of the 256 colors in the FLTK 1.3.x colormap or a direct RGB value on TrueColor screens.

```
Fl_Color fl_rgb_color(uchar r, uchar g, uchar b)
Fl_Color fl_rgb_color(uchar grayscale)
```

Generate `Fl_Color` out of specified 8-bit RGB values or one 8-bit grayscale value.

```
void fl_color(Fl_Color c)
void fl_color(int c)
```

Sets the color for all subsequent drawing operations. Please use the first form: the second form is only provided for back compatibility.

For colormapped displays, a color cell will be allocated out of `fl_colormap` the first time you use a color. If the colormap fills up then a least-squares algorithm is used to find the closest color.

`FI_Color fl_color()`

Returns the last color that was set using `fl_color()`. This can be used for state save/restore.

`void fl_color(uchar r, uchar g, uchar b)`

Set the color for all subsequent drawing operations. The closest possible match to the RGB color is used. The RGB color is used directly on TrueColor displays. For colormap visuals the nearest index in the gray ramp or color cube is used.

`unsigned FI::get_color(FI_Color i)`

`void FI::get_color(FI_Color i, uchar &red, uchar &green, uchar &blue)`

Generate RGB values from a colormap index value `i`. The first returns the RGB as a 32-bit unsigned integer, and the second decomposes the RGB into three 8-bit values.

`FI::get_system_colors()`

`FI::foreground()`

`FI::background()`

`FI::background2()`

The first gets color values from the user preferences or the system, and the other routines are used to apply those values.

`FI::own_colormap()`

`FI::free_color(FI_Color i, int overlay)`

`FI::set_color(FI_Color i, unsigned c)`

`FI::own_colormap()` is used to install a local colormap [X11 only].

`FI::free_color()` and `FI::set_color()` are used to remove and replace entries from the colormap.

There are two predefined graphical interfaces for choosing colors. The function `fl_show_colormap()` shows a table of colors and returns an `FI_Color` index value. The `FI_Color_Chooser` widget provides a standard RGB color chooser.

As the `FI_Color` encoding maps to a 32-bit unsigned integer representing RGBI, it is also possible to specify a color using a hex constant as a color map index:

```
// COLOR MAP INDEX
color(0x000000II)
    ----- |
    | |
    | | Color map index (8 bits)
    | | Must be zero

button->color(0x000000ff); // colormap index #255 (FL_WHITE)
```

or specify a color using a hex constant for the RGB components:

```
// RGB COLOR ASSIGNMENTS
color(0xRRGGBB00)
    | | | |
    | | | Must be zero
    | | Blue (8 bits)
    | Green (8 bits)
    Red (8 bits)

button->color(0xff000000); // RGB: red
button->color(0x00ff0000); // RGB: green
button->color(0x0000ff00); // RGB: blue
button->color(0xffffff00); // RGB: white
```

Note

If TrueColor is not available, any RGB colors will be set to the nearest entry in the colormap.

7.3.1 Line Dashes and Thickness

FLTK supports drawing of lines with different styles and widths. Full functionality is not available under Windows 95, 98, and Me due to the reduced drawing functionality these operating systems provide.

void [fl_line_style](#)(int style, int width, char* dashes)

Set how to draw lines (the "pen"). If you change this it is your responsibility to set it back to the default with `fl_line_style(0)`.

Note: Because of how line styles are implemented on MS Windows systems, you *must* set the line style *after* setting the drawing color. If you set the color after the line style you will lose the line style settings!

`style` is a bitmask which is a bitwise-OR of the following values. If you don't specify a dash type you will get a solid line. If you don't specify a cap or join type you will get a system-defined default of whatever value is fastest.

- FL_SOLID -----
- FL_DASH - - - - -
- FL_DOT
- FL_DASHDOT - . - .
- FL_DASHDOTDOT - . . -
- FL_CAP_FLAT
- FL_CAP_ROUND
- FL_CAP_SQUARE (extends past end point 1/2 line width)
- FL_JOIN_MITER (pointed)
- FL_JOIN_ROUND
- FL_JOIN_BEVEL (flat)

`width` is the number of pixels thick to draw the lines. Zero results in the system-defined default, which on both X and Windows is somewhat different and nicer than 1.

`dashes` is a pointer to an array of dash lengths, measured in pixels. The first location is how long to draw a solid portion, the next is how long to draw the gap, then the solid, etc. It is terminated with a zero-length entry. A `NULL` pointer or a zero-length array results in a solid line. Odd array sizes are not supported and result in undefined behavior.

Note: The dashes array does not work under Windows 95, 98, or Me, since those operating systems do not support complex line styles.

7.3.2 Drawing Fast Shapes

These functions are used to draw almost all the FLTK widgets. They draw on exact pixel boundaries and are as fast as possible. Their behavior is duplicated exactly on all platforms FLTK is ported. It is undefined whether these are affected by the [transformation matrix](#), so you should only call these while the matrix is set to the identity matrix (the default).

```
void fl_point(int x, int y)
```

Draw a single pixel at the given coordinates.

```
void fl_rectf(int x, int y, int w, int h)
void fl_rectf(int x, int y, int w, int h, FL_Color c)
```

Color a rectangle that exactly fills the given bounding box.

```
void fl_rectf(int x, int y, int w, int h, uchar r, uchar g, uchar b)
```

Color a rectangle with "exactly" the passed r, g, b color. On screens with less than 24 bits of color this is done by drawing a solid-colored block using `fl_draw_image()` so that the correct color shade is produced.

```
void fl_rect(int x, int y, int w, int h)
void fl_rect(int x, int y, int w, int h, FL_Color c)
```

Draw a 1-pixel border *inside* this bounding box.

```
void fl_line(int x, int y, int x1, int y1)
void fl_line(int x, int y, int x1, int y1, int x2, int y2)
```

Draw one or two lines between the given points.

```
void fl_loop(int x, int y, int x1, int y1, int x2, int y2)
void fl_loop(int x, int y, int x1, int y1, int x2, int y2, int x3, int y3)
```

Outline a 3 or 4-sided polygon with lines.

```
void fl_polygon(int x, int y, int x1, int y1, int x2, int y2)
void fl_polygon(int x, int y, int x1, int y1, int x2, int y2, int x3, int y3)
```

Fill a 3 or 4-sided polygon. The polygon must be convex.

```
void fl_xyline(int x, int y, int x1)
void fl_xyline(int x, int y, int x1, int y2)
void fl_xyline(int x, int y, int x1, int y2, int x3)
```

Draw horizontal and vertical lines. A horizontal line is drawn first, then a vertical, then a horizontal.

```
void fl_yxline(int x, int y, int y1)
void fl_yxline(int x, int y, int y1, int x2)
void fl_yxline(int x, int y, int y1, int x2, int y3)
```

Draw vertical and horizontal lines. A vertical line is drawn first, then a horizontal, then a vertical.

```
void fl_arc(int x, int y, int w, int h, double a1, double a2)
void fl_pie(int x, int y, int w, int h, double a1, double a2)
```

Draw ellipse sections using integer coordinates. These functions match the rather limited circle drawing code provided by X and MS Windows. The advantage over using `fl_arc()` with floating point coordinates is that they are faster because they often use the hardware, and they draw much nicer small circles, since the small sizes are often hard-coded bitmaps.

If a complete circle is drawn it will fit inside the passed bounding box. The two angles are measured in degrees counter-clockwise from 3'o'clock and are the starting and ending angle of the arc, `a2` must be greater or equal to `a1`.

`fl_arc()` draws a series of lines to approximate the arc. Notice that the integer version of `fl_arc()` has a different number of arguments to the other `fl_arc()` function described later in this chapter.

`fl_pie()` draws a filled-in pie slice. This slice may extend outside the line drawn by `fl_arc()`; to avoid this use `w-1` and `h-1`.

Todo add an `Fl_Draw_Area_Cb` typedef to allow `fl_scroll(...)` to be doxygenated?

```
void fl_scroll(int X, int Y, int W, int H, int dx, int dy, void (draw_area)(void, int,int,int,int), void* data)
```

Scroll a rectangle and draw the newly exposed portions. The contents of the rectangular area is first shifted by `dx` and `dy` pixels. The callback is then called for every newly exposed rectangular area,

7.3.3 Drawing Complex Shapes

The complex drawing functions let you draw arbitrary shapes with 2-D linear transformations. The functionality matches that found in the Adobe® PostScript™ language. The exact pixels that are filled are less defined than for the fast drawing functions so that FLTK can take advantage of drawing hardware. On both X and MS Windows the transformed vertices are rounded to integers before drawing the line segments: this severely limits the accuracy of these functions for complex graphics, so use OpenGL when greater accuracy and/or performance is required.

```
void fl_push_matrix()
void fl_pop_matrix()
```

Save and restore the current transformation. The maximum depth of the stack is 32 entries.

```
void fl_scale(double x,double y)
void fl_scale(double x)
void fl_translate(double x,double y)
void fl_rotate(double d)
void fl_mult_matrix(double a,double b,double c,double d,double x,double y)
```

Concatenate another transformation onto the current one. The rotation angle is in degrees (not radians) and is counter-clockwise.

```
double fl_transform_x(double x, double y)
double fl_transform_y(double x, double y)
double fl_transform_dx(double x, double y)
double fl_transform_dy(double x, double y)
void fl_transformed_vertex(double xf, double yf)
```

Transform a coordinate or a distance using the current transformation matrix. After transforming a coordinate pair, it can be added to the vertex list without any further translations using `fl_transformed_vertex()`.

```
void fl_begin_points()
void fl_end_points()
```

Start and end drawing a list of points. Points are added to the list with `fl_vertex()`.

```
void fl_begin_line()
void fl_end_line()
```

Start and end drawing lines.

```
void fl_begin_loop()
void fl_end_loop()
```

Start and end drawing a closed sequence of lines.

```
void fl_begin_polygon()
void fl_end_polygon()
```

Start and end drawing a convex filled polygon.

```
void fl_begin_complex_polygon()
void fl_gap()
void fl_end_complex_polygon()
```

Start and end drawing a complex filled polygon. This polygon may be concave, may have holes in it, or may be several disconnected pieces. Call `fl_gap()` to separate loops of the path. It is unnecessary but harmless to call `fl_gap()` before the first vertex, after the last one, or several times in a row.

`fl_gap()` should only be called between `fl_begin_complex_polygon()` and `fl_end_complex_polygon()`. To outline the polygon, use `fl_begin_loop()` and replace each `fl_gap()` with a `fl_end_loop();fl_begin_loop()` pair.

Note: For portability, you should only draw polygons that appear the same whether "even/odd" or "non-zero" winding rules are used to fill them. Holes should be drawn in the opposite direction of the outside loop.

```
void fl_vertex(double x,double y)
```

Add a single vertex to the current path.

```
void fl_curve(double X0, double Y0, double X1, double Y1, double X2, double Y2, double X3, double Y3)
```

Add a series of points on a Bezier curve to the path. The curve ends (and two of the points are) at $X0, Y0$ and $X3, Y3$.

```
void fl_arc(double x, double y, double r, double start, double end)
```

Add a series of points to the current path on the arc of a circle; you can get elliptical paths by using `scale` and `rotate` before calling `fl_arc()`. The center of the circle is given by `x` and `y`, and `r` is its radius. `fl_arc()` takes `start` and `end` angles that are measured in degrees counter-clockwise from 3 o'clock. If `end` is less than `start` then it draws the arc in a clockwise direction.

```
void fl_circle(double x, double y, double r)
```

`fl_circle(...)` is equivalent to `fl_arc(..., 0, 360)` but may be faster. It must be the *only* thing in the path: if you want a circle as part of a complex polygon you must use `fl_arc()`.

Note: `fl_circle()` draws incorrectly if the transformation is both rotated and non-square scaled.

7.3.4 Drawing Text

All text is drawn in the **current font**. It is undefined whether this location or the characters are modified by the current transformation.

```
void fl_draw(const char *, int x, int y)
void fl_draw(const char *, int n, int x, int y)
```

Draw a nul-terminated string or an array of `n` characters starting at the given location. Text is aligned to the left and to the baseline of the font. To align to the bottom, subtract `fl_descent()` from `y`. To align to the top, subtract `fl_descent()` and add `fl_height()`. This version of `fl_draw()` provides direct access to the text drawing function of the underlying OS. It does not apply any special handling to control characters.

```
void fl_draw(const char* str, int x, int y, int w, int h, FL_Align align, FL_Image* img, int draw_symbols)
```

Fancy string drawing function which is used to draw all the labels. The string is formatted and aligned inside the passed box. Handles `'t'` and `'n'`, expands all other control characters to `^X`, and aligns inside or against the edges of the box described by `x`, `y`, `w` and `h`. See `FL_Widget::align()` for values for `align`. The value `FL_ALIGN_INSIDE` is ignored, as this function always prints inside the box.

If `img` is provided and is not `NULL`, the image is drawn above or below the text as specified by the `align` value.

The `draw_symbols` argument specifies whether or not to look for symbol names starting with the "@" character.

void [fl_measure\(const char *str, int& w, int& h, int draw_symbols\)](#)

Measure how wide and tall the string will be when printed by the `fl_draw(...align)` function. This includes leading/trailing white space in the string, kerning, etc.

If the incoming `w` is non-zero it will wrap to that width.

This will probably give unexpected values unless you have called [fl_font\(\)](#) explicitly in your own code. Refer to the full documentation for [fl_measure\(\)](#) for details on usage and how to avoid common pitfalls.

See also

[fl_text_extents\(\)](#) – measure the 'inked' area of a string

[fl_width\(\)](#) – measure the pixel width of a string or single character

[fl_height\(\)](#) – measure the pixel height of the [current font](#)

[fl_descent\(\)](#) – the height of the descender for the [current font](#)

int [fl_height\(\)](#)

Recommended minimum line spacing for the [current font](#). You can also just use the value of `size` passed to [fl_font\(\)](#).

See also

[fl_text_extents\(\)](#), [fl_measure\(\)](#), [fl_width\(\)](#), [fl_descent\(\)](#)

int [fl_descent\(\)](#)

Recommended distance above the bottom of a [fl_height\(\)](#) tall box to draw the text at so it looks centered vertically in that box.

double [fl_width\(const char* txt\)](#)

double [fl_width\(const char* txt, int n\)](#)

double [fl_width\(unsigned int unicode_char\)](#)

Return the pixel width of a nul-terminated string, a sequence of `n` characters, or a single character in the [current font](#).

See also

[fl_measure\(\)](#), [fl_text_extents\(\)](#), [fl_height\(\)](#), [fl_descent\(\)](#)

```
void fl_text_extents(const char* txt, int& dx, int& dy, int& w, int& h)
```

Determines the minimum pixel dimensions of a nul-terminated string, ie. the 'inked area'.

Given a string "txt" drawn using `fl_draw(txt, x, y)` you would determine its pixel extents on the display using `fl_text_extents(txt, dx, dy, wo, ho)` such that a bounding box that exactly fits around the inked area of the text could be drawn with `fl_rect(x+dx, y+dy, wo, ho)`.

Refer to the full documentation for [fl_text_extents\(\)](#) for details on usage.

See also

[fl_measure\(\)](#), [fl_width\(\)](#), [fl_height\(\)](#), [fl_descent\(\)](#)

```
const char* fl_shortcut_label(int shortcut)
```

Unparse a shortcut value as used by [Fl_Button](#) or [Fl_Menu_Item](#) into a human-readable string like "Alt+N". This only works if the shortcut is a character key or a numbered function key. If the shortcut is zero an empty string is returned. The return value points at a static buffer that is overwritten with each call.

7.3.5 Fonts

FLTK supports a set of standard fonts based on the Times, Helvetica/Arial, Courier, and Symbol typefaces, as well as custom fonts that your application may load. Each font is accessed by an index into a font table.

Initially only the first 16 faces are filled in. There are symbolic names for them: `FL_HELVETICA`, `FL_TIMES`, `FL_COURIER`, and modifier values `FL_BOLD` and `FL_ITALIC` which can be added to these, and `FL_SYMBOL` and `FL_ZAPF_DINGBATS`. Faces greater than 255 cannot be used in [Fl_Widget](#) labels, since [Fl_Widget](#) stores the index as a byte.

One important thing to note about 'current font' is that there are so many paths through the GUI event handling code as widgets are partially or completely hidden, exposed and then re-drawn and therefore you can not guarantee that 'current font' contains the same value that you set on the other side of the event loop. Your value may have been superseded when a widget was redrawn. You are strongly advised to set the font explicitly before you draw any text or query the width and height of text strings, etc.

```
void fl_font(int face, int size)
```

Set the current font, which is then used by the routines described above. You may call this outside a draw context if necessary to call `fl_width()`, but on X this will open the display.

The font is identified by a `face` and a `size`. The size of the font is measured in `pixels` and not "points". Lines should be spaced `size` pixels apart or more.

```
int fl_font()
int fl_size()
```

Returns the face and size set by the most recent call to `fl_font(a,b)`. This can be used to save/restore the font.

7.3.6 Character Encoding

FLTK 1.3 expects all text in Unicode UTF-8 encoding. UTF-8 is ASCII compatible for the first 128 characters. International characters are encoded in multibyte sequences.

FLTK expects individual characters, characters that are not part of a string, in UCS-4 encoding, which is also ASCII compatible, but requires 4 bytes to store a Unicode character.

For more information about character encodings, see the chapter on [Unicode and UTF-8 Support](#).

7.3.7 Drawing Overlays

These functions allow you to draw interactive selection rectangles without using the overlay hardware. FLTK will XOR a single rectangle outline over a window.

```
void fl_overlay_rect(int x, int y, int w, int h)
void fl_overlay_clear()
```

`fl_overlay_rect()` draws a selection rectangle, erasing any previous rectangle by XOR'ing it first. `fl_overlay_clear()` will erase the rectangle without drawing a new one.

Using these functions is tricky. You should make a widget with both a `handle()` and `draw()` method. `draw()` should call `fl_overlay_clear()` before doing anything else. Your `handle()` method should call `window()->make_current()` and then `fl_overlay_rect()` after `FL_DRAG` events, and should call `fl_overlay_clear()` after a `FL_RELEASE` event.

7.4 Drawing Images

To draw images, you can either do it directly from data in your memory, or you can create a [FL_Image](#) object. The advantage of drawing directly is that it is more intuitive, and it is faster if the image data changes more often than it is redrawn. The advantage of using the object is that FLTK will cache translated forms of the image (on X it uses a server pixmap) and thus redrawing is *much* faster.

7.4.1 Direct Image Drawing

The behavior when drawing images when the current transformation matrix is not the identity is not defined, so you should only draw images when the matrix is set to the identity.

```
void fl_draw_image(const uchar *buf,int X,int Y,int W,int H,int D,int L)
void fl_draw_image_mono(const uchar *buf,int X,int Y,int W,int H,int D,int L)
```

Draw an 8-bit per color RGB or luminance image. The pointer points at the "r" data of the top-left pixel. Color data must be in *r, g, b* order. The top left corner is given by *X* and *Y* and the size of the image is given by *W* and *H*. *D* is the delta to add to the pointer between pixels, it may be any value greater or equal to 3, or it can be negative to flip the image horizontally. *L* is the delta to add to the pointer between lines (if 0 is passed it uses *W*D*), and may be larger than *W*D* to crop data, or negative to flip the image vertically.

It is highly recommended that you put the following code before the first `show()` of *any* window in your program to get rid of the dithering if possible:

```
Fl::visual (FL_RGB) ;
```

Gray scale (1-channel) images may be drawn. This is done if `abs(D)` is less than 3, or by calling `fl_↔draw_image_mono()`. Only one 8-bit sample is used for each pixel, and on screens with different numbers of bits for red, green, and blue only gray colors are used. Setting *D* greater than 1 will let you display one channel of a color image.

Note: The X version does not support all possible visuals. If FLTK cannot draw the image in the current visual it will abort. FLTK supports any visual of 8 bits or less, and all common TrueColor visuals up to 32 bits.

```
typedef void (*Fl_Draw_Image_Cb)(void *data,int x,int y,int w,uchar *buf)
void fl_draw_image(Fl_Draw_Image_Cb cb,void *data,int X,int Y,int W,int H,int D)
void fl_draw_image_mono(Fl_Draw_Image_Cb cb,void *data,int X,int Y,int W,int H,int D)
```

Call the passed function to provide each scan line of the image. This lets you generate the image as it is being drawn, or do arbitrary decompression of stored data, provided it can be decompressed to individual scan lines easily.

The callback is called with the `void*` user data pointer which can be used to point at a structure of information about the image, and the *x*, *y*, and *w* of the scan line desired from the image. 0,0 is the upper-left corner of the image, *not* *X*, *Y*. A pointer to a buffer to put the data into is passed. You must copy *w* pixels from scanline *y*, starting at pixel *x*, to this buffer.

Due to cropping, less than the whole image may be requested. So *x* may be greater than zero, the first *y* may be greater than zero, and *w* may be less than *W*. The buffer is long enough to store the entire *W*D* pixels, this is for convenience with some decompression schemes where you must decompress the entire line at once: decompress it into the buffer, and then if *x* is not zero, copy the data over so the *x*'*th* pixel is at the start of the buffer.

You can assume the `y`'s will be consecutive, except the first one may be greater than zero.

If `D` is 4 or more, you must fill in the unused bytes with zero.

```
int fl_draw_pixmap(char* const* data, int x, int y, FI_Color bg)
int fl_draw_pixmap(const char* const* cdata, int x, int y, FI_Color bg)
```

Draws XPM image data, with the top-left corner at the given position. The image is dithered on 8-bit displays so you won't lose color space for programs displaying both images and pixmaps. This function returns zero if there was any error decoding the XPM data.

To use an XPM, do:

```
#include "foo.xpm"
...
fl_draw_pixmap(foo, X, Y);
```

Transparent colors are replaced by the optional `FI_Color` argument. To draw with true transparency you must use the `FI_Pixmap` class.

```
int fl_measure_pixmap(char* const* data, int &w, int &h)
int fl_measure_pixmap(const char* const* cdata, int &w, int &h)
```

An XPM image contains the dimensions in its data. This function finds and returns the width and height. The return value is non-zero if the dimensions were parsed ok and zero if there was any problem.

7.4.2 Direct Image Reading

FLTK provides a single function for reading from the current window or off-screen buffer into a RGB(A) image buffer.

```
uchar* fl_read_image(uchar *p, int X, int Y, int W, int H, int alpha)
```

Read a RGB(A) image from the current window or off-screen buffer. The `p` argument points to a buffer that can hold the image and must be at least $W \times H \times 3$ bytes when reading RGB images and $W \times H \times 4$ bytes when reading RGBA images. If `NULL`, `fl_read_image()` will create an array of the proper size which can be freed using `delete[]`.

The `alpha` parameter controls whether an alpha channel is created and the value that is placed in the alpha channel. If 0, no alpha channel is generated.

7.4.3 Image Classes

FLTK provides a base image class called [FL_Image](#) which supports creating, copying, and drawing images of various kinds, along with some basic color operations. Images can be used as labels for widgets using the `image()` and `deimage()` methods or drawn directly.

The [FL_Image](#) class does almost nothing by itself, but is instead supported by three basic image types:

- [FL_Bitmap](#)
- [FL_Pixmap](#)
- [FL_RGB_Image](#)

The [FL_Bitmap](#) class encapsulates a mono-color bitmap image. The `draw()` method draws the image using the current drawing color.

The [FL_Pixmap](#) class encapsulates a colormapped image. The `draw()` method draws the image using the colors in the file, and masks off any transparent colors automatically.

The [FL_RGB_Image](#) class encapsulates a full-color (or grayscale) image with 1 to 4 color components. Images with an even number of components are assumed to contain an alpha channel that is used for transparency. The transparency provided by the `draw()` method is either a 24-bit blend against the existing window contents or a "screen door" transparency mask, depending on the platform and screen color depth.

char [fl_can_do_alpha_blending\(\)](#)

`fl_can_do_alpha_blending()` will return 1, if your platform supports true alpha blending for RGBA images, or 0, if FLTK will use screen door transparency.

FLTK also provides several image classes based on the three standard image types for common file formats:

- [FL_GIF_Image](#)
- [FL_JPEG_Image](#)
- [FL_PNG_Image](#)
- [FL_PNM_Image](#)
- [FL_XBM_Image](#)
- [FL_XPM_Image](#)

Each of these image classes loads a named file of the corresponding format. The [FL_Shared_Image](#) class can be used to load any type of image file - the class examines the file and constructs an image of the appropriate type. It can also be used to scale an image to a certain size in drawing units, independently from its size in pixels (see [FL_Shared_Image::scale\(\)](#)).

Finally, FLTK provides a special image class called [FL_Tiled_Image](#) to tile another image object in the specified area. This class can be used to tile a background image in a [FL_Group](#) widget, for example.

```
virtual void FL\_Image::copy\(\)  
virtual FL\_Image\* FL\_Image::copy\(int w, int h\)
```

The `copy()` method creates a copy of the image. The second form specifies the new size of the image - the image is resized using the nearest-neighbor algorithm (this is the default).

Note

As of FLTK 1.3.3 the image resizing algorithm can be changed. See [FI_Image::RGB_scaling\(FI_RGB_Scaling method\)](#)

virtual void [FI_Image::draw\(int x, int y, int w, int h, int ox, int oy\)](#)

The `draw()` method draws the image object. `x, y, w, h` indicates the destination rectangle. `ox, oy, w, h` is the source rectangle. This source rectangle is copied to the destination. The source rectangle may extend outside the image, i.e. `ox` and `oy` may be negative and `w` and `h` may be bigger than the image, and this area is left unchanged.

Note

See exceptions for [FI_Tiled_Image::draw\(\)](#) regarding arguments `ox`, `oy`, `w`, and `h`.

virtual void [FI_Image::draw\(int x, int y\)](#)

Draws the image with the upper-left corner at `x, y`. This is the same as doing `img->draw(x, y, img->w(), img->h(), 0, 0)` where `img` is a pointer to any [FI_Image](#) type.

7.4.4 Offscreen Drawing

Sometimes it can be very useful to generate a complex drawing in memory first and copy it to the screen at a later point in time. This technique can significantly reduce the amount of repeated drawing. Offscreen drawing functions are declared in `<FL/x.H>`.

[FI_Double_Window](#) uses offscreen rendering to avoid flickering on systems that don't support double-buffering natively.

FLTK can draw into an offscreen buffer at any time. There is no need to wait for an [FI_Widget::draw\(\)](#) to occur.

Note

The X11 platform requires an open display for offscreen drawing, i.e. you may need to call `fl_open_display()` prior to creating and using offscreen buffers, particularly if no window has been shown yet.

Note

In FLTK 1.3.x and earlier versions all offscreen drawing functions described below are implemented as macros and create certain temporary variables to save context information. You may need to create local scope blocks with curly braces { ... } if you use offscreen functions more than once in a function or method.

Example:

```
fl_open_display(); // necessary before showing the first window
Fl_Offscreen oscr = fl_create_offscreen(120, 120);
{ // begin block
  fl_begin_offscreen(oscr);
  fl_color(FL_WHITE);
  fl_rectf(0, 0, 120, 120);
  fl_end_offscreen();
} // end block
// other code here
{ // begin block
  fl_begin_offscreen(oscr);
  fl_color(FL_BLACK);
  fl_rectf(10, 10, 100, 100);
  fl_end_offscreen();
} // end block
// other code here
fl_delete_offscreen(oscr);
```

Note

In FLTK 1.4.0 and later neither calling `fl_open_display()` nor using local blocks is necessary since the offscreen functions described below are real functions (not macros as in 1.3.x).

`Fl_Offscreen fl_create_offscreen(int w, int h)`

Create an RGB offscreen buffer with `w*h` pixels.

`void fl_delete_offscreen(Fl_Offscreen)`

Delete a previously created offscreen buffer. All drawings are lost.

`void fl_begin_offscreen(Fl_Offscreen)`

Send all subsequent drawing commands to this offscreen buffer.

`void fl_end_offscreen()`

Quit sending drawing commands to this offscreen buffer.

`void fl_copy_offscreen(int x, int y, int w, int h, Fl_Offscreen oscr, int srcx, int srcy)`

Copy a rectangular area of the size `w*h` from `srcx,srcy` in the offscreen buffer into the current buffer at `x,y`.

Chapter 8

Handling Events

This chapter discusses the FLTK event model and how to handle events in your program or widget.

8.1 The FLTK Event Model

Every time a user moves the mouse pointer, clicks a button, or presses a key, an event is generated and sent to your application. Events can also come from other programs like the window manager.

Events are identified by the integer argument passed to a `handle()` method that overrides the `FL_Widget::handle()` virtual method. Other information about the most recent event is stored in static locations and acquired by calling the `FL::event_*` methods. This static information remains valid until the next event is read from the window system, so it is ok to look at it outside of the `handle()` method.

Event numbers can be converted to their actual names using the `fl_eventnames[]` array defined in `#include <FL/names.h>`; see next chapter for details.

In the next chapter, the `MyClass::handle()` example shows how to override the `FL_Widget::handle()` method to accept and process specific events.

8.2 Mouse Events

8.2.1 FL_PUSH

A mouse button has gone down with the mouse pointing at this widget. You can find out what button by calling `FL::event_button()`. You find out the mouse position by calling `FL::event_x()` and `FL::event_y()`.

A widget indicates that it "wants" the mouse click by returning non-zero from its `handle()` method, as in the `MyClass::handle()` example. It will then become the `FL::pushed()` widget and will get `FL_DRAG` and the matching `FL_RELEASE` events. If `handle()` returns zero then FLTK will try sending the `FL_PUSH` to another widget.

8.2.2 FL_DRAG

The mouse has moved with a button held down. The current button state is in `FL::event_state()`. The mouse position is in `FL::event_x()` and `FL::event_y()`.

In order to receive `FL_DRAG` events, the widget must return non-zero when handling `FL_PUSH`.

8.2.3 FL_RELEASE

A mouse button has been released. You can find out what button by calling `Fl::event_button()`.

In order to receive the `FL_RELEASE` event, the widget must return non-zero when handling `FL_PUSH`.

8.2.4 FL_MOVE

The mouse has moved without any mouse buttons held down. This event is sent to the `Fl::belowmouse()` widget.

In order to receive `FL_MOVE` events, the widget must return non-zero when handling `FL_ENTER`.

8.2.5 FL_MOUSEWHEEL

The user has moved the mouse wheel. The `Fl::event_dx()` and `Fl::event_dy()` methods can be used to find the amount to scroll horizontally and vertically.

8.3 Focus Events

8.3.1 FL_ENTER

The mouse has been moved to point at this widget. This can be used for highlighting feedback. If a widget wants to highlight or otherwise track the mouse, it indicates this by returning non-zero from its `handle()` method. It then becomes the `Fl::belowmouse()` widget and will receive `FL_MOVE` and `FL_LEAVE` events.

8.3.2 FL_LEAVE

The mouse has moved out of the widget.

In order to receive the `FL_LEAVE` event, the widget must return non-zero when handling `FL_ENTER`.

8.3.3 FL_FOCUS

This indicates an *attempt* to give a widget the keyboard focus.

If a widget wants the focus, it should change itself to display the fact that it has the focus, and return non-zero from its `handle()` method. It then becomes the `Fl::focus()` widget and gets `FL_KEYDOWN`, `FL_KEYUP`, and `FL_UNFOCUS` events.

The focus will change either because the window manager changed which window gets the focus, or because the user tried to navigate using tab, arrows, or other keys. You can check `Fl::event_key()` to figure out why it moved. For navigation it will be the key pressed and for interaction with the window manager it will be zero.

8.3.4 FL_UNFOCUS

This event is sent to the previous `Fl::focus()` widget when another widget gets the focus or the window loses focus.

8.4 Keyboard Events

8.4.1 FL_KEYBOARD, FL_KEYDOWN, FL_KEYUP

A key was pressed (FL_KEYDOWN) or released (FL_KEYUP). FL_KEYBOARD is a synonym for FL_KEYDOWN, and both names are used interchangeably in this documentation.

The key can be found in [Fl::event_key\(\)](#). The text that the key should insert can be found with [Fl::event_text\(\)](#) and its length is in [Fl::event_length\(\)](#).

If you use the key, then `handle()` should return 1. If you return zero then FLTK assumes you ignored the key and will then attempt to send it to a parent widget. If none of them want it, it will change the event into a FL_SHORTCUT event. FL_KEYBOARD events are also generated by the character palette/map.

To receive FL_KEYBOARD events you must also respond to the FL_FOCUS and FL_UNFOCUS events by returning 1. This way FLTK knows whether to bother sending your widget keyboard events. (Some widgets don't need them, e.g. [Fl_Box](#).)

If you are writing a text-editing widget you may also want to call the [Fl::compose\(\)](#) function to translate individual keystrokes into characters.

FL_KEYUP events are sent to the widget that currently has focus. This is not necessarily the same widget that received the corresponding FL_KEYDOWN event because focus may have changed between events.

Todo Add details on how to detect repeating keys, since on some X servers a repeating key will generate both FL_KEYUP and FL_KEYDOWN, such that to tell if a key is held, you need [Fl::event_key\(int\)](#) to detect if the key is being held down during FL_KEYUP or not.

8.4.2 FL_SHORTCUT

If the [Fl::focus\(\)](#) widget is zero or ignores an FL_KEYBOARD event then FLTK tries sending this event to every widget it can, until one of them returns non-zero. FL_SHORTCUT is first sent to the [Fl::belowmouse\(\)](#) widget, then its parents and siblings, and eventually to every widget in the window, trying to find an object that returns non-zero. FLTK tries really hard to not to ignore any keystrokes!

You can also make "global" shortcuts by using [Fl::add_handler\(\)](#). A global shortcut will work no matter what windows are displayed or which one has the focus.

8.5 Widget Events

8.5.1 FL_DEACTIVATE

This widget is no longer active, due to [deactivate\(\)](#) being called on it or one of its parents. Please note that although [active\(\)](#) may still return true for this widget after receiving this event, it is only truly active if [active\(\)](#) is true for both it and all of its parents. (You can use [active_r\(\)](#) to check this).

8.5.2 FL_ACTIVATE

This widget is now active, due to [activate\(\)](#) being called on it or one of its parents.

8.5.3 FL_HIDE

This widget is no longer visible, due to `hide()` being called on it or one of its parents, or due to a parent window being minimized. Please note that although `visible()` may still return true for this widget after receiving this event, it is only truly visible if `visible()` is true for both it and all of its parents. (You can use `visible_r()` to check this).

8.5.4 FL_SHOW

This widget is visible again, due to `show()` being called on it or one of its parents, or due to a parent window being restored. *A child `Fl_Window` will respond to this by actually creating the window if not done already, so if you subclass a window, be sure to pass `FL_SHOW` to the base class `handle()` method!*

Note

The events in this chapter ("Widget Events"), i.e. `FL_ACTIVATE`, `FL_DEACTIVATE`, `FL_SHOW`, and `FL_HIDE`, are the only events deactivated and invisible widgets can usually get, depending on their states. Under certain circumstances, there may also be `FL_LEAVE` or `FL_UNFOCUS` events delivered to deactivated or hidden widgets.

8.6 Clipboard Events

8.6.1 FL_PASTE

You should get this event some time after you call `Fl::paste()`. The contents of `Fl::event_text()` is the text to insert and the number of characters is in `Fl::event_length()`.

8.6.2 FL_SELECTIONCLEAR

The `Fl::selection_owner()` will get this event before the selection is moved to another widget. This indicates that some other widget or program has claimed the selection. Motif programs used this to clear the selection indication. Most modern programs ignore this.

8.7 Drag and Drop Events

FLTK supports drag and drop of text and files from any application on the desktop to an FLTK widget. Text is transferred using UTF-8 encoding. Files are received as a list of full path and file names, separated by newline.

On some X11 platforms, files are received as a URL-encoded UTF-8 string, that is, non-ASCII bytes (and a few others such as space and %) are replaced by the 3 bytes "%XY" where XY are the byte's hexadecimal value. The `fl_decode_uri()` function can be used to transform in-place the received string into a proper UTF-8 string. On these platforms, strings corresponding to dropped files are further prepended by `file://` (or other prefixes such as `computer://`).

See `Fl::dnd()` for drag and drop from an FLTK widget.

The drag and drop data is available in `Fl::event_text()` at the concluding `FL_PASTE`. On some platforms, the event text is also available for the `FL_DND_*` events, however application must not depend on that behavior because it depends on the protocol used on each platform.

`FL_DND_*` events cannot be used in widgets derived from `Fl_Group` or `Fl_Window`.

8.7.1 FL_DND_ENTER

The mouse has been moved to point at this widget. A widget that is interested in receiving drag'n'drop data must return 1 to receive `FL_DND_DRAG`, `FL_DND_LEAVE` and `FL_DND_RELEASE` events.

8.7.2 FL_DND_DRAG

The mouse has been moved inside a widget while dragging data. A widget that is interested in receiving drag'n'drop data should indicate the possible drop position.

8.7.3 FL_DND_LEAVE

The mouse has moved out of the widget.

8.7.4 FL_DND_RELEASE

The user has released the mouse button dropping data into the widget. If the widget returns 1, it will receive the data in the immediately following `FL_PASTE` event.

8.8 Other events

8.8.1 FL_SCREEN_CONFIGURATION_CHANGED

Sent whenever the screen configuration changes (a screen is added/removed, a screen resolution is changed, screens are moved). Use [Fl::add_handler\(\)](#) to be notified of this event.

8.8.2 FL_FULLSCREEN

The application window has been changed from normal to fullscreen, or from fullscreen to normal. If you are using a X window manager which supports Extended Window Manager Hints, this event will not be delivered until the change has actually happened.

8.9 Fl::event_*() methods

FLTK keeps the information about the most recent event in static storage. This information is good until the next event is processed. Thus it is valid inside `handle()` and `callback()` methods.

These are all trivial inline functions and thus very fast and small:

- [Fl::event_button\(\)](#)
- [Fl::event_clicks\(\)](#)
- [Fl::event_dx\(\)](#)
- [Fl::event_dy\(\)](#)
- [Fl::event_inside\(\)](#)
- [Fl::event_is_click\(\)](#)
- [Fl::event_key\(\)](#)
- [Fl::event_length\(\)](#)
- [Fl::event_state\(\)](#)
- [Fl::event_text\(\)](#)
- [Fl::event_x\(\)](#)
- [Fl::event_x_root\(\)](#)
- [Fl::event_y\(\)](#)
- [Fl::event_y_root\(\)](#)
- [Fl::get_key\(\)](#)
- [Fl::get_mouse\(\)](#)
- [Fl::test_shortcut\(\)](#)

8.10 Event Propagation

Widgets receive events via the virtual `handle()` function. The argument indicates the type of event that can be handled. The widget must indicate if it handled the event by returning 1. FLTK will then remove the event and wait for further events from the host. If the widget's handle function returns 0, FLTK may redistribute the event based on a few rules.

Most events are sent directly to the `handle()` method of the [Fl_Window](#) that the window system says they belong to. The window (actually the [Fl_Group](#) that [Fl_Window](#) is a subclass of) is responsible for sending the events on to any child widgets. To make the [Fl_Group](#) code somewhat easier, FLTK sends some events (`FL_DRAG`, `FL_RELEASE`, `FL_KEYBOARD`, `FL_SHORTCUT`, `FL_UNFOCUS`, and `FL_LEAVE`) directly to leaf widgets. These procedures control those leaf widgets:

- [Fl::add_handler\(\)](#)
- [Fl::belowmouse\(\)](#)
- [Fl::focus\(\)](#)

- [Fl::grab\(\)](#)
- [Fl::modal\(\)](#)
- [Fl::pushed\(\)](#)
- [Fl::release\(\)](#) (deprecated, see [Fl::grab\(0\)](#))
- [Fl_Widget::take_focus\(\)](#)

FLTK propagates events along the widget hierarchy depending on the kind of event and the status of the UI. Some events are injected directly into the widgets, others may be resent as new events to a different group of receivers.

Mouse click events are first sent to the window that caused them. The window then forwards the event down the hierarchy until it reaches the widget that is below the click position. If that widget uses the given event, the widget is marked "pushed" and will receive all following mouse motion (`FL_DRAG`) events until the mouse button is released.

Mouse motion (`FL_MOVE`) events are sent to the [Fl::belowmouse\(\)](#) widget, i.e. the widget that returned 1 on the last `FL_ENTER` event.

Mouse wheel events are sent to the window that caused the event. The window propagates the event down the tree, first to the widget that is below the mouse pointer, and if that does not succeed, to all other widgets in the group. This ensures that scroll widgets work as expected with the widget furthest down in the hierarchy getting the first opportunity to use the wheel event, but also giving scroll bars, that are not directly below the mouse a chance.

Keyboard events are sent directly to the widget that has keyboard focus. If the focused widget rejects the event, it is resent as a shortcut event, first to the top-most window, then to the widget below the mouse pointer, propagating up the hierarchy to all its parents. Those send the event also to all widgets that are not below the mouse pointer. Now if that did not work out, the shortcut is sent to all registered shortcut handlers.

If we are still unsuccessful, the event handler flips the case of the shortcut letter and starts over. Finally, if the key is "escape", FLTK sends a close event to the top-most window.

All other events are pretty much sent right away to the window that created the event.

Widgets can "grab" events. The grabbing window gets all events exclusively, but usually by the same rules as described above.

Windows can also request exclusivity in event handling by making the window modal.

8.11 FLTK Compose-Character Sequences

The character composition done by [Fl_Input](#) widget requires that you call the [Fl::compose\(\)](#) function if you are writing your own text editor widget.

Currently, all characters made by single key strokes with or without modifier keys, or by system-defined character compose sequences (that can involve dead keys or a compose key) can be input. You should call [Fl::compose\(\)](#) in case any enhancements to this processing are done in the future. The interface has been designed to handle arbitrary UTF-8 encoded text.

The following methods are provided for character composition:

- [Fl::compose\(\)](#)
- [Fl::compose_reset\(\)](#)

Under Mac OS X, FLTK "previews" partially composed sequences.

Chapter 9

Adding and Extending Widgets

This chapter describes how to add your own widgets or extend existing widgets in FLTK.

9.1 Subclassing

New widgets are created by *subclassing* an existing FLTK widget, typically [Fl_Widget](#) for controls and [Fl_Group](#) for composite widgets.

A control widget typically interacts with the user to receive and/or display a value of some sort.

A composite widget holds a list of child widgets and handles moving, sizing, showing, or hiding them as needed. [Fl_Group](#) is the main composite widget class in FLTK, and all of the other composite widgets ([Fl_Pack](#), [Fl_Scroll](#), [Fl_Tabs](#), [Fl_Tile](#), and [Fl_Window](#)) are subclasses of it.

You can also subclass other existing widgets to provide a different look or user-interface. For example, the button widgets are all subclasses of [Fl_Button](#) since they all interact with the user via a mouse button click. The only difference is the code that draws the face of the button.

9.2 Making a Subclass of Fl_Widget

Your subclasses can directly descend from [Fl_Widget](#) or any subclass of [Fl_Widget](#). [Fl_Widget](#) has only four virtual methods, and overriding some or all of these may be necessary.

9.3 The Constructor

The constructor should have the following arguments:

```
MyClass(int x, int y, int w, int h, const char *label = 0);
```

This will allow the class to be used in [FLUID](#) without problems.

The constructor must call the constructor for the base class and pass the same arguments:

```
MyClass::MyClass(int x, int y, int w, int h, const char *label)
: Fl_Widget(x, y, w, h, label) {
// do initialization stuff...
}
```

`Fl_Widget`'s protected constructor sets `x()`, `y()`, `w()`, `h()`, and `label()` to the passed values and initializes the other instance variables to:

```
type(0);
box(FL_NO_BOX);
color(FL_BACKGROUND_COLOR);
selection_color(FL_BACKGROUND_COLOR);
labeltype(FL_NORMAL_LABEL);
labelstyle(FL_NORMAL_STYLE);
labelsize(FL_NORMAL_SIZE);
labelcolor(FL_FOREGROUND_COLOR);
align(FL_ALIGN_CENTER);
callback(default_callback, 0);
flags(ACTIVE|VISIBLE);
image(0);
deimage(0);
```

9.4 Protected Methods of `Fl_Widget`

The following methods are provided for subclasses to use:

- `clear_visible()`
- `damage()`
- `draw_box()`
- `draw_focus()`
- `draw_label()`
- `set_flag()`
- `set_visible()`
- `test_shortcut()`
- `type()`

```
void Fl_Widget::damage(uchar mask)
void Fl_Widget::damage(uchar mask, int x, int y, int w, int h)
uchar Fl_Widget::damage()
```

The first form indicates that a partial update of the object is needed. The bits in `mask` are OR'd into `damage()`. Your `draw()` routine can examine these bits to limit what it is drawing. The public method `Fl_Widget::redraw()` simply does `Fl_Widget::damage(FL_DAMAGE_ALL)`, but the implementation of your widget can call the public `damage(n)`.

The second form indicates that a region is damaged. If only these calls are done in a window (no calls to `damage(n)`) then FLTK will clip to the union of all these calls before drawing anything. This can greatly speed up incremental displays. The mask bits are OR'd into `damage()` unless this is a `Fl_Window` widget.

The third form returns the bitwise-OR of all `damage(n)` calls done since the last `draw()`.

When redrawing your widgets you should look at the damage bits to see what parts of your widget need redrawing. The `handle()` method can then set individual damage bits to limit the amount of drawing that needs to be done:

```
MyClass::handle(int event) {
    ...
    if (change_to_part1) damage(1);
    if (change_to_part2) damage(2);
    if (change_to_part3) damage(4);
}
MyClass::draw() {
    if (damage() & FL_DAMAGE_ALL) {
        ... draw frame/box and other static stuff ...
    }
    if (damage() & (FL_DAMAGE_ALL | 1)) draw_part1();
    if (damage() & (FL_DAMAGE_ALL | 2)) draw_part2();
    if (damage() & (FL_DAMAGE_ALL | 4)) draw_part3();
}
```

Todo Clarify `Fl_Window::damage(uchar)` handling - seems confused/wrong? ORing value doesn't match setting behaviour in `Fl_Widget.H!`

```
void Fl_Widget::draw_box() const
void Fl_Widget::draw_box(Fl_Boxtype t, Fl_Color c) const
```

The first form draws this widget's `box()`, using the dimensions of the widget. The second form uses `t` as the box type and `c` as the color for the box.

```
void Fl_Widget::draw_focus()
void Fl_Widget::draw_focus(Fl_Boxtype t, int x, int y, int w, int h) const
```

Draws a focus box inside the widget's bounding box. The second form allows you to specify a different bounding box.

```
void Fl_Widget::draw_label() const
void Fl_Widget::draw_label(int x, int y, int w, int h) const
void Fl_Widget::draw_label(int x, int y, int w, int h, Fl_Align align) const
```

The first form is the usual function for a `draw()` method to call to draw the widget's label. It does not draw the label if it is supposed to be outside the box (on the assumption that the enclosing group will draw those labels).

The second form uses the passed bounding box instead of the widget's bounding box. This is useful so "centered" labels are aligned with some feature, like a moving slider.

The third form draws the label anywhere. It acts as though `FL_ALIGN_INSIDE` has been forced on so the label will appear inside the passed bounding box. This is designed for parent groups to draw labels with.

```
void Fl_Widget::set_flag(int c)
```

Calling `set_flag(SHORTCUT_LABEL)` modifies the behavior of `draw_label()` so that '&' characters cause an underscore to be printed under the next letter.

```
void FI_Widget::set_visible()
void FI_Widget::clear_visible()
```

Fast inline versions of `FI_Widget::hide()` and `FI_Widget::show()`. These do not send the `FL_HIDE` and `FL_↔_SHOW` events to the widget.

```
int FI_Widget::test_shortcut()
static int FI_Widget::test_shortcut(const char *s)
```

The first version tests `FI_Widget::label()` against the current event (which should be a `FL_SHORTCUT` event). If the label contains a '&' character and the character after it matches the keypress, this returns true. This returns false if the `SHORTCUT_LABEL` flag is off, if the label is `NULL`, or does not have a '&' character in it, or if the keypress does not match the character.

The second version lets you do this test against an arbitrary string.

Todo Clarify `FI_Widget::test_shortcut()` explanations. `FI_Widget.h` says Internal Use only, but subclassing chapter gives details!

```
uchar FI_Widget::type() const
void FI_Widget::type(uchar t)
```

The property `FI_Widget::type()` can return an arbitrary 8-bit identifier, and can be set with the protected method `type(uchar t)`. This value had to be provided for Forms compatibility, but you can use it for any purpose you want. Try to keep the value less than 100 to not interfere with reserved values.

FLTK does not use RTTI (Run Time Typing Information) to enhance portability. But this may change in the near future if RTTI becomes standard everywhere.

If you don't have RTTI you can use the clumsy FLTK mechanism, by having `type()` use a unique value. These unique values must be greater than the symbol `FL_RESERVED_TYPE` (which is 100) and less than `FL_WINDOW` (unless you make a subclass of `FI_Window`). Look through the header files for `FL_↔_RESERVED_TYPE` to find an unused number. If you make a subclass of `FI_Window` you must use `FL_↔_WINDOW + n` (where `n` must be in the range 1 to 7).

9.5 Handling Events

The virtual method `Fl_Widget::handle(int event)` is called to handle each event passed to the widget. It can:

- Change the state of the widget.
- Call `Fl_Widget::redraw()` if the widget needs to be redisplayed.
- Call `Fl_Widget::damage(uchar c)` if the widget needs a partial-update (assuming you provide support for this in your `draw()` method).
- Call `Fl_Widget::do_callback()` if a callback should be generated.
- Call `Fl_Widget::handle()` on child widgets.

Events are identified by the integer argument. Other information about the most recent event is stored in static locations and acquired by calling the `Fl::event_*` methods. This information remains valid until another event is handled.

Here is a sample `handle()` method for a widget that acts as a pushbutton and also accepts the keystroke 'x' to cause the callback:

```
int MyClass::handle(int event) {
    switch(event) {
        case FL_PUSH:
            highlight = 1;
            redraw();
            return 1;
        case FL_DRAG: {
            int t = Fl::event_inside(this);
            if (t != highlight) {
                highlight = t;
                redraw();
            }
        }
        return 1;
        case FL_RELEASE:
            if (highlight) {
                highlight = 0;
                redraw();
                do_callback();
                // never do anything after a callback, as the callback
                // may delete the widget!
            }
            return 1;
        case FL_SHORTCUT:
            if (Fl::event_key() == 'x') {
                do_callback();
                return 1;
            }
            return 0;
        default:
            return Fl_Widget::handle(event);
    }
}
```

You must return non-zero if your `handle()` method uses the event. If you return zero, the parent widget will try sending the event to another widget.

For debugging purposes, event numbers can be printed as their actual event names using the `fl_eventnames[]` array, e.g.:

```
#include <FL/names.h> // defines fl_eventnames[]
[...
int MyClass::handle(int e) {
    printf("Event was %s (%d)\n", fl_eventnames[e], e); // e.g. "Event was FL_PUSH (1)"
    [...
}
```

9.6 Drawing the Widget

The `draw()` virtual method is called when FLTK wants you to redraw your widget. It will be called if and only if `damage()` is non-zero, and `damage()` will be cleared to zero after it returns. The `draw()` method should be declared protected so that it can't be called from non-drawing code.

The `damage()` value contains the bitwise-OR of all the `damage(n)` calls to this widget since it was last drawn. This can be used for minimal update, by only redrawing the parts whose bits are set. FLTK will turn on the `FL_DAMAGE_ALL` bit if it thinks the entire widget must be redrawn, e.g. for an expose event.

Expose events (and the `damage(mask,x,y,w,h)` function described above) will cause `draw()` to be called with FLTK's clipping turned on. You can greatly speed up redrawing in some cases by testing `fl_not_clipped(x,y,w,h)` or `fl_clip_box()` and skipping invisible parts.

Besides the protected methods described above, FLTK provides a large number of basic drawing functions, which are described in the chapter [Drawing Things in FLTK](#).

9.7 Resizing the Widget

The `resize(x,y,w,h)` method is called when the widget is being resized or moved. The arguments are the new position, width, and height. `x()`, `y()`, `w()`, and `h()` still remain the old size. You must call `resize()` on your base class with the same arguments to get the widget size to actually change.

This should *not* call `redraw()`, at least if only the `x()` and `y()` change. This is because composite widgets like [FL_Scroll](#) may have a more efficient way of drawing the new position.

9.8 Making a Composite Widget

A "composite" widget contains one or more "child" widgets. To make a composite widget you should subclass [FL_Group](#). It is possible to make a composite object that is not a subclass of [FL_Group](#), but you'll have to duplicate the code in [FL_Group](#) anyways.

Instances of the child widgets may be included in the parent:

```
class MyClass : public FL_Group {
    FL_Button the_button;
    FL_Slider the_slider;
    ...
};
```

The constructor has to initialize these instances. They are automatically added to the group, since the [FL_Group](#) constructor does [FL_Group::begin\(\)](#). *Don't forget to call [FL_Group::end\(\)](#) or use the [FL_End](#) pseudo-class:*

```
MyClass::MyClass(int x, int y, int w, int h) :
    FL_Group(x, y, w, h),
    the_button(x + 5, y + 5, 100, 20),
    the_slider(x, y + 50, w, 20)
{
    ... (you could add dynamically created child widgets here) ...
    end(); // don't forget to do this!
}
```

The child widgets need callbacks. These will be called with a pointer to the children, but the widget itself may be found in the `parent()` pointer of the child. Usually these callbacks can be static private methods, with a matching private method:

```
void MyClass::static_slider_cb(FL_Widget* v, void *) { // static method
    ((MyClass*) (v->parent()))->slider_cb();
}
void MyClass::slider_cb() { // normal method
    use(the_slider->value());
}
```

If you make the `handle()` method, you can quickly pass all the events to the children using the `Fl_Group::handle()` method. You don't need to override `handle()` if your composite widget does nothing other than pass events to the children:

```
int MyClass::handle(int event) {
    if (Fl_Group::handle(event)) return 1;
    ... handle events that children don't want ...
}
```

If you override `draw()` you need to draw all the children. If `redraw()` or `damage()` is called on a child, `damage(FL_DAMAGE_CHILD)` is done to the group, so this bit of `damage()` can be used to indicate that a child needs to be drawn. It is fastest if you avoid drawing anything else in this case:

```
int MyClass::draw() {
    Fl_Widget *const*a = array();
    if (damage() == FL_DAMAGE_CHILD) { // only redraw some children
        for (int i = children(); i --; a++) update_child(**a);
    } else { // total redraw
        ... draw background graphics ...
        // now draw all the children atop the background:
        for (int i = children_; i --; a++) {
            draw_child(**a);
            draw_outside_label(**a); // you may not need to do this
        }
    }
}
```

`Fl_Group` provides some protected methods to make drawing easier:

- [draw_child\(\)](#)
- [draw_children\(\)](#)
- [draw_outside_label\(\)](#)
- [update_child\(\)](#)

void [Fl_Group::draw_child\(Fl_Widget &widget\) const](#)

This will force the child's `damage()` bits all to one and call `draw()` on it, then clear the `damage()`. You should call this on all children if a total redraw of your widget is requested, or if you draw something (like a background box) that damages the child. Nothing is done if the child is not `visible()` or if it is clipped.

void [Fl_Group::draw_children\(\)](#)

A convenience function that draws all children of the group. This is useful if you derived a widget from `Fl_Group` and want to draw a special border or background. You can call `draw_children()` from the derived `draw()` method after drawing the box, border, or background.

void [Fl_Group::draw_outside_label\(const Fl_Widget &widget\) const](#)

Draw the labels that are *not* drawn by [draw_label\(\)](#). If you want more control over the label positions you might want to call `child->draw_label(x, y, w, h, a)`.

void [Fl_Group::update_child\(Fl_Widget& widget\) const](#)

Draws the child only if its `damage()` is non-zero. You should call this on all the children if your own `damage` is equal to `FL_DAMAGE_CHILD`. Nothing is done if the child is not `visible()` or if it is clipped.

9.9 Cut and Paste Support

FLTK provides routines to cut and paste UTF-8 encoded text between applications:

- [Fl::copy\(\)](#)
- [Fl::paste\(\)](#)
- [Fl::selection\(\)](#)
- [Fl::selection_owner\(\)](#)

It is also possible to copy and paste image data between applications:

- [Fl_Copy_Surface](#)
- [Fl::clipboard_contains\(\)](#)
- [Fl::paste\(\)](#)

It may be possible to cut/paste other kinds of data by using [Fl::add_handler\(\)](#). Note that handling events beyond those provided by FLTK may be operating system specific. See [Operating System Issues](#) for more details.

9.10 Drag And Drop Support

FLTK provides routines to drag and drop UTF-8 encoded text between applications:

Drag'n'drop operations are initiated by copying data to the clipboard and calling the function [Fl::dnd\(\)](#).

Drop attempts are handled via the following events, already described under [Drag and Drop Events](#) in a previous chapter:

- `FL_DND_ENTER`
- `FL_DND_DRAG`
- `FL_DND_LEAVE`
- `FL_DND_RELEASE`
- `FL_PASTE`

9.11 Making a subclass of Fl_Window

You may want your widget to be a subclass of [Fl_Window](#), [Fl_Double_Window](#), or [Fl_Gl_Window](#). This can be useful if your widget wants to occupy an entire window, and can also be used to take advantage of system-provided clipping, or to work with a library that expects a system window ID to indicate where to draw.

Subclassing [Fl_Window](#) is almost exactly like subclassing [Fl_Group](#), and in fact you can easily switch a subclass back and forth. Watch out for the following differences:

1. [Fl_Window](#) is a subclass of [Fl_Group](#) so *make sure your constructor calls* `end()` unless you actually want children added to your window.
2. When handling events and drawing, the upper-left corner is at 0,0, not `x(), y()` as in other [Fl_Widget](#)'s. For instance, to draw a box around the widget, call `draw_box(0, 0, w(), h())`, rather than `draw_box(x(), y(), w(), h())`.

You may also want to subclass [Fl_Window](#) in order to get access to different visuals or to change other attributes of the windows. See the [Operating System Issues](#) chapter for more information.

Chapter 10

Using OpenGL

This chapter discusses using FLTK for your OpenGL applications.

10.1 Using OpenGL in FLTK

The easiest way to make an OpenGL display is to subclass `Fl_Gl_Window`. Your subclass must implement a `draw()` method which uses OpenGL calls to draw the display. Your main program should call `redraw()` when the display needs to change, and (somewhat later) FLTK will call `draw()`.

With a bit of care you can also use OpenGL to draw into normal FLTK windows. This allows you to use Gouraud shading for drawing your widgets. To do this you use the `gl_start()` and `gl_finish()` functions around your OpenGL code.

You must include FLTK's `<FL/gl.h>` header file. It will include the file `<GL/gl.h>`, define some extra drawing functions provided by FLTK, and include the `<windows.h>` header file needed by WIN32 applications.

Some simple coding rules (see [OpenGL and 'retina' displays](#)) allow to write cross-platform code that will draw high resolution OpenGL graphics if run on 'retina' displays with Mac OS X.

10.2 Making a Subclass of `Fl_Gl_Window`

To make a subclass of `Fl_Gl_Window`, you must provide:

- A class definition.
- A `draw()` method.
- A `handle()` method if you need to receive input from the user.

If your subclass provides static controls in the window, they must be redrawn whenever the `FL_DAMAGE_ALL` bit is set in the value returned by `damage()`. For double-buffered windows you will need to surround the drawing code with the following code to make sure that both buffers are redrawn:

```
#ifndef MESA
glDrawBuffer(GL_FRONT_AND_BACK);
#endif // !MESA
... draw stuff here ...
#ifndef MESA
glDrawBuffer(GL_BACK);
#endif // !MESA
```

Note:

If you are using the Mesa graphics library, the call to `glDrawBuffer()` is not required and will slow down drawing considerably. The preprocessor instructions shown above will optimize your code based upon the graphics library used.

10.2.1 Defining the Subclass

To define the subclass you just subclass the `Fl_Gl_Window` class:

```
class MyWindow : public Fl_Gl_Window {
    void draw();
    int handle(int);
public:
    MyWindow(int X, int Y, int W, int H, const char *L)
        : Fl_Gl_Window(X, Y, W, H, L) {}
};
```

The `draw()` and `handle()` methods are described below. Like any widget, you can include additional private and public data in your class (such as scene graph information, etc.)

10.2.2 The draw() Method

The `draw()` method is where you actually do your OpenGL drawing:

```
void MyWindow::draw() {
    if (!valid()) {
        ... set up projection, viewport, etc ...
        ... window size is in w() and h().
        ... valid() is turned on by FLTK after draw() returns
    }
    ... draw ...
}
```

10.2.3 The handle() Method

The `handle()` method handles mouse and keyboard events for the window:

```
int MyWindow::handle(int event) {
    switch(event) {
    case FL_PUSH:
        ... mouse down event ...
        ... position in Fl::event_x() and Fl::event_y()
        return 1;
    case FL_DRAG:
        ... mouse moved while down event ...
        return 1;
    case FL_RELEASE:
        ... mouse up event ...
        return 1;
    case FL_FOCUS :
    case FL_UNFOCUS :
        ... Return 1 if you want keyboard events, 0 otherwise
        return 1;
    case FL_KEYBOARD:
        ... keypress, key is in Fl::event_key(), ascii in Fl::event_text()
        ... Return 1 if you understand/use the keyboard event, 0 otherwise...
        return 1;
    case FL_SHORTCUT:
        ... shortcut, key is in Fl::event_key(), ascii in Fl::event_text()
        ... Return 1 if you understand/use the shortcut event, 0 otherwise...
        return 1;
    default:
        // pass other events to the base class...
        return Fl_Gl_Window::handle(event);
    }
}
```

When `handle()` is called, the OpenGL context is not set up! If your display changes, you should call `redraw()` and let `draw()` do the work. Don't call any OpenGL drawing functions from inside `handle()`!

You can call *some* OpenGL stuff like hit detection and texture loading functions by doing:

```
case FL_PUSH:
    make_current(); // make OpenGL context current
    if (!valid()) {
        ... set up projection exactly the same as draw ...
        valid(1); // stop it from doing this next time
    }
    ... ok to call NON-DRAWING OpenGL code here, such as hit
    detection, loading textures, etc...
```

Your main program can now create one of your windows by doing `new MyWindow(...)`.

You can also use your new window class in FLUID by:

1. Putting your class definition in a `MyWindow.H` file.
2. Creating a `Fl_Box` widget in FLUID.
3. In the widget panel fill in the "class" field with `MyWindow`. This will make FLUID produce constructors for your new class.
4. In the "Extra Code" field put `#include "MyWindow.H"`, so that the FLUID output file will compile.

You must put `glwindow->show()` in your main code after calling `show()` on the window containing the OpenGL window.

10.3 Using OpenGL in Normal FLTK Windows

You can put OpenGL code into the `draw()` method, as described in [Drawing the Widget](#) in the previous chapter, or into the code for a `boxtype` or other places with some care.

Most importantly, before you show *any* windows, including those that don't have OpenGL drawing, you **must** initialize FLTK so that it knows it is going to use OpenGL. You may use any of the symbols described for `Fl_Gl_Window::mode()` to describe how you intend to use OpenGL:

```
Fl::gl_visual(FL_RGB);
```

You can then put OpenGL drawing code anywhere you can draw normally by surrounding it with `gl_start()` and `gl_finish()` to set up, and later release, an OpenGL context with an orthographic projection so that 0,0 is the lower-left corner of the window and each pixel is one unit. The current clipping is reproduced with OpenGL `glScissor()` commands. These functions also synchronize the OpenGL graphics stream with the drawing done by other X, WIN32, or FLTK functions.

```
gl_start();
... put your OpenGL code here ...
gl_finish();
```

The same context is reused each time. If your code changes the projection transformation or anything else you should use `glPushMatrix()` and `glPopMatrix()` functions to put the state back before calling `gl_finish()`.

You may want to use `Fl_Window::current()->h()` to get the drawable height so that you can flip the Y coordinates.

Unfortunately, there are a bunch of limitations you must adhere to for maximum portability:

- You must choose a default visual with `Fl::gl_visual()`.
- You cannot pass `FL_DOUBLE` to `Fl::gl_visual()`.
- You cannot use `Fl_Double_Window` or `Fl_Overlay_Window`.

Do *not* call `gl_start()` or `gl_finish()` when drawing into an `Fl_Gl_Window`!

10.4 OpenGL Drawing Functions

FLTK provides some useful OpenGL drawing functions. They can be freely mixed with any OpenGL calls, and are defined by including `<FL/gl.h>` which you should include instead of the OpenGL header `<GL/gl.h>`.

```
void gl_color(FL_Color)
```

Sets the current OpenGL color to a FLTK color. *For color-index modes it will use `fl_xpixel(c)`, which is only right if this window uses the default colormap!*

```
void gl_rect(int x, int y, int w, int h)
```

```
void gl_rectf(int x, int y, int w, int h)
```

Outlines or fills a rectangle with the current color. If `FL_GL_Window::ortho()` has been called, then the rectangle will exactly fill the pixel rectangle passed.

```
void gl_font(FL_Font fontid, int size)
```

Sets the current OpenGL font to the same font you get by calling `fl_font()`.

```
int gl_height()
```

```
int gl_descent()
```

```
float gl_width(const char *s)
```

```
float gl_width(const char *s, int n)
```

```
float gl_width(uchar c)
```

Returns information about the current OpenGL font.

```
void gl_draw(const char *s)
```

```
void gl_draw(const char *s, int n)
```

Draws a nul-terminated string or an array of `n` characters in the current OpenGL font at the current raster position.

```
void gl_draw(const char *s, int x, int y)
```

```
void gl_draw(const char *s, int n, int x, int y)
```

```
void gl_draw(const char *s, float x, float y)
```

```
void gl_draw(const char *s, int n, float x, float y)
```

Draws a nul-terminated string or an array of `n` characters in the current OpenGL font at the given position.

```
void gl_draw(const char *s, int x, int y, int w, int h, FL_Align)
```

Draws a string formatted into a box, with newlines and tabs expanded, other control characters changed to `^X`, and aligned with the edges or center. Exactly the same output as `fl_draw()`.

10.5 Speeding up OpenGL

Performance of [Fl_Gl_Window](#) may be improved on some types of OpenGL implementations, in particular MESA and other software emulators, by setting the `GL_SWAP_TYPE` environment variable. This variable declares what is in the backbuffer after you do a `swapbuffers`.

- `setenv GL_SWAP_TYPE COPY`

This indicates that the back buffer is copied to the front buffer, and still contains its old data. This is true of many hardware implementations. Setting this will speed up emulation of overlays, and widgets that can do partial update can take advantage of this as `damage()` will not be cleared to -1.

- `setenv GL_SWAP_TYPE NODAMAGE`

This indicates that nothing changes the back buffer except drawing into it. This is true of MESA and Win32 software emulation and perhaps some hardware emulation on systems with lots of memory.

- All other values for `GL_SWAP_TYPE`, and not setting the variable, cause FLTK to assume that the back buffer must be completely redrawn after a swap.

This is easily tested by running the [gl_overlay](#) demo program and seeing if the display is correct when you drag another window over it or if you drag the window off the screen and back on. You have to exit and run the program again for it to see any changes to the environment variable.

10.6 Using OpenGL Optimizer with FLTK

[OpenGL Optimizer](#) is a scene graph toolkit for OpenGL available from Silicon Graphics for IRIX and Microsoft Windows. It allows you to view large scenes without writing a lot of OpenGL code.

OptimizerWindow Class Definition

To use [OpenGL Optimizer](#) with FLTK you'll need to create a subclass of `Fl_Gl_Widget` that includes several state variables:

```
class OptimizerWindow : public Fl_Gl_Window {
  csContext *context_; // Initialized to 0 and set by draw()...
  csDrawAction *draw_action_; // Draw action...
  csGroup *scene_; // Scene to draw...
  csCamera *camera_; // Viewport for scene...
  void draw();
public:
  OptimizerWindow(int X, int Y, int W, int H, const char *L)
    : Fl_Gl_Window(X, Y, W, H, L) {
    context_ = (csContext *)0;
    draw_action_ = (csDrawAction *)0;
    scene_ = (csGroup *)0;
    camera_ = (csCamera *)0;
  }
  void scene(csGroup *g) { scene_ = g; redraw(); }
  void camera(csCamera *c) {
    camera_ = c;
    if (context_) {
      draw_action_->setCamera(camera_);
      camera_>draw(draw_action_);
      redraw();
    }
  }
};
```

The camera() Method

The `camera()` method sets the camera (projection and viewpoint) to use when drawing the scene. The scene is redrawn after this call.

The draw() Method

The `draw()` method performs the needed initialization and does the actual drawing:

```
void OptimizerWindow::draw() {
    if (!context_) {
        // This is the first time we've been asked to draw; create the
        // Optimizer context for the scene...
#ifdef WIN32
        context_ = new csContext((HDC)fl_getHDC());
        context_>ref();
        context_>makeCurrent((HDC)fl_getHDC());
#else
        context_ = new csContext(fl_display, fl_visual);
        context_>ref();
        context_>makeCurrent(fl_display, fl_window);
#endif // WIN32
        ... perform other context setup as desired ...
        // Then create the draw action to handle drawing things...
        draw_action_ = new csDrawAction;
        if (camera_) {
            draw_action_>setCamera(camera_);
            camera_>draw(draw_action_);
        }
    } else {
#ifdef WIN32
        context_>makeCurrent((HDC)fl_getHDC());
#else
        context_>makeCurrent(fl_display, fl_window);
#endif // WIN32
    }
    if (!valid()) {
        // Update the viewport for this context...
        context_>setViewport(0, 0, w(), h());
    }
    // Clear the window...
    context_>clear(csContext::COLOR_CLEAR | csContext::DEPTH_CLEAR,
                 0.0f,          // Red
                 0.0f,          // Green
                 0.0f,          // Blue
                 1.0f);         // Alpha
    // Then draw the scene (if any)...
    if (scene_)
        draw_action_>apply(scene_);
}
```

The scene() Method

The `scene()` method sets the scene to be drawn. The scene is a collection of 3D objects in a `csGroup`. The scene is redrawn after this call.

10.7 Using OpenGL 3.0 (or higher versions)

The examples subdirectory contains `OpenGL3test.cxx`, a toy program showing how to use OpenGL 3.0 (or higher versions) with FLTK in a cross-platform fashion. It contains also `OpenGL3-glut-test.cxx` which shows how to use FLTK's GLUT compatibility and OpenGL 3.

To access OpenGL 3.0 (or higher versions), use the `FL_OPENGL3` flag when calling `FL_Gl_Window::mode(int a)` or `glutInitDisplayMode()`.

On the Windows and Unix/Linux platforms, FLTK creates contexts implementing the highest OpenGL version supported by the hardware. Such contexts may also be compatible with lower OpenGL versions. Access to functions from OpenGL versions above 1.1 requires to load function pointers at runtime on these platforms. FLTK recommends to use the GLEW library to perform this. It is therefore necessary to install the GLEW library (see below).

On the macOS platform, MacOS 10.7 or above is required; GLEW is possible but not necessary. FLTK creates contexts for OpenGL versions 1 and 2 without the `FL_OPENGL3` flag and for OpenGL versions 3.2 and above with it.

GLEW installation (Unix/Linux and MSWindows platforms)

GLEW is available as a package for most Linux distributions and in source form at <http://glew.sourceforge.net/>. For the MSWindows platform, a Visual Studio static library (`glew32.lib`) can be downloaded from the same web site; a MinGW-style static library (`libglew32.a`) can be built from source with the make command.

Source-level changes for OpenGL 3:

- Put this in all OpenGL-using source files (instead of `#include <FL/gl.h>`, and before `#include <FL/glut.h>` if you use GLUT):

```
#if defined(__APPLE__)
#   include <OpenGL/gl3.h> // defines OpenGL 3.0+ functions
#else
#   if defined(WIN32)
#       define GLEW_STATIC 1
#   endif
#   include <GL/glew.h>
#endif
```

- Add the `FL_OPENGL3` flag when calling `FL_Gl_Window::mode(int a)` or `glutInitDisplayMode()`.
- Put this in the `handle(int event)` member function of the first to be created among your `FL_Gl_Window`-derived classes:

```
#ifndef __APPLE__
    static int first = 1;
    if (first && event == FL_SHOW && shown()) {
        first = 0;
        make_current();
        glewInit(); // defines pters to functions of OpenGL V 1.2 and above
    }
#endif
```

- Alternatively, if you use GLUT, put

```
#ifndef __APPLE__
    glewInit(); // defines pters to functions of OpenGL V 1.2 and above
#endif
```

after the first `glutCreateWindow()` call.

If GLEW is installed on the Mac OS development platform, it is possible to use the same code for all platforms, with one exception: put

```
#ifndef __APPLE__
    glewExperimental = GL_TRUE;
#endif
```

before the `glewInit()` call.

Changes in the build process

Link with `libGLEW.so` (on Unix/Linux), `libglew32.a` (with MinGW) or `glew32.lib` (with MS Visual Studio); no change is needed on the Mac OS platform.

Chapter 11

Programming with FLUID

This chapter shows how to use the Fast Light User-Interface Designer ("FLUID") to create your GUIs.

Subchapters:

- [What is FLUID?](#)
- [Running FLUID Under UNIX](#)
- [Running FLUID Under Microsoft Windows](#)
- [Compiling .fl files](#)
- [A Short Tutorial](#)
- [FLUID Reference](#)
- [Internationalization with FLUID](#)
- [Known limitations](#)

11.1 What is FLUID?

The Fast Light User Interface Designer, or FLUID, is a graphical editor that is used to produce FLTK source code. FLUID edits and saves its state in `.fl` files. These files are text, and you can (with care) edit them in a text editor, perhaps to get some special effects.

FLUID can "compile" the `.fl` file into a `.cxx` and a `.h` file. The `.cxx` file defines all the objects from the `.fl` file and the `.h` file declares all the global ones. FLUID also supports localization ([Internationalization](#)) of label strings using message files and the GNU gettext or POSIX catgets interfaces.

A simple program can be made by putting all your code (including a `main()` function) into the `.fl` file and thus making the `.cxx` file a single source file to compile. Most programs are more complex than this, so you write other `.cxx` files that call the FLUID functions. These `.cxx` files must `#include` the `.h` file or they can `#include` the `.cxx` file so it still appears to be a single source file.

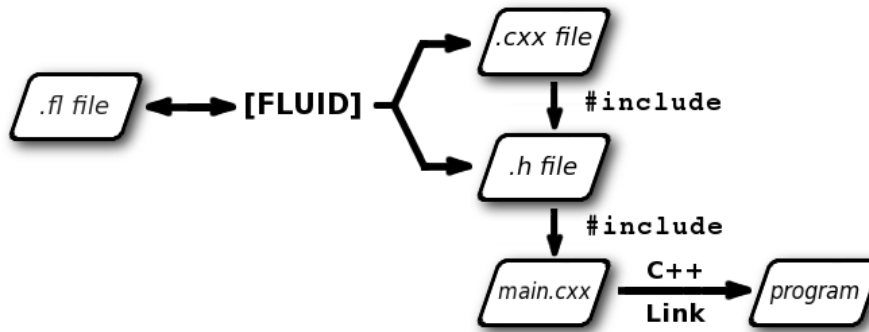


Figure 11.1 FLUID organization

Normally the FLUID file defines one or more functions or classes which output C++ code. Each function defines one or more FLTK windows, and all the widgets that go inside those windows.

Widgets created by FLUID are either "named", "complex named" or "unnamed". A named widget has a legal C++ variable identifier as its name (i.e. only alphanumeric and underscore). In this case FLUID defines a global variable or class member that will point at the widget after the function defining it is called. A complex named object has punctuation such as ' . ' or ' -> ' or any other symbols in its name. In this case FLUID assigns a pointer to the widget to the name, but does not attempt to declare it. This can be used to get the widgets into structures. An unnamed widget has a blank name and no pointer is stored.

Widgets may either call a named callback function that you write in another source file, or you can supply a small piece of C++ source and FLUID will write a private callback function into the .cxx file.

11.2 Running FLUID Under UNIX

To run FLUID under UNIX, type:

```
fluid filename.fl &
```

to edit the .fl file filename.fl. If the file does not exist you will get an error pop-up, but if you dismiss it you will be editing a blank file of that name. You can run FLUID without any name, in which case you will be editing an unnamed blank setup (but you can use save-as to write it to a file).

You can provide any of the standard FLTK switches before the filename:

```
-display host:n.n
-geometry WxH+X+Y
-title windowtitle
-name classname
-iconic
-fg color
-bg color
-bg2 color
-scheme schemename
```

Changing the colors may be useful to see what your interface will look at if the user calls it with the same switches. Similarly, using "-scheme plastic" will show how the interface will look using the "plastic" scheme.

In the current version, if you don't put FLUID into the background with ' & ' then you will be able to abort FLUID by typing CTRL-C on the terminal. It will exit immediately, losing any changes.

11.3 Running FLUID Under Microsoft Windows

To run FLUID under WIN32, double-click on the *FLUID.exe* file. You can also run FLUID from the Command Prompt window. FLUID always runs in the background under WIN32.

11.4 Compiling .fl files

FLUID can also be called as a command-line "compiler" to create the .cxx and .h file from a .fl file. To do this type:

```
fluid -c filename.fl
```

This is the same as the menu 'File/Write Code...'. It will read the `filename.fl` file and write `filename.cxx` and `filename.h`. Any leading directory on `filename.fl` will be stripped, so they are always written to the current directory. If there are any errors reading or writing the files, FLUID will print the error and exit with a non-zero code. You can use the following lines in a makefile to automate the creation of the source and header files:

```
my_panels.h my_panels.cxx: my_panels.fl
    fluid -c my_panels.fl
```

Most versions of make support rules that cause .fl files to be compiled:

```
.SUFFIXES: .fl .cxx .h
.fl.h .fl.cxx:
    fluid -c $<
```

If you use

```
fluid -cs filename.fl
```

FLUID will also write the "strings" for internationalization in file 'filename.txt' (menu: 'File/Write Strings...').

Finally there is another option which is useful for program developers who have many .fl files and want to upgrade them to the current FLUID version. FLUID will read the `filename.fl` file, save it, and exit immediately. This writes the file with current syntax and options and the current FLTK version in the header of the file. Use

```
fluid -u filename.fl
```

to 'upgrade' `filename.fl`. You may combine this with '-c' or '-cs'.

Note

All these commands overwrite existing files w/o warning. You should particularly take care when running 'fluid -u' since this overwrites the original .fl source file.

11.5 A Short Tutorial

FLUID is an amazingly powerful little program. However, this power comes at a price as it is not always obvious how to accomplish seemingly simple tasks with it. This tutorial will show you how to generate a complete user interface class with FLUID that is used for the CubeView program provided with FLTK.

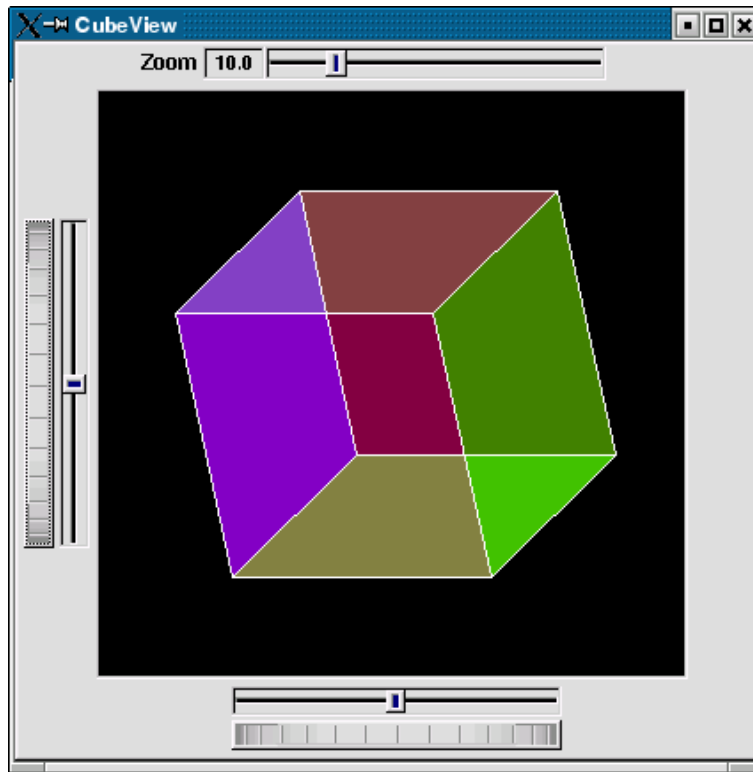


Figure 11.2 CubeView demo

The window is of class `CubeViewUI`, and is completely generated by FLUID, including class member functions. The central display of the cube is a separate subclass of `FI_GI_Window` called `CubeView`. `CubeViewUI` manages `CubeView` using callbacks from the various sliders and rollers to manipulate the viewing angle and zoom of `CubeView`.

At the completion of this tutorial you will (hopefully) understand how to:

1. Use FLUID to create a complete user interface class, including constructor and any member functions necessary.
2. Use FLUID to set callbacks member functions of a custom widget classes.
3. Subclass an `FI_GI_Window` to suit your purposes.

11.5.1 The CubeView Class

The `CubeView` class is a subclass of `FI_GI_Window`. It has methods for setting the zoom, the x and y pan, and the rotation angle about the x and y axes.

You can safely skip this section as long as you realize that `CubeView` is a subclass of `FI_GI_Window` and will respond to calls from `CubeViewUI`, generated by FLUID.

The CubeView Class Definition

Here is the CubeView class definition, as given by its header file "test/CubeView.h":

```
class CubeView : public Fl_Gl_Window {
public:
    CubeView(int x,int y,int w,int h,const char *l=0);
    // this value determines the scaling factor used to draw the cube.
    double size;
    /* Set the rotation about the vertical ( y ) axis.
 *
 * This function is called by the horizontal roller in CubeViewUI
 * and the initialize button in CubeViewUI.
 */
    void v_angle(float angle){vAng=angle;};
    // Return the rotation about the vertical ( y ) axis.
    float v_angle(){return vAng;};
    /* Set the rotation about the horizontal ( x ) axis.
 *
 * This function is called by the vertical roller in CubeViewUI
 * and the
 * initialize button in CubeViewUI.
 */
    void h_angle(float angle){hAng=angle;};
    // the rotation about the horizontal ( x ) axis.
    float h_angle(){return hAng;};
    /* Sets the x shift of the cube view camera.
 *
 * This function is called by the slider in CubeViewUI and the
 * initialize button in CubeViewUI.
 */
    void panx(float x){xshift=x;};
    /* Sets the y shift of the cube view camera.
 *
 * This function is called by the slider in CubeViewUI and the
 * initialize button in CubeViewUI.
 */
    void pany(float y){yshift=y;};
    /* The widget class draw() override.
 * The draw() function initialize Gl for another round of
 * drawing then calls specialized functions for drawing each
 * of the entities displayed in the cube view.
 */
    void draw();
private:
    /* Draw the cube boundaries
 * Draw the faces of the cube using the boxv[] vertices, using
 * GL_LINE_LOOP for the faces. The color is #defined by
 * CUBECOLOR.
 */
    void drawCube();

    float vAng,hAng; float xshift,yshift;
    float boxv0[3];float boxv1[3]; float boxv2[3];float boxv3[3];
    float boxv4[3];float boxv5[3]; float boxv6[3];float boxv7[3];
};
```

The CubeView Class Implementation

Here is the CubeView implementation. It is very similar to the "cube" demo included with FLTK.

```
#include "CubeView.h"
#include <math.h>
CubeView::CubeView(int x,int y,int w,int h,const char *l)
    : Fl_Gl_Window(x,y,w,h,l)
{
    vAng = 0.0; hAng=0.0; size=10.0;
    /* The cube definition. These are the vertices of a unit cube
 * centered on the origin.*/
    boxv0[0] = -0.5; boxv0[1] = -0.5; boxv0[2] = -0.5; boxv1[0] = 0.5;
    boxv1[1] = -0.5; boxv1[2] = -0.5; boxv2[0] = 0.5; boxv2[1] = 0.5;
    boxv2[2] = -0.5; boxv3[0] = -0.5; boxv3[1] = 0.5; boxv3[2] = -0.5;
    boxv4[0] = -0.5; boxv4[1] = -0.5; boxv4[2] = 0.5; boxv5[0] = 0.5;
    boxv5[1] = -0.5; boxv5[2] = 0.5; boxv6[0] = 0.5; boxv6[1] = 0.5;
    boxv6[2] = 0.5; boxv7[0] = -0.5; boxv7[1] = 0.5; boxv7[2] = 0.5;
};
// The color used for the edges of the bounding cube.
#define CUBECOLOR 255,255,255,255
void CubeView::drawCube() {
```

```

/* Draw a colored cube */
#define ALPHA 0.5
glShadeModel(GL_FLAT);
glBegin(GL_QUADS);
    glColor4f(0.0, 0.0, 1.0, ALPHA);
    glVertex3fv(boxv0);
    glVertex3fv(boxv1);
    glVertex3fv(boxv2);
    glVertex3fv(boxv3);
    glColor4f(1.0, 1.0, 0.0, ALPHA);
    glVertex3fv(boxv0);
    glVertex3fv(boxv4);
    glVertex3fv(boxv5);
    glVertex3fv(boxv1);
    glVertex3fv(boxv2);
    glColor4f(0.0, 1.0, 1.0, ALPHA);
    glVertex3fv(boxv2);
    glVertex3fv(boxv6);
    glVertex3fv(boxv7);
    glVertex3fv(boxv3);
    glColor4f(1.0, 0.0, 0.0, ALPHA);
    glVertex3fv(boxv4);
    glVertex3fv(boxv5);
    glVertex3fv(boxv6);
    glVertex3fv(boxv7);
    glColor4f(1.0, 0.0, 1.0, ALPHA);
    glVertex3fv(boxv0);
    glVertex3fv(boxv3);
    glVertex3fv(boxv7);
    glVertex3fv(boxv4);
    glColor4f(0.0, 1.0, 0.0, ALPHA);
    glVertex3fv(boxv1);
    glVertex3fv(boxv5);
    glVertex3fv(boxv6);
    glVertex3fv(boxv2);
glEnd();
glColor3f(1.0, 1.0, 1.0);
glBegin(GL_LINES);
    glVertex3fv(boxv0);
    glVertex3fv(boxv1);
    glVertex3fv(boxv1);
    glVertex3fv(boxv2);
    glVertex3fv(boxv2);
    glVertex3fv(boxv3);
    glVertex3fv(boxv3);
    glVertex3fv(boxv0);
    glVertex3fv(boxv4);
    glVertex3fv(boxv5);
    glVertex3fv(boxv5);
    glVertex3fv(boxv6);
    glVertex3fv(boxv6);
    glVertex3fv(boxv7);
    glVertex3fv(boxv7);
    glVertex3fv(boxv4);
    glVertex3fv(boxv0);
    glVertex3fv(boxv4);
    glVertex3fv(boxv1);
    glVertex3fv(boxv5);
    glVertex3fv(boxv2);
    glVertex3fv(boxv6);
    glVertex3fv(boxv3);
    glVertex3fv(boxv7);
glEnd();
}; //drawCube
void CubeView::draw() {
    if (!valid()) {
        glLoadIdentity(); glViewport(0,0,w(),h());
        glOrtho(-10,10,-10,10,-20000,10000); glEnable(GL_BLEND);
        glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA);
    }
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glPushMatrix(); glTranslatef(xshift, yshift, 0);
    glRotatef(hAng,0,1,0); glRotatef(vAng,1,0,0);
    glScalef(float(size),float(size),float(size)); drawCube();
    glPopMatrix();
};

```

11.5.2 The CubeViewUI Class

We will completely construct a window to display and control the CubeView defined in the previous section using FLUID.

Defining the CubeViewUI Class

Once you have started FLUID, the first step in defining a class is to create a new class within FLUID using the **New->Code->Class** menu item. Name the class "CubeViewUI" and leave the subclass blank. We do not need any inheritance for this window. You should see the new class declaration in the FLUID browser window.

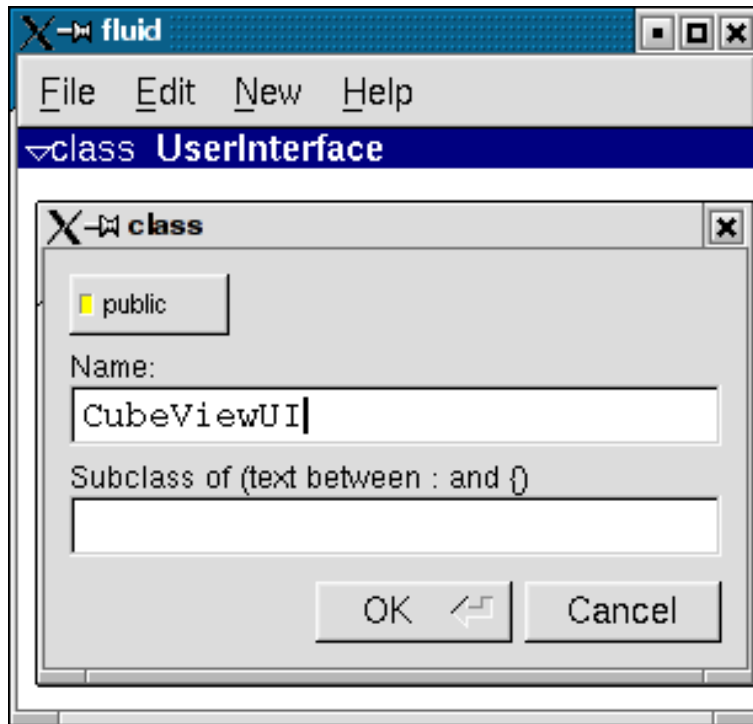


Figure 11.3 FLUID file for CubeView

Adding the Class Constructor

Click on the CubeViewUI class in the FLUID window and add a new method by selecting **New->Code->Function/Method**. The name of the function will also be CubeViewUI. FLUID will understand that this will be the constructor for the class and will generate the appropriate code. Make sure you declare the constructor public.

Then add a window to the CubeViewUI class. Highlight the name of the constructor in the FLUID browser window and click on **New->Group->Window**. In a similar manner add the following to the CubeViewUI constructor:

- A horizontal roller named `hrot`
- A vertical roller named `vrot`
- A horizontal slider named `xpan`
- A vertical slider named `ypan`
- A horizontal value slider named `zoom`

None of these additions need be public. And they shouldn't be unless you plan to expose them as part of the interface for CubeViewUI.

When you are finished you should have something like this:

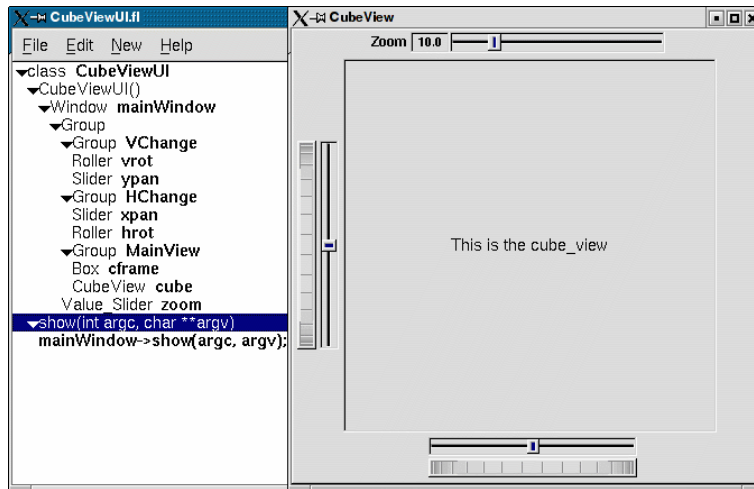


Figure 11.4 FLUID window containing CubeView demo

We will talk about the `show()` method that is highlighted shortly.

Adding the CubeView Widget

What we have is nice, but does little to show our cube. We have already defined the `CubeView` class and we would like to show it within the `CubeViewUI`.

The `CubeView` class inherits the `FI_GI_Window` class, which is created in the same way as a `FI_Box` widget. Use **New->Other->Box** to add a square box to the main window. This will be no ordinary box, however.

The Box properties window will appear. The key to letting `CubeViewUI` display `CubeView` is to enter `CubeView` in the **Class:** text entry box. This tells FLUID that it is not an `FI_Box`, but a similar widget with the same constructor.

In the **Extra Code:** field enter `#include "CubeView.h"`

This `#include` is important, as we have just included `CubeView` as a member of `CubeViewUI`, so any public `CubeView` methods are now available to `CubeViewUI`.

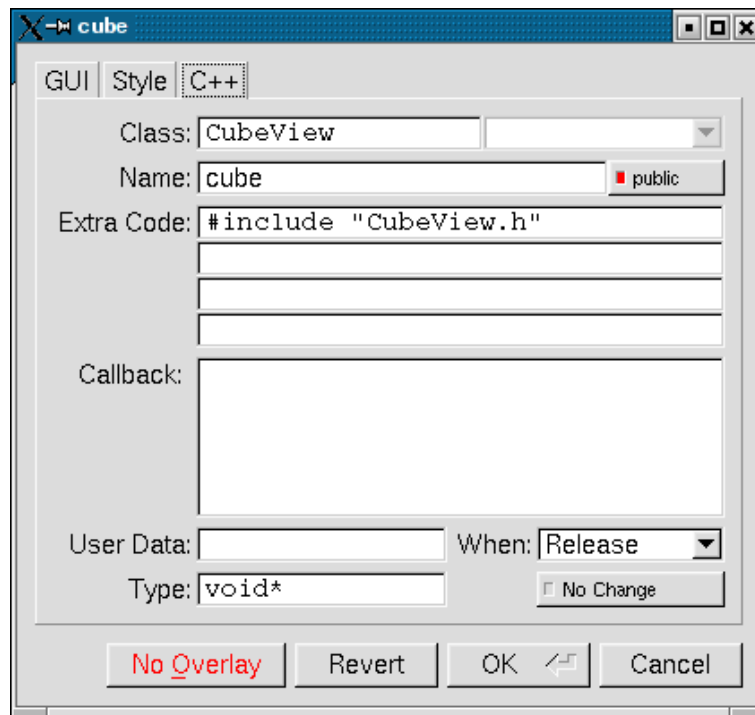


Figure 11.5 CubeView methods

Defining the Callbacks

Each of the widgets we defined before adding CubeView can have callbacks that call CubeView methods. You can call an external function or put in a short amount of code in the **Callback** field of the widget panel. For example, the callback for the `ypan` slider is:

```
cube->pany(((Fl_Slider *)o)->value());
cube->redraw();
```

We call `cube->redraw()` after changing the value to update the CubeView window. CubeView could easily be modified to do this, but it is nice to keep this exposed. In the case where you may want to do more than one view change only redrawing once saves a lot of time.

There is no reason to wait until after you have added CubeView to enter these callbacks. FLUID assumes you are smart enough not to refer to members or functions that don't exist.

Adding a Class Method

You can add class methods within FLUID that have nothing to do with the GUI. As an example add a `show` function so that CubeViewUI can actually appear on the screen.

Make sure the top level CubeViewUI is selected and select **New->Code->Function/Method**. Just use the name `show()`. We don't need a return value here, and since we will not be adding any widgets to this method FLUID will assign it a return type of `void`.

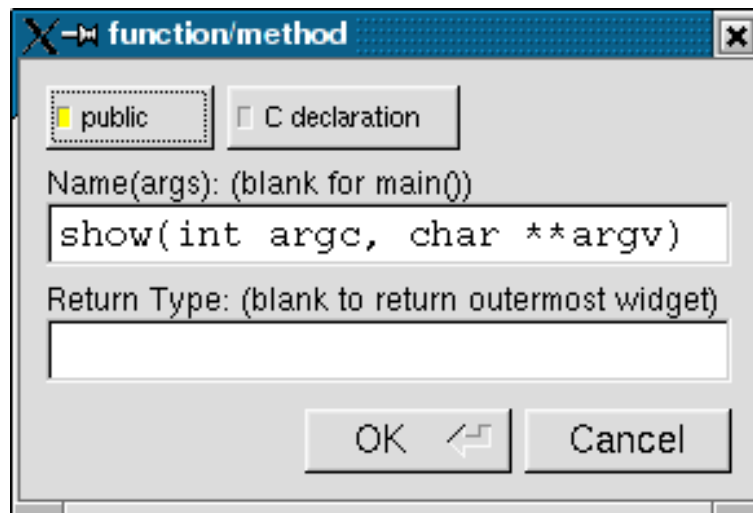


Figure 11.6 CubeView constructor

Once the new method has been added, highlight its name and select **New->Code->Code**. Enter the method's code in the code window.

11.5.3 Adding Constructor Initialization Code

If you need to add code to initialize a class, for example setting initial values of the horizontal and vertical angles in the CubeView, you can simply highlight the constructor and select **New->Code->Code**. Add any required code.

11.5.4 Generating the Code

Now that we have completely defined the CubeViewUI, we have to generate the code. There is one last trick to ensure this all works. Open the preferences dialog from **Edit->Preferences**.

At the bottom of the preferences dialog box is the key: **"Include Header from Code"**. Select that option and set your desired file extensions and you are in business. You can include the CubeViewUI.h (or whatever extension you prefer) as you would any other C++ class.

11.6 FLUID Reference

The following sections describe each of the windows in FLUID.

11.6.1 The Widget Browser

The main window shows a menu bar and a scrolling browser of all the defined widgets. The name of the `.fl` file being edited is shown in the window title.

The widgets are stored in a hierarchy. You can open and close a level by clicking the "triangle" at the left of a widget. The leftmost widgets are the *parents*, and all the widgets listed below them are their *children*. Parents don't have to have any children.

The top level of the hierarchy is composed of *functions* and *classes*. Each of these will produce a single C++ public function or class in the output `.cxx` file. Calling the function or instantiating the class will create all of the child widgets.

The second level of the hierarchy contains the *windows*. Each of these produces an instance of class `FI_Window`.

Below that are either *widgets* (subclasses of `FI_Widget`) or *groups* of widgets (including other groups). Plain groups are for layout, navigation, and resize purposes. *Tab groups* provide the well-known file-card tab interface.

Widgets are shown in the browser by either their *name* (such as "main_panel" in the example), or by their *type* and *label* (such as "Button "the green"").

You *select* widgets by clicking on their names, which highlights them (you can also select widgets from any displayed window). You can select many widgets by dragging the mouse across them, or by using Shift+Click to toggle them on and off. To select no widgets, click in the blank area under the last widget. Note that hidden children may be selected even when there is no visual indication of this.

You *open* widgets by double-clicking on them, or (to open several widgets you have picked) by typing the F1 key. A control panel will appear so you can change the widget(s).

11.6.2 Menu Items

The menu bar at the top is duplicated as a pop-up menu on any displayed window. The shortcuts for all the menu items work in any window. The menu items are:

File/Open... (Ctrl+o)

Discards the current editing session and reads in a different `.fl` file. You are asked for confirmation if you have changed the current file.

FLUID can also read `.fd` files produced by the Forms and XForms "fdesign" programs. It is best to File/↔ Merge them instead of opening them. FLUID does not understand everything in a `.fd` file, and will print a warning message on the controlling terminal for all data it does not understand. You will probably need to edit the resulting setup to fix these errors. Be careful not to save the file without changing the name, as FLUID will write over the `.fd` file with its own format, which fdesign cannot read!

File/Insert... (Ctrl+i)

Inserts the contents of another `.fl` file, without changing the name of the current `.fl` file. All the functions (even if they have the same names as the current ones) are added, and you will have to use cut/paste to put the widgets where you want.

File/Save (Ctrl+s)

Writes the current data to the `.fl` file. If the file is unnamed then FLUID will ask for a filename.

File/Save As... (Ctrl+Shift+S)

Asks for a new filename and saves the file.

File/Write Code (Ctrl+Shift+C)

"Compiles" the data into a `.cxx` and `.h` file. These are exactly the same as the files you get when you run FLUID with the `-c` switch.

The output file names are the same as the `.fl` file, with the leading directory and trailing ".fl" stripped, and ".h" or ".cxx" appended.

File/Write Strings (Ctrl+Shift+W)

Writes a message file for all of the text labels defined in the current file.

The output file name is the same as the `.fl` file, with the leading directory and trailing ".fl" stripped, and ".txt", ".po", or ".msg" appended depending on the [Internationalization Mode](#).

File/Quit (Ctrl+q)

Exits FLUID. You are asked for confirmation if you have changed the current file.

Edit/Undo (Ctrl+z)

This isn't implemented yet. You should do save often so you can recover from any mistakes you make.

Edit/Cut (Ctrl+x)

Deletes the selected widgets and all of their children. These are saved to a "clipboard" file and can be pasted back into any FLUID window.

Edit/Copy (Ctrl+c)

Copies the selected widgets and all of their children to the "clipboard" file.

Edit/Paste (Ctrl+v)

Pastes the widgets from the clipboard file.

If the widget is a window, it is added to whatever function is selected, or contained in the current selection.

If the widget is a normal widget, it is added to whatever window or group is selected. If none is, it is added to the window or group that is the parent of the current selection.

To avoid confusion, it is best to select exactly one widget before doing a paste.

Cut/paste is the only way to change the parent of a widget.

Edit/Select All (Ctrl+a)

Selects all widgets in the same group as the current selection.

If they are all selected already then this selects all widgets in that group's parent. Repeatedly typing Ctrl+a will select larger and larger groups of widgets until everything is selected.

Edit/Open... (F1 or double click)

Displays the current widget in the attributes panel. If the widget is a window and it is not visible then the window is shown instead.

Edit/Sort

Sorts the selected widgets into left to right, top to bottom order. You need to do this to make navigation keys in FLTK work correctly. You may then fine-tune the sorting with "Earlier" and "Later". This does not affect the positions of windows or functions.

Edit/Earlier (F2)

Moves all of the selected widgets one earlier in order among the children of their parent (if possible). This will affect navigation order, and if the widgets overlap it will affect how they draw, as the later widget is drawn on top of the earlier one. You can also use this to reorder functions, classes, and windows within functions.

Edit/Later (F3)

Moves all of the selected widgets one later in order among the children of their parent (if possible).

Edit/Group (F7)

Creates a new [FL_Group](#) and make all the currently selected widgets children of it.

Edit/Ungroup (F8)

Deletes the parent group if all the children of a group are selected.

Edit/Overlays on/off (Ctrl+Shift+O)

Toggles the display of the red overlays off, without changing the selection. This makes it easier to see box borders and how the layout looks. The overlays will be forced back on if you change the selection.

Edit/Project Settings... (Alt+p)

Displays the project settings panel.

Under the "Output" tab you control the extensions or names of the files that are generated by FLUID. If you check the "Include Header from Code" button the code file will include the header file automatically.

Under the "Internationalization" tab are the [internationalization](#) options, described later in this chapter.

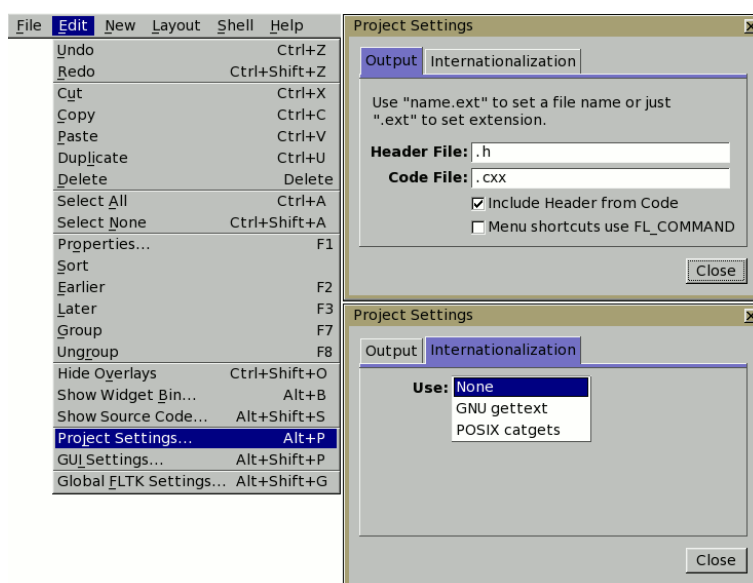


Figure 11.7 FLUID Project Settings Window

Edit/GUI Settings... (Shift+Alt+p)

Displays the GUI Settings panel, used to control the user interface settings.

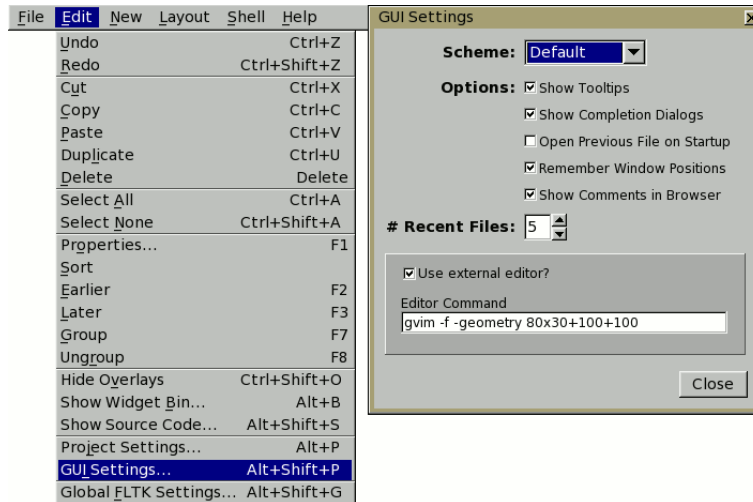


Figure 11.8 FLUID GUI Settings Window

Edit/Global FLTK Settings... (Shift+Alt+g)

Displays the FLTK Global Settings ("Preferences") panel, used to control fluid's user specific and/or system wide settings.

Tooltips provide descriptions of each option.

At the lower-right, "User Settings" causes changes to only affect the current user, "System Settings" causes changes to be applied to all users on the current machine.

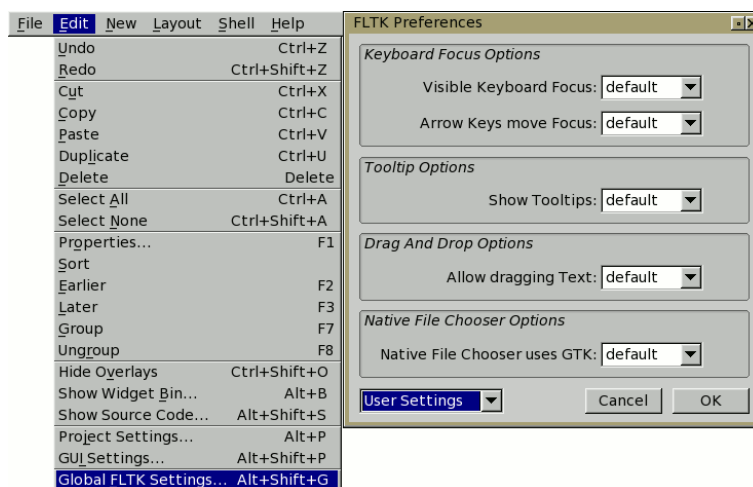


Figure 11.9 FLUID Global Settings Window

New/Code/Function

Creates a new C function. You will be asked for a name for the function. This name should be a legal C++ function template, without the return type. You can pass arguments which can be referred to by code you type into the individual widgets.

If the function contains any unnamed windows, it will be declared as returning a [Fl_Window](#) pointer. The unnamed window will be returned from it (more than one unnamed window is useless). If the function contains only named windows, it will be declared as returning nothing (`void`).

It is possible to make the `.cxx` output be a self-contained program that can be compiled and executed. This is done by deleting the function name so `main(argc, argv)` is used. The function will call `show()` on all the windows it creates and then call `Fl::run()`. This can also be used to test resize behavior or other parts of the user interface.

You can change the function name by double-clicking on the function.

New/Window

Creates a new [Fl_Window](#) widget. The window is added to the currently selected function, or to the function containing the currently selected item. The window will appear, sized to 100x100. You can resize it to whatever size you require.

The widget panel will also appear and is described later in this chapter.

New/...

All other items on the New menu are subclasses of [Fl_Widget](#). Creating them will add them to the currently selected group or window, or the group or window containing the currently selected widget. The initial dimensions and position are chosen by copying the current widget, if possible.

When you create the widget you will get the widget's control panel, which is described later in this chapter.

Layout/Align/...

Align all selected widgets to the first widget in the selection.

Layout/Space Evenly/...

Space all selected widgets evenly inside the selected space. Widgets will be sorted from first to last.

Layout/Make Same Size/...

Make all selected widgets the same size as the first selected widget.

Layout/Center in Group/...

Center all selected widgets relative to their parent widget

Layout/Grid and Size Settings... (Ctrl+g)

Displays the grid settings panel.

This panel controls the grid that all widgets snap to when you move and resize them, and for the "snap" which is how far a widget has to be dragged from its original position to actually change.

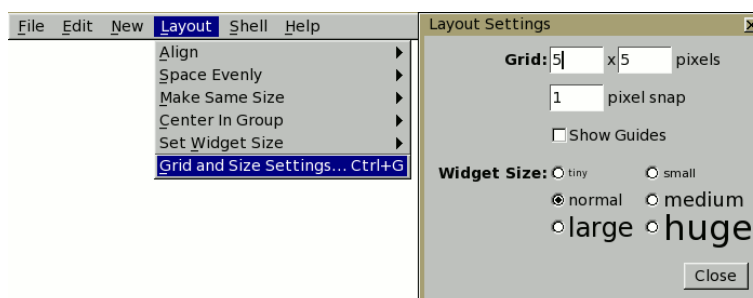


Figure 11.10 FLUID Layout/Grid Settings Window

Shell/Execute Command... (Alt+x)

Displays the shell command panel. The shell command is commonly used to run a 'make' script to compile the FLTK output.

Shell/Execute Again (Alt+g)

Run the shell command again.

Help/About FLUID

Pops up a panel showing the version of FLUID.

Help/On FLUID

Shows this chapter of the manual.

Help/Manual

Shows the contents page of the manual

11.6.3 The Widget Panel

When you double-click on a widget or a set of widgets you will get the "widget attribute panel".

When you change attributes using this panel, the changes are reflected immediately in the window. It is useful to hit the "no overlay" button (or type Ctrl+Shift+O) to hide the red overlay so you can see the widgets more accurately, especially when setting the box type.

If you have several widgets selected, they may have different values for the fields. In this case the value for *one* of the widgets is shown. But if you change this value, *all* of the selected widgets are changed to the new value.

Hitting "OK" makes the changes permanent. Selecting a different widget also makes the changes permanent. FLUID checks for simple syntax errors such as mismatched parenthesis in any code before saving any text.

"Revert" or "Cancel" put everything back to when you last brought up the panel or hit OK. However in the current version of FLUID, changes to "visible" attributes (such as the color, label, box) are not undone by revert or cancel. Changes to code like the callbacks are undone, however.

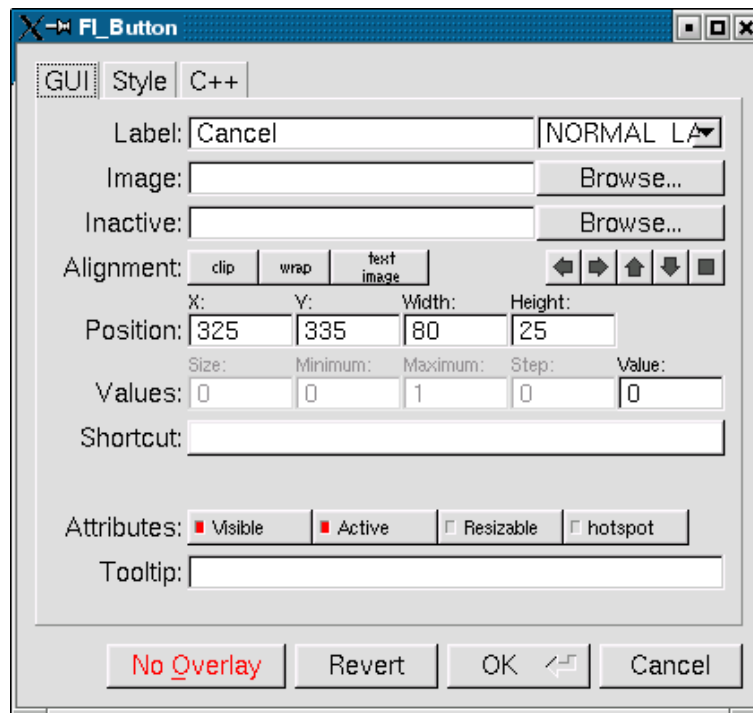


Figure 11.11 The FLUID widget GUI attributes

11.7 GUI Attributes

Label (text field)

String to print next to or inside the button. You can put newlines into the string to make multiple lines. The easiest way is by typing Ctrl+j.

[Symbols](#) can be added to the label using the at sign ("@").

Label (pull down menu)

How to draw the label. Normal, shadowed, engraved, and embossed change the appearance of the text.

Image

The active image for the widget. Click on the **Browse...** button to pick an image file using the file chooser.

Inactive

The inactive image for the widget. Click on the **Browse...** button to pick an image file using the file chooser.

Alignment (buttons)

Where to draw the label. The arrows put it on that side of the widget, you can combine them to put it in the corner. The "box" button puts the label inside the widget, rather than outside.

The **clip** button clips the label to the widget box, the **wrap** button wraps any text in the label, and the **text image** button puts the text over the image instead of under the image.

Position (text fields)

The position fields show the current position and size of the widget box. Enter new values to move and/or resize a widget.

Values (text fields)

The values and limits of the current widget. Depending on the type of widget, some or all of these fields may be inactive.

Shortcut

The shortcut key to activate the widget. Click on the shortcut button and press any key sequence to set the shortcut.

Attributes (buttons)

The **Visible** button controls whether the widget is visible (on) or hidden (off) initially. Don't change this for windows or for the immediate children of a Tabs group.

The **Active** button controls whether the widget is activated (on) or deactivated (off) initially. Most widgets appear greyed out when deactivated.

The **Resizable** button controls whether the window is resizable. In addition all the size changes of a window or group will go "into" the resizable child. If you have a large data display surrounded by buttons, you probably want that data area to be resizable. You can get more complex behavior by making invisible boxes the resizable widget, or by using hierarchies of groups. Unfortunately the only way to test it is to compile the program. Resizing the FLUID window is *not* the same as what will happen in the user program.

The **Hotspot** button causes the parent window to be positioned with that widget centered on the mouse. This position is determined *when the FLUID function is called*, so you should call it immediately before showing the window. If you want the window to hide and then reappear at a new position, you should have your program set the hotspot itself just before `show()`.

The **Border** button turns the window manager border on or off. On most window managers you will have to close the window and reopen it to see the effect.

X Class (text field)

The string typed into here is passed to the X window manager as the class. This can change the icon or window decorations. On most (all?) window managers you will have to close the window and reopen it to see the effect.

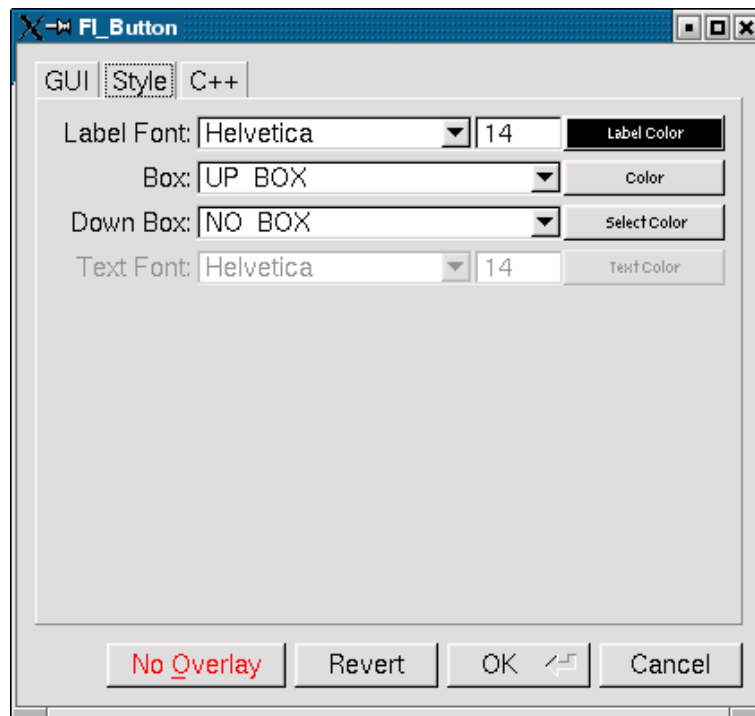


Figure 11.12 The FLUID widget Style attributes

11.7.1 Style Attributes

Label Font (pulldown menu)

Font to draw the label in. Ignored by symbols, bitmaps, and pixmaps. Your program can change the actual font used by these "slots" in case you want some font other than the 16 provided.

Label Size (pulldown menu)

Pixel size (height) for the font to draw the label in. Ignored by symbols, bitmaps, and pixmaps. To see the result without dismissing the panel, type the new number and then Tab.

Label Color (button)

Color to draw the label. Ignored by pixmaps (bitmaps, however, do use this color as the foreground color).

Box (pulldown menu)

The boxtype to draw as a background for the widget.

Many widgets will work, and draw faster, with a "frame" instead of a "box". A frame does not draw the colored interior, leaving whatever was already there visible. Be careful, as FLUID may draw this ok but the real program may leave unwanted stuff inside the widget.

If a window is filled with child widgets, you can speed up redrawing by changing the window's box type to "NO_BOX". FLUID will display a checkerboard for any areas that are not colored in by boxes. Note that this checkerboard is not drawn by the resulting program. Instead random garbage will be displayed.

Down Box (pulldown menu)

The boxtype to draw when a button is pressed or for some parts of other widgets like scrollbars and valuator.

Color (button)

The color to draw the box with.

Select Color (button)

Some widgets will use this color for certain parts. FLUID does not always show the result of this: this is the color buttons draw in when pushed down, and the color of input fields when they have the focus.

Text Font, Size, and Color

Some widgets display text, such as input fields, pull-down menus, and browsers.

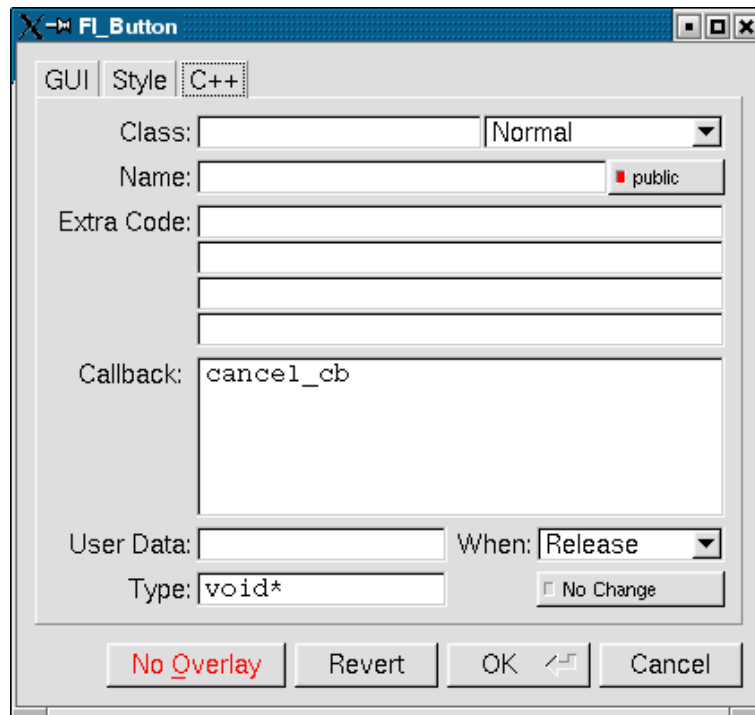


Figure 11.13 The FLUID widget C++ attributes

11.7.2 C++ Attributes

Class

This is how you use your own subclasses of `Fl_Widget`. Whatever identifier you type in here will be the class that is instantiated.

In addition, no `#include` header file is put in the `.h` file. You must provide a `#include` line as the first line of the "Extra Code" which declares your subclass.

The class must be similar to the class you are spoofing. It does not have to be a subclass. It is sometimes useful to change this to another FLTK class. Currently the only way to get a double-buffered window is to change this field for the window to "Fl_Double_Window" and to add

```
#include <FL/Fl_Double_Window.h>
```

to the extra code.

Type (upper-right pulldown menu)

Some classes have subtypes that modify their appearance or behavior. You pick the subtype off of this menu.

Name (text field)

Name of a variable to declare, and to store a pointer to this widget into. This variable will be of type "`<class>*`". If the name is blank then no variable is created.

You can name several widgets with "name[0]", "name[1]", "name[2]", etc. This will cause FLUID to declare an array of pointers. The array is big enough that the highest number found can be stored. All widgets in the array must be the same type.

Public (button)

Controls whether the widget is publicly accessible. When embedding widgets in a C++ class, this controls whether the widget is `public` or `private` in the class. Otherwise it controls whether the widget is declared `static` or `global` (`extern`).

Extra Code (text fields)

These four fields let you type in literal lines of code to dump into the `.h` or `.cxx` files.

If the text starts with a `#` or the word `extern` then FLUID thinks this is an "include" line, and it is written to the `.h` file. If the same include line occurs several times then only one copy is written.

All other lines are "code" lines. The current widget is pointed to by the local variable `o`. The window being constructed is pointed to by the local variable `w`. You can also access any arguments passed to the function here, and any named widgets that are before this one.

FLUID will check for matching parenthesis, braces, and quotes, but does not do much other error checking. Be careful here, as it may be hard to figure out what widget is producing an error in the compiler. If you need more than four lines you probably should call a function in your own `.cxx` code.

Callback (text field)

This can either be the name of a function, or a small snippet of code. If you enter anything other than letters, numbers, and the underscore then FLUID treats it as code.

A name refers to a function in your own code. It must be declared as `void name(<class>*, void*)`.

A code snippet is inserted into a static function in the `.cxx` output file. The function prototype is `void name(class *o, void *v)` so that you can refer to the widget as `o` and the `user_data()` as `v`. FLUID will check for matching parenthesis, braces, and quotes, but does not do much other error checking. Be careful here, as it may be hard to figure out what widget is producing an error in the compiler.

If the callback is blank then no callback is set.

User Data (text field)

This is a value for the `user_data()` of the widget. If blank the default value of zero is used. This can be any piece of C code that can be cast to a `void` pointer.

Type (text field)

The `void*` in the callback function prototypes is replaced with this. You may want to use `long` for old XForms code. Be warned that anything other than `void*` is not guaranteed to work! However on most architectures other pointer types are ok, and `long` is usually ok, too.

When (pulldown menu)

When to do the callback. This can be **Never**, **Changed**, **Release**, or **Enter Key**. The value of **Enter Key** is only useful for text input fields.

There are other rare but useful values for the `when()` field that are not in the menu. You should use the extra code fields to put these values in.

No Change (button)

The **No Change** button means the callback is done on the matching event even if the data is not changed.

11.8 Selecting and Moving Widgets

Double-clicking a window name in the browser will display it, if not displayed yet. From this display you can select widgets, sets of widgets, and move or resize them. To close a window either double-click it or type `ESC`.

To select a widget, click it. To select several widgets drag a rectangle around them. Holding down shift will toggle the selection of the widgets instead.

You cannot pick hidden widgets. You also cannot choose some widgets if they are completely overlapped by later widgets. Use the browser to select these widgets.

The selected widgets are shown with a red "overlay" line around them. You can move the widgets by dragging this box. Or you can resize them by dragging the outer edges and corners. Hold down the Alt key while dragging the mouse to defeat the snap-to-grid effect for fine positioning.

If there is a tab box displayed you can change which child is visible by clicking on the file tabs. The child you pick is selected.

The arrow, tab, and shift+tab keys "navigate" the selection. Left, right, tab, or shift+tab move to the next or previous widgets in the hierarchy. Hit the right arrow enough and you will select every widget in the window. Up/down widgets move to the previous/next widgets that overlap horizontally. If the navigation does not seem to work you probably need to "Sort" the widgets. This is important if you have input fields, as FLTK uses the same rules when using arrow keys to move between input fields.

To "open" a widget, double click it. To open several widgets select them and then type F1 or pick "Edit/Open" off the pop-up menu.

Type `Ctrl+o` to temporarily toggle the overlay off without changing the selection, so you can see the widget borders.

You can resize the window by using the window manager border controls. FLTK will attempt to round the window size to the nearest multiple of the grid size and makes it big enough to contain all the widgets (it does this using illegal X methods, so it is possible it will barf with some window managers!). Notice that the actual window in your program may not be resizable, and if it is, the effect on child widgets may be different.

The panel for the window (which you get by double-clicking it) is almost identical to the panel for any other [Fl_Widget](#). There are three extra items:

11.9 Image Labels

The *contents* of the image files in the **Image** and **Inactive** text fields are written to the `.cxx` file. If many widgets share the same image then only one copy is written. Since the image data is embedded in the generated source code, you need only distribute the C++ code and not the image files themselves.

However, the *filenames* are stored in the `.fl` file so you will need the image files as well to read the `.fl` file. Filenames are relative to the location of the `.fl` file and not necessarily the current directory. We recommend you either put the images in the same directory as the `.fl` file, or use absolute path names.

Notes for All Image Types

FLUID runs using the default visual of your X server. This may be 8 bits, which will give you dithered images. You may get better results in your actual program by adding the code "Fl::visual(FL_RGB)" to your code right before the first window is displayed.

All widgets with the same image on them share the same code and source X pixmap. Thus once you have put an image on a widget, it is nearly free to put the same image on many other widgets.

If you edit an image at the same time you are using it in FLUID, the only way to convince FLUID to read the image file again is to remove the image from all widgets that are using it or re-load the `.fl` file.

Don't rely on how FLTK crops images that are outside the widget, as this may change in future versions! The cropping of inside labels will probably be unchanged.

To more accurately place images, make a new "box" widget and put the image in that as the label.

XBM (X Bitmap) Files

FLUID reads X bitmap files which use C source code to define a bitmap. Sometimes they are stored with the ".h" or ".bm" extension rather than the standard ".xpm" extension.

FLUID writes code to construct an [Fl_Bitmap](#) image and use it to label the widget. The '1' bits in the bitmap are drawn using the label color of the widget. You can change this color in the FLUID widget attributes panel. The '0' bits are transparent.

The program "bitmap" on the X distribution does an adequate job of editing bitmaps.

XPM (X Pixmap) Files

FLUID reads X pixmap files as used by the `libxpm` library. These files use C source code to define a pixmap. The filenames usually have the ".xpm" extension.

FLUID writes code to construct an [FL_Pixmap](#) image and use it to label the widget. The label color of the widget is ignored, even for 2-color images that could be a bitmap. XPM files can mark a single color as being transparent, and FLTK uses this information to generate a transparency mask for the image.

We have not found any good editors for small iconic pictures. For pixmaps we have used [XPaint](#) and the KDE icon editor.

BMP Files

FLUID reads Windows BMP image files which are often used in WIN32 applications for icons. FLUID converts BMP files into (modified) XPM format and uses a [FL_BMP_Image](#) image to label the widget. Transparency is handled the same as for XPM files. All image data is uncompressed when written to the source file, so the code may be much bigger than the `.bmp` file.

GIF Files

FLUID reads GIF image files which are often used in HTML documents to make icons. FLUID converts GIF files into (modified) XPM format and uses a [FL_GIF_Image](#) image to label the widget. Transparency is handled the same as for XPM files. All image data is uncompressed when written to the source file, so the code may be much bigger than the `.gif` file. Only the first image of an animated GIF file is used.

JPEG Files

If FLTK is compiled with JPEG support, FLUID can read JPEG image files which are often used for digital photos. FLUID uses a [FL_JPEG_Image](#) image to label the widget, and writes uncompressed RGB or grayscale data to the source file.

PNG (Portable Network Graphics) Files

If FLTK is compiled with PNG support, FLUID can read PNG image files which are often used in HTML documents. FLUID uses a [FL_PNG_Image](#) image to label the widget, and writes uncompressed RGB or grayscale data to the source file. PNG images can provide a full alpha channel for partial transparency, and FLTK supports this as best as possible on each platform.

11.10 Internationalization with FLUID

FLUID supports internationalization (I18N for short) of label strings used by widgets. The preferences window (Ctrl+p) provides access to the I18N options.

11.10.1 I18N Methods

FLUID supports three methods of I18N: use none, use GNU gettext, and use POSIX catgets. The "use none" method is the default and just passes the label strings as-is to the widget constructors.

The "GNU gettext" method uses GNU gettext (or a similar text-based I18N library) to retrieve a localized string before calling the widget constructor.

The "POSIX catgets" method uses the POSIX catgets function to retrieve a numbered message from a message catalog before calling the widget constructor.

11.10.2 Using GNU gettext for I18N

FLUID's code support for GNU gettext is limited to calling a function or macro to retrieve the localized label; you still need to call `setlocale()` and `textdomain()` or `bindtextdomain()` to select the appropriate language and message file.

To use GNU gettext for I18N, open the preferences window and choose "GNU gettext" from the **Use:** chooser. Two new input fields will then appear to control the include file and function/macro name to use when retrieving the localized label strings.

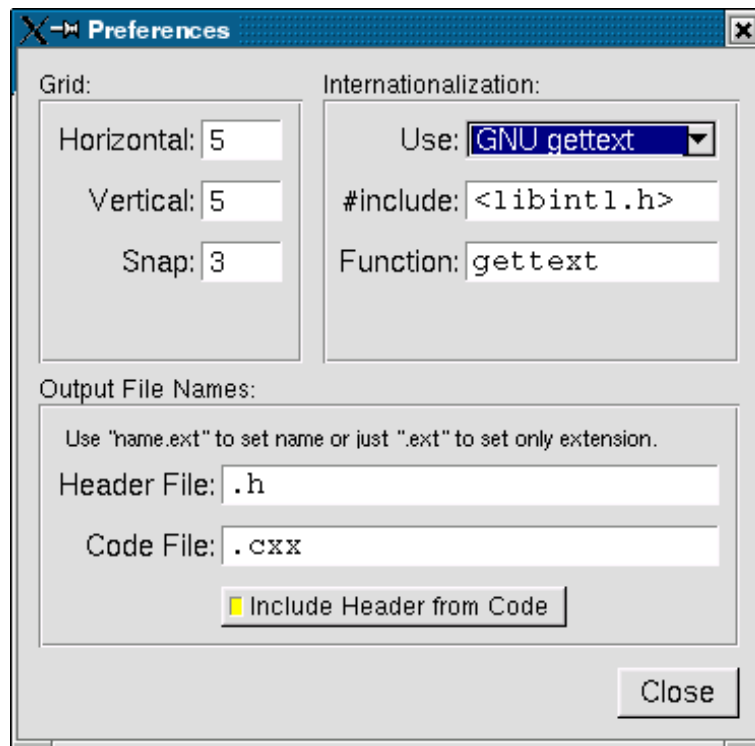


Figure 11.14 Internationalization using GNU gettext

The **#include** field controls the header file to include for I18N; by default this is `<libintl.h>`, the standard I18N file for GNU gettext.

The **Function:** field controls the function (or macro) that will retrieve the localized message; by default the `gettext` function will be called.

11.10.3 Using POSIX catgets for I18N

FLUID's code support for POSIX catgets allows you to use a global message file for all interfaces or a file specific to each `.fl` file; you still need to call `setlocale()` to select the appropriate language.

To use POSIX catgets for I18N, open the preferences window and choose "POSIX catgets" from the **Use:** chooser. Three new input fields will then appear to control the include file, catalog file, and set number for retrieving the localized label strings.

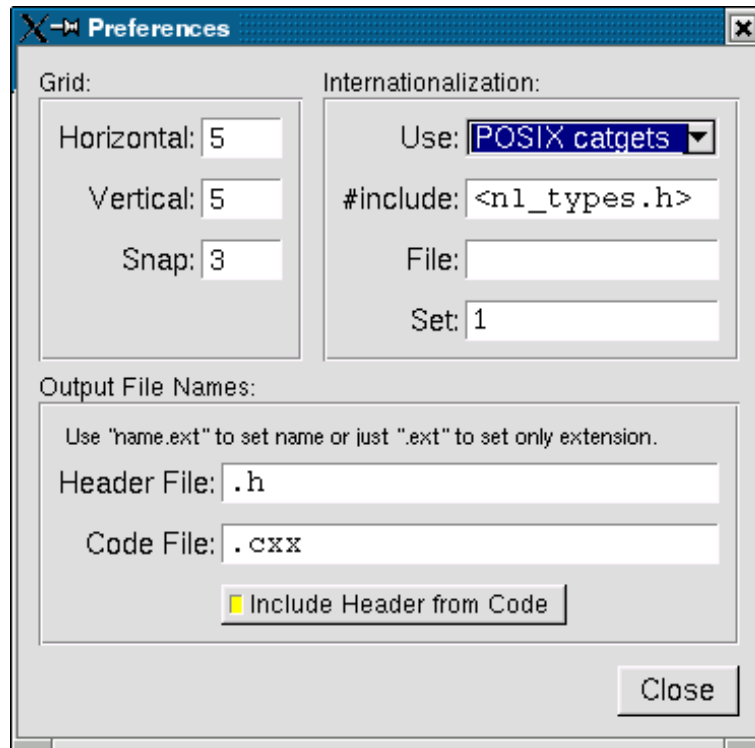


Figure 11.15 Internationalization using POSIX catgets

The **#include** field controls the header file to include for I18N; by default this is `<nl_types.h>`, the standard I18N file for POSIX catgets.

The **File:** field controls the name of the catalog file variable to use when retrieving localized messages; by default the file field is empty which forces a local (static) catalog file to be used for all of the windows defined in your `.fl` file.

The **Set:** field controls the set number in the catalog file. The default set is 1 and rarely needs to be changed.

11.11 Known limitations

Declaration Blocks can be used to temporarily block out already designed code using `#if 0` and `#endif` type construction. This will effectively avoid compilation of blocks of code. However, static code and data generated by this segment (menu items, images, include statements, etc.) will still be generated and likely cause compile-time warnings.

Chapter 12

Advanced FLTK

This chapter explains advanced programming and design topics that will help you to get the most out of FLTK.

12.1 Multithreading

FLTK can be used to implement a GUI for a multithreaded application but, as with multithreaded programming generally, there are some concepts and caveats that must be kept in mind.

Key amongst these is that, for many of the target platforms on which FLTK is supported, only the `main()` thread of the process is permitted to handle system events, create or destroy windows and open or close windows. Further, only the `main()` thread of the process can safely write to the display.

To support this in a portable way, all FLTK `draw()` methods are executed in the `main()` thread. A worker thread may update the state of an existing widget, but it may not do any rendering directly, nor create or destroy a window. (**NOTE:** A special case exists for [FL_Gl_Window](#) where it can, with suitable precautions, be possible to safely render to an existing GL context from a worker thread.)

Creating portable threads

We do not provide a threading interface as part of the library. A simple example showing how threads can be implemented, for all supported platforms, can be found in `test/threads.h` and `test/threads.cxx`.

FLTK has been used with a variety of thread interfaces, so if the simple example shown in `test/threads.cxx` does not cover your needs, you might want to select a third-party library that provides the features you require.

12.2 FLTK multithread locking - `Fl::lock()` and `Fl::unlock()`

In a multithreaded program, drawing of widgets (in the `main()` thread) happens asynchronously to widgets being updated by worker threads, so no drawing can occur safely whilst a widget is being modified (and no widget should be modified whilst drawing is in progress).

FLTK supports multithreaded applications using a locking mechanism internally. This allows a worker thread to lock the rendering context, preventing any drawing from taking place, whilst it changes the value of its widget.

Note

The converse is also true; whilst a worker thread holds the lock, the `main()` thread may not be able to process any drawing requests, nor service any events. So a worker thread that holds the FLTK lock **must** contrive to do so for the shortest time possible or it could impair operation of the application.

The lock operates broadly as follows.

Using the FLTK library, the `main()` thread holds the lock whenever it is processing events or redrawing the display. It acquires (locks) and releases (unlocks) the FLTK lock automatically and no "user intervention" is required. Indeed, a function that runs in the context of the `main()` thread ideally should **not** acquire / release the FLTK lock explicitly. (Though note that the lock calls are recursive, so calling `Fl::lock()` from a thread that already holds the lock, including the `main()` thread, is benign. The only constraint is that every call to `Fl::lock()` **must** be balanced by a corresponding call to `Fl::unlock()` to ensure the lock count is preserved.)

The `main()` thread **must** call `Fl::lock()` **once** before any windows are shown, to enable the internal lock (it is "off" by default since it is not useful in single-threaded applications) but thereafter the `main()` thread lock is managed by the library internally.

A worker thread, when it wants to alter the value of a widget, can acquire the lock using `Fl::lock()`, update the widget, then release the lock using `Fl::unlock()`. Acquiring the lock ensures that the worker thread can update the widget, without any risk that the `main()` thread will attempt to redraw the widget whilst it is being updated.

Note that acquiring the lock is a blocking action; the worker thread will stall for as long as it takes to acquire the lock. If the `main()` thread is engaged in some complex drawing operation this may block the worker thread for a long time, effectively serializing what ought to be parallel operations. (This frequently comes as a surprise to coders less familiar with multithreaded programming issues; see the discussion of "lockless programming" later for strategies for managing this.)

To incorporate the locking mechanism in the library, FLTK must be compiled with `--enable-threads` set during the `configure` process. IDE-based versions of FLTK are automatically compiled with the locking mechanism incorporated if possible. Since version 1.3, the `configure` script that builds the FLTK library also sets `--enable-threads` by default.

12.3 Simple multithreaded examples using `Fl::lock`

In `main()`, call `Fl::lock()` once before `Fl::run()` or `Fl::wait()` to enable the lock and start the runtime multithreading support for your program. All callbacks and derived functions like `handle()` and `draw()` will now be properly locked.

This might look something like this:

```
int main(int argc, char **argv) {
    /* Create your windows and widgets here */
    Fl::lock(); /* "start" the FLTK lock mechanism */
    /* show your window */
    main_win->show(argc, argv);
    /* start your worker threads */
    ... start threads ...
    /* Run the FLTK main loop */
    int result = Fl::run();
    /* terminate any pending worker threads */
    ... stop threads ...
    return result;
}
```

You can start as many threads as you like. From within a thread (other than the `main()` thread) FLTK calls must be wrapped with calls to `Fl::lock()` and `Fl::unlock()`:

```
void my_thread(void) {
    while (thread_still_running) {
        /* do thread work */
        ...
        /* compute new values for widgets */
        ...
        Fl::lock(); // acquire the lock
        my_widget->update(values);
        Fl::unlock(); // release the lock; allow other threads to access FLTK again
        Fl::awake(); // use Fl::awake() to signal main thread to refresh the GUI
    }
}
```

Note

To trigger a refresh of the GUI from a worker thread, the worker code should call `Fl::awake()`

Using `Fl::awake` thread messages

You can send messages from worker threads to the `main()` thread using `Fl::awake(void* message)`. If using this thread message interface, your `main()` might look like this:

```
int main(int argc, char **argv) {
    /* Create your windows and widgets here */
    Fl::lock(); /* "start" the FLTK lock mechanism */
    /* show your window */
    main_win->show(argc, argv);
    /* start your worker threads */
    ... start threads ...
    /* Run the FLTK loop and process thread messages */
    while (Fl::wait() > 0) {
        if ((next_message = Fl::thread_message()) != NULL) {
            /* process your data, update widgets, etc. */
            ...
        }
    }
    /* terminate any pending worker threads */
    ... stop threads ...
    return 0;
}
```

Your worker threads can send messages to the `main()` thread using `Fl::awake(void* message)`:

```
void *msg;          // "msg" is a pointer to your message
Fl::awake(msg);    // send "msg" to main thread
```

A message can be anything you like. The `main()` thread can retrieve the message by calling `Fl::thread_message()`.

Using `Fl::awake` callback messages

You can also request that the `main()` thread call a function on behalf of the worker thread by using `Fl::awake(Fl_Awake_Handler cb, void* userdata)`.

The `main()` thread will execute the callback "as soon as possible" when next processing the pending events. This can be used by a worker thread to perform operations (for example showing or hiding windows) that are prohibited in a worker thread.

```
void do_something_cb(void *userdata) {
    // Will run in the context of the main thread
    ... do_stuff ...
}
// running in worker thread
void *data;          // "data" is a pointer to your user data
Fl::awake(do_something_cb, data); // call to execute cb in main thread
```

Note

The `main()` thread will execute the `Fl_Awake_Handler` callback `do_something_cb` asynchronously to the worker thread, at some short but indeterminate time after the worker thread registers the request. When it executes the `Fl_Awake_Handler` callback, the `main()` thread will use the contents of `*userdata` **at the time of execution**, not necessarily the contents that `*userdata` had at the time that the worker thread posted the callback request. The worker thread should therefore contrive **not** to alter the contents of `*userdata` once it posts the callback, since the worker thread does not know when the `main()` thread will consume that data. It is often useful that `userdata` point to a struct, one member of which the `main()` thread can modify to indicate that it has consumed the data, thereby allowing the worker thread to re-use or update `userdata`.

Warning

The mechanisms used to deliver `Fl::awake(void* message)` and `Fl::awake(Fl_Awake_Handler cb, void* userdata)` events to the `main()` thread can interact in unexpected ways on some platforms. Therefore, for reliable operation, it is advised that a program use either `Fl::awake(Fl_Awake_Handler cb, void* userdata)` or `Fl::awake(void* message)`, but that they never be intermixed. Calling `Fl::awake()` with no parameters should be safe in either case.

If you have to choose between using the `Fl::awake(void* message)` and `Fl::awake(Fl_Awake_Handler cb, void* userdata)` mechanisms and don't know which to choose, then try the `Fl::awake(Fl_Awake_Handler cb, void* userdata)` method first as it tends to be more powerful in general.

12.4 FLTK multithreaded "lockless programming"

The simple multithreaded examples shown above, using the FLTK lock, work well for many cases where multiple threads are required. However, when that model is extended to more complex programs, it often produces results that the developer did not anticipate.

A typical case might go something like this. A developer creates a program to process a huge data set. The program has a `main()` thread and 7 worker threads and is targeted to run on an 8-core computer. When it runs, the program divides the data between the 7 worker threads, and as they process their share of the data, each thread updates its portion of the GUI with the results, locking and unlocking as they do so.

But when this program runs, it is much slower than expected and the developer finds that only one of the eight CPU cores seems to be utilised, despite there being 8 threads in the program. What happened?

The threads in the program all run as expected, but they end up being serialized (that is, not able to run in parallel) because they all depend on the single FLTK lock. Acquiring (and releasing) that lock has an associated cost, and is a **blocking** action if the lock is already held by any other worker thread or by the `main()` thread.

If the worker threads are acquiring the lock "too often", then the lock will **always** be held **somewhere** and every attempt by any other thread (even `main()`) to lock will cause that other thread (including `main()`) to block. And blocking `main()` also blocks event handling, display refresh...

As a result, only one thread will be running at any given time, and the multithreaded program is effectively reduced to being a (complicated and somewhat less efficient) single thread program.

A "solution" is for the worker threads to lock "less often", such that they do not block each other or the `main()` thread. But judging what constitutes locking "too often" for any given configuration, and hence will block, is a very tricky question. What works well on one machine, with a given graphics card and CPU configuration may behave very differently on another target machine.

There are "interesting" variations on this theme, too: for example it is possible that a "faulty" multithreaded program such as described above will work adequately on a single-core machine (where all threads are inherently serialized anyway and so are less likely to block each other) but then stall or even deadlock in unexpected ways on a multicore machine when the threads do interfere with each other. (I have seen this - it really happens.)

The "better" solution is to avoid using the FLTK lock so far as possible. Instead, the code should be designed so that the worker threads do not update the GUI themselves and therefore never need to acquire the FLTK lock. This would be FLTK multithreaded "lockless programming".

There are a number of ways this can be achieved (or at least approximated) in practice but the most direct approach is for the worker threads to make use of the `Fl::awake(Fl_Awake_Handler cb, void* userdata)` method so that GUI updates can all run in the context of the `main()` thread, alleviating the need for the worker thread to ever lock. The onus is then on the worker threads to manage the `userdata` so that it is delivered safely to the `main()` thread, but there are many ways that can be done.

Note

Using `Fl::awake` is not, strictly speaking, entirely "lockless" since the awake handler mechanism incorporates resource locking internally to protect the queue of pending awake messages. These resource locks are held transiently and generally do not trigger the pathological blocking issues described here.

However, aside from using `Fl::awake`, there are many other ways that a "lockless" design can be implemented, including message passing, various forms of IPC, etc.

If you need high performing multithreaded programming, then take some time to study the options and understand the advantages and disadvantages of each; we can't even begin to scratch the surface of this huge topic here!

And of course occasional, sparse, use of the FLTK lock from worker threads will do no harm; it is "excessive" locking (whatever that might be) that triggers the failing behaviour.

It is always a Good Idea to update the GUI at the lowest rate that is acceptable when processing bulk data (or indeed, in all cases!) Updating at a few frames per second is probably adequate for providing feedback during a long calculation. At the upper limit, anything faster than the frame rate of your monitor and the updates will never even be displayed; why waste CPU computing pixels that you will never show?

12.5 FLTK multithreaded Constraints

FLTK supports multiple platforms, some of which allow only the `main()` thread to handle system events and open or close windows. The safe thing to do is to adhere to the following rules for threads on all operating systems:

- Don't `show()` or `hide()` anything that contains `Fl_Window` based widgets from a worker thread. This includes any windows, dialogs, file choosers, subwindows or widgets using `Fl_Gl_Window`. Note that this constraint also applies to non-window widgets that have tooltips, since the tooltip will contain a `Fl_Window` object. The safe and portable approach is **never** to call `show()` or `hide()` on any widget from the context of a worker thread. Instead you can use the `Fl_Awake_Handler` variant of `Fl::awake()` to request the `main()` thread to create, destroy, show or hide the widget on behalf of the worker thread.
- Don't call `Fl::run()`, `Fl::wait()`, `Fl::flush()`, `Fl::check()` or any related methods that will handle system messages from a worker thread
- Don't intermix use of `Fl::awake(Fl_Awake_Handler cb, void* userdata)` and `Fl::awake(void* message)` calls in the same program as they may interact unpredictably on some platforms; choose one or other style of `Fl::awake(<thing>)` mechanism and use that. (Intermixing calls to `Fl::awake()` should be safe with either however.)
- Don't start or cancel timers from a worker thread
- Don't change window decorations or titles from a worker thread
- The `make_current()` method will probably not work well for regular windows, but should always work for a `Fl_Gl_Window` to allow for high speed rendering on graphics cards with multiple pipelines. Managing thread-safe access to the GL pipelines is left as an exercise for the reader! (And may be target specific...)

See also: `Fl::lock()`, `Fl::unlock()`, `Fl::awake()`, `Fl::awake(Fl_Awake_Handler cb, void* userdata)`, `Fl::awake(void* message)`, `Fl::thread_message()`.

Chapter 13

Unicode and UTF-8 Support

This chapter explains how FLTK handles international text via Unicode and UTF-8.

Unicode support was only recently added to FLTK and is still incomplete. This chapter is Work in Progress, reflecting the current state of Unicode support.

13.1 About Unicode, ISO 10646 and UTF-8

The summary of Unicode, ISO 10646 and UTF-8 given below is deliberately brief and provides just enough information for the rest of this chapter.

For further information, please see:

- <http://www.unicode.org>
- <http://www.iso.org>
- <http://en.wikipedia.org/wiki/Unicode>
- <http://www.cl.cam.ac.uk/~mgk25/unicode.html>
- <http://www.apps.ietf.org/rfc/rfc3629.html>

The Unicode Standard

The Unicode Standard was originally developed by a consortium of mainly US computer manufacturers and developers of multi-lingual software. It has now become a defacto standard for character encoding and is supported by most of the major computing companies in the world.

Before Unicode, many different systems, on different platforms, had been developed for encoding characters for different languages, but no single encoding could satisfy all languages. Unicode provides access to over 100,000 characters used in all the major languages written today, and is independent of platform and language.

Unicode also provides higher-level concepts needed for text processing and typographic publishing systems, such as algorithms for sorting and comparing text, composite character and text rendering, right-to-left and bi-directional text handling.

Note

There are currently no plans to add this extra functionality to FLTK.

ISO 10646

The International Organisation for Standardization (ISO) had also been trying to develop a single unified character set. Although both ISO and the Unicode Consortium continue to publish their own standards, they have agreed to coordinate their work so that specific versions of the Unicode and ISO 10646 standards are compatible with each other.

The international standard ISO 10646 defines the **Universal Character Set** (UCS) which contains the characters required for almost all known languages. The standard also defines three different implementation levels specifying how these characters can be combined.

Note

There are currently no plans for handling the different implementation levels or the combining characters in FLTK.

In UCS, characters have a unique numerical code and an official name, and are usually shown using 'U+' and the code in hexadecimal, e.g. U+0041 is the "Latin capital letter A". The UCS characters U+0000 to U+007F correspond to US-ASCII, and U+0000 to U+00FF correspond to ISO 8859-1 (Latin1).

ISO 10646 was originally designed to handle a 31-bit character set from U+00000000 to U+7FFFFFFF, but the current idea is that 21 bits will be sufficient for all future needs, giving characters up to U+10FFFF. The complete character set is sub-divided into *planes*. *Plane 0*, also known as the **Basic Multilingual Plane** (BMP), ranges from U+0000 to U+FFFF and consists of the most commonly used characters from previous encoding standards. Other planes contain characters for specialist applications.

Todo Do we need this info about planes?

The UCS also defines various methods of encoding characters as a sequence of bytes. UCS-2 encodes Unicode characters into two bytes, which is wasteful if you are only dealing with ASCII or Latin1 text, and insufficient if you need characters above U+00FFFF. UCS-4 uses four bytes, which lets it handle higher characters, but this is even more wasteful for ASCII or Latin1.

UTF-8

The Unicode standard defines various UCS Transformation Formats (UTF). UTF-16 and UTF-32 are based on units of two and four bytes. UCS characters requiring more than 16 bits are encoded using "surrogate pairs" in UTF-16.

UTF-8 encodes all Unicode characters into variable length sequences of bytes. Unicode characters in the 7-bit ASCII range map to the same value and are represented as a single byte, making the transformation to Unicode quick and easy.

All UCS characters above U+007F are encoded as a sequence of several bytes. The top bits of the first byte are set to show the length of the byte sequence, and subsequent bytes are always in the range 0x80 to 0xBF. This combination provides some level of synchronisation and error detection.

Unicode range	Byte sequences
U+00000000 - U+0000007F	0xxxxxxx
U+00000080 - U+000007FF	110xxxxx 10xxxxxx
U+00000800 - U+0000FFFF	1110xxxx 10xxxxxx 10xxxxxx
U+00010000 - U+001FFFFF	11110xxx 10xxxxxx 10xxxxxx 10xxxxxx
U+00200000 - U+03FFFFFF	111110xx 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx
U+04000000 - U+7FFFFFFF	1111110x 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx

Moving from ASCII encoding to Unicode will allow all new FLTK applications to be easily internationalized and used all over the world. By choosing UTF-8 encoding, FLTK remains largely source-code compatible to previous iterations of the library.

13.2 Unicode in FLTK

Todo Work through the code and this documentation to harmonize the `[OksiD]` and `[fltk2]` functions.

FLTK will be entirely converted to Unicode using UTF-8 encoding. If a different encoding is required by the underlying operating system, FLTK will convert the string as needed.

It is important to note that the initial implementation of Unicode and UTF-8 in FLTK involves three important areas:

- provision of Unicode character tables and some simple related functions;
- conversion of `char*` variables and function parameters from single byte per character representation to UTF-8 variable length sequences;
- modifications to the display font interface to accept general Unicode character or UCS code numbers instead of just ASCII or Latin1 characters.

The current implementation of Unicode / UTF-8 in FLTK will impose the following limitations:

- An implementation note in the `[OksiD]` code says that all functions are LIMITED to 24 bit Unicode values, but also says that only 16 bits are really used under linux and win32. **[Can we verify this?]**
- The `[fltk2]` `fl_utf8encode()` and `fl_utf8decode()` functions are designed to handle Unicode characters in the range U+000000 to U+10FFFF inclusive, which covers all UTF-16 characters, as specified in RFC 3629. *Note that the user must first convert UTF-16 surrogate pairs to UCS.*
- FLTK will only handle single characters, so composed characters consisting of a base character and floating accent characters will be treated as multiple characters.
- FLTK will only compare or sort strings on a byte by byte basis and not on a general Unicode character basis.
- FLTK will not handle right-to-left or bi-directional text.

Todo Verify 16/24 bit Unicode limit for different character sets? OksiD's code appears limited to 16-bit whereas the FLTK2 code appears to handle a wider set. What about illegal characters? See comments in `fl_utf8fromwc()` and `fl_utf8toUtf16()`.

13.3 Illegal Unicode and UTF-8 Sequences

Three pre-processor variables are defined in the source code [1] that determine how `fl_utf8decode()` handles illegal UTF-8 sequences:

- if `ERRORS_TO_CP1252` is set to 1 (the default), `fl_utf8decode()` will assume that a byte sequence starting with a byte in the range 0x80 to 0x9f represents a Microsoft CP1252 character, and will return the value of an equivalent UCS character. Otherwise, it will be processed as an illegal byte value as described below.
- if `STRICT_RFC3629` is set to 1 (not the default!) then UTF-8 sequences that correspond to illegal UCS values are treated as errors. Illegal UCS values include those above U+10FFFF, or corresponding to UTF-16 surrogate pairs. Illegal byte values are handled as described below.
- if `ERRORS_TO_ISO8859_1` is set to 1 (the default), the illegal byte value is returned unchanged, otherwise 0xFFFD, the Unicode REPLACEMENT CHARACTER, is returned instead.

[1] Since FLTK 1.3.4 you may set these three pre-processor variables on your compile command line with `-D"variable=value"` (value: 0 or 1) to avoid editing the source code.

`fl_utf8encode()` is less strict, and only generates the UTF-8 sequence for 0xFFFD, the Unicode REPLACEMENT CHARACTER, if it is asked to encode a UCS value above U+10FFFF.

Many of the `[fltk2]` functions below use `fl_utf8decode()` and `fl_utf8encode()` in their own implementation, and are therefore somewhat protected from bad UTF-8 sequences.

The `[OksiD]` `fl_utf8len()` function assumes that the byte it is passed is the first byte in a UTF-8 sequence, and returns the length of the sequence. Trailing bytes in a UTF-8 sequence will return -1.

- **WARNING:** `fl_utf8len()` can not distinguish between single bytes representing Microsoft CP1252 characters 0x80-0x9f and those forming part of a valid UTF-8 sequence. You are strongly advised not to use `fl_utf8len()` in your own code unless you know that the byte sequence contains only valid UTF-8 sequences.
- **WARNING:** Some of the `[OksiD]` functions below still use `fl_utf8len()` in their implementations. These may need further validation.

Please see the individual function description for further details about error handling and return values.

13.4 FLTK Unicode and UTF-8 Functions

This section currently provides a brief overview of the functions. For more details, consult the main text for each function via its link.

`int fl_utf8locale()` **FLTK2**

`fl_utf8locale()` returns true if the "locale" seems to indicate that UTF-8 encoding is used.

It is highly recommended that you change your system so this does return true!

`int fl_utf8test(const char *src, unsigned len)` **FLTK2**

`fl_utf8test()` examines the first `len` bytes of `src`. It returns 0 if there are any illegal UTF-8 sequences; 1 if `src` contains plain ASCII or if `len` is zero; or 2, 3 or 4 to indicate the range of Unicode characters found.

`int fl_utf_nb_char(const unsigned char *buf, int len)` **OksiD**

Returns the number of UTF-8 characters in the first `len` bytes of `buf`.

`int fl_unichar_to_utf8_size(Fl_Unichar)`
`int fl_utf8bytes(unsigned ucs)`

Returns the number of bytes needed to encode `ucs` in UTF-8.

`int fl_utf8len(char c)` **OksiD**

If `c` is a valid first byte of a UTF-8 encoded character sequence, `fl_utf8len()` will return the number of bytes in that sequence. It returns -1 if `c` is not a valid first byte.

`unsigned int fl_nonspacing(unsigned int ucs)` **OksiD**

Returns true if `ucs` is a non-spacing character.

`const char* fl_utf8back(const char *p, const char *start, const char *end)` **FLTK2**
`const char* fl_utf8fwd(const char *p, const char *start, const char *end)` **FLTK2**

If `p` already points to the start of a UTF-8 character sequence, these functions will return `p`. Otherwise `fl_utf8back()` searches backwards from `p` and `fl_utf8fwd()` searches forwards from `p`, within the `start` and `end` limits, looking for the start of a UTF-8 character.

`unsigned int fl_utf8decode(const char *p, const char *end, int *len)` **FLTK2**
`int fl_utf8encode(unsigned ucs, char *buf)` **FLTK2**

`fl_utf8decode()` attempts to decode the UTF-8 character that starts at `p` and may not extend past `end`. It returns the Unicode value, and the length of the UTF-8 character sequence is returned via the `len` argument. `fl_utf8encode()` writes the UTF-8 encoding of `ucs` into `buf` and returns the number of bytes in the sequence. See the main documentation for the treatment of illegal Unicode and UTF-8 sequences.

unsigned int `fl_utf8froma(char *dst, unsigned dstlen, const char *src, unsigned srclen)` **FLTK2**

unsigned int `fl_utf8toa(const char *src, unsigned srclen, char *dst, unsigned dstlen)` **FLTK2**

`fl_utf8froma()` converts a character string containing single bytes per character (i.e. ASCII or ISO-8859-1) into UTF-8. If the `src` string contains only ASCII characters, the return value will be the same as `srclen`.

`fl_utf8toa()` converts a string containing UTF-8 characters into single byte characters. UTF-8 characters that do not correspond to ASCII or ISO-8859-1 characters below 0xFF are replaced with '?'.

Both functions return the number of bytes that would be written, not counting the null terminator. `dstlen` provides a means of limiting the number of bytes written, so setting `dstlen` to zero is a means of measuring how much storage would be needed before doing the real conversion.

char* `fl_utf2mbs(const char *src)` **OksiD**

converts a UTF-8 string to a local multi-byte character string. **[More info required here!]**

unsigned int `fl_utf8fromwc(char *dst, unsigned dstlen, const wchar_t *src, unsigned srclen)` **FLTK2**

unsigned int `fl_utf8towc(const char *src, unsigned srclen, wchar_t *dst, unsigned dstlen)` **FLTK2**

unsigned int `fl_utf8toUtf16(const char *src, unsigned srclen, unsigned short *dst, unsigned dstlen)` **FLTK2**

These routines convert between UTF-8 and `wchar_t` or "wide character" strings. The difficulty lies in the fact that `sizeof(wchar_t)` is 2 on Windows and 4 on Linux and most other systems. Therefore some "wide characters" on Windows may be represented as "surrogate pairs" of more than one `wchar_t`.

`fl_utf8fromwc()` converts from a "wide character" string to UTF-8. Note that `srclen` is the number of `wchar_t` elements in the source string and on Windows this might be larger than the number of characters. `dstlen` specifies the maximum number of **bytes** to copy, including the null terminator.

`fl_utf8towc()` converts a UTF-8 string into a "wide character" string. Note that on Windows, some "wide characters" might result in "surrogate pairs" and therefore the return value might be more than the number of characters. `dstlen` specifies the maximum number of **wchar_t** elements to copy, including a zero terminating element. **[Is this all worded correctly?]**

`fl_utf8toUtf16()` converts a UTF-8 string into a "wide character" string using UTF-16 encoding to handle the "surrogate pairs" on Windows. `dstlen` specifies the maximum number of `wchar_t` elements to copy, including a zero terminating element. **[Is this all worded correctly?]**

These routines all return the number of elements that would be required for a full conversion of the `src` string, including the zero terminator. Therefore setting `dstlen` to zero is a way of measuring how much storage would be needed before doing the real conversion.

```
unsigned int fl_utf8from_mb(char *dst, unsigned dstlen, const char *src, unsigned srclen) FLTK2
unsigned int fl_utf8to_mb(const char *src, unsigned srclen, char *dst, unsigned dstlen) FLTK2
```

These functions convert between UTF-8 and the locale-specific multi-byte encodings used on some systems for filenames, etc. If `fl_utf8locale()` returns true, these functions don't do anything useful. **[Is this all worded correctly?]**

```
int fl_tolower(unsigned int ucs) OksiD
int fl_toupper(unsigned int ucs) OksiD
int fl_utf_tolower(const unsigned char *str, int len, char *buf) OksiD
int fl_utf_toupper(const unsigned char *str, int len, char *buf) OksiD
```

`fl_tolower()` and `fl_toupper()` convert a single Unicode character from upper to lower case, and vice versa. `fl_utf_tolower()` and `fl_utf_toupper()` convert a string of bytes, some of which may be multi-byte UTF-8 encodings of Unicode characters, from upper to lower case, and vice versa.

Warning: to be safe, `buf` length must be at least `3*len` [for 16-bit Unicode]

```
int fl_utf_strcasecmp(const char *s1, const char *s2) OksiD
int fl_utf_strncasecmp(const char *s1, const char *s2, int n) OksiD
```

`fl_utf_strcasecmp()` is a UTF-8 aware string comparison function that converts the strings to lower case Unicode as part of the comparison. `fl_utf_strncasecmp()` only compares the first `n` characters [bytes?]

13.5 FLTK Unicode Versions of System Calls

- `int fl_access(const char* f, int mode)` **OksiD**
- `int fl_chmod(const char* f, int mode)` **OksiD**
- `int fl_execvp(const char* file, char* const* argv)` **OksiD**
- `FILE* fl_fopen(const char* f, const char* mode)` **OksiD**
- `char* fl_getcwd(char* buf, int maxlen)` **OksiD**
- `char* fl_getenv(const char* name)` **OksiD**
- `char fl_make_path(const char* path)` - returns char ? **OksiD**
- `void fl_make_path_for_file(const char* path)` **OksiD**
- `int fl_mkdir(const char* f, int mode)` **OksiD**
- `int fl_open(const char* f, int o, ...)` **OksiD**
- `int fl_rename(const char* f, const char* t)` **OksiD**
- `int fl_rmdir(const char* f)` **OksiD**
- `int fl_stat(const char* path, struct stat* buffer)` **OksiD**
- `int fl_system(const char* f)` **OksiD**
- `int fl_unlink(const char* f)` **OksiD**

TODO:

- more doc on unicode, add links
- write something about filename encoding on OS X...
- explain the `fl_utf8_...` commands
- explain issues with [FL_Preferences](#)
- why FLTK has no `FL_String` class

Chapter 14

FLTK Enumerations

Note

This file is not actively maintained any more, but is left here as a reference, until the doxygen documentation is completed.

See also

[FL/Enumerations.H](#).

This appendix lists the enumerations provided in the [<FL/Enumerations.H>](#) header file, organized by section. Constants whose value are zero are marked with "(0)", this is often useful to know when programming.

14.1 Version Numbers

The FLTK version number is stored in a number of compile-time constants:

- `FL_MAJOR_VERSION` - The major release number, currently 1
- `FL_MINOR_VERSION` - The minor release number, currently 3
- `FL_PATCH_VERSION` - The patch release number, currently 6
- `FL_VERSION` - [Deprecated] A combined floating-point version number for the major, minor, and patch release numbers, currently 1.0306
- `FL_API_VERSION` - A combined integer version number for the major, minor, and patch release numbers, currently 10306 (use this instead of `FL_VERSION`, if possible)
- `FL_ABI_VERSION` - A combined integer version number for the application binary interface (ABI) major, minor, and patch release numbers, currently 10300 (default)

Note

The ABI version (`FL_ABI_VERSION`) is usually constant throughout one major/minor release version, for instance 10300 if `FL_API_VERSION` is 10305. Hence the ABI is constant if only the patch version is changed. You can change this with `configure` or `CMake` though if you want the latest enhancements (called "ABI features", see `CHANGES`).

14.2 Events

Events are identified by an [FL_Event](#) enumeration value. The following events are currently defined:

- `FL_NO_EVENT` - No event (or an event fltk does not understand) occurred (0).
- `FL_PUSH` - A mouse button was pushed.
- `FL_RELEASE` - A mouse button was released.
- `FL_ENTER` - The mouse pointer entered a widget.
- `FL_LEAVE` - The mouse pointer left a widget.
- `FL_DRAG` - The mouse pointer was moved with a button pressed.
- `FL_FOCUS` - A widget should receive keyboard focus.
- `FL_UNFOCUS` - A widget loses keyboard focus.
- `FL_KEYBOARD` - A key was pressed.
- `FL_CLOSE` - A window was closed.
- `FL_MOVE` - The mouse pointer was moved with no buttons pressed.
- `FL_SHORTCUT` - The user pressed a shortcut key.
- `FL_DEACTIVATE` - The widget has been deactivated.
- `FL_ACTIVATE` - The widget has been activated.
- `FL_HIDE` - The widget has been hidden.
- `FL_SHOW` - The widget has been shown.
- `FL_PASTE` - The widget should paste the contents of the clipboard.
- `FL_SELECTIONCLEAR` - The widget should clear any selections made for the clipboard.
- `FL_MOUSEWHEEL` - The horizontal or vertical mousewheel was turned.
- `FL_DND_ENTER` - The mouse pointer entered a widget dragging data.
- `FL_DND_DRAG` - The mouse pointer was moved dragging data.
- `FL_DND_LEAVE` - The mouse pointer left a widget still dragging data.
- `FL_DND_RELEASE` - Dragged data is about to be dropped.
- `FL_SCREEN_CONFIGURATION_CHANGED` - The screen configuration (number, positions) was changed.
- `FL_FULLSCREEN` - The fullscreen state of the window has changed.

14.3 Callback "When" Conditions

The following constants determine when a callback is performed:

- `FL_WHEN_NEVER` - Never call the callback (0).
- `FL_WHEN_CHANGED` - Do the callback only when the widget value changes.
- `FL_WHEN_NOT_CHANGED` - Do the callback whenever the user interacts with the widget.
- `FL_WHEN_RELEASE` - Do the callback when the button or key is released and the value changes.
- `FL_WHEN_ENTER_KEY` - Do the callback when the user presses the ENTER key and the value changes.
- `FL_WHEN_RELEASE_ALWAYS` - Do the callback when the button or key is released, even if the value doesn't change.
- `FL_WHEN_ENTER_KEY_ALWAYS` - Do the callback when the user presses the ENTER key, even if the value doesn't change.

14.4 Fl::event_button() Values

The following constants define the button numbers for `FL_PUSH` and `FL_RELEASE` events:

- `FL_LEFT_MOUSE` - the left mouse button
- `FL_MIDDLE_MOUSE` - the middle mouse button
- `FL_RIGHT_MOUSE` - the right mouse button

14.5 Fl::event_key() Values

The following constants define the non-ASCII keys on the keyboard for `FL_KEYBOARD` and `FL_SHORTCUT` events:

- `FL_Button` - A mouse button; use `Fl_Button + n` for mouse button `n`.
- `FL_BackSpace` - The backspace key.
- `FL_Tab` - The tab key.
- `FL_Enter` - The enter key.
- `FL_Pause` - The pause key.
- `FL_Scroll_Lock` - The scroll lock key.
- `FL_Escape` - The escape key.
- `FL_Home` - The home key.
- `FL_Left` - The left arrow key.
- `FL_Up` - The up arrow key.
- `FL_Right` - The right arrow key.

- `FL_Down` - The down arrow key.
- `FL_Page_Up` - The page-up key.
- `FL_Page_Down` - The page-down key.
- `FL_End` - The end key.
- `FL_Print` - The print (or print-screen) key.
- `FL_Insert` - The insert key.
- `FL_Menu` - The menu key.
- `FL_Num_Lock` - The num lock key.
- `FL_KP` - One of the keypad numbers; use `FL_KP + n` for number `n`.
- `FL_KP_Enter` - The enter key on the keypad.
- `FL_F` - One of the function keys; use `FL_F + n` for function key `n`.
- `FL_Shift_L` - The lefthand shift key.
- `FL_Shift_R` - The righthand shift key.
- `FL_Control_L` - The lefthand control key.
- `FL_Control_R` - The righthand control key.
- `FL_Caps_Lock` - The caps lock key.
- `FL_Meta_L` - The left meta/Windows key.
- `FL_Meta_R` - The right meta/Windows key.
- `FL_Alt_L` - The left alt key.
- `FL_Alt_R` - The right alt key.
- `FL_Delete` - The delete key.

14.6 Fl::event_state() Values

The following constants define bits in the `Fl::event_state()` value:

- `FL_SHIFT` - One of the shift keys is down.
- `FL_CAPS_LOCK` - The caps lock is on.
- `FL_CTRL` - One of the ctrl keys is down.
- `FL_ALT` - One of the alt keys is down.
- `FL_NUM_LOCK` - The num lock is on.
- `FL_META` - One of the meta/Windows keys is down.
- `FL_COMMAND` - An alias for `FL_CTRL` on WIN32 and X11, or `FL_META` on MacOS X.
- `FL_SCROLL_LOCK` - The scroll lock is on.
- `FL_BUTTON1` - Mouse button 1 is pushed.
- `FL_BUTTON2` - Mouse button 2 is pushed.
- `FL_BUTTON3` - Mouse button 3 is pushed.
- `FL_BUTTONS` - Any mouse button is pushed.
- `FL_BUTTON(n)` - Mouse button `n` (where `n > 0`) is pushed.

14.7 Alignment Values

The following constants define bits that can be used with `FL_Widget::align()` to control the positioning of the label:

- `FL_ALIGN_CENTER` - The label is centered (0).
- `FL_ALIGN_TOP` - The label is top-aligned.
- `FL_ALIGN_BOTTOM` - The label is bottom-aligned.
- `FL_ALIGN_LEFT` - The label is left-aligned.
- `FL_ALIGN_RIGHT` - The label is right-aligned.
- `FL_ALIGN_CLIP` - The label is clipped to the widget.
- `FL_ALIGN_WRAP` - The label text is wrapped as needed.
- `FL_ALIGN_TOP_LEFT` - The label appears at the top of the widget, aligned to the left.
- `FL_ALIGN_TOP_RIGHT` - The label appears at the top of the widget, aligned to the right.
- `FL_ALIGN_BOTTOM_LEFT` - The label appears at the bottom of the widget, aligned to the left.
- `FL_ALIGN_BOTTOM_RIGHT` - The label appears at the bottom of the widget, aligned to the right.
- `FL_ALIGN_LEFT_TOP` - The label appears to the left of the widget, aligned at the top. Outside labels only.
- `FL_ALIGN_RIGHT_TOP` - The label appears to the right of the widget, aligned at the top. Outside labels only.
- `FL_ALIGN_LEFT_BOTTOM` - The label appears to the left of the widget, aligned at the bottom. Outside labels only.
- `FL_ALIGN_RIGHT_BOTTOM` - The label appears to the right of the widget, aligned at the bottom. Outside labels only.
- `FL_ALIGN_INSIDE` - 'or' this with other values to put label inside the widget.
- `FL_ALIGN_TEXT_OVER_IMAGE` - Label text will appear above the image.
- `FL_ALIGN_IMAGE_OVER_TEXT` - Label text will be below the image.
- `FL_ALIGN_IMAGE_NEXT_TO_TEXT` - The image will appear to the left of the text.
- `FL_ALIGN_TEXT_NEXT_TO_IMAGE` - The image will appear to the right of the text.
- `FL_ALIGN_IMAGE_BACKDROP` - The image will be used as a background for the widget.

14.8 Fonts

The following constants define the standard FLTK fonts:

- `FL_HELVETICA` - Helvetica (or Arial) normal (0).
- `FL_HELVETICA_BOLD` - Helvetica (or Arial) bold.
- `FL_HELVETICA_ITALIC` - Helvetica (or Arial) oblique.
- `FL_HELVETICA_BOLD_ITALIC` - Helvetica (or Arial) bold-oblique.
- `FL_COURIER` - Courier normal.

- FL_COURIER_BOLD - Courier bold.
- FL_COURIER_ITALIC - Courier italic.
- FL_COURIER_BOLD_ITALIC - Courier bold-italic.
- FL_TIMES - Times roman.
- FL_TIMES_BOLD - Times bold.
- FL_TIMES_ITALIC - Times italic.
- FL_TIMES_BOLD_ITALIC - Times bold-italic.
- FL_SYMBOL - Standard symbol font.
- FL_SCREEN - Default monospaced screen font.
- FL_SCREEN_BOLD - Default monospaced bold screen font.
- FL_ZAPF_DINGBATS - Zapf-dingbats font.

14.9 Colors

The FL_Color enumeration type holds a FLTK color value. Colors are either 8-bit indexes into a [virtual colormap](#) or 24-bit RGB color values. Color indices occupy the lower 8 bits of the value, while RGB colors occupy the upper 24 bits, for a byte organization of RGBI.

14.9.1 Color Constants

Constants are defined for the user-defined foreground and background colors, as well as specific colors and the start of the grayscale ramp and color cube in the [virtual colormap](#). Inline functions are provided to retrieve specific grayscale, color cube, or RGB color values.

The following color constants can be used to access the user-defined colors:

- FL_BACKGROUND_COLOR - the default background color
- FL_BACKGROUND2_COLOR - the default background color for text, list, and valuator widgets
- FL_FOREGROUND_COLOR - the default foreground color (0) used for labels and text
- FL_INACTIVE_COLOR - the inactive foreground color
- FL_SELECTION_COLOR - the default selection/highlight color

The following color constants can be used to access the colors from the FLTK standard color cube:

- FL_BLACK
- FL_BLUE
- FL_CYAN
- FL_DARK_BLUE
- FL_DARK_CYAN

- FL_DARK_GREEN
- FL_DARK_MAGENTA
- FL_DARK_RED
- FL_DARK_YELLOW
- FL_GREEN
- FL_MAGENTA
- FL_RED
- FL_WHITE
- FL_YELLOW

The following are named values within the standard grayscale:

- FL_GRAY0
- FL_DARK3
- FL_DARK2
- FL_DARK1
- FL_LIGHT1
- FL_LIGHT2
- FL_LIGHT3

The inline methods for getting a grayscale, color cube, or RGB color value are described in the [Colors](#) section of the [Drawing Things in FLTK](#) chapter.

14.10 Cursors

The following constants define the mouse cursors that are available in FLTK. The double-headed arrows are bitmaps provided by FLTK on X, the others are provided by system-defined cursors.

- FL_CURSOR_DEFAULT - the default cursor, usually an arrow (0)
- FL_CURSOR_ARROW - an arrow pointer
- FL_CURSOR_CROSS - crosshair
- FL_CURSOR_WAIT - watch or hourglass
- FL_CURSOR_INSERT - I-beam
- FL_CURSOR_HAND - hand (uparrow on MSWindows)
- FL_CURSOR_HELP - question mark
- FL_CURSOR_MOVE - 4-pointed arrow
- FL_CURSOR_NS - up/down arrow
- FL_CURSOR_WE - left/right arrow
- FL_CURSOR_NWSE - diagonal arrow
- FL_CURSOR_NESW - diagonal arrow
- FL_CURSOR_NONE - invisible

14.11 FD "When" Conditions

- `FL_READ` - Call the callback when there is data to be read.
- `FL_WRITE` - Call the callback when data can be written without blocking.
- `FL_EXCEPT` - Call the callback if an exception occurs on the file.

14.12 Damage Masks

The following damage mask bits are used by the standard FLTK widgets:

- `FL_DAMAGE_CHILD` - A child needs to be redrawn.
- `FL_DAMAGE_EXPOSE` - The window was exposed.
- `FL_DAMAGE_SCROLL` - The [Fl_Scroll](#) widget was scrolled.
- `FL_DAMAGE_OVERLAY` - The overlay planes need to be redrawn.
- `FL_DAMAGE_USER1` - First user-defined damage bit.
- `FL_DAMAGE_USER2` - Second user-defined damage bit.
- `FL_DAMAGE_ALL` - Everything needs to be redrawn.

Chapter 15

GLUT Compatibility

This appendix describes the GLUT compatibility header file supplied with FLTK.

FLTK's GLUT compatibility is based on the original GLUT 3.7 and the follow-on FreeGLUT 2.4.0 libraries.

15.1 Using the GLUT Compatibility Header File

You should be able to compile existing GLUT source code by including `<FL/glut.H>` instead of `<GL/glut.h>`. This can be done by editing the source, by changing the `-I` switches to the compiler, or by providing a symbolic link from `GL/glut.h` to `FL/glut.H`.

All files calling GLUT procedures must be compiled with C++. You may have to alter them slightly to get them to compile without warnings, and you may have to rename them to get make to use the C++ compiler.

You must link with the FLTK library. Most of `FL/glut.H` is inline functions. You should take a look at it (and maybe at `test/glpuzzle.cxx` in the FLTK source) if you are having trouble porting your GLUT program.

This has been tested with most of the demo programs that come with the GLUT and FreeGLUT distributions.

15.2 Known Problems

The following functions and/or arguments to functions are missing, and you will have to replace them or comment them out for your code to compile:

- `glutGet (GLUT_ELAPSED_TIME)`
- `glutGet (GLUT_SCREEN_HEIGHT_MM)`
- `glutGet (GLUT_SCREEN_WIDTH_MM)`
- `glutGet (GLUT_WINDOW_NUM_CHILDREN)`
- `glutInitDisplayMode (GLUT_LUMINANCE)`
- `glutKeyboardUpFunc(void(*callback)(unsigned char key, int x, int y))`
- `glutLayerGet (GLUT_HAS_OVERLAY)`

- `glutLayerGet (GLUT_LAYER_IN_USE)`
- `glutPushWindow ()`
- `glutSetColor (), glutGetColor (), glutCopyColormap ()`
- `glutVideoResize ()` missing.
- `glutWarpPointer ()`
- `glutWindowStatusFunc ()`
- Spaceball, buttonbox, dials, and tablet functions

Most of the symbols/enumerations have different values than GLUT uses. This will break code that relies on the actual values. The only symbols guaranteed to have the same values are true/false pairs like `GLUT_DOWN` and `GLUT_UP`, mouse buttons `GLUT_LEFT_BUTTON`, `GLUT_MIDDLE_BUTTON`, `GLUT_RIGHT_BUTTON`, and `GLUT_KEY_F1` thru `GLUT_KEY_F12`.

The strings passed as menu labels are not copied.

`glutPostRedisplay ()` does not work if called from inside a display function. You must use `glutIdleFunc ()` if you want your display to update continuously.

`glutSwapBuffers ()` does not work from inside a display function. This is on purpose, because FLTK swaps the buffers for you.

`glutUseLayer ()` does not work well, and should only be used to initialize transformations inside a resize callback. You should redraw overlays by using `glutOverlayDisplayFunc ()`.

Overlays are cleared before the overlay display function is called. `glutLayerGet (GLUT_OVERLAY_DAMAGED)` always returns true for compatibility with some GLUT overlay programs. You must rewrite your code so that `gl_color ()` is used to choose colors in an overlay, or you will get random overlay colors.

`glutSetCursor (GLUT_CURSOR_FULL_CROSSHAIR)` just results in a small crosshair.

The fonts used by `glutBitmapCharacter ()` and `glutBitmapWidth ()` may be different.

`glutInit (argc, argv)` will consume different switches than GLUT does. It accepts the switches recognized by `Fl::args()`, and will accept any abbreviation of these switches (such as `"-di"` for `"-display"`).

15.3 Mixing GLUT and FLTK Code

You can make your GLUT window a child of a `Fl_Window` with the following scheme. The biggest trick is that GLUT insists on a call to `show ()` the window at the point it is created, which means the `Fl_Window` parent window must already be shown.

- Don't call `glutInit ()`.
- Create your `Fl_Window`, and any FLTK widgets. Leave a blank area in the window for your GLUT window.
- `show ()` the `Fl_Window`. Perhaps call `show (argc, argv)`.
- Call `window->begin ()` so that the GLUT window will be automatically added to it.
- Use `glutInitWindowSize ()` and `glutInitWindowPosition ()` to set the location in the parent window to put the GLUT window.
- Put your GLUT code next. It probably does not need many changes. Call `window->end ()` immediately after the `glutCreateWindow ()` !
- You can call either `glutMainLoop ()`, `Fl::run()`, or loop calling `Fl::wait()` to run the program.

15.4 class Fl_Glut_Window

15.4.1 Class Hierarchy

```
Fl_Gl_Window
|
+----Fl_Glut_Window
```

15.4.2 Include Files

```
#include <FL/glut.H>
```

15.4.3 Description

Each GLUT window is an instance of this class. You may find it useful to manipulate instances directly rather than use GLUT window id's. These may be created without opening the display, and thus can fit better into FLTK's method of creating windows.

The current GLUT window is available in the global variable `glut_window`.

`new Fl_Glut_Window(...)` is the same as `glutCreateWindow()` except it does not `show()` the window or make the window current.

`window->make_current()` is the same as `glutSetWindow(number)`. If the window has not had `show()` called on it yet, some functions that assume an OpenGL context will not work. If you do `show()` the window, call `make_current()` again to set the context.

`~Fl_Glut_Window()` is the same as `glutDestroyWindow()`.

15.4.4 Members

The `Fl_Glut_Window` class contains several public members that can be altered directly:

member	description
<code>display</code>	A pointer to the function to call to draw the normal planes.
<code>entry</code>	A pointer to the function to call when the mouse moves into or out of the window.
<code>keyboard</code>	A pointer to the function to call when a regular key is pressed.
<code>menu[3]</code>	The menu to post when one of the mouse buttons is pressed.
<code>mouse</code>	A pointer to the function to call when a button is pressed or released.
<code>motion</code>	A pointer to the function to call when the mouse is moved with a button down.
<code>overlaydisplay</code>	A pointer to the function to call to draw the overlay planes.
<code>passivemotion</code>	A pointer to the function to call when the mouse is moved with no buttons down.
<code>reshape</code>	A pointer to the function to call when the window is resized.
<code>special</code>	A pointer to the function to call when a special key is pressed.
<code>visibility</code>	A pointer to the function to call when the window is iconified or restored (made visible.)

15.4.5 Methods

```
FI_Glut_Window::FI_Glut_Window(int x, int y, int w, int h, const char *title = 0)  
FI_Glut_Window::FI_Glut_Window(int w, int h, const char *title = 0)
```

The first constructor takes 4 int arguments to create the window with a preset position and size. The second constructor with 2 arguments will create the window with a preset size, but the window manager will choose the position according to its own whims.

```
virtual FI_Glut_Window::~FI_Glut_Window()
```

Destroys the GLUT window.

```
void FI_Glut_Window::make_current()
```

Switches all drawing functions to the GLUT window.

Chapter 16

Forms Compatibility

This appendix describes the Forms compatibility included with FLTK.

Warning: The Forms compatibility is deprecated and no longer maintained in FLTK 1.3, and is likely to be removed completely in FLTK 1.4

16.1 Importing Forms Layout Files

`FLUID` can read the `.fd` files put out by all versions of Forms and XForms fdesign. However, it will mangle them a bit, but it prints a warning message about anything it does not understand. `FLUID` cannot write fdesign files, so you should save to a new name so you don't write over the old one.

You will need to edit your main code considerably to get it to link with the output from `FLUID`. If you are not interested in this you may have more immediate luck with the forms compatibility header, `<FL/forms.H>`.

16.2 Using the Compatibility Header File

You should be able to compile existing Forms or XForms source code by changing the include directory switch to your compiler so that the `forms.h` file supplied with FLTK is included. The `forms.h` file simply pulls in `<FL/forms.H>` so you don't need to change your source code. Take a look at `<FL/forms.H>` to see how it works, but the basic trick is lots of inline functions. Most of the XForms demo programs work without changes.

You will also have to compile your Forms or XForms program using a C++ compiler. The FLTK library does not provide C bindings or header files.

Although FLTK was designed to be compatible with the GL Forms library (version 0.3 or so), XForms has bloated severely and its interface is X-specific. Therefore, XForms compatibility is no longer a goal of FLTK. Compatibility was limited to things that were free, or that would add code that would not be linked in if the feature is unused, or that was not X-specific.

To use any new features of FLTK, you should rewrite your code to not use the inline functions and instead use "pure" FLTK. This will make it a lot cleaner and make it easier to figure out how to call the FLTK functions. Unfortunately this conversion is harder than expected and even Digital Domain's inhouse code still uses `forms.H` a lot.

16.3 Problems You Will Encounter

Many parts of XForms use X-specific structures like `XEvent` in their interface. I did not emulate these! Unfortunately these features (such as the "canvas" widget) are needed by most large programs. You will need to rewrite these to use FLTK subclasses.

`Fl_Free` widgets emulate the *old* Forms "free" widget. It may be useful for porting programs that change the `handle()` function on widgets, but you will still need to rewrite things.

`Fl_Timer` widgets are provided to emulate the XForms timer. These work, but are quite inefficient and inaccurate compared to using `Fl::add_timeout()`.

All instance variables are hidden. If you directly refer to the `x`, `y`, `w`, `h`, `label`, or other fields of your Forms widgets you will have to add empty parenthesis after each reference. The easiest way to do this is to globally replace `"->x"` with `"->x()"`, etc. Replace `"boxtype"` with `"box()"`.

`const char *` arguments to most FLTK methods are simply stored, while Forms would `strdup()` the passed string. This is most noticeable with the label of widgets. Your program must always pass static data such as a string constant or malloc'd buffer to `label()`. If you are using labels to display program output you may want to try the `Fl_Output` widget.

The default fonts and sizes are matched to the older GL version of Forms, so all labels will draw somewhat larger than an XForms program does.

`fdesign` outputs a setting of a "fdui" instance variable to the main window. I did not emulate this because I wanted all instance variables to be hidden. You can store the same information in the `user_data()` field of a window. To do this, search through the `fdesign` output for all occurrences of `"->fdui"` and edit to use `"->user_data()"` instead. This will require casts and is not trivial.

The prototype for the functions passed to `fl_add_timeout()` and `fl_set_idle_callback()` callback are different.

All the following XForms calls are missing:

- `FL_REVISION, fl_library_version()`
- `FL_RETURN_DBLCLICK` (use `Fl::event_clicks()`)
- `fl_add_signal_callback()`
- `fl_set_form_atactivate()` `fl_set_form_atdeactivate()`
- `fl_set_form_property()`
- `fl_set_app_mainform()`, `fl_get_app_mainform()`
- `fl_set_form_minsize()`, `fl_set_form_maxsize()`
- `fl_set_form_event_cmask()`, `fl_get_form_event_cmask()`
- `fl_set_form_dblbuffer()`, `fl_set_object_dblbuffer()` (use an `Fl_Double_Window` instead)
- `fl_adjust_form_size()`
- `fl_register_raw_callback()`
- `fl_set_object_bw()`, `fl_set_border_width()`
- `fl_set_object_resize()`, `fl_set_object_gravity()`
- `fl_set_object_shortcutkey()`

- `fl_set_object_automatic()`
- `fl_get_object_bbox()` (maybe FLTK should do this)
- `fl_set_object_prehandler()`, `fl_set_object_posthandler()`
- `fl_enumerate_fonts()`
- Most drawing functions
- `fl_set_coordunit()` (FLTK uses pixels all the time)
- `fl_ringbell()`
- `fl_gettime()`
- `fl_win*()` (all these functions)
- `fl_initialize(argc, argv, x, y, z)` ignores last 3 arguments
- `fl_read_bitmapfile()`, `fl_read_pixmapfile()`
- `fl_addto_browser_chars()`
- `FL_MENU_BUTTON` just draws normally
- `fl_set_bitmapbutton_file()`, `fl_set_pixmapbutton_file()`
- `FL_CANVAS` objects
- `FL_DIGITAL_CLOCK` (comes out analog)
- `fl_create_bitmap_cursor()`, `fl_set_cursor_color()`
- `fl_set_dial_angles()`
- `fl_show_oneliner()`
- `fl_set_choice_shortcut(a, b, c)`
- command log
- Only some of file selector is emulated
- `FL_DATE_INPUT`
- `fl_pup*()` (all these functions)
- textbox object (should be easy but I had no sample programs)
- xyplot object

16.4 Additional Notes

These notes were written for porting programs written with the older IRISGL version of Forms. Most of these problems are the same ones encountered when going from old Forms to XForms:

Does Not Run In Background

The IRISGL library always forked when you created the first window, unless `foreground()` was called. FLTK acts like `foreground()` is called all the time. If you really want the fork behavior do `if (fork()) exit(0)` right at the start of your program.

You Cannot Use IRISGL Windows or fl_queue

If a Forms (not XForms) program if you wanted your own window for displaying things you would create a IRISGL window and draw in it, periodically calling Forms to check if the user hit buttons on the panels. If the user did things to the IRISGL window, you would find this out by having the value `FL_EVENT` returned from the call to Forms.

None of this works with FLTK. Nor will it compile, the necessary calls are not in the interface.

You have to make a subclass of `Fl_Gl_Window` and write a `draw()` method and `handle()` method. This may require anywhere from a trivial to a major rewrite.

If you draw into the overlay planes you will have to also write a `draw_overlay()` method and call `redraw_overlay()` on the OpenGL window.

One easy way to hack your program so it works is to make the `draw()` and `handle()` methods on your window set some static variables, storing what event happened. Then in the main loop of your program, call `Fl::wait()` and then check these variables, acting on them as though they are events read from `fl_queue`.

You Must Use OpenGL to Draw Everything

The file `<FL/gl.h>` defines replacements for a lot of IRISGL calls, translating them to OpenGL. There are much better translators available that you might want to investigate.

You Cannot Make Forms Subclasses

Programs that call `fl_make_object` or directly setting the handle routine will not compile. You have to rewrite them to use a subclass of `Fl_Widget`. It is important to note that the `handle()` method is not exactly the same as the `handle()` function of Forms. Where a Forms `handle()` returned non-zero, your `handle()` must call `do_callback()`. And your `handle()` must return non-zero if it "understood" the event.

An attempt has been made to emulate the "free" widget. This appears to work quite well. It may be quicker to modify your subclass into a "free" widget, since the "handle" functions match.

If your subclass draws into the overlay you are in trouble and will have to rewrite things a lot.

You Cannot Use `<device.h>`

If you have written your own "free" widgets you will probably get a lot of errors about "getvaluator". You should substitute:

Forms	FLTK
MOUSE_X	<code>Fl::event_x_root()</code>
MOUSE_Y	<code>Fl::event_y_root()</code>
LEFTSHIFTKEY,RIGHTSHIFTKEY	<code>Fl::event_shift()</code>
CAPSLCKKEY	<code>Fl::event_capslock()</code>
LEFTCTRLKEY,RIGHTCTRLKEY	<code>Fl::event_ctrl()</code>
LEFTALTKEY,RIGHTALTKEY	<code>Fl::event_alt()</code>
MOUSE1,RIGHTMOUSE	<code>Fl::event_state()</code>
MOUSE2,MIDDLEMOUSE	<code>Fl::event_state()</code>
MOUSE3,LEFTMOUSE	<code>Fl::event_state()</code>

Anything else in `getvaluator` and you are on your own...

Font Numbers Are Different

The "style" numbers have been changed because I wanted to insert bold-italic versions of the normal fonts. If you use Times, Courier, or Bookman to display any text you will get a different font out of FLTK. If you are really desperate to fix this use the following code:

```
fl_font_name(3, "*courier-medium-r-no*");  
fl_font_name(4, "*courier-bold-r-no*");  
fl_font_name(5, "*courier-medium-o-no*");  
fl_font_name(6, "*times-medium-r-no*");  
fl_font_name(7, "*times-bold-r-no*");  
fl_font_name(8, "*times-medium-i-no*");  
fl_font_name(9, "*bookman-light-r-no*");  
fl_font_name(10, "*bookman-demi-r-no*");  
fl_font_name(11, "*bookman-light-i-no*");
```


Chapter 17

Operating System Issues

This appendix describes the operating system specific interfaces in FLTK:

- [Accessing the OS Interfaces](#)
- [The UNIX \(X11\) Interface](#)
- [The Windows \(WIN32\) Interface](#)
- [The Apple OS X Interface](#)

17.1 Accessing the OS Interfaces

All programs that need to access the operating system specific interfaces must include the following header file:

```
#include <FL/x.H>
```

Despite the name, this header file will define the appropriate interface for your environment.

Note

This header file name "x.H" is changed in FLTK 1.4.0 to the better name "platform.H". Since FLTK 1.3.5 there is a compatibility header file [FL/platform.H](#) that includes [FL/x.H](#) to help you move to FLTK 1.4.0. If your code is targeted at FLTK 1.3.5 or higher you can safely change it to include [FL/platform.H](#) instead. FLTK 1.4.x will keep the file "x.H" for a few releases for backwards compatibility.

The pages that follow describe the functionality that is provided for each operating system.

WARNING:

The interfaces provided by this header file may change radically in new FLTK releases. Use them only when an existing generic FLTK interface is not sufficient.

17.2 The UNIX (X11) Interface

The UNIX interface provides access to the X Window System state information and data structures.

17.2.1 Handling Other X Events

void [Fl::add_handler](#)(int (*f)(int))

Installs a function to parse unrecognized events. If FLTK cannot figure out what to do with an event, it calls each of these functions (most recent first) until one of them returns non-zero. If none of them returns non-zero then the event is ignored.

FLTK calls this for any X events it does not recognize, or X events with a window ID that FLTK does not recognize. You can look at the X event in the `fl_xevent` variable.

The argument is the FLTK event type that was not handled, or zero for unrecognized X events. These handlers are also called for global shortcuts and some other events that the widget they were passed to did not handle, for example `FL_SHORTCUT`.

extern XEvent *fl_xevent

This variable contains the most recent X event.

extern ulong fl_event_time

This variable contains the time stamp from the most recent X event that reported it; not all events do. Many X calls like cut and paste need this value.

Window fl_xid(const Fl_Window *)

Returns the XID for a window, or zero if not `shown()`.

[Fl_Window](#) *fl_find(ulong xid)

Returns the [Fl_Window](#) that corresponds to the given XID, or `NULL` if not found. This function uses a cache so it is slightly faster than iterating through the windows yourself.

int fl_handle(const XEvent &)

This call allows you to supply the X events to FLTK, which may allow FLTK to cooperate with another toolkit or library. The return value is non-zero if FLTK understood the event. If the window does not belong to FLTK and the `add_handler()` functions all return 0, this function will return false.

Besides feeding events your code should call [Fl::flush\(\)](#) periodically so that FLTK redraws its windows.

This function will call the callback functions. It will not return until they complete. In particular, if a callback pops up a modal window by calling [fl_ask\(\)](#), for instance, it will not return until the modal function returns.

17.2.2 Drawing using Xlib

The following global variables are set before `FL_Widget::draw()` is called, or by `FL_Window::make_current()`:

```
extern Display *fl_display;
extern Window fl_window;
extern GC fl_gc;
extern int fl_screen;
extern XVisualInfo *fl_visual;
extern Colormap fl_colormap;
```

You must use them to produce Xlib calls. Don't attempt to change them. A typical X drawing call is written like this:

```
XDrawSomething(fl_display, fl_window, fl_gc, ...);
```

Other information such as the position or size of the X window can be found by looking at `FL_Window::current()`, which returns a pointer to the `FL_Window` being drawn.

```
unsigned long fl_xpixel(FL_Color i)
unsigned long fl_xpixel(uchar r, uchar g, uchar b)
```

Returns the X pixel number used to draw the given FLTK color index or RGB color. This is the X pixel that `fl_color()` would use.

```
int fl_parse_color(const char* p, uchar& r, uchar& g, uchar& b)
```

Convert a name into the red, green, and blue values of a color by parsing the X11 color names. On other systems, `fl_parse_color()` can only convert names in hexadecimal encoding, for example `#ff8083`.

```
extern XFontStruct *fl_xfont
```

Points to the font selected by the most recent `fl_font()`. This is not necessarily the current font of `fl_gc`, which is not set until `fl_draw()` is called. If FLTK was compiled with Xft support, `fl_xfont` will usually be 0 and `fl_xfont` will contain a pointer to the `XftFont` structure instead.

```
extern void *fl_xffont
```

If FLTK was compiled with Xft support enabled, `fl_xffont` points to the xft font selected by the most recent `fl_font()`. Otherwise it will be 0. `fl_xffont` should be cast to `XftFont*`.

17.2.3 Changing the Display, Screen, or X Visual

FLTK uses only a single display, screen, X visual, and X colormap. This greatly simplifies its internal structure and makes it much smaller and faster. You can change which it uses by setting global variables *before the first `FL_Window::show()` is called*. You may also want to call `Fl::visual()`, which is a portable interface to get a full color and/or double buffered visual.

```
int Fl::display(const char *)
```

Set which X display to use. This actually does `putenv("DISPLAY=...")` so that child programs will display on the same screen if called with `exec()`. This must be done before the display is opened. This call is provided under MacOS and WIN32 but it has no effect.

```
extern Display *fl_display
```

The open X display. This is needed as an argument to most Xlib calls. Don't attempt to change it! This is `NULL` before the display is opened.

```
void fl_open_display()
```

Opens the display. Does nothing if it is already open. This will make sure `fl_display` is non-zero. You should call this if you wish to do X calls and there is a chance that your code will be called before the first `show()` of a window.

This may call `Fl::abort()` if there is an error opening the display.

```
void fl_close_display()
```

This closes the X connection. You do *not* need to call this to exit, and in fact it is faster to not do so! It may be useful to call this if you want your program to continue without the X connection. You cannot open the display again, and probably cannot call any FLTK functions.

```
extern int fl_screen
```

Which screen number to use. This is set by `fl_open_display()` to the default screen. You can change it by setting this to a different value immediately afterwards. It can also be set by changing the last number in the `Fl::display()` string to "host:0.#".

```
extern XVisualInfo *fl_visual
extern Colormap fl_colormap
```

The visual and colormap that FLTK will use for all windows. These are set by `fl_open_display()` to the default visual and colormap. You can change them before calling `show()` on the first window. Typical code for changing the default visual is:

```
Fl::args(argc, argv); // do this first so $DISPLAY is set
fl_open_display();
fl_visual = find_a_good_visual(fl_display, fl_screen);
if (!fl_visual) Fl::abort("No good visual");
fl_colormap = make_a_colormap(fl_display, fl_visual->visual, fl_visual->depth);
// it is now ok to show() windows:
window->show(argc, argv);
```


17.2.4 Using a Subclass of `Fl_Window` for Special X Stuff

FLTK can manage an X window on a different screen, visual and/or colormap, you just can't use FLTK's drawing routines to draw into it. But you can write your own `draw()` method that uses Xlib (and/or OpenGL) calls only.

FLTK can also manage XID's provided by other libraries or programs, and call those libraries when the window needs to be redrawn.

To do this, you need to make a subclass of `Fl_Window` and override some of these virtual functions:

virtual void `Fl_Window::show()`

If the window is already `shown()` this must cause it to be raised, this can usually be done by calling `Fl_Window::show()`. If not `shown()` your implementation must call either `Fl_X::set_xid()` or `Fl_X::make_xid()`.

An example:

```
void MyWindow::show() {
    if (shown()) {Fl_Window::show(); return;} // you must do this!
    fl_open_display(); // necessary if this is first window
    // we only calculate the necessary visual colormap once:
    static XVisualInfo *visual;
    static Colormap colormap;
    if (!visual) {
        visual = figure_out_visual();
        colormap = XCreateColormap(fl_display, RootWindow(fl_display, fl_screen),
                                   vis->visual, AllocNone);
    }
    Fl_X::make_xid(this, visual, colormap);
}
```

```
Fl_X *Fl_X::set_xid(Fl_Window*, Window xid)
```

Allocate a hidden class called an `Fl_X`, put the XID into it, and set a pointer to it from the `Fl_Window`. This causes `Fl_Window::shown()` to return true.

```
void Fl_X::make_xid(Fl_Window*, XVisualInfo* = fl_visual, Colormap = fl_colormap)
```

This static method does the most onerous parts of creating an X window, including setting the label, resize limitations, etc. It then does `Fl_X::set_xid()` with this new window and maps the window.

virtual void `Fl_Window::flush()`

This virtual function is called by `Fl::flush()` to update the window. For FLTK's own windows it does this by setting the global variables `fl_window` and `fl_gc` and then calling the `draw()` method. For your own windows you might just want to put all the drawing code in here.

The X region that is a combination of all `damage()` calls done so far is in `Fl_X::i(this)->region`. If `NULL` then you should redraw the entire window. The undocumented function `fl_clip_region()` (↔ `XRegion`) will initialize the FLTK clip stack with a region or `NULL` for no clipping. You must set region to `NULL` afterwards as `fl_clip_region()` will own and delete it when done.

If `damage()` & `FL_DAMAGE_EXPOSE` then only X expose events have happened. This may be useful if you have an undamaged image (such as a backing buffer) around.

Here is a sample where an undamaged image is kept somewhere:

```
void MyWindow::flush() {
    fl_clip_region(Fl_X::i(this)->region);
    Fl_X::i(this)->region = 0;
    if (damage() != 2) {... draw things into backing store ...}
    ... copy backing store to window ...
}
```

virtual void `Fl_Window::hide()`

Destroy the window server copy of the window. Usually you will destroy contexts, pixmaps, or other resources used by the window, and then call `Fl_Window::hide()` to get rid of the main window identified by `xid()`. If you override this, you must also override the destructor as shown:

```
void MyWindow::hide() {
    if (mypixmap) {
        XFreePixmap(fl_display, mypixmap);
        mypixmap = 0;
    }
    Fl_Window::hide(); // you must call this
}
```

virtual void `Fl_Window::~~Fl_Window()`

Because of the way C++ works, if you override `hide()` you *must* override the destructor as well (otherwise only the base class `hide()` is called):

```
MyWindow::~~MyWindow() {
    hide();
}
```

Note

Access to the `Fl_X` hidden class requires to `#define FL_INTERNALS` before compilation.

17.2.5 Setting the Icon of a Window

FLTK currently supports setting a window's icon **before** it is shown using the `Fl_Window::icon()` method.

void `Fl_Window::icon(const void *)`

Sets the icon for the window to the passed pointer. You will need to cast the icon `Pixmap` to a `char*` when calling this method. To set a monochrome icon using a bitmap compiled with your application use:

```
#include "icon.xbm"
fl_open_display(); // needed if display has not been previously opened
Pixmap p = XCreateBitmapFromData(fl_display, DefaultRootWindow(fl_display),
                                icon_bits, icon_width, icon_height);
window->icon((const void*)p);
```

To use a multi-colored icon, the XPM format and library should be used as follows:

```
#include <X11/xpm.h>
#include "icon.xpm"
fl_open_display(); // needed if display has not been previously opened
Pixmap p, mask;
XpmCreatePixmapFromData(fl_display, DefaultRootWindow(fl_display),
                        icon_xpm, &p, &mask, NULL);
window->icon((const void *)p);
```

When using the Xpm library, be sure to include it in the list of libraries that are used to link the application (usually `-lXpm`).

NOTE:

You must call `FL_Window::show(int argc, char** argv)` for the icon to be used. The `FL_Window::show()` method does not bind the icon to the window.

17.2.6 X Resources

When the `FL_Window::show(int argc, char** argv)` method is called, FLTK looks for the following X resources:

- `background` - The default background color for widgets (color).
- `dndTextOps` - The default setting for drag and drop text operations (boolean).
- `foreground` - The default foreground (label) color for widgets (color).
- `scheme` - The default scheme to use (string).
- `selectBackground` - The default selection color for menus, etc. (color).
- `Text.background` - The default background color for text fields (color).
- `tooltips` - The default setting for tooltips (boolean).
- `visibleFocus` - The default setting for visible keyboard focus on non-text widgets (boolean).

Resources associated with the first window's `FL_Window::xclass()` string are queried first, or if no class has been specified then the class `"fltk"` is used (e.g. `fltk.background`). If no match is found, a global search is done (e.g. `*background`).

17.3 The Windows (WIN32) Interface

The Windows interface provides access to the WIN32 GDI state information and data structures.

17.3.1 Using filenames with non-ASCII characters

In FLTK, all strings, including filenames, are UTF-8 encoded. The utility functions `fl_fopen()` and `fl_open()` allow to open files potentially having non-ASCII names in a cross-platform fashion, whereas the standard `fopen()/open()` functions fail to do so.

17.3.2 Responding to WM_QUIT

FLTK will intercept WM_QUIT messages that are directed towards the thread that runs the main loop. These are converted to SIGTERM signals via `raise()`. This allows you to deal with outside termination requests with the same code on both Windows and UNIX systems. Other processes can send this message via `PostThreadMessage()` in order to request, rather than force your application to terminate.

17.3.3 Handling Other WIN32 Messages

By default a single WNDCLASSEX called "FLTK" is created. All `Fl_Window` 's are of this class unless you use `Fl_Window::xclass()`. The window class is created the first time `Fl_Window::show()` is called.

You can probably combine FLTK with other libraries that make their own WIN32 window classes. The easiest way is to call `Fl::wait()`, as it will call `DispatchMessage()` for all messages to the other windows. If necessary you can let the other library take over as long as it calls `DispatchMessage()`, but you will have to arrange for the function `Fl::flush()` to be called regularly so that widgets are updated, timeouts are handled, and the idle functions are called.

```
extern MSG fl_msg
```

This variable contains the most recent message read by `GetMessage()`, which is called by `Fl::wait()`. This may not be the most recent message sent to an FLTK window, because silly WIN32 calls the handle procedures directly for some events (sigh).

```
void Fl::add_handler(int (*f)(int))
```

Installs a function to parse unrecognized messages sent to FLTK windows. If FLTK cannot figure out what to do with a message, it calls each of these functions (most recent first) until one of them returns non-zero. The argument passed to the functions is the FLTK event that was not handled or zero for unknown messages. If all the handlers return zero then FLTK calls `DefWindowProc()`.

```
HWND fl_xid(const Fl_Window *)
```

Returns the window handle for a `Fl_Window`, or zero if not `shown()`.

```
Fl_Window *fl_find(HWND xid)
```

Returns the `Fl_Window` that corresponds to the given window handle, or `NULL` if not found. This function uses a cache so it is slightly faster than iterating through the windows yourself.

17.3.4 Drawing Things Using the WIN32 GDI

When the virtual function `FL_Widget::draw()` is called, FLTK stores all the extra arguments you need to make a proper GDI call in some global variables:

```
extern HINSTANCE fl_display;
extern HWND fl_window;
extern HDC fl_gc;
COLORREF fl_RGB();
HPEN fl_pen();
HBRUSH fl_brush();
```

These global variables are set before `FL_Widget::draw()` is called, or by `FL_Window::make_current()`. You can refer to them when needed to produce GDI calls, but don't attempt to change them. The functions return GDI objects for the current color set by `fl_color()` and are created as needed and cached. A typical GDI drawing call is written like this:

```
DrawSomething(fl_gc, ..., fl_brush());
```

It may also be useful to refer to `FL_Window::current()` to get the window's size or position.

17.3.5 Setting the Icon of a Window

FLTK currently supports setting a window's icon *before* it is shown using the `FL_Window::icon()` method.

```
void FL_Window::icon(const void *)
```

Sets the icon for the window to the passed pointer. You will need to cast the `HICON` handle to a `char*` when calling this method. To set the icon using an icon resource compiled with your application use:

```
window->icon((const void *)LoadIcon(fl_display, MAKEINTRESOURCE(IDI_ICON)));
```

You can also use the `LoadImage()` and related functions to load specific resolutions or create the icon from bitmap data.

NOTE:

You must call `FL_Window::show(int argc, char** argv)` for the icon to be used. The `FL_Window::show()` method does not bind the icon to the window.

17.3.6 How to Not Get a MSDOS Console Window

WIN32 has a really stupid mode switch stored in the executables that controls whether or not to make a console window.

To always get a console window you simply create a console application (the `"/SUBSYSTEM:CONSOLE"` option for the linker). For a GUI-only application create a WIN32 application (the `"/SUBSYSTEM:WINDOWS"` option for the linker).

FLTK includes a `WinMain()` function that calls the ANSI standard `main()` entry point for you. *This function creates a console window when you use the debug version of the library.*

WIN32 applications without a console cannot write to `stdout` or `stderr`, even if they are run from a console window. Any output is silently thrown away. Additionally, WIN32 applications are run in the background by the console, although you can use `"start /wait program"` to run them in the foreground.

17.3.7 Known WIN32 Bugs and Problems

The following is a list of known bugs and problems in the WIN32 version of FLTK:

- If a program is deactivated, `Fl::wait()` does not return until it is activated again, even though many events are delivered to the program. This can cause idle background processes to stop unexpectedly. This also happens while the user is dragging or resizing windows or otherwise holding the mouse down. We were forced to remove most of the efficiency FLTK uses for redrawing in order to get windows to update while being moved. This is a design error in WIN32 and probably impossible to get around.
- `Fl_Gl_Window::can_do_overlay()` returns true until the first time it attempts to draw an overlay, and then correctly returns whether or not there is overlay hardware.
- `SetCapture` (used by `Fl::grab()`) doesn't work, and the main window title bar turns gray while menus are popped up.
- Compilation with `gcc 3.4.4` and `-Os` exposes an optimisation bug in `gcc`. The symptom is that when drawing filled circles only the perimeter is drawn. This can for instance be seen in the symbols demo. Other optimisation options such as `-O2` and `-O3` seem to work OK. More details can be found in STR#1656

17.4 The Apple OS X Interface

FLTK supports Apple OS X using the Apple Cocoa library. Older versions of MacOS are no longer supported.

Control, Option, and Command Modifier Keys

FLTK maps the Mac 'control' key to `FL_CTRL`, the 'option' key to `FL_ALT` and the 'Apple' key to `FL_META`. Furthermore, `FL_COMMAND` designates the 'Apple' key on Mac OS X and the 'control' key on other platforms. Keyboard events return the key name in `Fl::event_key()` and the keystroke translation in `Fl::event_text()`. For example, typing Option-Y on a Mac US keyboard will set `FL_ALT` in `Fl::event_state()`, set `Fl::event_key()` to 'y' and return the Yen symbol in `Fl::event_text()`.

Right Click simulation with Ctrl Click

The Apple HIG guidelines indicate applications should support 'Ctrl Click' to simulate 'Right Click' for e.g. context menus, so users with one-button mice and one-click trackpads can still access right-click features. However, paraphrasing [Manolo's comment on the fltk.coredev newsgroup](#):

- *FLTK does /not/ support Ctrl-Click == Right Click itself because Mac OS X event processing doesn't support this at the system level: the system reports left-clicks with the ctrl modifier when the user ctrl-clicks, and OS X system preferences don't allow changing this behavior. Therefore, applications must handle simulation of Right Click with Ctrl Click in the application code.*

Ian MacArthur provided the following `handle()` method code snippet showing an example of how to do this:

```

case FL_PUSH:
{
    int btn = Fl::event_button();
#ifdef __APPLE__
    int ev_state = Fl::event_state();
#endif
    //
    // Context menu can be called up in one of two ways: -
    // 1 - right click, as normally used on Windows and Linux
    // 2 - Ctrl + left click, as sometimes used on Mac
    //
#ifdef __APPLE__
    // On apple, check right click, and ctrl+left click
    if ((btn == FL_RIGHT_MOUSE) || (ev_state == (FL_CTRL | FL_BUTTON1)))
#else
    // On other platforms, only check right click as ctrl+left is used for selections
    if (btn == FL_RIGHT_MOUSE)
#endif
    {
        // Did we right click on the object?..
    }
}

```

There is a thread about this subject on `fltk.coredev` (Aug 1-14, 2014) entitled "[RFC] Right click emulation for one button mouse on Mac".

Apple "Quit" Event

When the user presses `Cmd-Q` or requests a termination of the application, FLTK reacts sending an `FL_CLOSE` event to all open windows. If any window remains open, the termination request aborts, and the app continues. If all windows close, FLTK default behaviour is to terminate the application immediately, without letting `Fl::run()` return. Consequently, potential cleanup code placed after the `Fl::run()` call does not run, and potential global destructors that would run after `main()` would return do not run. All code that should run so the app cleanly terminates must therefore be placed in window callbacks (which run when windows are closed) or in `atexit()` functions. Alternatively, FLTK can be directed to just terminate the event loop and therefore let potential cleanup code placed after return from `Fl::run()` and from `main()` execute. This is obtained setting global variable `fl_mac_quit_early` to 0.

Apple "Open" Event

Whenever the user drops a file onto an application icon, OS X generates an Apple Event of the type "Open". You can have FLTK notify you of an Open event by calling the `fl_open_callback` function.

`void fl_open_display()`

Opens the display. Does nothing if it is already open. You should call this if you wish to do Cocoa or Quartz calls and there is a chance that your code will be called before the first `show()` of a window.

Window `fl_xid(const Fl_Window *)`

Returns the window reference for an `Fl_Window`, or `NULL` if the window has not been shown. This reference is a pointer to an instance of the subclass `FLWindow` of Cocoa's `NSWindow` class.

`Fl_Window *fl_find(Window xid)`

Returns the `FL_Window` that corresponds to the given window reference, or `NULL` if not found.

void `fl_mac_set_about(FL_Callback *cb, void *user_data, int shortcut)`

Attaches the callback `cb` to the "About myprog" item of the system application menu. `cb` will be called with `NULL` first argument and `user_data` second argument.

`FL_Sys_Menu_Bar` class

The `FL_Sys_Menu_Bar` class allows to build menu bars that, on Mac OS X, are placed in the system menu bar (at top-left of display), and, on other platforms, at a user-chosen location of a user-chosen window.

17.4.1 Setting the icon of an application

- First, create a `.icns` file containing several copies of your icon of decreasing sizes. This can be done using the Preview application or the Icon Composer application available in "Graphics Tools for Xcode". To create a high resolution icon file, it is necessary to use the `iconutil` command-line utility.
- Put your `.icns` file in the Resources subdirectory of your application bundle.
- Add these two lines to the `Info.plist` file of your application bundle

```
<key>CFBundleIconFile</key>
<string>foo.icns</string>
```

replacing `foo` by your application name. If you use Xcode, just add your `.icns` file to your application target.

17.4.2 Drawing Things Using Quartz

All code inside `FL_Widget::draw()` is expected to call Quartz drawing functions. The Quartz coordinate system is flipped to match FLTK's coordinate system. The origin for all drawing is in the top left corner of the enclosing `FL_Window`. The global variable `fl_gc` (of type `CGContextRef`) is the appropriate Quartz 2D drawing environment. Include `FL/x.H` to declare the `fl_gc` variable.

17.4.3 Internationalization

All FLTK programs contain an application menu with, e.g., the About xxx, Hide xxx, and Quit xxx items. This menu can be internationalized/localized by any of two means.

- using the `FL_Mac_App_Menu` class.
- using the standard Mac OS X localization procedure. Create a language-specific `.lproj` directory (e.g., `German.lproj`) in the Resources subdirectory of the application bundle. Create therein a `Localizable.strings` file that translates all menu items to this language. The German `Localizable.strings` file, for example, contains:

```
"About %@" = "Über %@";
"Print Front Window"="Frontfenster drucken";
"Services" = "Dienste";
"Hide %@"="%@ ausblenden";
"Hide Others"="Andere ausblenden";
"Show All"="Alle einblenden";
"Quit %@"="%@ beenden";
```

Set `"Print Front Window" = ""`; therein so the application menu doesn't show a "Print Front Window" item. To localize the application name itself, create a file `InfoPlist.strings` in each `.lproj` directory and put `CFBundleName = "localized name"`; in each such file.

17.4.4 OpenGL and 'retina' displays

It is possible to have OpenGL produce graphics at the high pixel resolution allowed by the so-called 'retina' displays present on recent Apple hardware. For this, call

```
Fl::use_high_res_GL(1);
```

before any `Fl_Gl_Window` is shown. Also, adapt your `Fl_Gl_Window::draw()` and `Fl_Gl_Window::draw_overlay()` methods replacing

```
glViewport(0, 0, w(), h());
```

by

```
glViewport(0, 0, pixel_w(), pixel_h());
```

making use of the `Fl_Gl_Window::pixel_w()` and `Fl_Gl_Window::pixel_h()` methods that return the width and height of the GL scene in pixels: if the `Fl_Gl_Window` is mapped on a retina display, these methods return twice as much as reported by `Fl_Widget::w()` and `Fl_Widget::h()`; if it's mapped on a regular display, they return the same values as `w()` and `h()`. These methods dynamically change their values if the window is moved into/out from a retina display. If `Fl::use_high_res_GL(1)` is not called, all `Fl_Gl_Window`'s are drawn at low resolution. These methods are synonyms of `w()` and `h()` on non-Mac OS X platforms, so the source code remains cross-platform.

The `Fl_Gl_Window::pixels_per_unit()` method is useful when the OpenGL code depends on the pixel dimension of the GL scene. This occurs, e.g., if a window's `handle()` method uses `Fl::event_x()` and `Fl::event_y()` whose returned values should be multiplied by `Fl_Gl_Window::pixels_per_unit()` to obtain the adequate pixel units. This method may also be useful, for example, to adjust the width of a line in a high resolution GL scene.

17.4.5 Fl_Double_Window

OS X double-buffers all windows automatically. On OS X, `Fl_Window` and `Fl_Double_Window` are handled internally in the same way.

17.4.6 Mac File System Specifics

Resource Forks

FLTK does not access the resource fork of an application. However, a minimal resource fork must be created for OS X applications. Starting with OS X 10.6, resource forks are no longer needed.

Caution (OS X 10.2 and older):

When using UNIX commands to copy or move executables, OS X will NOT copy any resource forks! For copying and moving use `CpMac` and `MvMac` respectively. For creating a tar archive, all executables need to be stripped from their Resource Fork before packing, e.g. "DeRez fluid > fluid.r". After unpacking the Resource Fork needs to be reattached, e.g. "Rez fluid.r -o fluid".

It is advisable to use the Finder for moving and copying and Mac archiving tools like Sit for distribution as they will handle the Resource Fork correctly.

Mac File Paths

FLTK uses UTF-8-encoded UNIX-style filenames and paths.

See also

[Mac OS X-specific symbols](#)

Chapter 18

Migrating Code from FLTK 1.0 to 1.1

This appendix describes the differences between the FLTK 1.0.x and FLTK 1.1.x functions and classes.

18.1 Color Values

Color values are now stored in a 32-bit unsigned integer instead of the unsigned character in 1.0.x. This allows for the specification of 24-bit RGB values or 8-bit FLTK color indices.

`FL_BLACK` and `FL_WHITE` now remain black and white, even if the base color of the gray ramp is changed using `Fl::background()`. `FL_DARK3` and `FL_LIGHT3` can be used instead to draw a very dark or a very bright background hue.

Widgets use the new color symbols `FL_FOREGROUND_COLOR`, `FL_BACKGROUND_COLOR`, `FL_BACKGROUND2_COLOR`, `FL_INACTIVE_COLOR`, and `FL_SELECTION_COLOR`. More details can be found in the chapter [FLTK Enumerations](#).

18.2 Cut and Paste Support

The FLTK clipboard is now broken into two parts - a local selection value and a cut-and-paste value. This allows FLTK to support things like highlighting and replacing text that was previously cut or copied, which makes FLTK applications behave like traditional GUI applications.

18.3 File Chooser

The file chooser in FLTK 1.1.x is significantly different than the one supplied with FLTK 1.0.x. Any code that directly references the old `FCB` class or members will need to be ported to the new [Fl_File_Chooser](#) class.

18.4 Function Names

Some function names have changed from FLTK 1.0.x to 1.1.x in order to avoid name space collisions. You can still use the old function names by defining the `FLTK_1_0_COMPAT` symbol on the command-line when you compile (`-DFLTK_1_0_COMPAT`) or in your source, e.g.:

```
#define FLTK_1_0_COMPAT
#include <FL/Fl.H>
#include <FL/Enumerations.H>
#include <FL/filename.H>
```

The following table shows the old and new function names:

Old 1.0.x Name	New 1.1.x Name
contrast()	fl_contrast()
down()	fl_down()
filename_absolute()	fl_filename_absolute()
filename_expand()	fl_filename_expand()
filename_ext()	fl_filename_ext()
filename_isdir()	fl_filename_isdir()
filename_list()	fl_filename_list()
filename_match()	fl_filename_match()
filename_name()	fl_filename_name()
filename_relative()	fl_filename_relative()
filename_settext()	fl_filename_settext()
frame()	fl_frame()
inactive()	fl_inactive()
numeric_sort()	fl_numeric_sort()

18.5 Image Support

Image support in FLTK has been significantly revamped in 1.1.x. The [Fl_Image](#) class is now a proper base class, with the core image drawing functionality in the [Fl_Bitmap](#), [Fl_Pixmap](#), and [Fl_RGB_Image](#) classes.

BMP, GIF, JPEG, PNG, XBM, and XPM image files can now be loaded using the appropriate image classes, and the [Fl_Shared_Image](#) class can be used to cache images in memory.

Image labels are no longer provided as an add-on label type. If you use the old `label()` methods on an image, the widget's `image()` method is called to set the image as the label.

Image labels in menu items must still use the old `labeltype` mechanism to preserve source compatibility.

18.6 Keyboard Navigation

FLTK 1.1.x now supports keyboard navigation and control with all widgets. To restore the old FLTK 1.0.x behavior so that only text widgets get keyboard focus, call the [Fl::visible_focus\(\)](#) method to disable it:

```
Fl::visible_focus(0);
```

Chapter 19

Migrating Code from FLTK 1.1 to 1.3

This appendix describes the differences between the FLTK 1.1.x and FLTK 1.3.x functions and classes.

19.1 Migrating From FLTK 1.0

If you want to migrate your code from FLTK 1.0 to FLTK 1.3, then you should first consult [Appendix Migrating Code from FLTK 1.0 to 1.1](#).

19.2 FI_Scroll Widget

`FI_Scroll::scroll_to(int x, int y)` replaces `FI_Scroll::position(int x, int y)`.

This change was needed because `FI_Scroll::position(int,int)` redefined `FI_Widget::position(int,int)`, but with a completely different function (moving the scrollbars instead of moving the widget).

Please be aware that you need to change your application's code for all `FI_Scroll`-derived widgets, if you used `FI_Scroll::position(int x, int y)` to position **the scrollbars** (not the widget itself).

The compiler will not detect any errors, because your calls to `position(int x, int y)` will be calling `FI_Widget::position(int x, int y)`.

19.3 Unicode (UTF-8)

FLTK 1.3 uses Unicode (UTF-8) encoding internally. If you are only using characters in the ASCII range (32-127), there is a high probability that you don't need to modify your code. However, if you use international characters (128-255), encoded as e.g. Windows codepage 1252, ISO-8859-1, ISO-8859-15 or any other encoding, then you will need to update your character string constants and widget input data accordingly.

Please refer to the [Unicode and UTF-8 Support](#) chapter for more details.

Note

It is important that, although your software uses only ASCII characters for input to FLTK widgets, the user may enter non-ASCII characters, and FLTK will return these characters with UTF-8 encoding to your application, e.g. via `FI_Input::value()`. You **will** need to re-encode them to **your** (non-UTF-8) encoding, otherwise you might see or print garbage in your data.

19.4 Widget Coordinate Representation

FLTK 1.3 changed all Widget coordinate variables and methods, e.g. `Fl_Widget::x()`, `Fl_Widget::y()`, `Fl_Widget::w()`, `Fl_Widget::h()`, from short (16-bit) to int (32-bit) representation. This should not affect any existing code, but makes it possible to use bigger scroll areas (e.g. `Fl_Scroll` widget).

Chapter 20

Developer Information

This chapter describes FLTK development and documentation.

Example

```
/** \file
    Fl_Clock, Fl_Clock_Output widgets. */

/**
    \class Fl_Clock_Output
    \brief This widget can be used to display a program-supplied time.

    The time shown on the clock is not updated. To display the current time,
    use Fl_Clock instead.

    \image html clock.png
    \image latex clock.png "" width=10cm
    \image html round_clock.png
    \image latex clock.png "" width=10cm
    \image html round_clock.png "" width=10cm */

/**
    Returns the displayed time.
    Returns the time in seconds since the UNIX epoch (January 1, 1970).
    \see value(ulong)
    */
ulong value() const {return value_;}

/**
    Set the displayed time.
    Set the time in seconds since the UNIX epoch (January 1, 1970).
    \param[in] v seconds since epoch
    \see value()
    */
void Fl_Clock_Output::value(ulong v) {
    [...]
}

/**
    Create an Fl_Clock widget using the given position, size, and label string.
    The default boxtype is \c FL_NO_BOX.
    \param[in] X, Y, W, H position and size of the widget
    \param[in] L widget label, default is no label
    */
Fl_Clock::Fl_Clock(int X, int Y, int W, int H, const char *L)
    : Fl_Clock_Output(X, Y, W, H, L) {}

/**
```

```

    Create an Fl_Clock widget using the given boxtype, position, size, and
    label string.
    \param[in] t boxtype
    \param[in] X, Y, W, H position and size of the widget
    \param[in] L widget label, default is no label
    */
Fl_Clock::Fl_Clock(uchar t, int X, int Y, int W, int H, const char *L)
    : Fl_Clock_Output(X, Y, W, H, L) {
    type(t);
    box(t==FL_ROUND_CLOCK ? FL_NO_BOX : FL_UP_BOX);
}

```

Note

From Duncan: (will be removed later, just for now as a reminder)

I've just added comments for the `fl_color_chooser()` functions, and in order to keep them and the general Function Reference information for them together, I created a new doxygen group, and used `\ingroup` in the three comment blocks. This creates a new Modules page (which may not be what we want) with links to it from the File Members and [Fl_Color_Chooser.H](#) pages. It needs a bit more experimentation on my part unless someone already knows how this should be handled. (Maybe we can add it to a `functions.dox` file that defines a functions group and do that for all of the function documentation?)

Update: the trick is not to create duplicate entries in a new group, but to move the function information into the doxygen comments for the class, and use the navigation links provided. Simply using `\relatesalso` as the first doxygen command in the function's comment puts it in the appropriate place. There is no need to have `\defgroup` and `\ingroup` as well, and indeed they don't work. So, to summarize:

```

Gizmo.H
/** \class Gizmo
    A gizmo that does everything
    */
class Gizmo {
    etc
};
extern int popup_gizmo(...);

Gizmo.cxx:
/** \relatesalso Gizmo
    Pops up a gizmo dialog with a Gizmo in it
    */
int popup_gizmo(...);

```

Comments Within Doxygen Comment Blocks

You can use HTML comment statements to embed comments in doxygen comment blocks. These comments will not be visible in the generated document.

```

The following text is a developer comment.
<!-- *** This *** is *** invisible *** -->
This will be visible again.

```

will be shown as:

```

The following text is a developer comment.
<!-- *** This *** is *** invisible *** -->
This will be visible again.

```


Different Headlines

You can use HTML tags `<H1> ... <H4>` for headlines with different sizes. As of doxygen 1.8.x there must not be more than three spaces at the beginning of the line for this to work. Currently (doxygen 1.8.6) there seems to be no difference in the font sizes of `<H3>` and `<H4>` in the pdf output, whereas the html output uses different font sizes.

```
<H1>Headline in big text (H1)</H1>
<H2>Headline in big text (H2)</H2>
<H3>Headline in big text (H3)</H3>
<H4>Headline in big text (H4)</H4>
```

Headline in big text (H1)

Headline in big text (H2)

Headline in big text (H3)

Headline in big text (H4)

20.1 Non-ASCII Characters

Doxygen understands many HTML quoting characters like `"`, `ü`, `ç`, `Ç`, but not all HTML quoting characters.

This will appear in the document:

```
Doxygen understands many HTML quoting characters like
&quot;; &uuml;; &ccedil;; &Ccedil;; but not all HTML quoting characters.
```

For further informations about HTML quoting characters see

<http://www.doxygen.org/htmlcmds.html>

Alternatively you can use **UTF-8** encoding within Doxygen comments.

20.2 Document Structure

- `\page` creates a named page
- `\section` creates a named section within that page
- `\subsection` creates a named subsection within the current section
- `\subsubsection` creates a named subsubsection within the current subsection

All these statements take a "name" as their first argument, and a title as their second argument. The title can contain spaces.

The page, section, and subsection titles are formatted in blue color and a size like "<H1>", "<H2>", and "<H3>", and "<H4>", respectively.

By **FLTK documentation convention**, a file like this one with a doxygen documentation chapter has the name "<chapter>.dox". The `\page` statement at the top of the page is "`\page <chapter> This is the title`". Sections within a documentation page must be called "<chapter>_<section>", where "<chapter>" is the name part of the file, and "<section>" is a unique section name within the page that can be referenced in links. The same for subsections and subsubsections.

These doxygen page and section commands work only in special documentation chapters, not within normal source or header documentation blocks. However, links **from** normal (e.g. class) documentation **to** documentation sections **do work**.

This page has

```
\page development I - Developer Information
```

at its top.

This section is

```
\section development_structure Document Structure
```

The following section is

```
\section development_links Creating Links
```

20.3 Creating Links

Links to other documents and external links can be embedded with

- doxygen `\ref` links to other doxygen `\page`, `\section`, `\subsection` and `\anchor` locations
- HTML links without markup - doxygen creates "`http://...`" links automatically
- standard, non-Doxygen, HTML links

```
- see chapter \ref unicode creates a link to the named chapter
  unicode that has been created with a \page statement.
- For further informations about quoting see
  http://www.doxygen.org/htmlcmds.html
- see <a href="http://www.nedit.org/">Nedit</a> creates
  a standard HTML link
```

appears as:

- see chapter [Unicode and UTF-8 Support](#) creates a link to the named chapter unicode that has been created with a `\page` statement.
- For further informations about quoting see <http://www.doxygen.org/htmlcmds.html>
- see [Nedit](#) creates a standard HTML link

20.4 Paragraph Layout

There is no real need to use HTML `<P>` and `</P>` tags within the text to tell doxygen to start or stop a paragraph. In most cases, when doxygen encounters a blank line or some, but not all, `\commands` in the text it knows that it has reached the start or end of a paragraph. Doxygen also offers the `\par` command for special paragraph handling. It can be used to provide a paragraph title and also to indent a paragraph. Unfortunately `\par` won't do what you expect if you want to have doxygen links and sometimes html tags don't work either.

```
\par Normal Paragraph with title

This paragraph will have a title, but because there is a blank line
between the \par and the text, it will have the normal layout.

\par Indented Paragraph with title
This paragraph will also have a title, but because there is no blank
line between the \par and the text, it will be indented.

\par
It is also possible to have an indented paragraph without title.
This is how you indent subsequent paragraphs.

\par No link to Fl_Widget::draw()
Note that the paragraph title is treated as plain text.
Doxygen type links will not work.
HTML characters and tags may or may not work.

Fl_Widget::draw() links and "html" tags work<br>
\par
Use a single line ending with <br> for complicated paragraph titles.
```

The above code produces the following paragraphs:

Normal Paragraph with title

This paragraph will have a title, but because there is a blank line between the `\par` and the text, it will have the normal layout.

Indented Paragraph with title

This paragraph will also have a title, but because there is no blank line between the `\par` and the text, it will be indented.

It is also possible to have an indented paragraph without title. This is how you indent subsequent paragraphs.

No link to `Fl_Widget::draw()`

Note that the paragraph title is treated as plain text. Doxygen type links will not work. HTML characters and tags may or may not work.

[Fl_Widget::draw\(\)](#) links and "html" tags work

Use a single line ending with `
` for complicated paragraph titles.

20.5 Navigation Elements

Each introduction (tutorial) page ends with navigation elements. These elements must only be included in the html documentation, therefore they must be separated with `\htmlonly` and `\endhtmlonly`.

The following code gives the navigation bar at the bottom of this page:

```
\htmlonly
<hr>
<table summary="navigation bar" width="100%" border="0">
<tr>
  <td width="45%" align="LEFT">
    <a class="el" href="migration_1_3.html">
      [Prev]
      Migrating Code from FLTK 1.1 to 1.3
    </a>
  </td>
  <td width="10%" align="CENTER">
    <a class="el" href="index.html">[Index]</a>
  </td>
  <td width="45%" align="RIGHT">
    <a class="el" href="license.html">
      Software License
      [Next]
    </a>
  </td>
</tr>
</table>
\endhtmlonly
```

Chapter 21

Software License

December 11, 2001

The FLTK library and included programs are provided under the terms of the GNU Library General Public License (LGPL) with the following exceptions:

1. Modifications to the FLTK configure script, config header file, and makefiles by themselves to support a specific platform do not constitute a modified or derivative work.

The authors do request that such modifications be contributed to the FLTK project - send all contributions through the "Software Trouble Report" on the following page: <http://www.fltk.org/str.php>

2. Widgets that are subclassed from FLTK widgets do not constitute a derivative work.
3. Static linking of applications and widgets to the FLTK library does not constitute a derivative work and does not require the author to provide source code for the application or widget, use the shared FLTK libraries, or link their applications or widgets against a user-supplied version of FLTK.

If you link the application or widget to a modified version of FLTK, then the changes to FLTK must be provided under the terms of the LGPL in sections 1, 2, and 4.

4. You do not have to provide a copy of the FLTK license with programs that are linked to the FLTK library, nor do you have to identify the FLTK license in your program or documentation as required by section 6 of the LGPL.

However, programs must still identify their use of FLTK. The following example statement can be included in user documentation to satisfy this requirement:

[program/widget] is based in part on the work of the FLTK project (<http://www.fltk.org>).

GNU LIBRARY GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1991 Free Software Foundation, Inc.

59 Temple Place - Suite 330, Boston, MA 02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed. [This is the first released version of the library GPL. It is numbered 2 because it goes with version 2 of the ordinary GPL.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users.

This license, the Library General Public License, applies to some specially designated Free Software Foundation software, and to any other libraries whose authors decide to use it. You can use it for your libraries, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library, or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link a program with the library, you must provide complete object files to the recipients so that they can relink them with the library, after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

Our method of protecting your rights has two steps: (1) copyright the library, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the library.

Also, for each distributor's protection, we want to make certain that everyone understands that there is no warranty for this free library. If the library is modified by someone else and passed on, we want its recipients to know that what they have is not the original version, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that companies distributing free software will individually obtain patent licenses, thus in effect transforming the program into proprietary software. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License, which was designed for utility programs. This license, the GNU Library General Public License, applies to certain designated libraries. This license is quite different from the ordinary one; be sure to read it in full, and don't assume that anything in it is the same as in the ordinary license.

The reason we have a separate public license for some libraries is that they blur the distinction we usually make between modifying or adding to a program and simply using it. Linking a program with a library, without changing the library, is in some sense simply using the library, and is analogous to running a utility program or application program. However, in a textual and legal sense, the linked executable is a combined work, a derivative of the original library, and the ordinary General Public License treats it as such.

Because of this blurred distinction, using the ordinary General Public License for libraries did not effectively promote software sharing, because most developers did not use the libraries. We concluded that weaker conditions might promote sharing better.

However, unrestricted linking of non-free programs would deprive the users of those programs of all benefit from the free status of the libraries themselves. This Library General Public License is intended to permit developers of non-free programs to use free libraries, while preserving your freedom as a user of such programs to change the free libraries that are incorporated in them. (We have not seen how to achieve this as regards changes in header files, but we have achieved it as regards changes in the actual functions of the Library.) The hope is that this will lead to faster development of free libraries.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, while the latter only works together with the library.

Note that it is possible for a library to be covered by the ordinary General Public License rather than by this special one.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Library General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given

copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also compile or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

c) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

d) Verify that the user has already received a copy of these materials or that you have already sent this user a copy. For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Library General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE

COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Chapter 22

Example Source Code

The FLTK distribution contains over 60 sample applications written in, or ported to, FLTK.

If the FLTK archive you received does not contain either an 'examples' or 'test' directory, you can download the complete FLTK distribution from <http://fltk.org/software.php>.

Most of the example programs were created while testing a group of widgets. They are not meant to be great achievements in clean C++ programming, but merely a test platform to verify the functionality of the FLTK library.

Note that extra example programs are also available in an additional 'examples' directory, but these are **NOT** built automatically when you build FLTK, unlike those in the 'test' directory shown below.

22.1 Example Applications

adjuster	arc	ask	bitmap	blocks	boxtype
browser	button	buttons	checkers	clock	colbrowser
color_chooser	cube	CubeView	cursor	curve	demo
device	doublebuffer	editor	fast_slow	file_chooser	fluid
fonts	forms	fractals	fullscreen	gl_overlay	glpuzzle
hello	help	iconize	image	inactive	input
input_choice	keyboard	label	line_style	list_visuals	mandelbrot
menubar	message	minimum	navigation	output	overlay
pack	pixmap_browser	pixmap	preferences	radio	resizebox
resize	scroll	shape	subwindow	sudoku	symbols
tabs	threads	tile	tiled_image	unittests	utf8
valuators					

22.1.1 adjuster

`adjuster` shows a nifty little widget for quickly setting values in a great range.

22.1.2 arc

The `arc` demo explains how to derive your own widget to generate some custom drawings. The sample drawings use the matrix based arc drawing for some fun effects.

22.1.3 ask

`ask` shows some of FLTK's standard dialog boxes. Click the correct answers or you may end up in a loop, or you may end up in a loop, or you... .

22.1.4 bitmap

This simple test shows the use of a single color bitmap as a label for a box widget. Bitmaps are stored in the X11 '.bmp' file format and can be part of the source code.

22.1.5 blocks

A wonderful and addictive game that shows the usage of FLTK timers, graphics, and how to implement sound on all platforms. `blocks` is also a good example for the Mac OS X specific bundle format.

22.1.6 boxtype

`boxtype` gives an overview of readily available boxes and frames in FLTK. More types can be added by the application programmer. When using themes, FLTK shuffles boxtypes around to give your program a new look.

22.1.7 browser

`browser` shows the capabilities of the [FI_Browser](#) widget. Important features tested are loading of files, line formatting, and correct positioning of the browser data window.

22.1.8 button

The `button` test is a simple demo of push-buttons and callbacks.

22.1.9 buttons

`buttons` shows a sample of FLTK button types.

22.1.10 checkers

Written by Steve Poulsen in early 1979, `checkers` shows how to convert a VT100 text-terminal based program into a neat application with a graphical UI. Check out the code that drags the pieces, and how the pieces are drawn by layering. Then tell me how to beat the computer at Checkers.

22.1.11 clock

The `clock` demo shows two analog clocks. The innards of the [FI_Clock](#) widget are pretty interesting, explaining the use of timeouts and matrix based drawing.

22.1.12 colbrowser

`colbrowser` runs only on X11 systems. It reads `/usr/lib/X11/rgb.txt` to show the color representation of every text entry in the file. This is beautiful, but only moderately useful unless your UI is written in *Motif*.

22.1.13 color_chooser

The `color_chooser` gives a short demo of FLTK's palette based color chooser and of the RGB based color wheel.

22.1.14 cube

The `cube` demo shows the speed of OpenGL. It also tests the ability to render two OpenGL buffers into a single window, and shows OpenGL text.

22.1.15 CubeView

`CubeView` shows how to create a UI containing OpenGL with Fluid.

22.1.16 cursor

The `cursor` demo shows all mouse cursor shapes that come standard with FLTK. The `fgcolor` and `bgcolor` sliders work only on few systems (some version of Irix for example).

22.1.17 curve

`curve` draws a nice Bezier curve into a custom widget. The `points` option for splines is not supported on all platforms.

22.1.18 demo

This tool allows quick access to all programs in the `test` directory. `demo` is based on the visuals of the IrixGL demo program. The menu tree can be changed by editing `test/demo.menu`.

22.1.19 device

Exercises the `Fl_Image_Surface`, `Fl_Copy_Surface`, and `Fl_Printer` classes to draw to an `Fl_Image` object, copy graphical data to the clipboard, and for print support.

Note

The `clipboard.cxx` program of the 'examples' directory is a clipboard watching application that continuously displays the textual or graphical content of the system clipboard (a.k.a pasteboard on Mac OS X) exercising `Fl::paste()`.

22.1.20 doublebuffer

The `doublebuffer` demo shows the difference between a single buffered window, which may flicker during a slow redraw, and a double buffered window, which never flickers, but uses twice the amount of RAM. Some modern OS's double buffer all windows automatically to allow transparency and shadows on the desktop. FLTK is smart enough to not tripple buffer a window in that case.

22.1.21 editor

FLTK has two very different text input widgets. `Fl_Input` and derived classes are rather light weight, however `Fl_Text_Editor` is a complete port of `ncedit` (with permission). The `editor` test is almost a full application, showing custom syntax highlighting and dialog creation.

22.1.22 fast_slow

`fast_slow` shows how an application can use the `Fl_Widget::when()` setting to receive different kinds of callbacks.

22.1.23 file_chooser

The standard FLTK `file_chooser` is the result of many iterations, trying to find a middle ground between a complex browser and a fast light implementation.

22.1.24 fonts

`fonts` shows all available text fonts on the host system. If your machine still has some pixmap based fonts, the supported sizes will be shown in bold face. Only the first 256 fonts will be listed.

22.1.25 forms

`forms` is an XForms program with very few changes. Search for "fltk" to find all changes necessary to port to fltk. This demo shows the different boxtypes. Note that some boxtypes are not appropriate for some objects.

22.1.26 fractals

`fractals` shows how to mix OpenGL, Glut and FLTK code. FLTK supports a rather large subset of Glut, so that many Glut applications compile just fine.

22.1.27 fullscreen

This demo shows how to do many of the window manipulations that are popular for games. You can toggle the border on/off, switch between single- and double-buffered rendering, and take over the entire screen. More information in the source code.

22.1.28 gl_overlay

`gl_overlay` shows OpenGL overlay plane rendering. If no hardware overlay plane is available, FLTK will simulate it for you.

22.1.29 glpuzzle

The `glpuzzle` test shows how most Glut source code compiles easily under FLTK.

22.1.30 hello

`hello`: Hello, World. Need I say more? Well, maybe. This tiny demo shows how little is needed to get a functioning application running with FLTK. Quite impressive, I'd say.

22.1.31 help

`help` displays the built-in FLTK help browser. The `Fl_Help_Dialog` understands a subset of html and renders various image formats. This widget makes it easy to provide help pages to the user without depending on the operating system's html browser.

22.1.32 iconize

`iconize` demonstrates the effect of the window functions `hide()`, `iconize()`, and `show()`.

22.1.33 image

The `image` demo shows how an image can be created on the fly. This generated image contains an alpha (transparency) channel which lets previous renderings 'shine through', either via true transparency or by using screen door transparency (pixelation).

22.1.34 inactive

`inactive` tests the correct rendering of inactive widgets. To see the inactive version of images, you can check out the `pixmap` or `image` test.

22.1.35 input

This tool shows and tests different types of text input fields based on `FL_Input_`. The `input` program also tests various settings of `FL_Input::when()`.

22.1.36 input_choice

`input_choice` tests the latest addition to FLTK1, a text input field with an attached pulldown menu. Windows users will recognize similarities to the 'ComboBox'. `input_choice` starts up in 'plastic' scheme, but the traditional scheme is also supported.

22.1.37 keyboard

FLTK unifies keyboard events for all platforms. The `keyboard` test can be used to check the return values of `Fl::event_key()` and `Fl::event_text()`. It is also great to see the modifier buttons and the scroll wheel at work. Quit this application by closing the window. The ESC key will not work.

22.1.38 label

Every FLTK widget can have a label attached to it. The `label` demo shows alignment, clipping, and wrapping of text labels. Labels can contain symbols at the start and end of the text, like `@FLTK` or `@circle uh-huh @square`.

22.1.39 line_style

Advanced line drawing can be tested with `line_style`. Not all platforms support all line styles.

22.1.40 list_visuals

This little app finds all available pixel formats for the current X11 screen. But since you are now an FLTK user, you don't have to worry about any of this.

22.1.41 mandelbrot

`mandelbrot` shows two advanced topics in one test. It creates grayscale images on the fly, updating them via the `idle` callback system. This is one of the few occasions where the `idle` callback is very useful by giving all available processor time to the application without blocking the UI or other apps.

22.1.42 menubar

The `menubar` tests many aspects of FLTK's popup menu system. Among the features are radio buttons, menus taller than the screen, arbitrary sub menu depth, and global shortcuts.

22.1.43 message

`message` pops up a few of FLTK's standard message boxes.

22.1.44 minimum

The `minimum` test program verifies that the update regions are set correctly. In a real life application, the trail would be avoided by choosing a smaller label or by setting label clipping differently.

22.1.45 navigation

`navigation` demonstrates how the text cursor moves from text field to text field when using the arrow keys, tab, and shift-tab.

22.1.46 output

`output` shows the difference between the single line and multi line mode of the [FI_Output](#) widget. Fonts can be selected from the FLTK standard list of fonts.

22.1.47 overlay

The `overlay` test app shows how easy an FLTK window can be layered to display cursor and manipulator style elements. This example derives a new class from [FI_Overlay_Window](#) and provides a new function to draw custom overlays.

22.1.48 pack

The `pack` test program demonstrates the resizing and repositioning of children of the [FI_Pack](#) group. Putting an [FI_Pack](#) into an [FI_Scroll](#) is a useful way to create a browser for large sets of data.

22.1.49 pixmap_browser

`pixmap_browser` tests the shared-image interface. When using the same image multiple times, [FI_Shared_Image](#) will keep it only once in memory.

22.1.50 pixmap

This simple test shows the use of a LUT based pixmap as a label for a box widget. Pixmapes are stored in the X11 `!xpm` file format and can be part of the source code. Pixmapes support one transparent color.

22.1.51 preferences

I do have my `preferences` in the morning, but sometimes I just can't remember a thing. This is where the [FI_Preferences](#) come in handy. They remember any kind of data between program launches.

22.1.52 radio

The `radio` tool was created entirely with *fluid*. It shows some of the available button types and tests radio button behavior.

22.1.53 resizebox

`resizebox` shows some possible ways of FLTK's automatic resize behavior.

22.1.54 resize

The `resize` demo tests size and position functions with the given window manager.

22.1.55 scroll

`scroll` shows how to scroll an area of widgets, one of them being a slow custom drawing. [FI_Scroll](#) uses clipping and smart window area copying to improve redraw speed. The buttons at the bottom of the window control decoration rendering and updates.

22.1.56 shape

`shape` is a very minimal demo that shows how to create your own OpenGL rendering widget. Now that you know that, go ahead and write that flight simulator you always dreamt of.

22.1.57 subwindow

The `subwindow` demo tests messaging and drawing between the main window and 'true' sub windows. A sub window is different to a group by resetting the FLTK coordinate system to 0, 0 in the top left corner. On Win32 and X11, subwindows have their own operating system specific handle.

22.1.58 sudoku

Another highly addictive game - don't play it, I warned you. The implementation shows how to create application icons, how to deal with OS specifics, and how to generate sound.

22.1.59 symbols

`symbols` are a speciality of FLTK. These little vector drawings can be integrated into labels. They scale and rotate, and with a little patience, you can define your own. The rotation number refers to 45 degree rotations if you were looking at a numeric keypad (2 is down, 6 is right, etc.).

22.1.60 tabs

The `tabs` tool was created with *fluid*. It tests correct hiding and redisplaying of tabs, navigation across tabs, resize behavior, and no unneeded redrawing of invisible widgets.

The `tabs` application shows the [FI_Tabs](#) widget on the left and the [FI_Wizard](#) widget on the right side for direct comparison of these two panel management widgets.

22.1.61 threads

FLTK can be used in a multithreading environment. There are some limitations, mostly due to the underlying operating system. `threads` shows how to use `Fl::lock()`, `Fl::unlock()`, and `Fl::awake()` in secondary threads to keep FLTK happy. Although locking works on all platforms, this demo is not available on every machine.

22.1.62 tile

The `tile` tool shows a nice way of using `Fl_Tile`. To test correct resizing of subwindows, the widget for region 1 is created from an `Fl_Window` class.

22.1.63 tiled_image

The `tiled_image` demo uses an image as the background for a window by repeating it over the full size of the widget. The window is resizable and shows how the image gets repeated.

22.1.64 unittests

`unittests` exercises all of FLTK's drawing features (e.g., text, lines, circles, images), as well as scrollbars and schemes.

22.1.65 utf8

`utf8` shows all fonts available to the platform that runs it, and how each font draws each of the Unicode code points ranging between U+0020 and U+FFFF.

22.1.66 valuator

`valuator` shows all of FLTK's nifty widgets to change numeric values.

22.1.67 fluid

`fluid` is not only a big test program, but also a very useful visual UI designer. Many parts of `fluid` were created using `fluid`. See the [Fluid Tutorial](#) for more details.

Chapter 23

FAQ (Frequently Asked Questions)

A list of frequently asked questions about FLTK.

This appendix describes various frequently asked questions regarding FLTK.

- [Where do I start learning FLTK?](#)
- [How do I make a box with text?](#)
- [Can I use FLTK to make closed-source commercial applications?](#)
- [Hitting the 'Escape' key closes windows - how do I prevent this?](#)

23.1 Where do I start learning FLTK?

It is assumed you know C++, which is the language all FLTK programs are written in, including FLTK itself.

If you like reading manuals to work your way into things, a good start is the FLTK documentation's [Introduction to FLTK](#). Under the [FLTK Basics](#) section there's an example 'hello world' program that includes a line-by-line description.

If you like looking at simple code first to pique your interest, and then read up from there, start with the example programs in the `test/` and `examples/` directory that is included with the source code. A good place to start is the 'hello world' program in `test/hello.cxx`. Also do a google search for "FLTK example programs". "Erco's Cheat Page" is one that shows many simple examples of how to do specific things.

If you like to run example programs and look for ones that are like yours and then read them, download and build FLTK from the source, then run the `test/demo` program. Also, go into the 'examples/' directory and run 'make', then run some of those programs.

If you prefer watching TV to reading books and code, google search for "FLTK video tutorials" which has some introductory examples of how to write FLTK programs in C++ and build them.

23.2 How do I make a box with text?

The 'hello world' program shows how to make a box with text. All widgets have labels, so picking a simple widget like [Fl_Box](#) and setting its `label()` and using `align()` to align the label and `labelfont()` to set the font, and `labelsize()` to set the size, you can get text just how you want.

Labels are not selectable though; if you want selectable text, you can use [Fl_Output](#) or [Fl_Multiline_Output](#) for simple text that doesn't include scrollbars. For more complex text that might want scrollbars and multiple colors/fonts, use either [Fl_Text_Display](#) which handles plain text, or [Fl_Help_View](#) which handles simple HTML formatted text.

23.3 Can I use FLTK to make closed-source commercial applications?

Yes. The FLTK [Software License](#) is standard LGPL, but also includes a special clause ("exception") to allow for static linking. Specifically:

```
[from the top of the FLTK LGPL License section on exceptions]
```

```
3. Static linking of applications and widgets to the FLTK library does
```

not constitute a derivative work and does not require the author to provide source code for the application or widget, use the shared FLTK libraries, or link their applications or widgets against a user-supplied version of FLTK.

If you link the application or widget to a modified version of FLTK, then the changes to FLTK must be provided under the terms of the LGPL in sections 1, 2, and 4.

4. You do not have to provide a copy of the FLTK license with programs that are linked to the FLTK library, nor do you have to identify the FLTK license in your program or documentation as required by section 6 of the LGPL.

However, programs must still identify their use of FLTK. The following example statement can be included in user documentation to satisfy this requirement:

```
[program/widget] is based in part on the work of the
FLTK project (http://www.fltk.org).
```

23.4 Hitting the 'Escape' key closes windows - how do I prevent this?

[From FLTK article #378]

1. FLTK has a "global event handler" that makes Escape try to close the window, the same as clicking the close box. To disable this everywhere you can install your own that pretends it wants the escape key and thus stops the default one from seeing it (this may not be what you want, see below about the callbacks):

```
static int my_handler(int event) {
    if (event == FL_SHORTCUT) return 1; // eat all shortcut keys
    return 0;
}
...in main():
    Fl::add_handler(my_handler);
...
```

1. Attempts to close a window (both clicking the close box or typing Escape) call that window's callback. The default version of the callback does hide(). To make the window not close or otherwise do something different you replace the callback. To make the main window exit the program:

```
void my_callback(Fl_Widget*, void*) {
    exit(0);
}
...
main_window->callback(my_callback);
...
```

If you don't want Escape to close the main window and exit you can check for and ignore it. This is better than replacing the global handler because Escape will still close pop-up windows:

```
void my_callback(Fl_Widget*, void*) {
    if (Fl::event() == FL_SHORTCUT && Fl::event_key() == FL_Escape)
        return; // ignore Escape
    exit(0);
}
```

It is very common to ask for confirmation before exiting, this can be done with:

```
void my_callback(Fl_Widget*, void*) {
    if (fl_ask("Are you sure you want to quit?"))
        exit(0);
}
```

Chapter 24

Todo List

Page [Adding and Extending Widgets](#)

Clarify [FI_Window::damage\(uchar\)](#) handling - seems confused/wrong? ORing value doesn't match setting behaviour in [FL_Widget.H!](#)

Clarify [FI_Widget::test_shortcut\(\)](#) explanations. [FI_Widget.h](#) says Internal Use only, but subclassing chapter gives details!

Page [Drawing Things in FLTK](#)

add an [FI_Draw_Area_Cb](#) typedef to allow [fl_scroll\(...\)](#) to be doxygenated?

Member [FI_Browser::scrollbar_width](#) (int width)

This method should eventually be removed in 1.4+

Member [FI_Browser::scrollbar_width](#) () const

This method should eventually be removed in 1.4+

Member [FI_Browser::sort](#) (int flags=0)

Add a flag to ignore case

Class [FI_Button](#)

Refactor the doxygen comments for [FI_Button when\(\)](#) documentation.

Refactor the doxygen comments for [FI_Button type\(\)](#) documentation.

Class [FI_Chart](#)

Refactor [FI_Chart::type\(\)](#) information.

Class [FI_Choice](#)

Refactor the doxygen comments for [FI_Choice changed\(\)](#) documentation.

Class [FI_Counter](#)

Refactor the doxygen comments for [FI_Counter type\(\)](#) documentation.

Member [FI_Cursor](#)

enum [FI_Cursor](#) needs maybe an image.

Member [FI_File_Input::errorcolor](#) () const

Better docs for [FI_File_Input::errorcolor\(\)](#) - is it even used?

Member [FI_Group::sizes](#) ()

Should the internal representation of the [sizes\(\)](#) array be documented?

Member [fl_height](#) (int font, int size)

In the future, when the XFT issues are resolved, this function should simply return the 'size' value.

Member [FI_Input::handle_mouse](#) (int, int, int, int, int keepmark=0)

Add comment and parameters

Member [FI_Input::handletext](#) (int e, int, int, int, int)

Add comment and parameters

Member `fl_intptr_t`

typedef's `fl_intptr_t` and `fl_uintptr_t` should be documented.

Class `FI_Label`

There is an aspiration that the `FI_Label` type will become a widget by itself. That way we will be avoiding a lot of code duplication by handling labels in a similar fashion to widgets containing text. We also provide an easy interface for very complex labels, containing html or vector graphics. However, this re-factoring is not in place in this release.

Member `FI_Labeltype`

The doxygen comments are incomplete, and some labeltypes start with an underscore. Also, there are three external functions undocumented (yet):

- `fl_define_FL_SHADOW_LABEL()`
- `fl_define_FL_ENGRAVED_LABEL()`
- `fl_define_FL_EMBOSSSED_LABEL()`

Member `FI_Menu::add (const char *, int shortcut, FI_Callback *, void **=0, int=0)`

Raw integer shortcut needs examples. Dependent on responses to <http://fltk.org/newsgroups.php?gfltk.development+v:10086> and results of STR#2344

Member `fl_old_shortcut (const char *s)`

Fix these silly legacy issues in a future release to support more predictable behavior for the modifier keys.

Member `FI_Preferences::get (const char *entry, void *value, const void *defaultValue, int defaultSize, int maxSize)`

`maxSize` should receive the number of bytes that were read.

Member `fl_reset_spot (void)`

provide user documentation for `fl_reset_spot` function

Member `FI_Scroll::bbox (int &, int &, int &, int &)`

The visibility of the scrollbars ought to be checked/calculated outside of the `draw()` method (STR #1895).

Member `fl_set_spot (int font, int size, int X, int Y, int W, int H, FI_Window *win=0)`

provide user documentation for `fl_set_spot` function

Member `fl_set_status (int X, int Y, int W, int H)`

provide user documentation for `fl_set_status` function

Member `FI_String`

FIXME: temporary (?) typedef to mark UTF-8 and Unicode conversions

Member `FI_Text_Display::display_insert ()`

Unicode?

Member `FI_Text_Display::extend_range_for_styles (int *start, int *end)`

Unicode?

Member `FI_Text_Display::handle_vline (int mode, int lineStart, int lineLen, int leftChar, int rightChar, int topClip, int bottomClip, int leftClip, int rightClip) const`

we need to handle hidden hyphens and tabs here!

we handle all styles and selections

we must provide code to get pixel positions of the middle of a character as well

Member `FI_Text_Display::overstrike (const char *text)`

Unicode? Find out exactly what we do here and simplify.

Member `FI_Text_Display::position_to_line (int pos, int *lineNum) const`

What does this do?

Member `FI_Text_Display::position_to_linecol (int pos, int *lineNum, int *column) const`

a column number makes little sense in the UTF-8/variable font width environment. We will have to further define what exactly we want to return. Please check the functions that call this particular function.

Member `Fl_Text_Display::scroll` (int topLineNum, int horizOffset)

Column numbers make little sense here.

Member `Fl_Text_Display::shortcut` (int s)

FIXME : get set methods pointing on `shortcut_` have no effects as `shortcut_` is unused in this class and derived!

Member `Fl_Text_Display::shortcut` () const

FIXME : get set methods pointing on `shortcut_` have no effects as `shortcut_` is unused in this class and derived!

Member `Fl_Text_Display::wrap_mode` (int wrap, int wrap_margin)

we need new wrap modes to wrap at the window edge and based on pixel width or average character width.

Member `Fl_Text_Display::wrapped_column` (int row, int column) const

What does this do and how is it useful? Column numbers mean little in this context. Which functions depend on this one?

Unicode?

Member `Fl_Text_Display::wrapped_row` (int row) const

What does this do and how is it useful? Column numbers mean little in this context. Which functions depend on this one?

Member `Fl_Tiled_Image::Fl_Tiled_Image` (Fl_Image *i, int W=0, int H=0)

Fix `Fl_Tiled_Image` as background image for widgets and windows and fix the implementation of `Fl::scheme(const char *)`.

Member `Fl_Tree::handle` (int e)

add `Fl_Widget_Tracker` (see `Fl_Browser_cxx::handle()`)

Member `Fl_Tree::is_scrollbar` (Fl_Widget *w)

should be const

Member `Fl_Tree::show_self` ()

should be const

Member `Fl_When`

doxygen comments for values are incomplete and maybe wrong or unclear

Member `Fl_Widget::argument` () const

The user data value must be implemented using `intptr_t` or similar to avoid 64-bit machine incompatibilities.

Member `Fl_Widget::argument` (long v)

The user data value must be implemented using `intptr_t` or similar to avoid 64-bit machine incompatibilities.

Member `Fl_Widget::type` () const

Explain "simulate RTTI" (currently only used to decide if a widget is a window, i.e. `type() >= FL_WINDOW` ?). Is `type()` really used in a way that ensures "Forms compatibility" ?

Member `Fl_Window::show` (int argc, char **argv)

explain which system parameters are set up.

Member `Fl_Window::show` ()

Check if we can remove resetting the current group in a later FLTK version (after 1.3.x). This may break "already broken" programs though if they rely on this "feature".

Page Handling Events

Add details on how to detect repeating keys, since on some X servers a repeating key will generate both `FL_↔` `KEYUP` and `FL_KEYDOWN`, such that to tell if a key is held, you need `Fl::event_key(int)` to detect if the key is being held down during `FL_KEYUP` or not.

Page Unicode and UTF-8 Support

Verify 16/24 bit Unicode limit for different character sets? OksiD's code appears limited to 16-bit whereas the FLTK2 code appears to handle a wider set. What about illegal characters? See comments in `fl_utf8fromwc()` and `fl_utf8toUtf16()`.

Work through the code and this documentation to harmonize the `[OksiD]` and `[fltk2]` functions.

Do we need this info about planes?

Chapter 25

Deprecated List

Member `Fl::release ()`

Use `Fl::grab(0)` instead.

Member `Fl::set_idle (Fl_Old_Idle_Handler cb)`

This method is obsolete - use the `add_idle()` method instead.

Member `Fl::version ()`

Use `int Fl::api_version()` instead.

Member `fl_ask (const char *fmt,...)`

`fl_ask()` is deprecated since it uses "Yes" and "No" for the buttons which does not conform to the current FLTK Human Interface Guidelines. Use `fl_choice()` with the appropriate verbs instead.

Member `fl_clip`

`fl_clip(int, int, int, int)` is deprecated and will be removed from future releases. Please use `fl_push_clip(int x, int y, int w, int h)` instead.

Member `Fl_Group::focus (Fl_Widget *W)`

This is for backwards compatibility only. You should use `W->take_focus()` instead.

Member `Fl_Menu_Item::check ()`

.

Member `Fl_Menu_Item::checked () const`

.

Member `Fl_Menu_Item::setonly ()`

This method is dangerous if radio items are first in the menu. Use `Fl_Menu_::setonly(Fl_Menu_Item*)` instead.

Member `Fl_Menu_Item::uncheck ()`

.

Member `Fl_Spinner::maximum () const`

Member `Fl_Spinner::minimum () const`

Member `Fl_Tree::first_visible ()`

in 1.3.3 ABI – use `first_visible_item()` instead.

Member `Fl_Tree::item_clicked (Fl_Tree_Item *val)`

in 1.3.3 ABI – use `callback_item()` instead.

Member `Fl_Tree::item_clicked ()`

in 1.3.3 ABI – use `callback_item()` instead.

Member `Fl_Tree::last_visible ()`

in 1.3.3 – use `last_visible_item()` instead.

Member `FI_Tree_Item::FI_Tree_Item (const FI_Tree_Prefs &prefs)`

in 1.3.3 ABI – you must use `FI_Tree_Item(FI_Tree*)` for proper horizontal scrollbar behavior.

Member `FI_Tree_Item::next_displayed (FI_Tree_Prefs &prefs)`

in 1.3.3 for confusing name, use `next_visible()` instead

Member `FI_Tree_Item::prev_displayed (FI_Tree_Prefs &prefs)`

in 1.3.3 for confusing name, use `prev_visible()`

Member `FL_VERSION`

This `double` version number is retained for compatibility with existing program code. New code should use `int FL_API_VERSION` instead. `FL_VERSION` is deprecated because comparisons of floating point values may fail due to rounding errors. However, there are currently no plans to remove this deprecated constant.

Member `FI_Widget::color2 (unsigned a)`

Use `selection_color(unsigned)` instead.

Member `FI_Widget::color2 () const`

Use `selection_color()` instead.

Member `FI_Window::free_position ()`

please use `force_position(0)` instead

Member `FI_Window::icon () const`

in 1.3.3

Member `FI_Window::icon (const void *ic)`

in 1.3.3

Chapter 26

Module Index

26.1 Modules

Here is a list of all modules:

Callback function typedefs	213
Windows handling functions	214
Events handling functions	216
Selection & Clipboard functions	230
Screen functions	233
Color & Font functions	238
Drawing functions	248
Multithreading support functions	272
Safe widget deletion support functions	274
Cairo Support Functions and Classes	277
Unicode and UTF-8 functions	278
Mac OS X-specific symbols	292
Common Dialogs classes and functions	294
File names and URI utility functions	306

Chapter 27

Hierarchical Index

27.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

FI_Preferences::Entry	315
FI	315
FI_Cairo_State	386
FL_CHART_ENTRY	392
FI_Device	417
FI_Graphics_Driver	470
FI_GDI_Graphics_Driver	451
FI_GDI_Printer_Graphics_Driver	456
FI_PostScript_Graphics_Driver	634
FI_Quartz_Graphics_Driver	673
FI_Xlib_Graphics_Driver	977
FI_Surface_Device	720
FI_Copy_Surface	411
FI_Display_Device	422
FI_Image_Surface	528
FI_Paged_Device	612
FI_PostScript_File_Device	629
FI_PostScript_Printer	649
FI_Printer	664
FI_System_Printer	727
FI_End	425
FI_File_Chooser	428
FI_File_Icon	434
FI_FLTK_File_Chooser	444
FI_GTK_File_Chooser	510
FI_Font_Descriptor	445
FI_Fontdesc	445
FI_GI_Choice	459
FI_Glut_Bitmap_Font	467
FI_Glut_StrokeChar	468
FI_Glut_StrokeFont	468
FI_Glut_StrokeStrip	468
FI_Glut_StrokeVertex	468
FI_Help_Block	510
FI_Help_Dialog	510
FI_Help_Font_Stack	512
FI_Help_Font_Style	513
FI_Help_Link	513
FI_Help_Target	514

FI_Image	523
FI_Bitmap	339
FI_XBM_Image	976
FI_Pixmap	618
FI_GIF_Image	458
FI_XPM_Image	983
FI_RGB_Image	685
FI_BMP_Image	341
FI_JPEG_Image	557
FI_PNG_Image	624
FI_PNM_Image	626
FI_Shared_Image	704
FI_Tiled_Image	824
FI_Label	558
FI_Mac_App_Menu	562
FI_Menu_Item	582
FI_Multi_Label	596
FI_Native_File_Chooser	599
FI_Plugin	622
FI_Device_Plugin	418
FI_Preferences	650
FI_Plugin_Manager	623
FI_Scroll::FI_Region_LRTB	681
FI_Scroll::FI_Region_XYWH	681
FI_Scroll::FI_Scrollbar_Data	701
FI_Text_Buffer	757
FI_Text_Selection	817
FI_Tooltip	829
FI_Tree_Item	876
FI_Tree_Item_Array	897
FI_Tree_Prefs	900
FI_Widget	920
FI_Box	342
FI_Button	381
FI_Light_Button	560
FI_Check_Button	395
FI_Radio_Light_Button	680
FI_Round_Button	691
FI_Radio_Round_Button	680
FI_Radio_Button	679
FI_Repeat_Button	682
FI_Return_Button	683
FI_Toggle_Button	828
FI_Chart	388
FI_Clock_Output	402
FI_Clock	400
FI_Round_Clock	692
FI_FormsBitmap	445
FI_FormsPixmap	446
FI_FormsText	448
FI_Free	449
FI_Group	500
FI_Browser_	365
FI_Browser	344
FI_File_Browser	426
FI_Hold_Browser	520

FI_Multi_Browser	595
FI_Select_Browser	703
FI_Check_Browser	393
FI_Color_Chooser	406
FI_Help_View	514
FI_Input_Choice	553
FI_Pack	610
FI_Scroll	692
FI_Spinner	716
FI_Table	731
FI_Table_Row	748
FI_Tabs	751
FI_Text_Display	773
FI_Text_Editor	809
FI_Tile	821
FI_Tree	834
FI_Window	957
FI_Double_Window	423
FI_Cairo_Window	387
FI_Overlay_Window	607
FI_Gl_Window	459
FI_Glut_Window	469
FI_Single_Window	711
FI_Menu_Window	593
FI_Wizard	975
FI_Input_	534
FI_Input	531
FI_File_Input	439
FI_Float_Input	443
FI_Int_Input	556
FI_Multiline_Input	597
FI_Output	606
FI_Multiline_Output	598
FI_Secret_Input	702
FI_Menu_	563
FI_Choice	396
FI_Menu_Bar	577
FI_Sys_Menu_Bar	722
FI_Menu_Button	579
FI_Positioner	626
FI_Progress	671
FI_Timer	826
FI_Valuator	905
FI_Adjuster	337
FI_Counter	413
FI_Simple_Counter	711
FI_Dial	419
FI_Fill_Dial	442
FI_Line_Dial	562
FI_Roller	689
FI_Slider	713
FI_Fill_Slider	442
FI_Hor_Fill_Slider	521
FI_Hor_Nice_Slider	522
FI_Hor_Slider	522
FI_Nice_Slider	605
FI_Scrollbar	698

FI_Value_Slider	917
FI_Hor_Value_Slider	523
FI_Value_Input	910
FI_Value_Output	914
FI_Widget_Tracker	956
FI_XColor	977
FI_Text_Editor::Key_Binding	983
FI_Graphics_Driver::matrix	984
FI_Preferences::Name	984
FI_Preferences::Node	985
FI_Paged_Device::page_format	986
FI_Preferences::RootNode	986
FI_Scroll::ScrollInfo	986
FI_Window::shape_data_type	987
FI_Text_Display::Style_Table_Entry	988

Chapter 28

Class Index

28.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

FI_Preferences::Entry	315
FI	
The FI is the FLTK global (static) class containing state information and global methods for the current application	315
FI_Adjuster	
Was stolen from Prisms, and has proven to be very useful for values that need a large dynamic range	337
FI_Bitmap	
Supports caching and drawing of mono-color (bitmap) images	339
FI_BMP_Image	
Supports loading, caching, and drawing of Windows Bitmap (BMP) image files	341
FI_Box	
This widget simply draws its box, and possibly its label	342
FI_Browser	
Displays a scrolling list of text lines, and manages all the storage for the text	344
FI_Browser_	
This is the base class for browsers	365
FI_Button	
Buttons generate callbacks when they are clicked by the user	381
FI_Cairo_State	
Contains all the necessary info on the current cairo context	386
FI_Cairo_Window	
This defines a pre-configured cairo fltk window	387
FI_Chart	
FI_Chart displays simple charts	388
FL_CHART_ENTRY	
For internal use only	392
FI_Check_Browser	
Displays a scrolling list of text lines that may be selected and/or checked by the user	393
FI_Check_Button	
A button with a "checkmark" to show its status	395
FI_Choice	
A button that is used to pop up a menu	396
FI_Clock	
This widget provides a round analog clock display	400
FI_Clock_Output	
This widget can be used to display a program-supplied time	402
FI_Color_Chooser	
Standard RGB color chooser	406

FI_Copy_Surface	Supports copying of graphical data to the clipboard	411
FI_Counter	Controls a single floating point value with button (or keyboard) arrows	413
FI_Device	All graphical output devices and all graphics systems	417
FI_Device_Plugin	This plugin socket allows the integration of new device drivers for special window or screen types	418
FI_Dial	Circular dial to control a single floating point value	419
FI_Display_Device	A display to which the computer can draw	422
FI_Double_Window	The FI_Double_Window provides a double-buffered window	423
FI_End	This is a dummy class that allows you to end a FI_Group in a constructor list of a class:	425
FI_File_Browser	Displays a list of filenames, optionally with file-specific icons	426
FI_File_Chooser	Displays a standard file selection dialog that supports various selection modes	428
FI_File_Icon	Manages icon images that can be used as labels in other widgets and as icons in the FileBrowser widget	434
FI_File_Input	This widget displays a pathname in a text input field	439
FI_Fill_Dial	Draws a dial with a filled arc	442
FI_Fill_Slider	Widget that draws a filled horizontal slider, useful as a progress or value meter	442
FI_Float_Input	Subclass of FI_Input that only allows the user to type floating point numbers (sign, digits, decimal point, more digits, 'E' or 'e', sign, digits)	443
FI_FLTK_File_Chooser	444
FI_Font_Descriptor	This a structure for an actual system font, with junk to help choose it and info on character sizes	445
FI_Fontdesc	445
FI_FormsBitmap	Forms compatibility Bitmap Image Widget	445
FI_FormsPixmap	Forms pixmap drawing routines	446
FI_FormsText	448
FI_Free	Emulation of the Forms "free" widget	449
FI_GDI_Graphics_Driver	The MSWindows-specific graphics class	451
FI_GDI_Printer_Graphics_Driver	The graphics driver used when printing on MSWindows	456
FI_GIF_Image	Supports loading, caching, and drawing of Compuserve GIF SM images	458
FI_GI_Choice	459
FI_GI_Window	Sets things up so OpenGL works	459
FI_Glut_Bitmap_Font	Fltk glut font/size attributes used in the glutXXX functions	467
FI_Glut_StrokeChar	468
FI_Glut_StrokeFont	468
FI_Glut_StrokeStrip	468
FI_Glut_StrokeVertex	468

FI_Glut_Window	GLUT is emulated using this window class and these static variables (plus several more static variables hidden in <code>glut_compatibility.cxx</code>):	469
FI_Graphics_Driver	A virtual class subclassed for each graphics driver FLTK uses	470
FI_Group	FLTK container widget	500
FI_GTK_File_Chooser		510
FI_Help_Block		510
FI_Help_Dialog	Displays a standard help dialog window using the FI_Help_View widget	510
FI_Help_Font_Stack		512
FI_Help_Font_Style	FI_Help_View font stack element definition	513
FI_Help_Link	Definition of a link for the html viewer	513
FI_Help_Target	FI_Help_Target structure	514
FI_Help_View	Displays HTML text	514
FI_Hold_Browser	The FI_Hold_Browser is a subclass of FI_Browser which lets the user select a single item, or no items by clicking on the empty space	520
FI_Hor_Fill_Slider		521
FI_Hor_Nice_Slider		522
FI_Hor_Slider	Horizontal Slider class	522
FI_Hor_Value_Slider		523
FI_Image	Base class for image caching and drawing	523
FI_Image_Surface	Directs all graphics requests to an FI_Image	528
FI_Input	This is the FLTK text input widget	531
FI_Input_	This class provides a low-overhead text input field	534
FI_Input_Choice	A combination of the input widget and a menu button	553
FI_Int_Input	Subclass of FI_Input that only allows the user to type decimal digits (or hex numbers of the form 0xaeF)	556
FI_JPEG_Image	Supports loading, caching, and drawing of Joint Photographic Experts Group (JPEG) File Interchange Format (JFIF) images	557
FI_Label	This struct stores all information for a text or mixed graphics label	558
FI_Light_Button	This subclass displays the "on" state by turning on a light, rather than drawing pushed in	560
FI_Line_Dial		562
FI_Mac_App_Menu	Mac OS-specific class allowing to customize and localize the application menu	562
FI_Menu_	Base class of all widgets that have a menu in FLTK	563
FI_Menu_Bar	This widget provides a standard menubar interface	577
FI_Menu_Button	This is a button that when pushed pops up a menu (or hierarchy of menus) defined by an array of FI_Menu_Item objects	579

FI_Menu_Item	The FI_Menu_Item structure defines a single menu item that is used by the FI_Menu_ class . . .	582
FI_Menu_Window	Window type used for menus	593
FI_Multi_Browser	Subclass of FI_Browser which lets the user select any set of the lines	595
FI_Multi_Label	This struct allows multiple labels to be added to objects that might normally have only one label	596
FI_Multiline_Input	This input field displays '\n' characters as new lines rather than ^J, and accepts the Return, Tab, and up and down arrow keys	597
FI_Multiline_Output	This widget is a subclass of FI_Output that displays multiple lines of text	598
FI_Native_File_Chooser	This class lets an FLTK application easily and consistently access the operating system's native file chooser	599
FI_Nice_Slider		605
FI_Output	This widget displays a piece of text	606
FI_Overlay_Window	This window provides double buffering and also the ability to draw the "overlay" which is another picture placed on top of the main image	607
FI_Pack	This widget was designed to add the functionality of compressing and aligning widgets	610
FI_Paged_Device	Represents page-structured drawing surfaces	612
FI_Pixmap	Supports caching and drawing of colormap (pixmap) images, including transparency	618
FI_Plugin	FI_Plugin allows link-time and run-time integration of binary modules	622
FI_Plugin_Manager	FI_Plugin_Manager manages link-time and run-time plugin binaries	623
FI_PNG_Image	Supports loading, caching, and drawing of Portable Network Graphics (PNG) image files	624
FI_PNM_Image	Supports loading, caching, and drawing of Portable Anymap (PNM, PBM, PGM, PPM) image files	626
FI_Positioner	This class is provided for Forms compatibility	626
FI_PostScript_File_Device	To send graphical output to a PostScript file	629
FI_PostScript_Graphics_Driver	PostScript graphical backend	634
FI_PostScript_Printer	Print support under Unix/Linux	649
FI_Preferences	FI_Preferences provides methods to store user settings between application starts	650
FI_Printer	OS-independent print support	664
FI_Progress	Displays a progress bar for the user	671
FI_Quartz_Graphics_Driver	The Mac OS X-specific graphics class	673
FI_Radio_Button		679
FI_Radio_Light_Button		680
FI_Radio_Round_Button		680
FI_Scroll::FI_Region_LRTB	A local struct to manage a region defined by left/right/top/bottom	681

FI_Scroll::FI_Region_XYWH	A local struct to manage a region defined by xywh	681
FI_Repeat_Button	The FI_Repeat_Button is a subclass of FI_Button that generates a callback when it is pressed and then repeatedly generates callbacks as long as it is held down	682
FI_Return_Button	The FI_Return_Button is a subclass of FI_Button that generates a callback when it is pressed or when the user presses the Enter key	683
FI_RGB_Image	Supports caching and drawing of full-color images with 1 to 4 channels of color information	685
FI_Roller	"dolly" control commonly used to move 3D objects	689
FI_Round_Button	Buttons generate callbacks when they are clicked by the user	691
FI_Round_Clock	A clock widget of type FL_ROUND_CLOCK	692
FI_Scroll	This container widget lets you maneuver around a set of widgets much larger than your window	692
FI_Scrollbar	Displays a slider with arrow buttons at the ends of the scrollbar	698
FI_Scroll::FI_Scrollbar_Data	A local struct to manage a scrollbar's xywh region and tab values	701
FI_Secret_Input	Subclass of FI_Input that displays its input as a string of placeholders	702
FI_Select_Browser	The class is a subclass of FI_Browser which lets the user select a single item, or no items by clicking on the empty space	703
FI_Shared_Image	This class supports caching, loading, scaling, and drawing of image files	704
FI_Simple_Counter	This widget creates a counter with only 2 arrow buttons	711
FI_Single_Window	This is the same as FI_Window	711
FI_Slider	Sliding knob inside a box	713
FI_Spinner	This widget is a combination of the input widget and repeat buttons	716
FI_Surface_Device	A drawing surface that's susceptible to receive graphical output	720
FI_Sys_Menu_Bar	A class to create, modify and delete menus that appear on Mac OS X in the menu bar at the top of the screen	722
FI_System_Printer	Print support under MSWindows and Mac OS	727
FI_Table	A table of widgets or other content	731
FI_Table_Row	A table with row selection capabilities	748
FI_Tabs	"file card tabs" interface that allows you to put lots and lots of buttons and switches in a panel, as popularized by many toolkits	751
FI_Text_Buffer	This class manages Unicode text displayed in one or more FI_Text_Display widgets	757
FI_Text_Display	Rich text display widget	773
FI_Text_Editor	This is the FLTK text editor widget	809

FI_Text_Selection	This is an internal class for FI_Text_Buffer to manage text selections	817
FI_Tile	Lets you resize its children by dragging the border between them	821
FI_Tiled_Image	This class supports tiling of images over a specified area	824
FI_Timer	This is provided only to emulate the Forms Timer widget	826
FI_Toggle_Button	The toggle button is a push button that needs to be clicked once to toggle on, and one more time to toggle off	828
FI_Tooltip	Tooltip support for all FLTK widgets	829
FI_Tree	Tree widget	834
FI_Tree_Item	Tree widget item	876
FI_Tree_Item_Array	Manages an array of FI_Tree_Item pointers	897
FI_Tree_Prefs	Tree widget's preferences	900
FI_Valuator	Controls a single floating-point value and provides a consistent interface to set the value, range, and step, and insures that callbacks are done the same for every object	905
FI_Value_Input	Displays a numeric value	910
FI_Value_Output	Displays a floating point value	914
FI_Value_Slider	FI_Slider widget with a box displaying the current value	917
FI_Widget	FI_Widget is the base class for all widgets in FLTK	920
FI_Widget_Tracker	This class should be used to control safe widget deletion	956
FI_Window	This widget produces an actual window	957
FI_Wizard	This widget is based off the FI_Tabs widget, but instead of displaying tabs it only changes "tabs" under program control	975
FI_XBM_Image	Supports loading, caching, and drawing of X Bitmap (XBM) bitmap files	976
FI_XColor	977
FI_Xlib_Graphics_Driver	The Xlib-specific graphics class	977
FI_XPM_Image	Supports loading, caching, and drawing of X Pixmap (XPM) images, including transparency	983
FI_Text_Editor::Key_Binding	Simple linked list item associating a key/state to a function	983
FI_Graphics_Driver::matrix	A 2D coordinate transformation matrix	984
FI_Preferences::Name	'Name' provides a simple method to create numerical or more complex procedural names for entries and groups on the fly	984
FI_Preferences::Node	985
FI_Paged_Device::page_format	Width, height and name of a page format	986
FI_Preferences::RootNode	986

FI_Scroll::ScrollInfo	
Structure to manage scrollbar and widget interior sizes	986
FI_Window::shape_data_type	
Data supporting a non-rectangular window shape	987
FI_Text_Display::Style_Table_Entry	
This structure associates the color, font, and font size of a string to draw with an attribute mask matching attr	988

Chapter 29

File Index

29.1 File List

Here is a list of all documented files with brief descriptions:

abi-version.h	989
dirent.h	989
Enumerations.H	
This file contains type definitions and general enumerations	989
filename.H	
File names and URI utility functions	1016
FI.H	
FI static class	1019
FI_Adjuster.H	1026
fl_ask.H	
API for common dialogs	1026
FI_Bitmap.H	1029
FI_BMP_Image.H	1030
FI_Box.H	1030
FI_Browser.H	1031
FI_Browser_.H	1033
FI_Button.H	1035
FI_Cairo.H	1036
FI_Cairo_Window.H	1037
FI_Chart.H	1037
FI_Check_Browser.H	1038
FI_Check_Button.H	1040
FI_Choice.H	1040
FI_Clock.H	1041
FI_Color_Chooser.H	
FI_Color_Chooser widget	1042
FI_Copy_Surface.H	1043
FI_Counter.H	1045
FI_Device.H	
Declaration of classes FI_Device , FI_Graphics_Driver , FI_Surface_Device , FI_Display_Device , FI_Device_Plugin	1046
FI_Dial.H	1052
FI_Double_Window.H	1053
fl_draw.H	
Utility header to pull drawing functions together	1053
FI_Export.H	1062
FI_File_Browser.H	1062
FI_File_Chooser.H	1063
FI_File_Icon.H	1065
FI_File_Input.H	1067

FI_Fill_Dial.H	1068
FI_Fill_Slider.H	1068
FI_Float_Input.H	1069
FI_FormsBitmap.H	1069
FI_FormsPixmap.H	1070
FI_Free.H	1070
FI_GIF_Image.H	1071
FI_Gl_Window.H	1071
FI_Group.H	1073
FI_Help_Dialog.H	1074
FI_Help_View.H	1075
FI_Hold_Browser.H	1078
FI_Hor_Fill_Slider.H	1079
FI_Hor_Nice_Slider.H	1079
FI_Hor_Slider.H	1080
FI_Hor_Value_Slider.H	1080
FI_Image.H	
FI_Image, FI_RGB_Image classes	1081
FI_Image_Surface.H	1083
FI_Input.H	1084
FI_Input_H	1085
FI_Input_Choice.H	1088
FI_Int_Input.H	1090
FI_JPEG_Image.H	1090
FI_Light_Button.H	1091
FI_Line_Dial.H	1091
FI_Menu.H	1092
FI_Menu_H	1092
FI_Menu_Bar.H	1093
FI_Menu_Button.H	1094
FI_Menu_Item.H	1094
FI_Menu_Window.H	1098
fl_message.H	1098
FI_Multi_Browser.H	1099
FI_Multi_Label.H	1099
FI_Multiline_Input.H	1100
FI_Multiline_Output.H	1100
FI_Native_File_Chooser.H	
FI_Native_File_Chooser widget	1101
FI_Nice_Slider.H	1104
FI_Object.H	1105
FI_Output.H	1105
FI_Overlay_Window.H	1106
FI_Pack.H	1106
FI_Paged_Device.H	
Declaration of class FI_Paged_Device	1107
FI_Pixmap.H	1109
FI_Plugin.H	1110
FI_PNG_Image.H	1111
FI_PNM_Image.H	1111
FI_Positioner.H	1112
FI_PostScript.H	
Declaration of classes FI_PostScript_Graphics_Driver, FI_PostScript_File_Device	1112
FI_Preferences.H	1116
FI_Printer.H	
Declaration of classes FI_Printer, FI_System_Printer and FI_PostScript_Printer	1118
FI_Progress.H	1120
FI_Radio_Button.H	1121

FI_Radio_Light_Button.H	1122
FI_Radio_Round_Button.H	1122
FI_Repeat_Button.H	1123
FI_Return_Button.H	1123
FI_RGB_Image.H	1124
FI_Roller.H	1124
FI_Round_Button.H	1124
FI_Round_Clock.H	1125
FI_Scroll.H	1125
FI_Scrollbar.H	1127
FI_Secret_Input.H	1127
FI_Select_Browser.H	1128
FI_Shared_Image.H	
FI_Shared_Image class	1128
fl_show_colormap.H	
The fl_show_colormap() function hides the implementation classes used to provide the popup window and color selection mechanism	1130
fl_show_input.H	1131
FI_Simple_Counter.H	1131
FI_Single_Window.H	1132
FI_Slider.H	1132
FI_Spinner.H	1133
FI_Sys_Menu_Bar.H	1136
FI_Table.H	1137
FI_Table_Row.H	1143
FI_Tabs.H	1145
FI_Text_Buffer.H	1146
FI_Text_Display.H	1149
FI_Text_Editor.H	1153
FI_Tile.H	1155
FI_Tiled_Image.H	1155
FI_Timer.H	1156
FI_Toggle_Button.H	1157
FI_Toggle_Light_Button.H	1157
FI_Toggle_Round_Button.H	1158
FI_Tooltip.H	1158
FI_Tree.H	
This file contains the definitions of the FI_Tree class	1159
FI_Tree_Item.H	
This file contains the definitions for FI_Tree_Item	1164
FI_Tree_Item_Array.H	
This file defines a class that manages an array of FI_Tree_Item pointers	1168
FI_Tree_Prefs.H	
This file contains the definitions for FI_Tree 's preferences	1170
fl_types.h	
This file contains simple "C"-style type definitions	1176
fl_utf8.h	
Header for Unicode and UTF-8 character handling	1177
FI_Valuator.H	1182
FI_Value_Input.H	1183
FI_Value_Output.H	1183
FI_Value_Slider.H	1184
FI_Widget.H	
FI_Widget , FI_Label classes	1185
FI_Window.H	
FI_Window widget	1190
FI_Wizard.H	1193
FI_XBM_Image.H	1194

Fl_XPM_Image.H	1195
forms.H	1195
gl.h	
This file defines wrapper functions for OpenGL in FLTK	1205
gl2opengl.h	1209
gl_draw.H	1209
glu.h	1210
glut.H	1211
mac.H	
Mac OS X-specific symbols	1216
math.h	1220
names.h	1221
platform.H	1222
win32.H	1222
x.H	1224
cgdebug.h	1226
fastarrow.h	1229
fl_arc.cxx	
Utility functions for drawing arcs and circles	1229
fl_arci.cxx	
Utility functions for drawing circles using integers	1229
fl_ask.cxx	
Utility Functions for Common Dialogs	1229
fl_boxtype.cxx	
Drawing code for common box types	1231
fl_cmap.h	1232
fl_color.cxx	
Color handling	1236
Fl_compose.cxx	
Utility functions to support text input	1237
fl_curve.cxx	
Utility for drawing Bezier curves, adding the points to the current fl_begin/fl_vertex/fl_end path	1237
Fl_Double_Window.cxx	
Fl_Double_Window implementation	1237
Fl_Font.H	1238
Fl_Gl_Choice.H	1239
fl_line_style.cxx	
Line style drawing utility hiding different platforms	1240
Fl_Paged_Device.cxx	
Implementation of class Fl_Paged_Device	1241
fl_rect.cxx	
Drawing and clipping routines for rectangles	1241
fl_vertex.cxx	
Portable drawing code for drawing arbitrary shapes with simple 2D transformations	1241
Fl_XColor.H	1242
flstring.h	1242
freeglut_teapot_data.h	1243
mediumarrow.h	1245
print_panel.h	1245
slowarrow.h	1246
Xutf8.h	1246
case.h	1248
dingbats_.h	1268
spacing.h	1275
symbol_.h	1298
armscii_8.h	1311
ascii.h	1312
big5.h	1312

big5_emacs.h	1360
cp1133.h	1362
cp1251.h	1363
cp1255.h	1364
cp1256.h	1366
cp936ext.h	1368
gb2312.h	1439
georgian_academy.h	1469
georgian_ps.h	1470
iso8859_1.h	1471
iso8859_10.h	1472
iso8859_11.h	1473
iso8859_13.h	1474
iso8859_14.h	1475
iso8859_15.h	1476
iso8859_16.h	1477
iso8859_2.h	1478
iso8859_3.h	1479
iso8859_4.h	1481
iso8859_5.h	1482
iso8859_6.h	1483
iso8859_7.h	1484
iso8859_8.h	1485
iso8859_9.h	1486
iso8859_9e.h	1487
jisx0201.h	1488
jisx0208.h	1489
jisx0212.h	1516
koi8_c.h	1541
koi8_r.h	1543
koi8_u.h	1544
ksc5601.h	1546
mulelao.h	1581
tatar_cyr.h	1582
tcvn.h	1583
tis620.h	1585
ucs2be.h	1586
utf8.h	1586
viscii.h	1587
Ximint.h	1589
Xlibint.h	1589

Chapter 30

Module Documentation

30.1 Callback function typedefs

Typedefs defined in [<FL/FI.H>](#) for callback or handler functions passed as function parameters.

Typedefs

- typedef void(* **FI_Abort_Handler**) (const char *format,...)
Signature of set_abort functions passed as parameters.
- typedef int(* **FI_Args_Handler**) (int argc, char **argv, int &i)
Signature of args functions passed as parameters.
- typedef void(* **FI_Atclose_Handler**) ([FI_Window](#) *window, void *data)
Signature of set_atclose functions passed as parameters.
- typedef void(* **FI_Awake_Handler**) (void *data)
Signature of some wakeup callback functions passed as parameters.
- typedef void() **FI_Box_Draw_F**(int x, int y, int w, int h, [FI_Color](#) color)
Signature of some box drawing functions passed as parameters.
- typedef void(* **FI_Clipboard_Notify_Handler**) (int source, void *data)
Signature of add_clipboard_notify functions passed as parameters.
- typedef int(* [FI_Event_Dispatch](#)) (int event, [FI_Window](#) *w)
Signature of event_dispatch functions passed as parameters.
- typedef int(* **FI_Event_Handler**) (int event)
Signature of add_handler functions passed as parameters.
- typedef void(* **FI_FD_Handler**) ([FL_SOCKET](#) fd, void *data)
Signature of add_fd functions passed as parameters.
- typedef void(* **FI_Idle_Handler**) (void *data)
Signature of add_idle callback functions passed as parameters.
- typedef void() **FI_Label_Draw_F**(const [FI_Label](#) *label, int x, int y, int w, int h, [FI_Align](#) align)
Signature of some label drawing functions passed as parameters.
- typedef void() **FI_Label_Measure_F**(const [FI_Label](#) *label, int &width, int &height)
Signature of some label measurement functions passed as parameters.
- typedef void(* **FI_Old_Idle_Handler**) ()
Signature of set_idle callback functions passed as parameters.
- typedef int(* **FI_System_Handler**) (void *event, void *data)
Signature of add_system_handler functions passed as parameters.
- typedef void(* **FI_Timeout_Handler**) (void *data)
Signature of some timeout callback functions passed as parameters.

30.1.1 Detailed Description

Typedefs defined in [<FL/FL.H>](#) for callback or handler functions passed as function parameters.

FLTK uses callback functions as parameters for some function calls, e.g. to set up global event handlers ([Fl::add_handler\(\)](#)), to add a timeout handler ([Fl::add_timeout\(\)](#)), and many more.

The typedefs defined in this group describe the function parameters used to set up or clear the callback functions and should also be referenced to define the callback function to handle such events in the user's code.

See also

[Fl::add_handler\(\)](#), [Fl::add_timeout\(\)](#), [Fl::repeat_timeout\(\)](#), [Fl::remove_timeout\(\)](#) and others

30.1.2 Typedef Documentation

30.1.2.1 Fl_Event_Dispatch

```
typedef int (* Fl_Event_Dispatch) (int event, Fl_Window *w)
```

Signature of event_dispatch functions passed as parameters.

See also

[Fl::event_dispatch\(Fl_Event_Dispatch\)](#)

30.2 Windows handling functions

Windows and standard dialogs handling declared in [<FL/FL.H>](#)

Functions

- static void [Fl::default_atclose](#) ([Fl_Window *](#), void *)
Default callback for window widgets.
- static [Fl_Window *](#) [Fl::first_window](#) ()
Returns the first top-level window in the list of shown() windows.
- static void [Fl::first_window](#) ([Fl_Window *](#))
Sets the window that is returned by [first_window\(\)](#).
- static [Fl_Window *](#) [Fl::grab](#) ()
Returns the window that currently receives all events.
- static void [Fl::grab](#) ([Fl_Window *](#))
Selects the window to grab.
- static [Fl_Window *](#) [Fl::modal](#) ()
Returns the top-most [modal\(\)](#) window currently shown.
- static [Fl_Window *](#) [Fl::next_window](#) (const [Fl_Window *](#))
Returns the next top-level window in the list of shown() windows.
- static void [Fl::set_abort](#) ([Fl_Abort_Handler](#) f)
For back compatibility, sets the void [Fl::fatal](#) handler callback.
- static void [Fl::set_atclose](#) ([Fl_Atclose_Handler](#) f)
For back compatibility, sets the [Fl::atclose](#) handler callback.

Variables

- static void(* [Fl::atclose](#))([Fl_Window *](#), void *)
Back compatibility: default window callback handler.

30.2.1 Detailed Description

Windows and standard dialogs handling declared in [<FL/FL.H>](#)

30.2.2 Function Documentation

30.2.2.1 default_atclose()

```
void Fl::default_atclose (
    Fl_Window * window,
    void * v ) [static]
```

Default callback for window widgets.

It hides the window and then calls the default widget callback.

30.2.2.2 first_window() [1/2]

```
Fl_Window * Fl::first_window ( ) [static]
```

Returns the first top-level window in the list of shown() windows.

If a modal() window is shown this is the top-most modal window, otherwise it is the most recent window to get an event.

30.2.2.3 first_window() [2/2]

```
void Fl::first_window (
    Fl_Window * window ) [static]
```

Sets the window that is returned by first_window().

The window is removed from wherever it is in the list and inserted at the top. This is not done if Fl::modal() is on or if the window is not shown(). Because the first window is used to set the "parent" of modal windows, this is often useful.

30.2.2.4 grab() [1/2]

```
static Fl_Window * Fl::grab ( ) [inline], [static]
```

Returns the window that currently receives all events.

Returns

The window that currently receives all events, or NULL if event grabbing is currently OFF.

30.2.2.5 grab() [2/2]

```
void Fl::grab (
    Fl_Window * win ) [static]
```

Selects the window to grab.

This is used when pop-up menu systems are active.

Send all events to the passed window no matter where the pointer or focus is (including in other programs). The window *does not have to be shown()*, this lets the handle() method of a "dummy" window override all event handling and allows you to map and unmap a complex set of windows (under both X and WIN32 *some* window must be mapped because the system interface needs a window id).

If grab() is on it will also affect show() of windows by doing system-specific operations (on X it turns on override-redirect). These are designed to make menus popup reliably and faster on the system.

To turn off grabbing do Fl::grab(0).

Be careful that your program does not enter an infinite loop while grab() is on. On X this will lock up your screen! To avoid this potential lockup, all newer operating systems seem to limit mouse pointer grabbing to the time during which a mouse button is held down. Some OS's may not support grabbing at all.

30.2.2.6 modal()

```
static Fl_Window * Fl::modal ( ) [inline], [static]
```

Returns the top-most modal() window currently shown.

This is the most recently shown() window with `modal()` true, or NULL if there are no `modal()` windows shown(). The `modal()` window has its `handle()` method called for all events, and no other windows will have `handle()` called (`grab()` overrides this).

30.2.2.7 next_window()

```
Fl_Window * Fl::next_window (
    const Fl_Window * window ) [static]
```

Returns the next top-level window in the list of shown() windows. You can use this call to iterate through all the windows that are shown().

Parameters

in	<code>window</code>	must be shown and not NULL
----	---------------------	----------------------------

30.2.2.8 set_atclose()

```
static void Fl::set_atclose (
    Fl_Atclose_Handler f ) [inline], [static]
```

For back compatibility, sets the `Fl::atclose` handler callback. You can now simply change the callback for the window instead.

See also

[Fl_Window::callback\(Fl_Callback*\)](#)

30.2.3 Variable Documentation

30.2.3.1 atclose

```
void(* Fl::atclose)(Fl_Window *, void *)=default_atclose [static], [default]
```

Back compatibility: default window callback handler.

See also

[Fl::set_atclose\(\)](#)

30.3 Events handling functions

Fl class events handling API declared in `<FL/Fl.H>`

Functions

- static void `Fl::add_handler` (`Fl_Event_Handler h`)
Install a function to parse unrecognized events.
- static void `Fl::add_system_handler` (`Fl_System_Handler h`, `void *data`)
Install a function to intercept system events.
- static `Fl_Widget *` `Fl::belowmouse` ()
Gets the widget that is below the mouse.
- static void `Fl::belowmouse` (`Fl_Widget *`)
Sets the widget that is below the mouse.
- static int `Fl::compose` (`int &del`)
Any text editing widget should call this for each FL_KEYBOARD event.
- static void `Fl::compose_reset` ()

- If the user moves the cursor, be sure to call `Fl::compose_reset()`.*
- static void `Fl::disable_im ()`
Disables the system input methods facilities.
 - static void `Fl::enable_im ()`
Enables the system input methods facilities.
 - static int `Fl::event ()`
Returns the last event that was processed.
 - static int `Fl::event_alt ()`
Returns non-zero if the Alt key is pressed.
 - static int `Fl::event_button ()`
Gets which particular mouse button caused the current event.
 - static int `Fl::event_button1 ()`
Returns non-zero if mouse button 1 is currently held down.
 - static int `Fl::event_button2 ()`
Returns non-zero if button 2 is currently held down.
 - static int `Fl::event_button3 ()`
Returns non-zero if button 3 is currently held down.
 - static int `Fl::event_buttons ()`
Returns the mouse buttons state bits; if non-zero, then at least one button is pressed now.
 - static int `Fl::event_clicks ()`
Returns non zero if we had a double click event.
 - static void `Fl::event_clicks (int i)`
Manually sets the number returned by `Fl::event_clicks()`.
 - static void * `Fl::event_clipboard ()`
During an `FL_PASTE` event of non-textual data, returns a pointer to the pasted data.
 - static const char * `Fl::event_clipboard_type ()`
Returns the type of the pasted data during an `FL_PASTE` event.
 - static int `Fl::event_command ()`
Returns non-zero if the `FL_COMMAND` key is pressed, either `FL_CTRL` or on OSX `FL_META`.
 - static int `Fl::event_ctrl ()`
Returns non-zero if the Control key is pressed.
 - static `Fl_Event_Dispatch Fl::event_dispatch ()`
Return the current event dispatch function.
 - static void `Fl::event_dispatch (Fl_Event_Dispatch d)`
Set a new event dispatch function.
 - static int `Fl::event_dx ()`
Returns the current horizontal mouse scrolling associated with the `FL_MOUSEWHEEL` event.
 - static int `Fl::event_dy ()`
Returns the current vertical mouse scrolling associated with the `FL_MOUSEWHEEL` event.
 - static int `Fl::event_inside (const Fl_Widget *)`
Returns whether or not the mouse event is inside a given child widget.
 - static int `Fl::event_inside (int, int, int, int)`
Returns whether or not the mouse event is inside the given rectangle.
 - static int `Fl::event_is_click ()`
Returns non-zero if the mouse has not moved far enough and not enough time has passed since the last `FL_PUSH` or `FL_KEYBOARD` event for it to be considered a "drag" rather than a "click".
 - static void `Fl::event_is_click (int i)`
Clears the value returned by `Fl::event_is_click()`.
 - static int `Fl::event_key ()`
Gets which key on the keyboard was last pushed.
 - static int `Fl::event_key (int key)`

- Returns true if the given `key` was held down (or pressed) during the last event.*

 - static int `Fl::event_length ()`
Returns the length of the text in `Fl::event_text()`.
 - static int `Fl::event_original_key ()`
Returns the keycode of the last key event, regardless of the NumLock state.
 - static int `Fl::event_shift ()`
Returns non-zero if the Shift key is pressed.
 - static int `Fl::event_state ()`
Returns the keyboard and mouse button states of the last event.
 - static int `Fl::event_state (int mask)`
Returns non-zero if any of the passed event state bits are turned on.
 - static const char * `Fl::event_text ()`
Returns the text associated with the current event, including `FL_PASTE` or `FL_DND_RELEASE` events.
 - static int `Fl::event_x ()`
Returns the mouse position of the event relative to the `Fl_Window` it was passed to.
 - static int `Fl::event_x_root ()`
Returns the mouse position on the screen of the event.
 - static int `Fl::event_y ()`
Returns the mouse position of the event relative to the `Fl_Window` it was passed to.
 - static int `Fl::event_y_root ()`
Returns the mouse position on the screen of the event.
 - static `Fl_Widget *` `Fl::focus ()`
Gets the current `Fl::focus()` widget.
 - static void `Fl::focus (Fl_Widget *)`
Sets the widget that will receive `FL_KEYBOARD` events.
 - static int `Fl::get_key (int key)`
Returns true if the given `key` is held down now.
 - static void `Fl::get_mouse (int &, int &)`
Return where the mouse is on the screen by doing a round-trip query to the server.
 - static int `Fl::handle (int, Fl_Window *)`
Handle events from the window system.
 - static int `Fl::handle_ (int, Fl_Window *)`
Handle events from the window system.
 - static `Fl_Widget *` `Fl::pushed ()`
Gets the widget that is being pushed.
 - static void `Fl::pushed (Fl_Widget *)`
Sets the widget that is being pushed.
 - static void `Fl::remove_handler (Fl_Event_Handler h)`
Removes a previously added event handler.
 - static void `Fl::remove_system_handler (Fl_System_Handler h)`
Removes a previously added system event handler.
 - static int `Fl::test_shortcut (Fl_Shortcut)`
Tests the current event, which must be an `FL_KEYBOARD` or `FL_SHORTCUT`, against a shortcut value (described in `Fl_Button`).

Variables

- const char *const `fl_eventnames []`
This is an array of event names you can use to convert event numbers into names.
- const char *const `fl_fontnames []`
This is an array of font names you can use to convert font numbers into names.

30.3.1 Detailed Description

Fl class events handling API declared in [<FL/Fl.H>](#)

30.3.2 Function Documentation

30.3.2.1 add_handler()

```
void Fl::add_handler (
    Fl_Event_Handler ha ) [static]
```

Install a function to parse unrecognized events.

If FLTK cannot figure out what to do with an event, it calls each of these functions (most recent first) until one of them returns non-zero. If none of them returns non-zero then the event is ignored. Events that cause this to be called are:

- [FL_SHORTCUT](#) events that are not recognized by any widget. This lets you provide global shortcut keys.
- [FL_SCREEN_CONFIGURATION_CHANGED](#) events. Under X11, this event requires the libXrandr.so shared library to be loadable at run-time and the X server to implement the RandR extension.
- [FL_FULLSCREEN](#) events sent to a window that enters or leaves fullscreen mode.
- System events that FLTK does not recognize. See [fl_xevent](#).
- *Some* other events when the widget FLTK selected returns zero from its [handle\(\)](#) method. Exactly which ones may change in future versions, however.

See also

[Fl::remove_handler\(Fl_Event_Handler\)](#)
[Fl::event_dispatch\(Fl_Event_Dispatch d\)](#)
[Fl::handle\(int, Fl_Window*\)](#)

30.3.2.2 add_system_handler()

```
void Fl::add_system_handler (
    Fl_System_Handler ha,
    void * data ) [static]
```

Install a function to intercept system events.

FLTK calls each of these functions as soon as a new system event is received. The processing will stop at the first function to return non-zero. If all functions return zero then the event is passed on for normal handling by FLTK.

Each function will be called with a pointer to the system event as the first argument and *data* as the second argument. The system event pointer will always be void *, but will point to different objects depending on the platform:

- X11: XEvent
- Windows: MSG
- OS X: NSEvent

Parameters

<i>ha</i>	The event handler function to register
<i>data</i>	User data to include on each call

See also

[Fl::remove_system_handler\(Fl_System_Handler\)](#)

30.3.2.3 belowmouse() [1/2]

```
static Fl_Widget * Fl::belowmouse ( ) [inline], [static]
```

Gets the widget that is below the mouse.

See also

[belowmouse\(Fl_Widget*\)](#)

30.3.2.4 belowmouse() [2/2]

```
void Fl::belowmouse (
    Fl_Widget * o ) [static]
```

Sets the widget that is below the mouse.

This is for highlighting buttons. It is not used to send FL_PUSH or FL_MOVE directly, for several obscure reasons, but those events typically go to this widget. This is also the first widget tried for FL_SHORTCUT events.

If you change the belowmouse widget, the previous one and all parents (that don't contain the new widget) are sent FL_LEAVE events. Changing this does *not* send FL_ENTER to this or any widget, because sending FL_ENTER is supposed to *test* if the widget wants the mouse (by it returning non-zero from [handle\(\)](#)).

30.3.2.5 compose()

```
int Fl::compose (
    int & del ) [static]
```

Any text editing widget should call this for each FL_KEYBOARD event.

Use of this function is very simple.

If *true* is returned, then it has modified the [Fl::event_text\(\)](#) and [Fl::event_length\(\)](#) to a set of *bytes* to insert (it may be of zero length!). It will also set the "del" parameter to the number of *bytes* to the left of the cursor to delete, this is used to delete the results of the previous call to [Fl::compose\(\)](#).

If *false* is returned, the keys should be treated as function keys, and del is set to zero. You could insert the text anyways, if you don't know what else to do.

On the Mac OS platform, text input can involve marked text, that is, temporary text replaced by other text during the input process. This occurs, e.g., when using dead keys or when entering CJK characters. Text editing widgets should preferentially signal marked text, usually underlining it. Widgets can use `int Fl::compose_state` after having called [Fl::compose\(int&\)](#) to obtain the length in bytes of marked text that always finishes at the current insertion point. It's the widget's task to underline marked text. Widgets should also call `void Fl::reset_marked_text()` when processing FL_UNFOCUS events. Optionally, widgets can also call `void Fl::insertion_point_location(int x, int y, int height)` to indicate the window coordinates of the bottom of the current insertion point and the line height. This way, auxiliary windows that help choosing among alternative characters appear just below the insertion point. If widgets don't do that, auxiliary windows appear at the widget's bottom. The [Fl_Input](#) and [Fl_Text_Editor](#) widgets underline marked text. If none of this is done by a user-defined text editing widget, text input will work, but will not signal to the user what text is marked. Finally, text editing widgets should call `set_flag(MAC_USE_ACCENTS_MENU)`; in their constructor if they want to use the feature introduced with Mac OS 10.7 "Lion" where pressing and holding a key on the keyboard opens an accented-character menu window.

Though the current implementation returns immediately, future versions may take quite awhile, as they may pop up a window or do other user-interface things to allow characters to be selected.

30.3.2.6 compose_reset()

```
void Fl::compose_reset ( ) [static]
```

If the user moves the cursor, be sure to call [Fl::compose_reset\(\)](#).

The next call to [Fl::compose\(\)](#) will start out in an initial state. In particular it will not set "del" to non-zero. This call is very fast so it is ok to call it many times and in many places.

30.3.2.7 disable_im()

```
static void Fl::disable_im ( ) [static]
```

Disables the system input methods facilities.

See also

[enable_im\(\)](#)

30.3.2.8 enable_im()

```
static void Fl::enable_im ( ) [static]
```

Enables the system input methods facilities.
This is the default.

See also

[disable_im\(\)](#)

30.3.2.9 event()

```
static int Fl::event ( ) [inline], [static]
```

Returns the last event that was processed.

This can be used to determine if a callback is being done in response to a keypress, mouse click, etc.

30.3.2.10 event_button()

```
static int Fl::event_button ( ) [inline], [static]
```

Gets which particular mouse button caused the current event.

This returns garbage if the most recent event was not a FL_PUSH or FL_RELEASE event.

Return values

<i>FL_LEFT_MOUSE</i>	
<i>FL_MIDDLE_MOUSE</i>	
<i>FL_RIGHT_MOUSE.</i>	

See also

[Fl::event_buttons\(\)](#)

30.3.2.11 event_button1()

```
static int Fl::event_button1 ( ) [inline], [static]
```

Returns non-zero if mouse button 1 is currently held down.

For more details, see [Fl::event_buttons\(\)](#).

30.3.2.12 event_button2()

```
static int Fl::event_button2 ( ) [inline], [static]
```

Returns non-zero if button 2 is currently held down.

For more details, see [Fl::event_buttons\(\)](#).

30.3.2.13 event_button3()

```
static int Fl::event_button3 ( ) [inline], [static]
```

Returns non-zero if button 3 is currently held down.
For more details, see [Fl::event_buttons\(\)](#).

30.3.2.14 event_buttons()

```
static int Fl::event_buttons ( ) [inline], [static]
```

Returns the mouse buttons state bits; if non-zero, then at least one button is pressed now.

This function returns the button state at the time of the event. During an FL_RELEASE event, the state of the released button will be 0. To find out, which button caused an FL_RELEASE event, you can use [Fl::event_button\(\)](#) instead.

Returns

a bit mask value like { [FL_BUTTON1] | [FL_BUTTON2] | [FL_BUTTON3] }

30.3.2.15 event_clicks() [1/2]

```
static int Fl::event_clicks ( ) [inline], [static]
```

Returns non zero if we had a double click event.

Return values

<i>Non-zero</i>	if the most recent FL_PUSH or FL_KEYBOARD was a "double click".
<i>N-1</i>	for N clicks. A double click is counted if the same button is pressed again while event_is_click() is true.

30.3.2.16 event_clicks() [2/2]

```
static void Fl::event_clicks (
    int i ) [inline], [static]
```

Manually sets the number returned by [Fl::event_clicks\(\)](#).

This can be used to set it to zero so that later code does not think an item was double-clicked.

Parameters

in	<i>i</i>	corresponds to no double-click if 0, i+1 mouse clicks otherwise
----	----------	---

See also

int [event_clicks\(\)](#)

30.3.2.17 event_clipboard()

```
static void * Fl::event_clipboard ( ) [inline], [static]
```

During an FL_PASTE event of non-textual data, returns a pointer to the pasted data.

The returned data is an [Fl_Image *](#) when the result of [Fl::event_clipboard_type\(\)](#) is [Fl::clipboard_image](#).

30.3.2.18 event_clipboard_type()

```
static const char * Fl::event_clipboard_type ( ) [inline], [static]
```

Returns the type of the pasted data during an FL_PASTE event.

This type can be [Fl::clipboard_plain_text](#) or [Fl::clipboard_image](#).

30.3.2.19 event_dispatch()

```
void Fl::event_dispatch (
    Fl_Event_Dispatch d ) [static]
```

Set a new event dispatch function.

The event dispatch function is called after native events are converted to FLTK events, but before they are handled by FLTK. If the dispatch function `Fl_Event_Dispatch d` is set, it is up to the dispatch function to call `Fl::handle_(int, Fl_Window*)` or to ignore the event.

The dispatch function itself must return 0 if it ignored the event, or non-zero if it used the event. If you call `Fl::handle_()`, then this will return the correct value.

The event dispatch can be used to handle exceptions in FLTK events and callbacks before they reach the native event handler:

```
int myHandler(int e, Fl_Window *w) {
    try {
        return Fl::handle_(e, w);
    } catch () {
        ...
    }
}
main() {
    Fl::event_dispatch(myHandler);
    ...
    Fl::run();
}
```

Parameters

<i>d</i>	new dispatch function, or NULL
----------	--------------------------------

See also

[Fl::add_handler\(Fl_Event_Handler\)](#)

[Fl::handle\(int, Fl_Window*\)](#)

[Fl::handle_\(int, Fl_Window*\)](#)

30.3.2.20 event_dx()

```
static int Fl::event_dx ( ) [inline], [static]
```

Returns the current horizontal mouse scrolling associated with the `FL_MOUSEWHEEL` event.

Right is positive.

30.3.2.21 event_dy()

```
static int Fl::event_dy ( ) [inline], [static]
```

Returns the current vertical mouse scrolling associated with the `FL_MOUSEWHEEL` event.

Down is positive.

30.3.2.22 event_inside() [1/2]

```
int Fl::event_inside (
    const Fl_Widget * o ) [static]
```

Returns whether or not the mouse event is inside a given child widget.

Returns non-zero if the current [Fl::event_x\(\)](#) and [Fl::event_y\(\)](#) put it inside the given child widget's bounding box.

This method can only be used to check whether the mouse event is inside a **child** widget of the window that handles the event, and there must not be an intermediate subwindow (i.e. the widget must not be inside a subwindow of the current window). However, it is valid if the widget is inside a nested [Fl_Group](#).

You must not use it with the window itself as the `o` argument in a window's [handle\(\)](#) method.

Note

The mentioned restrictions are necessary, because this method does not transform coordinates of child widgets, and thus the given widget `o` must be within the *same* window that is handling the current event. Otherwise the results are undefined.

You should always call this rather than doing your own comparison so you are consistent about edge effects.

See also

[Fl::event_inside\(int, int, int, int\)](#)

Parameters

<code>in</code>	<code>o</code>	child widget to be tested
-----------------	----------------	---------------------------

Returns

non-zero, if mouse event is inside the widget

30.3.2.23 event_inside() [2/2]

```
int Fl::event_inside (
    int xx,
    int yy,
    int ww,
    int hh ) [static]
```

Returns whether or not the mouse event is inside the given rectangle.

Returns non-zero if the current [Fl::event_x\(\)](#) and [Fl::event_y\(\)](#) put it inside the given arbitrary bounding box.

You should always call this rather than doing your own comparison so you are consistent about edge effects.

To find out, whether the event is inside a child widget of the current window, you can use [Fl::event_inside\(const Fl_Widget *\)](#).

Parameters

<code>in</code>	<code>xx,yy,ww,hh</code>	bounding box
-----------------	--------------------------	--------------

Returns

non-zero, if mouse event is inside

30.3.2.24 event_is_click() [1/2]

```
static int Fl::event_is_click ( ) [inline], [static]
```

Returns non-zero if the mouse has not moved far enough and not enough time has passed since the last `FL_PUSH` or `FL_KEYBOARD` event for it to be considered a "drag" rather than a "click".

You can test this on `FL_DRAG`, `FL_RELEASE`, and `FL_MOVE` events.

30.3.2.25 event_is_click() [2/2]

```
static void Fl::event_is_click (
    int i ) [inline], [static]
```

Clears the value returned by [Fl::event_is_click\(\)](#).

Useful to prevent the *next* click from being counted as a double-click or to make a popup menu pick an item with a single click. Don't pass non-zero to this.

30.3.2.26 event_key() [1/2]

```
static int Fl::event_key ( ) [inline], [static]
```

Gets which key on the keyboard was last pushed.

The returned integer 'key code' is not necessarily a text equivalent for the keystroke. For instance: if someone presses '5' on the numeric keypad with numlock on, [Fl::event_key\(\)](#) may return the 'key code' for this key, and NOT the character '5'. To always get the '5', use [Fl::event_text\(\)](#) instead.

Returns

an integer 'key code', or 0 if the last event was not a key press or release.

See also

int [event_key\(int\)](#), [event_text\(\)](#), [compose\(int&\)](#).

30.3.2.27 event_key() [2/2]

```
int Fl::event_key (
    int key ) [static]
```

Returns true if the given *key* was held down (or pressed) *during* the last event.

This is constant until the next event is read from the server.

[Fl::get_key\(int\)](#) returns true if the given key is held down *now*. Under X this requires a round-trip to the server and is *much* slower than [Fl::event_key\(int\)](#).

Keys are identified by the *unshifted* values. FLTK defines a set of symbols that should work on most modern machines for every key on the keyboard:

- All keys on the main keyboard producing a printable ASCII character use the value of that ASCII character (as though shift, ctrl, and caps lock were not on). The space bar is 32.
- All keys on the numeric keypad producing a printable ASCII character use the value of that ASCII character plus FL_KP. The highest possible value is FL_KP_Last so you can range-check to see if something is on the keypad.
- All numbered function keys use the number on the function key plus FL_F. The highest possible number is FL_F_Last, so you can range-check a value.
- Buttons on the mouse are considered keys, and use the button number (where the left button is 1) plus FL_Button.
- All other keys on the keypad have a symbol: FL_Escape, FL_BackSpace, FL_Tab, FL_Enter, FL_Print, FL_↵, FL_Scroll_Lock, FL_Pause, FL_Insert, FL_Home, FL_Page_Up, FL_Delete, FL_End, FL_Page_Down, FL_Left, FL_Up, FL_Right, FL_Down, FL_Iso_Key, FL_Shift_L, FL_Shift_R, FL_Control_L, FL_Control_R, FL_Caps_↵, FL_Lock, FL_Alt_L, FL_Alt_R, FL_Meta_L, FL_Meta_R, FL_Menu, FL_Num_Lock, FL_KP_Enter. Be careful not to confuse these with the very similar, but all-caps, symbols used by [Fl::event_state\(\)](#).

On X [Fl::get_key\(FL_Button+n\)](#) does not work.

On WIN32 [Fl::get_key\(FL_KP_Enter\)](#) and [Fl::event_key\(FL_KP_Enter\)](#) do not work.

30.3.2.28 event_length()

```
static int Fl::event_length ( ) [inline], [static]
```

Returns the length of the text in [Fl::event_text\(\)](#).

There will always be a nul at this position in the text. However there may be a nul before that if the keystroke translates to a nul character or you paste a nul character.

30.3.2.29 event_original_key()

```
static int Fl::event_original_key ( ) [inline], [static]
```

Returns the keycode of the last key event, regardless of the NumLock state.

If NumLock is deactivated, FLTK translates events from the numeric keypad into the corresponding arrow key events. [event_key\(\)](#) returns the translated key code, whereas [event_original_key\(\)](#) returns the keycode before NumLock translation.

30.3.2.30 event_state() [1/2]

```
static int Fl::event_state ( ) [inline], [static]
```

Returns the keyboard and mouse button states of the last event.

This is a bitfield of what shift states were on and what mouse buttons were held down during the most recent event. The legal event state bits are:

- FL_SHIFT
- FL_CAPS_LOCK
- FL_CTRL
- FL_ALT
- FL_NUM_LOCK
- FL_META
- FL_SCROLL_LOCK
- FL_BUTTON1
- FL_BUTTON2
- FL_BUTTON3

X servers do not agree on shift states, and FL_NUM_LOCK, FL_META, and FL_SCROLL_LOCK may not work. The values were selected to match the XFree86 server on Linux. In addition there is a bug in the way X works so that the shift state is not correctly reported until the first event *after* the shift key is pressed or released.

30.3.2.31 event_state() [2/2]

```
static int Fl::event_state (
    int mask ) [inline], [static]
```

Returns non-zero if any of the passed event state bits are turned on.

Use `mask` to pass the event states you're interested in. The legal event state bits are defined in [Fl::event_state\(\)](#).

30.3.2.32 event_text()

```
static const char * Fl::event_text ( ) [inline], [static]
```

Returns the text associated with the current event, including FL_PASTE or FL_DND_RELEASE events.

This can be used in response to FL_KEYUP, FL_KEYDOWN, FL_PASTE, and FL_DND_RELEASE.

When responding to FL_KEYUP/FL_KEYDOWN, use this function instead of [Fl::event_key\(\)](#) to get the text equivalent of keystrokes suitable for inserting into strings and text widgets.

The returned string is guaranteed to be NULL terminated. However, see [Fl::event_length\(\)](#) for the actual length of the string, in case the string itself contains NULLs that are part of the text data.

Returns

A NULL terminated text string equivalent of the last keystroke.

30.3.2.33 event_x_root()

```
static int Fl::event_x_root ( ) [inline], [static]
```

Returns the mouse position on the screen of the event.

To find the absolute position of an [Fl_Window](#) on the screen, use the difference between [event_x_root\(\)](#), [event_y_root\(\)](#) and [event_x\(\)](#), [event_y\(\)](#).

30.3.2.34 event_y_root()

```
static int Fl::event_y_root ( ) [inline], [static]
```

Returns the mouse position on the screen of the event.

To find the absolute position of an [Fl_Window](#) on the screen, use the difference between [event_x_root\(\)](#), [event_y_root\(\)](#) and [event_x\(\)](#), [event_y\(\)](#).

30.3.2.35 focus() [1/2]

```
static Fl_Widget * Fl::focus ( ) [inline], [static]
```

Gets the current [Fl::focus\(\)](#) widget.

See also

[Fl::focus\(Fl_Widget*\)](#)

30.3.2.36 focus() [2/2]

```
void Fl::focus (
    Fl_Widget * o ) [static]
```

Sets the widget that will receive FL_KEYBOARD events.

If you change [Fl::focus\(\)](#), the previous widget and all parents (that don't contain the new widget) are sent FL_↔ UNFOCUS events. Changing the focus does *not* send FL_FOCUS to this or any widget, because sending FL_↔ FOCUS is supposed to *test* if the widget wants the focus (by it returning non-zero from [handle\(\)](#)).

See also

[Fl_Widget::take_focus\(\)](#)

30.3.2.37 get_key()

```
int Fl::get_key (
    int key ) [static]
```

Returns true if the given key is held down *now*.

Under X this requires a round-trip to the server and is *much* slower than [Fl::event_key\(int\)](#).

See also

[event_key\(int\)](#)

30.3.2.38 get_mouse()

```
static void Fl::get_mouse (
    int & ,
    int & ) [static]
```

Return where the mouse is on the screen by doing a round-trip query to the server.

You should use [Fl::event_x_root\(\)](#) and [Fl::event_y_root\(\)](#) if possible, but this is necessary if you are not sure if a mouse event has been processed recently (such as to position your first window). If the display is not open, this will open it.

30.3.2.39 handle()

```
int Fl::handle (
    int e,
    Fl_Window * window ) [static]
```

Handle events from the window system.

This is called from the native event dispatch after native events have been converted to FLTK notation. This function calls `Fl::handle_(int, Fl_Window*)` unless the user sets a dispatch function. If a user dispatch function is set, the user must make sure that `Fl::handle_()` is called, or the event will be ignored.

Parameters

<code>e</code>	the event type (<code>Fl::event_number()</code> is not yet set)
<code>window</code>	the window that caused this event

Returns

0 if the event was not handled

See also

[Fl::add_handler\(Fl_Event_Handler\)](#)

[Fl::event_dispatch\(Fl_Event_Dispatch\)](#)

30.3.2.40 `handle_()`

```
int Fl::handle_ (
    int e,
    Fl_Window * window ) [static]
```

Handle events from the window system.

This function is called from the native event dispatch, unless the user sets another dispatch function. In that case, the user dispatch function must decide when to call `Fl::handle_(int, Fl_Window*)`

Parameters

<code>e</code>	the event type (<code>Fl::event_number()</code> is not yet set)
<code>window</code>	the window that caused this event

Returns

0 if the event was not handled

See also

[Fl::event_dispatch\(Fl_Event_Dispatch\)](#)

30.3.2.41 `pushed()` [1/2]

```
static Fl_Widget * Fl::pushed ( ) [inline], [static]
```

Gets the widget that is being pushed.

See also

[void pushed\(Fl_Widget*\)](#)

30.3.2.42 `pushed()` [2/2]

```
void Fl::pushed (
    Fl_Widget * o ) [static]
```

Sets the widget that is being pushed.

FL_DRAG or FL_RELEASE (and any more FL_PUSH) events will be sent to this widget.

If you change the pushed widget, the previous one and all parents (that don't contain the new widget) are sent FL_RELEASE events. Changing this does *not* send FL_PUSH to this or any widget, because sending FL_PUSH is supposed to *test* if the widget wants the mouse (by it returning non-zero from [handle\(\)](#)).

30.3.2.43 remove_handler()

```
void Fl::remove_handler (
    Fl_Event_Handler ha ) [static]
```

Removes a previously added event handler.

See also

[Fl::handle\(int, Fl_Window*\)](#)

30.3.2.44 remove_system_handler()

```
void Fl::remove_system_handler (
    Fl_System_Handler ha ) [static]
```

Removes a previously added system event handler.

Parameters

<i>ha</i>	The event handler function to remove
-----------	--------------------------------------

See also

[Fl::add_system_handler\(Fl_System_Handler\)](#)

30.3.2.45 test_shortcut()

```
int Fl::test_shortcut (
    Fl_Shortcut shortcut ) [static]
```

Tests the current event, which must be an FL_KEYBOARD or FL_SHORTCUT, against a shortcut value (described in [Fl_Button](#)).

Not to be confused with [Fl_Widget::test_shortcut\(\)](#).

Returns

non-zero if there is a match.

30.3.3 Variable Documentation

30.3.3.1 fl_eventnames

```
const char* const fl_eventnames[]
```

This is an array of event names you can use to convert event numbers into names.

The array gets defined inline wherever your '#include <[FL/names.h](#)>' appears.

Example:

```
#include <FL/names.h> // array will be defined here
int MyClass::handle(int e) {
    printf("Event was %s (%d)\n", fl_eventnames[e], e);
    // ..resulting output might be e.g. "Event was FL_PUSH (1)"..
    [..]
}
```

30.3.3.2 fl_fontnames

```
const char* const fl_fontnames[]
```

Initial value:

```
=
{
    "FL_HELVETICA",
    "FL_HELVETICA_BOLD",
    "FL_HELVETICA_ITALIC",
    "FL_HELVETICA_BOLD_ITALIC",
    "FL_COURIER",
    "FL_COURIER_BOLD",
    "FL_COURIER_ITALIC",
    "FL_COURIER_BOLD_ITALIC",
    "FL_TIMES",
    "FL_TIMES_BOLD",
    "FL_TIMES_ITALIC",
    "FL_TIMES_BOLD_ITALIC",
    "FL_SYMBOL",
    "FL_SCREEN",
    "FL_SCREEN_BOLD",
    "FL_ZAPF_DINGBATS",
}
```

This is an array of font names you can use to convert font numbers into names. The array gets defined inline wherever your '#include <FL/names.h>' appears.

Example:

```
#include <FL/names.h> // array will be defined here
int MyClass::my_callback(FL_Widget *w, void*) {
    int fnum = w->labelfont();
    // Resulting output might be e.g. "Label's font is FL_HELVETICA (0)"
    printf("Label's font is %s (%d)\n", fl_fontnames[fnum], fnum);
    // ..resulting output might be e.g. "Label's font is FL_HELVETICA (0)"..
    [...]
}
```

30.4 Selection & Clipboard functions

FLTK global copy/cut/paste functions declared in <FL/Fl.H>

Functions

- static void [Fl::add_clipboard_notify](#) ([Fl_Clipboard_Notify_Handler](#) h, void *data=0)
 - FLTK will call the registered callback whenever there is a change to the selection buffer or the clipboard.*
- static int [Fl::clipboard_contains](#) (const char *type)
 - Returns non 0 if the clipboard contains data matching type.*
- static void [Fl::copy](#) (const char *stuff, int len, int destination=0, const char *type=[Fl::clipboard_plain_text](#))
 - Copies the data pointed to by stuff to the selection buffer (destination is 0), the clipboard (destination is 1), or both (destination is 2).*
- static int [Fl::dnd](#) ()
 - Initiate a Drag And Drop operation.*
- static void [Fl::paste](#) ([Fl_Widget](#) &receiver)
 - Backward compatibility only.*
- static void [Fl::paste](#) ([Fl_Widget](#) &receiver, int source, const char *type=[Fl::clipboard_plain_text](#))
 - Pastes the data from the selection buffer (source is 0) or the clipboard (source is 1) into receiver.*
- static void [Fl::remove_clipboard_notify](#) ([Fl_Clipboard_Notify_Handler](#) h)
 - Stop calling the specified callback when there are changes to the selection buffer or the clipboard.*
- static void [Fl::selection](#) ([Fl_Widget](#) &owner, const char *, int len)
 - Changes the current selection.*
- static [Fl_Widget](#) * [Fl::selection_owner](#) ()
 - back-compatibility only: Gets the widget owning the current selection*
- static void [Fl::selection_owner](#) ([Fl_Widget](#) *)
 - Back-compatibility only: The single-argument call can be used to move the selection to another widget or to set the owner to NULL, without changing the actual text of the selection.*

Variables

- static char const *const **Fl::clipboard_image** = "image"
Denotes image data.
- static char const *const **Fl::clipboard_plain_text** = "text/plain"
Denotes plain textual data.

30.4.1 Detailed Description

FLTK global copy/cut/paste functions declared in <FL/FL.H>

30.4.2 Function Documentation

30.4.2.1 add_clipboard_notify()

```
void Fl::add_clipboard_notify (
    Fl_Clipboard_Notify_Handler h,
    void * data = 0 ) [static]
```

FLTK will call the registered callback whenever there is a change to the selection buffer or the clipboard. The source argument indicates which of the two has changed. Only changes by other applications are reported.

Example:

```
void clip_callback(int source, void *data) {
    if ( source == 0 ) printf("CLIP CALLBACK: selection buffer changed\n");
    if ( source == 1 ) printf("CLIP CALLBACK: clipboard changed\n");
}
[..]
int main() {
    [..]
    Fl::add_clipboard_notify(clip_callback);
    [..]
}
```

Note

Some systems require polling to monitor the clipboard and may therefore have some delay in detecting changes.

30.4.2.2 clipboard_contains()

```
static int Fl::clipboard_contains (
    const char * type ) [static]
```

Returns non 0 if the clipboard contains data matching `type`.

`type` can be [Fl::clipboard_plain_text](#) or [Fl::clipboard_image](#).

30.4.2.3 copy()

```
static void Fl::copy (
    const char * stuff,
    int len,
    int destination = 0,
    const char * type = Fl::clipboard_plain_text ) [static]
```

Copies the data pointed to by `stuff` to the selection buffer (`destination` is 0), the clipboard (`destination` is 1), or both (`destination` is 2).

Copying to both is only relevant on X11, on other platforms it maps to the clipboard (1). `len` is the number of relevant bytes in `stuff`. `type` is always [Fl::clipboard_plain_text](#). The selection buffer is used for middle-mouse pastes and for drag-and-drop selections. The clipboard is used for traditional copy/cut/paste operations.

Note

This function is, at present, intended only to copy UTF-8 encoded textual data. To copy graphical data, use the [Fl_Copy_Surface](#) class. The `type` argument may allow in the future to copy other kinds of data.

30.4.2.4 dnd()

```
int Fl::dnd ( ) [static]
```

Initiate a Drag And Drop operation.

The selection buffer should be filled with relevant data before calling this method. FLTK will then initiate the system wide drag and drop handling. Dropped data will be marked as *text*.

Create a selection first using: `Fl::copy(const char *stuff, int len, 0)`

30.4.2.5 paste() [1/2]

```
void Fl::paste (
    Fl_Widget & receiver ) [static]
```

Backward compatibility only.

This calls `Fl::paste(receiver, 0)`;

See also

[Fl::paste\(Fl_Widget &receiver, int clipboard, const char* type\)](#)

30.4.2.6 paste() [2/2]

```
static void Fl::paste (
    Fl_Widget & receiver,
    int source,
    const char * type = Fl::clipboard_plain_text ) [static]
```

Pastes the data from the selection buffer (`source` is 0) or the clipboard (`source` is 1) into `receiver`.

The selection buffer (`source` is 0) is used for middle-mouse pastes and for drag-and-drop selections. The clipboard (`source` is 1) is used for copy/cut/paste operations.

If `source` is 1, the optional `type` argument indicates what type of data is requested from the clipboard. At present, [Fl::clipboard_plain_text](#) (requesting text data) and [Fl::clipboard_image](#) (requesting image data) are possible. Set things up so the handle function of the `receiver` widget will be called with an `FL_PASTE` event some time in the future if the clipboard does contain data of the requested type. While processing the `FL_PASTE` event:

- if `type` is [Fl::clipboard_plain_text](#), the text string from the specified `source` is in [Fl::event_text\(\)](#) with UTF-8 encoding, and the number of bytes in [Fl::event_length\(\)](#). If [Fl::paste\(\)](#) gets called during the drop step of a files-drag-and-drop operation, [Fl::event_text\(\)](#) contains a list of filenames (see [Drag and Drop Events](#)).
- if `type` is [Fl::clipboard_image](#), the pointer returned by [Fl::event_clipboard\(\)](#) can be safely cast to type [Fl_Image *](#) to obtain a pointer to the pasted image. Furthermore, starting with FLTK 1.3.4, the image is of type [Fl_RGB_Image](#) across all platforms. If `receiver` accepts the clipboard image, `receiver.handle()` should return 1 and the application should take ownership of this image (that is, delete it after use). Conversely, if `receiver.handle()` returns 0, the application must not use the image.

The receiver should be prepared to be called *directly* by this, or for it to happen *later*, or possibly *not at all*. This allows the window system to take as long as necessary to retrieve the paste buffer (or even to screw up completely) without complex and error-prone synchronization code in FLTK.

Platform details for image data:

- Unix/Linux platform: Clipboard images in PNG or BMP formats are recognized. Requires linking with the `fttk_images` library.
- MSWindows platform: Both bitmap and vectorial (Enhanced metafile) data from clipboard can be pasted as image data.
- Mac OS X platform: Both bitmap (TIFF) and vectorial (PDF) data from clipboard can be pasted as image data.

30.4.2.7 `selection()`

```
void Fl::selection (
    Fl_Widget & owner,
    const char * text,
    int len ) [static]
```

Changes the current selection.

The block of text is copied to an internal buffer by FLTK (be careful if doing this in response to an `FL_PASTE` as this may be the same buffer returned by `event_text()`). The `selection_owner()` widget is set to the passed owner.

30.4.2.8 `selection_owner()` [1/2]

```
static Fl_Widget * Fl::selection_owner ( ) [inline], [static]
```

back-compatibility only: Gets the widget owning the current selection

See also

`Fl_Widget*` [selection_owner\(Fl_Widget*\)](#)

30.4.2.9 `selection_owner()` [2/2]

```
void Fl::selection_owner (
    Fl_Widget * owner ) [static]
```

Back-compatibility only: The single-argument call can be used to move the selection to another widget or to set the owner to `NULL`, without changing the actual text of the selection.

`FL_SELECTIONCLEAR` is sent to the previous selection owner, if any.

Copying the buffer every time the selection is changed is obviously wasteful, especially for large selections. An interface will probably be added in a future version to allow the selection to be made by a callback function. The current interface will be emulated on top of this.

30.5 Screen functions

fl global screen functions declared in `<FL/FL.H>`

Functions

- static int `Fl::h ()`
Returns the height in pixels of the main screen work area.
- static int `Fl::screen_count ()`
Gets the number of available screens.
- static void `Fl::screen_dpi (float &h, float &v, int n=0)`
Gets the screen resolution in dots-per-inch for the given screen.
- static int `Fl::screen_num (int x, int y)`
*Gets the screen number of a screen that contains the specified screen position *x*, *y*.*

- static int `Fl::screen_num` (int `x`, int `y`, int `w`, int `h`)
Gets the screen number for the screen which intersects the most with the rectangle defined by `x`, `y`, `w`, `h`.
- static void `Fl::screen_work_area` (int &`X`, int &`Y`, int &`W`, int &`H`)
Gets the bounding box of the work area of the screen that contains the mouse pointer.
- static void `Fl::screen_work_area` (int &`X`, int &`Y`, int &`W`, int &`H`, int `mx`, int `my`)
Gets the bounding box of the work area of a screen that contains the specified screen position `mx`, `my`.
- static void `Fl::screen_work_area` (int &`X`, int &`Y`, int &`W`, int &`H`, int `n`)
Gets the bounding box of the work area of the given screen.
- static void `Fl::screen_xywh` (int &`X`, int &`Y`, int &`W`, int &`H`)
Gets the bounding box of a screen that contains the mouse pointer.
- static void `Fl::screen_xywh` (int &`X`, int &`Y`, int &`W`, int &`H`, int `mx`, int `my`)
Gets the bounding box of a screen that contains the specified screen position `mx`, `my`.
- static void `Fl::screen_xywh` (int &`X`, int &`Y`, int &`W`, int &`H`, int `mx`, int `my`, int `mw`, int `mh`)
Gets the screen bounding rect for the screen which intersects the most with the rectangle defined by `mx`, `my`, `mw`, `mh`.
- static void `Fl::screen_xywh` (int &`X`, int &`Y`, int &`W`, int &`H`, int `n`)
Gets the screen bounding rect for the given screen.
- static int `Fl::w` ()
Returns the width in pixels of the main screen work area.
- static int `Fl::x` ()
Returns the leftmost x coordinate of the main screen work area.
- static int `Fl::y` ()
Returns the topmost y coordinate of the main screen work area.

30.5.1 Detailed Description

fl global screen functions declared in <[FL/Fl.H](#)>

30.5.2 Function Documentation

30.5.2.1 screen_dpi()

```
void Fl::screen_dpi (
    float & h,
    float & v,
    int n = 0 ) [static]
```

Gets the screen resolution in dots-per-inch for the given screen.

Parameters

out	<code>h,v</code>	horizontal and vertical resolution
in	<code>n</code>	the screen number (0 to <code>Fl::screen_count()</code> - 1)

See also

void [screen_xywh](#)(int &`x`, int &`y`, int &`w`, int &`h`, int `mx`, int `my`)

30.5.2.2 screen_num() [1/2]

```
int Fl::screen_num (
    int x,
    int y ) [static]
```

Gets the screen number of a screen that contains the specified screen position x , y .

Parameters

in	<i>x,y</i>	the absolute screen position
----	------------	------------------------------

30.5.2.3 screen_num() [2/2]

```
int Fl::screen_num (
    int x,
    int y,
    int w,
    int h ) [static]
```

Gets the screen number for the screen which intersects the most with the rectangle defined by *x*, *y*, *w*, *h*.

Parameters

in	<i>x,y,w,h</i>	the rectangle to search for intersection with
----	----------------	---

30.5.2.4 screen_work_area() [1/3]

```
static void Fl::screen_work_area (
    int & X,
    int & Y,
    int & W,
    int & H ) [inline], [static]
```

Gets the bounding box of the work area of the screen that contains the mouse pointer.

Parameters

out	<i>X,Y,W,H</i>	the work area bounding box
-----	----------------	----------------------------

See also

[void screen_work_area\(int &x, int &y, int &w, int &h, int mx, int my\)](#)

30.5.2.5 screen_work_area() [2/3]

```
void Fl::screen_work_area (
    int & X,
    int & Y,
    int & W,
    int & H,
    int mx,
    int my ) [static]
```

Gets the bounding box of the work area of a screen that contains the specified screen position *mx*, *my*.

Parameters

out	<i>X,Y,W,H</i>	the work area bounding box
in	<i>mx,my</i>	the absolute screen position

30.5.2.6 screen_work_area() [3/3]

```
void Fl::screen_work_area (
    int & X,
    int & Y,
    int & W,
    int & H,
    int n ) [static]
```

Gets the bounding box of the work area of the given screen.

Parameters

out	<i>X,Y,W,H</i>	the work area bounding box
in	<i>n</i>	the screen number (0 to <code>Fl::screen_count()</code> - 1)

See also

void [screen_xywh\(int &x, int &y, int &w, int &h, int mx, int my\)](#)

30.5.2.7 screen_xywh() [1/4]

```
static void Fl::screen_xywh (
    int & X,
    int & Y,
    int & W,
    int & H ) [inline], [static]
```

Gets the bounding box of a screen that contains the mouse pointer.

Parameters

out	<i>X,Y,W,H</i>	the corresponding screen bounding box
-----	----------------	---------------------------------------

See also

void [screen_xywh\(int &x, int &y, int &w, int &h, int mx, int my\)](#)

30.5.2.8 screen_xywh() [2/4]

```
void Fl::screen_xywh (
    int & X,
    int & Y,
    int & W,
    int & H,
    int mx,
    int my ) [static]
```

Gets the bounding box of a screen that contains the specified screen position `mx, my`.

Parameters

out	<i>X,Y,W,H</i>	the corresponding screen bounding box
in	<i>mx,my</i>	the absolute screen position

30.5.2.9 screen_xywh() [3/4]

```
void Fl::screen_xywh (
    int & X,
    int & Y,
    int & W,
    int & H,
    int mx,
    int my,
    int mw,
    int mh ) [static]
```

Gets the screen bounding rect for the screen which intersects the most with the rectangle defined by `mx`, `my`, `mw`, `mh`.

Parameters

out	<i>X,Y,W,H</i>	the corresponding screen bounding box
in	<i>mx,my,mw,mh</i>	the rectangle to search for intersection with

See also

void [screen_xywh\(int &X, int &Y, int &W, int &H, int n\)](#)

30.5.2.10 screen_xywh() [4/4]

```
void Fl::screen_xywh (
    int & X,
    int & Y,
    int & W,
    int & H,
    int n ) [static]
```

Gets the screen bounding rect for the given screen.

Under MSWindows, Mac OS X, and the Gnome desktop, screen #0 contains the menubar/taskbar

Parameters

out	<i>X,Y,W,H</i>	the corresponding screen bounding box
in	<i>n</i>	the screen number (0 to Fl::screen_count() - 1)

See also

void [screen_xywh\(int &x, int &y, int &w, int &h, int mx, int my\)](#)

30.6 Color & Font functions

fl global color, font functions.

Functions

- [Fl_Color fl_color \(\)](#)
Returns the last [fl_color\(\)](#) that was set.
- void [fl_color \(Fl_Color c\)](#)
Sets the color for all subsequent drawing operations.
- void [fl_color \(int c\)](#)
for back compatibility - use [fl_color\(Fl_Color c\)](#) instead

- void `fl_color` (`uchar` r, `uchar` g, `uchar` b)

Sets the color for all subsequent drawing operations.
- `FL_Color` `fl_color_average` (`FL_Color` color1, `FL_Color` color2, float weight)

Returns the weighted average color between the two given colors.
- `FL_Color` `fl_contrast` (`FL_Color` fg, `FL_Color` bg)

Returns a color that contrasts with the background color.
- int `fl_descent` ()

Returns the recommended distance above the bottom of a `fl_height()` tall box to draw the text at so it looks centered vertically in that box.
- `FL_Font` `fl_font` ()

*Returns the *face* set by the most recent call to `fl_font()`.*
- void `fl_font` (`FL_Font` face, `FL_Fontsize` fsize)

Sets the current font, which is then used in various drawing routines.
- int `fl_height` ()

Returns the recommended minimum line spacing for the current font.
- `FL_EXPORT` int `fl_height` (int font, int size)

*This function returns the actual height of the specified *font* and *size*.*
- `FL_Color` `fl_inactive` (`FL_Color` c)

Returns the inactive, dimmed version of the given color.
- `FL_EXPORT` const char * `fl_latin1_to_local` (const char *t, int n=-1)

Converts text from Windows/X11 latin1 character set to local encoding.
- `FL_EXPORT` const char * `fl_local_to_latin1` (const char *t, int n=-1)

Converts text from local encoding to Windows/X11 latin1 character set.
- `FL_EXPORT` const char * `fl_local_to_mac_roman` (const char *t, int n=-1)

Converts text from local encoding to Mac Roman character set.
- `FL_EXPORT` const char * `fl_mac_roman_to_local` (const char *t, int n=-1)

Converts text from Mac Roman character set to local encoding.
- `FL_EXPORT` `FL_Color` `fl_show_colormap` (`FL_Color` oldcol)

Pops up a window to let the user pick a colormap entry.
- `FL_Fontsize` `fl_size` ()

*Returns the *size* set by the most recent call to `fl_font()`.*
- `FL_EXPORT` void `fl_text_extents` (const char *, int &dx, int &dy, int &w, int &h)

Determines the minimum pixel dimensions of a nul-terminated string.
- void `fl_text_extents` (const char *t, int n, int &dx, int &dy, int &w, int &h)

*Determines the minimum pixel dimensions of a sequence of *n* characters.*
- `FL_EXPORT` double `fl_width` (const char *txt)

Returns the typographical width of a nul-terminated string using the current font face and size.
- double `fl_width` (const char *txt, int n)

*Returns the typographical width of a sequence of *n* characters using the current font face and size.*
- double `fl_width` (unsigned int c)

Returns the typographical width of a single character using the current font face and size.
- `ulong` `fl_xpixel` (`FL_Color` i)

Returns the X pixel number used to draw the given FLTK color index.
- `ulong` `fl_xpixel` (`uchar` r, `uchar` g, `uchar` b)

Returns the X pixel number used to draw the given rgb color.
- static void `FL::free_color` (`FL_Color` i, int overlay=0)

Frees the specified color from the colormap, if applicable.
- static unsigned `FL::get_color` (`FL_Color` i)

Returns the RGB value(s) for the given FLTK color index.
- static void `FL::get_color` (`FL_Color` i, `uchar` &red, `uchar` &green, `uchar` &blue)

Returns the RGB value(s) for the given FLTK color index.

- static const char * `Fl::get_font (Fl_Font)`
Gets the string for this face.
- static const char * `Fl::get_font_name (Fl_Font, int *attributes=0)`
Get a human-readable string describing the family of this face.
- static int `Fl::get_font_sizes (Fl_Font, int *&sizep)`
*Return an array of sizes in *sizep*.*
- static void `Fl::set_color (Fl_Color i, unsigned c)`
*Sets an entry in the *fl_color* index table.*
- static void `Fl::set_color (Fl_Color, uchar, uchar, uchar)`
*Sets an entry in the *fl_color* index table.*
- static void `Fl::set_font (Fl_Font, const char *)`
Changes a face.
- static void `Fl::set_font (Fl_Font, Fl_Font)`
Copies one face to another.
- static `Fl_Font Fl::set_fonts (const char **=0)`
FLTK will open the display, and add every fonts on the server to the face table.

30.6.1 Detailed Description

fl global color, font functions.

These functions are declared in [<FL/Fl.H>](#) or [<FL/fl_draw.H>](#).

30.6.2 Function Documentation

30.6.2.1 fl_color() [1/3]

```
Fl_Color fl_color (
    void ) [inline]
```

Returns the last `fl_color()` that was set.

This can be used for state save/restore.

30.6.2.2 fl_color() [2/3]

```
void fl_color (
    Fl_Color c ) [inline]
```

Sets the color for all subsequent drawing operations.

For colormapped displays, a color cell will be allocated out of `fl_colormap` the first time you use a color. If the colormap fills up then a least-squares algorithm is used to find the closest color. If no valid graphical context (`fl_gc`) is available, the foreground is not set for the current window.

Parameters

in	c	color
----	---	-------

30.6.2.3 fl_color() [3/3]

```
void fl_color (
    uchar r,
    uchar g,
    uchar b ) [inline]
```

Sets the color for all subsequent drawing operations.

The closest possible match to the RGB color is used. The RGB color is used directly on TrueColor displays. For

colormap visuals the nearest index in the gray ramp or color cube is used. If no valid graphical context (`fl_gc`) is available, the foreground is not set for the current window.

Parameters

in	<i>r,g,b</i>	color components
----	--------------	------------------

30.6.2.4 fl_color_average()

```
Fl_Color fl_color_average (
    Fl_Color color1,
    Fl_Color color2,
    float weight )
```

Returns the weighted average color between the two given colors.

The red, green and blue values are averages using the following formula:

$$\text{color} = \text{color1} * \text{weight} + \text{color2} * (1 - \text{weight})$$

Thus, a `weight` value of 1.0 will return the first color, while a value of 0.0 will return the second color.

Parameters

in	<i>color1,color2</i>	boundary colors
in	<i>weight</i>	weighting factor

30.6.2.5 fl_contrast()

```
Fl_Color fl_contrast (
    Fl_Color fg,
    Fl_Color bg )
```

Returns a color that contrasts with the background color.

This will be the foreground color if it contrasts sufficiently with the background color. Otherwise, returns `FL_WHITE` or `FL_BLACK` depending on which color provides the best contrast.

Parameters

in	<i>fg,bg</i>	foreground and background colors
----	--------------	----------------------------------

Returns

contrasting color

30.6.2.6 fl_font() [1/2]

```
Fl_Font fl_font (
    void ) [inline]
```

Returns the `face` set by the most recent call to `fl_font()`.

This can be used to save/restore the font.

30.6.2.7 fl_font() [2/2]

```
void fl_font (
    Fl_Font face,
    Fl_Fontsize fsize ) [inline]
```

Sets the current font, which is then used in various drawing routines.

You may call this outside a draw context if necessary to call `fl_width()`, but on X this will open the display.

The font is identified by a `face` and a `size`. The size of the font is measured in pixels and not "points". Lines should be spaced `size` pixels apart or more.

30.6.2.8 `fl_height()` [1/2]

```
int fl_height ( ) [inline]
```

Returns the recommended minimum line spacing for the current font.

You can also use the value of `size` passed to `fl_font()`

30.6.2.9 `fl_height()` [2/2]

```
FL_EXPORT int fl_height (
    int font,
    int size )
```

This function returns the actual height of the specified `font` and `size`.

Normally the font height should always be 'size', but with the advent of XFT, there are (currently) complexities that seem to only be solved by asking the font what its actual font height is. (See STR#2115)

This function was originally undocumented in 1.1.x, and was used only by `FL_Text_Display`. We're now documenting it in 1.3.x so that apps that need precise height info can get it with this function.

Returns

the height of the font in pixels.

Todo In the future, when the XFT issues are resolved, this function should simply return the 'size' value.

30.6.2.10 `fl_latin1_to_local()`

```
FL_EXPORT const char * fl_latin1_to_local (
    const char * t,
    int n = -1 )
```

Converts text from Windows/X11 latin1 character set to local encoding.

Parameters

in	<i>t</i>	character string (latin1 encoding)
in	<i>n</i>	optional number of characters to convert (default is all)

Returns

pointer to internal buffer containing converted characters

30.6.2.11 `fl_local_to_latin1()`

```
FL_EXPORT const char * fl_local_to_latin1 (
    const char * t,
    int n = -1 )
```

Converts text from local encoding to Windows/X11 latin1 character set.

Parameters

in	<i>t</i>	character string (local encoding)
in	<i>n</i>	optional number of characters to convert (default is all)

Returns

pointer to internal buffer containing converted characters

30.6.2.12 fl_local_to_mac_roman()

```
FL_EXPORT const char * fl_local_to_mac_roman (
    const char * t,
    int n = -1 )
```

Converts text from local encoding to Mac Roman character set.

Parameters

in	<i>t</i>	character string (local encoding)
in	<i>n</i>	optional number of characters to convert (default is all)

Returns

pointer to internal buffer containing converted characters

30.6.2.13 fl_mac_roman_to_local()

```
FL_EXPORT const char * fl_mac_roman_to_local (
    const char * t,
    int n = -1 )
```

Converts text from Mac Roman character set to local encoding.

Parameters

in	<i>t</i>	character string (Mac Roman encoding)
in	<i>n</i>	optional number of characters to convert (default is all)

Returns

pointer to internal buffer containing converted characters

30.6.2.14 fl_show_colormap()

```
FL_EXPORT Fl_Color fl_show_colormap (
    Fl_Color oldcol )
```

Pops up a window to let the user pick a colormap entry.



Figure 30.1 fl_show_colormap

Parameters

<code>in</code>	<code>oldcol</code>	color to be highlighted when grid is shown.
-----------------	---------------------	---

Return values

<code>FL_Color</code>	value of the chosen colormap entry.
-----------------------	-------------------------------------

See also

[FL_Color_Chooser](#)

30.6.2.15 fl_size()

```
FL_Fontsize fl_size ( ) [inline]
```

Returns the `size` set by the most recent call to `fl_font()`.

This can be used to save/restore the font.

30.6.2.16 fl_text_extents() [1/2]

```
FL_EXPORT void fl_text_extents (
    const char * c,
    int & dx,
    int & dy,
    int & w,
    int & h )
```

Determines the minimum pixel dimensions of a nul-terminated string.

Usage: given a string "txt" drawn using `fl_draw(txt, x, y)` you would determine its pixel extents on the display using `fl_text_extents(txt, dx, dy, wo, ho)` such that a bounding box that exactly fits around the text could be drawn with

`fl_rect(x+dx, y+dy, wo, ho)`. Note the `dx`, `dy` values hold the offset of the first "colored in" pixel of the string, from the draw origin.

No FLTK symbol expansion will be performed.

30.6.2.17 `fl_text_extents()` [2/2]

```
void fl_text_extents (
    const char * t,
    int n,
    int & dx,
    int & dy,
    int & w,
    int & h ) [inline]
```

Determines the minimum pixel dimensions of a sequence of `n` characters.

See also

[fl_text_extents\(const char*, int& dx, int& dy, int& w, int& h\)](#)

30.6.2.18 `fl_width()`

```
double fl_width (
    unsigned int c ) [inline]
```

Returns the typographical width of a single character using the current font face and size.

Note

if a valid `fl_gc` is NOT found then it uses the first window `gc`, or the screen `gc` if no fltk window is available when called.

30.6.2.19 `fl_xpixel()` [1/2]

```
ulong fl_xpixel (
    Fl_Color i )
```

Returns the X pixel number used to draw the given FLTK color index. This is the X pixel that `fl_color()` would use.

Parameters

<code>in</code>	<code>i</code>	color index
-----------------	----------------	-------------

Returns

X pixel number

30.6.2.20 `fl_xpixel()` [2/2]

```
ulong fl_xpixel (
    uchar r,
    uchar g,
    uchar b )
```

Returns the X pixel number used to draw the given rgb color. This is the X pixel that `fl_color()` would use.

Parameters

in	<i>r,g,b</i>	color components
----	--------------	------------------

Returns

X pixel number

30.6.2.21 free_color()

```
void Fl::free_color (
    Fl_Color i,
    int overlay = 0 ) [static]
```

Frees the specified color from the colormap, if applicable.

Free color *i* if used, and clear mapping table entry.

If *overlay* is non-zero then the color is freed from the overlay colormap.

Parameters

in	<i>i</i>	color index
in	<i>overlay</i>	0 for normal, 1 for overlay color

30.6.2.22 get_color() [1/2]

```
unsigned Fl::get_color (
    Fl_Color i ) [static]
```

Returns the RGB value(s) for the given FLTK color index.

This form returns the RGB values packed in a 32-bit unsigned integer with the red value in the upper 8 bits, the green value in the next 8 bits, and the blue value in bits 8-15. The lower 8 bits will always be 0.

30.6.2.23 get_color() [2/2]

```
void Fl::get_color (
    Fl_Color i,
    uchar & red,
    uchar & green,
    uchar & blue ) [static]
```

Returns the RGB value(s) for the given FLTK color index.

This form returns the red, green, and blue values separately in referenced variables.

See also unsigned [get_color\(Fl_Color c\)](#)

30.6.2.24 get_font()

```
const char * Fl::get_font (
    Fl_Font fnum ) [static]
```

Gets the string for this face.

This string is different for each face. Under X this value is passed to XListFonts to get all the sizes of this face.

30.6.2.25 get_font_name()

```
const char * Fl::get_font_name (
    Fl_Font fnum,
    int * attributes = 0 ) [static]
```

Get a human-readable string describing the family of this face.

This is useful if you are presenting a choice to the user. There is no guarantee that each face has a different name. The return value points to a static buffer that is overwritten each call.

The integer pointed to by `attributes` (if the pointer is not zero) is set to zero, `FL_BOLD` or `FL_ITALIC` or `FL_↔_BOLD | FL_ITALIC`. To locate a "family" of fonts, search forward and back for a set with non-zero attributes, these faces along with the face with a zero attribute before them constitute a family.

30.6.2.26 `get_font_sizes()`

```
int Fl::get_font_sizes (
    Fl_Font fnum,
    int *& sizes ) [static]
```

Return an array of sizes in `sizes`.

The return value is the length of this array. The sizes are sorted from smallest to largest and indicate what sizes can be given to `fl_font()` that will be matched exactly (`fl_font()` will pick the closest size for other sizes). A zero in the first location of the array indicates a scalable font, where any size works, although the array may list sizes that work "better" than others. Warning: the returned array points at a static buffer that is overwritten each call. Under X this will open the display.

30.6.2.27 `set_color()` [1/2]

```
void Fl::set_color (
    Fl_Color i,
    unsigned c ) [static]
```

Sets an entry in the `fl_color` index table.

Set color mapping table entry `i` to color `c`.

You can set it to any 8-bit RGB color. The color is not allocated until `fl_color(i)` is used.

Parameters

in	<i>i</i>	color index
in	<i>c</i>	color

30.6.2.28 `set_color()` [2/2]

```
void Fl::set_color (
    Fl_Color i,
    uchar red,
    uchar green,
    uchar blue ) [static]
```

Sets an entry in the `fl_color` index table.

You can set it to any 8-bit RGB color. The color is not allocated until `fl_color(i)` is used.

30.6.2.29 `set_font()`

```
void Fl::set_font (
    Fl_Font fnum,
    const char * name ) [static]
```

Changes a face.

The string pointer is simply stored, the string is not copied, so the string must be in static memory.

30.6.2.30 `set_fonts()`

```
Fl_Font Fl::set_fonts (
    const char * xstarname = 0 ) [static]
```

FLTK will open the display, and add every fonts on the server to the face table.

It will attempt to put "families" of faces together, so that the normal one is first, followed by bold, italic, and bold italic.

The optional argument is a string to describe the set of fonts to add. Passing NULL will select only fonts that have the ISO8859-1 character set (and are thus usable by normal text). Passing "-*" will select all fonts with any encoding as long as they have normal X font names with dashes in them. Passing "*" will list every font that exists (on X this may produce some strange output). Other values may be useful but are system dependent. With WIN32 NULL selects fonts with ISO8859-1 encoding and non-NULL selects all fonts.

The return value is how many faces are in the table after this is done.

30.7 Drawing functions

FLTK global graphics and GUI drawing functions.

Macros

- `#define fl_clip fl_push_clip`
Intersects the current clip region with a rectangle and pushes this new region onto the stack (deprecated).

Enumerations

- enum {
`FL_SOLID = 0` , `FL_DASH = 1` , `FL_DOT = 2` , `FL_DASHDOT = 3` ,
`FL_DASHDOTDOT = 4` , `FL_CAP_FLAT = 0x100` , `FL_CAP_ROUND = 0x200` , `FL_CAP_SQUARE = 0x300` ,
`FL_JOIN_MITER = 0x1000` , `FL_JOIN_ROUND = 0x2000` , `FL_JOIN_BEVEL = 0x3000` }

Functions

- void `Fl_Quartz_Graphics_Driver::copy_offscreen` (int x, int y, int w, int h, Fl_Offscreen pixmap, int srcx, int srcy)
see fl_copy_offscreen()
- FL_EXPORT int `fl_add_symbol` (const char *name, void(*drawit)(Fl_Color), int scalable)
Adds a symbol to the system.
- void `fl_arc` (double x, double y, double r, double start, double end)
Adds a series of points to the current path on the arc of a circle.
- void `fl_arc` (int x, int y, int w, int h, double a1, double a2)
Draw ellipse sections using integer coordinates.
- void `fl_begin_complex_polygon` ()
Starts drawing a complex filled polygon.
- void `fl_begin_line` ()
Starts drawing a list of lines.
- void `fl_begin_loop` ()
Starts drawing a closed sequence of lines.
- void `fl_begin_offscreen` (Fl_Offscreen ctx)
Send all subsequent drawing commands to this offscreen buffer.
- void `fl_begin_points` ()
Starts drawing a list of points.
- void `fl_begin_polygon` ()
Starts drawing a convex filled polygon.
- FL_EXPORT char `fl_can_do_alpha_blending` ()
Checks whether platform supports true alpha blending for RGBA images.
- FL_EXPORT void `fl_chord` (int x, int y, int w, int h, double a1, double a2)
fl_chord declaration is a place holder - the function does not yet exist
- void `fl_circle` (double x, double y, double r)
fl_circle() is equivalent to fl_arc(x,y,r,0,360), but may be faster.
- int `fl_clip_box` (int x, int y, int w, int h, int &X, int &Y, int &W, int &H)

- Intersects the rectangle with the current clip region and returns the bounding box of the result.*

 - `FL_Region fl_clip_region ()`
Returns the current clipping region.
 - `void fl_clip_region (FL_Region r)`
Replaces the top of the clipping stack with a clipping region of any shape.
 - `void fl_copy_offscreen (int x, int y, int w, int h, FL_Offscreen pixmap, int srcx, int srcy)`
Copy a rectangular area of the given offscreen buffer into the current drawing destination.
 - `FL_Offscreen fl_create_offscreen (int w, int h)`
Creation of an offscreen graphics buffer.
 - `FL_EXPORT void fl_cursor (FL_Cursor)`
Sets the cursor for the current window to the specified shape and colors.
 - `FL_EXPORT void fl_cursor (FL_Cursor, FL_Color fg, FL_Color bg=FL_WHITE)`
 - `void fl_curve (double X0, double Y0, double X1, double Y1, double X2, double Y2, double X3, double Y3)`
Adds a series of points on a Bezier curve to the path.
 - `void fl_delete_offscreen (FL_Offscreen ctx)`
Deletion of an offscreen graphics buffer.
 - `void fl_draw (const char *str, int n, int x, int y)`
Draws starting at the given x, y location a UTF-8 string of length n bytes.
 - `FL_EXPORT void fl_draw (const char *str, int x, int y)`
Draws a nul-terminated UTF-8 string starting at the given x, y location.
 - `FL_EXPORT void fl_draw (const char *str, int x, int y, int w, int h, FL_Align align, FL_Image *img=0, int draw↔_symbols=1)`
Fancy string drawing function which is used to draw all the labels.
 - `FL_EXPORT void fl_draw (const char *str, int x, int y, int w, int h, FL_Align align, void(*callthis)(const char *, int, int, int), FL_Image *img=0, int draw_symbols=1)`
The same as fl_draw(const char,int,int,int,int,FL_Align,FL_Image*,int) with the addition of the callthis parameter, which is a pointer to a text drawing function such as fl_draw(const char*, int, int, int) to do the real work.*
 - `void fl_draw (int angle, const char *str, int n, int x, int y)`
Draws at the given x, y location a UTF-8 string of length n bytes rotating angle degrees counter-clockwise.
 - `FL_EXPORT void fl_draw (int angle, const char *str, int x, int y)`
Draws a nul-terminated UTF-8 string starting at the given x, y location and rotating angle degrees counter-clockwise.
 - `FL_EXPORT void fl_draw_box (FL_Boxtype, int x, int y, int w, int h, FL_Color)`
Draws a box using given type, position, size and color.
 - `void fl_draw_image (const uchar *buf, int X, int Y, int W, int H, int D=3, int L=0)`
Draws an 8-bit per color RGB or luminance image.
 - `void fl_draw_image (FL_Draw_Image_Cb cb, void *data, int X, int Y, int W, int H, int D=3)`
Draws an image using a callback function to generate image data.
 - `void fl_draw_image_mono (const uchar *buf, int X, int Y, int W, int H, int D=1, int L=0)`
Draws a gray-scale (1 channel) image.
 - `void fl_draw_image_mono (FL_Draw_Image_Cb cb, void *data, int X, int Y, int W, int H, int D=1)`
Draws a gray-scale image using a callback function to generate image data.
 - `FL_EXPORT int fl_draw_pixmap (char *const *data, int x, int y, FL_Color=FL_GRAY)`
Draw XPM image data, with the top-left corner at the given position.
 - `FL_EXPORT int fl_draw_pixmap (const char *const *cdata, int x, int y, FL_Color=FL_GRAY)`
Draw XPM image data, with the top-left corner at the given position.
 - `FL_EXPORT int fl_draw_symbol (const char *label, int x, int y, int w, int h, FL_Color)`
Draw the named symbol in the given rectangle using the given color.
 - `void fl_end_complex_polygon ()`
Ends complex filled polygon, and draws.
 - `void fl_end_line ()`

- Ends list of lines, and draws.*

 - void **fl_end_loop** ()

Ends closed sequence of lines, and draws.
- void **fl_end_offscreen** ()

Quit sending drawing commands to the current offscreen buffer.
- void **fl_end_points** ()

Ends list of points, and draws.
- void **fl_end_polygon** ()

Ends convex filled polygon, and draws.
- FL_EXPORT const char * **fl_expand_text** (const char *from, char *buf, int maxbuf, double maxw, int &n, double &width, int wrap, int draw_symbols=0)

Copy from to buf, replacing control characters with ^X.
- FL_EXPORT void **fl_frame** (const char *s, int x, int y, int w, int h)

Draws a series of line segments around the given box.
- FL_EXPORT void **fl_frame2** (const char *s, int x, int y, int w, int h)

Draws a series of line segments around the given box.
- void **fl_gap** ()

Call fl_gap() to separate loops of the path.
- void **fl_line** (int x, int y, int x1, int y1)

Draws a line from (x,y) to (x1,y1)
- void **fl_line** (int x, int y, int x1, int y1, int x2, int y2)

Draws a line from (x,y) to (x1,y1) and another from (x1,y1) to (x2,y2)
- void **fl_line_style** (int style, int width=0, char *dashes=0)

Sets how to draw lines (the "pen").
- void **fl_loop** (int x, int y, int x1, int y1, int x2, int y2)

Outlines a 3-sided polygon with lines.
- void **fl_loop** (int x, int y, int x1, int y1, int x2, int y2, int x3, int y3)

Outlines a 4-sided polygon with lines.
- FL_EXPORT void **fl_measure** (const char *str, int &x, int &y, int draw_symbols=1)

Measure how wide and tall the string will be when printed by the fl_draw() function with align parameter.
- FL_EXPORT int **fl_measure_pixmap** (char *const *data, int &w, int &h)

Get the dimensions of a pixmap.
- FL_EXPORT int **fl_measure_pixmap** (const char *const *cdata, int &w, int &h)

Get the dimensions of a pixmap.
- void **fl_mult_matrix** (double a, double b, double c, double d, double x, double y)

Concatenates another transformation onto the current one.
- int **fl_not_clipped** (int x, int y, int w, int h)

Does the rectangle intersect the current clip region?
- FL_EXPORT unsigned int **fl_old_shortcut** (const char *s)

Emulation of XForms named shortcuts.
- FL_EXPORT void **fl_overlay_clear** ()

Erase a selection rectangle without drawing a new one.
- FL_EXPORT void **fl_overlay_rect** (int x, int y, int w, int h)

Draws a selection rectangle, erasing a previous one by XOR'ing it first.
- void **fl_pie** (int x, int y, int w, int h, double a1, double a2)

Draw filled ellipse sections using integer coordinates.
- void **fl_point** (int x, int y)

Draws a single pixel at the given coordinates.
- void **fl_polygon** (int x, int y, int x1, int y1, int x2, int y2)

Fills a 3-sided polygon.
- void **fl_polygon** (int x, int y, int x1, int y1, int x2, int y2, int x3, int y3)

- Fills a 4-sided polygon.*

 - void `fl_pop_clip` ()

Restores the previous clip region.
- void `fl_pop_matrix` ()

Restores the current transformation matrix from the stack.
- void `fl_push_clip` (int x, int y, int w, int h)

Intersects the current clip region with a rectangle and pushes this new region onto the stack.
- void `fl_push_matrix` ()

Saves the current transformation matrix on the stack.
- void `fl_push_no_clip` ()

Pushes an empty clip region onto the stack so nothing will be clipped.
- FL_EXPORT `uchar * fl_read_image` (`uchar *p`, int X, int Y, int W, int H, int alpha=0)

Reads an RGB(A) image from the current window or off-screen buffer.
- void `fl_rect` (int x, int y, int w, int h)

Draws a 1-pixel border inside the given bounding box.
- void `fl_rect` (int x, int y, int w, int h, `FL_Color` c)

Draws with passed color a 1-pixel border inside the given bounding box.
- void `fl_rectf` (int x, int y, int w, int h)

Colors with current color a rectangle that exactly fills the given bounding box.
- void `fl_rectf` (int x, int y, int w, int h, `FL_Color` c)

Colors with passed color a rectangle that exactly fills the given bounding box.
- FL_EXPORT void `fl_rectf` (int x, int y, int w, int h, `uchar` r, `uchar` g, `uchar` b)

*Colors a rectangle with "exactly" the passed *r, g, b* color.*
- FL_EXPORT void `fl_reset_spot` (void)
- void `fl_restore_clip` ()

Undoes any clobbering of clip done by your program.
- void `fl_rotate` (double d)

Concatenates rotation transformation onto the current one.
- void `fl_rtl_draw` (const char *str, int n, int x, int y)

*Draws a UTF-8 string of length *n* bytes right to left starting at the given *x, y* location.*
- void `fl_scale` (double x)

Concatenates scaling transformation onto the current one.
- void `fl_scale` (double x, double y)

Concatenates scaling transformation onto the current one.
- FL_EXPORT void `fl_scroll` (int X, int Y, int W, int H, int dx, int dy, void(*draw_area)(void *, int, int, int, int), void *data)

Scroll a rectangle and draw the newly exposed portions.
- FL_EXPORT void `fl_set_spot` (int font, int size, int X, int Y, int W, int H, `FL_Window` *win=0)
- FL_EXPORT void `fl_set_status` (int X, int Y, int W, int H)
- FL_EXPORT const char * `fl_shortcut_label` (unsigned int shortcut)

Get a human-readable string from a shortcut value.
- FL_EXPORT const char * `fl_shortcut_label` (unsigned int shortcut, const char **eom)

Get a human-readable string from a shortcut value.
- double `fl_transform_dx` (double x, double y)

Transforms distance using current transformation matrix.
- double `fl_transform_dy` (double x, double y)

Transforms distance using current transformation matrix.
- double `fl_transform_x` (double x, double y)

Transforms coordinate using the current transformation matrix.
- double `fl_transform_y` (double x, double y)

Transforms coordinate using the current transformation matrix.

- void `fl_transformed_vertex` (double xf, double yf)
Adds coordinate pair to the vertex list without further transformations.
- void `fl_translate` (double x, double y)
Concatenates translation transformation onto the current one.
- void `fl_vertex` (double x, double y)
Adds a single vertex to the current path.
- void `fl_xyline` (int x, int y, int x1)
Draws a horizontal line from (x,y) to (x1,y)
- void `fl_xyline` (int x, int y, int x1, int y2)
Draws a horizontal line from (x,y) to (x1,y), then vertical from (x1,y) to (x1,y2)
- void `fl_xyline` (int x, int y, int x1, int y2, int x3)
Draws a horizontal line from (x,y) to (x1,y), then a vertical from (x1,y) to (x1,y2) and then another horizontal from (x1,y2) to (x3,y2)
- void `fl_yxline` (int x, int y, int y1)
Draws a vertical line from (x,y) to (x,y1)
- void `fl_yxline` (int x, int y, int y1, int x2)
Draws a vertical line from (x,y) to (x,y1), then a horizontal from (x,y1) to (x2,y1)
- void `fl_yxline` (int x, int y, int y1, int x2, int y3)
Draws a vertical line from (x,y) to (x,y1) then a horizontal from (x,y1) to (x2,y1), then another vertical from (x2,y1) to (x2,y3)

Variables

- const int `stack_max` = 16

30.7.1 Detailed Description

FLTK global graphics and GUI drawing functions.

These functions are declared in `<FL/fl_draw.H>`, and in `<FL/x.H>` for offscreen buffer-related ones.

30.7.2 Macro Definition Documentation

30.7.2.1 fl_clip

```
#define fl_clip fl_push_clip
```

Intersects the current clip region with a rectangle and pushes this new region onto the stack (deprecated).

Parameters

in	x,y,w,h	position and size
----	---------	-------------------

Deprecated `fl_clip(int, int, int, int)` is deprecated and will be removed from future releases. Please use `fl_push_clip(int x, int y, int w, int h)` instead.

30.7.3 Enumeration Type Documentation

30.7.3.1 anonymous enum

```
anonymous enum
```

Enumerator

FL_SOLID	line style: _____
FL_DASH	line style: _ _ _ _ _ _
FL_DOT	line style:
FL_DASHDOT	line style: _ . _ . _ .
FL_DASHDOTDOT	line style: _ . . _ . .
FL_CAP_FLAT	cap style: end is flat
FL_CAP_ROUND	cap style: end is round
FL_CAP_SQUARE	cap style: end wraps end point
FL_JOIN_MITER	join style: line join extends to a point
FL_JOIN_ROUND	join style: line join is rounded
FL_JOIN_BEVEL	join style: line join is tidied

30.7.4 Function Documentation

30.7.4.1 `copy_offscreen()`

```
void Fl_Quartz_Graphics_Driver::copy_offscreen (
    int x,
    int y,
    int w,
    int h,
    Fl_Offscreen pixmap,
    int srcx,
    int srcy ) [virtual]
```

see [fl_copy_offscreen\(\)](#)

Reimplemented from [Fl_Graphics_Driver](#).

30.7.4.2 `fl_add_symbol()`

```
FL_EXPORT int fl_add_symbol (
    const char * name,
    void(*) (Fl_Color) drawit,
    int scalable )
```

Adds a symbol to the system.

Parameters

in	<i>name</i>	name of symbol (without the "@")
in	<i>drawit</i>	function to draw symbol
in	<i>scalable</i>	set to 1 if <i>drawit</i> uses scalable vector drawing

Returns

1 on success, 0 on failure

30.7.4.3 `fl_arc()` [1/2]

```
void fl_arc (
    double x,
```

```

double y,
double r,
double start,
double end ) [inline]

```

Adds a series of points to the current path on the arc of a circle.

You can get elliptical paths by using `scale` and `rotate` before calling `fl_arc()`.

Parameters

in	<i>x,y,r</i>	center and radius of circular arc
in	<i>start,end</i>	angles of start and end of arc measured in degrees counter-clockwise from 3 o'clock. If <code>end</code> is less than <code>start</code> then it draws the arc in a clockwise direction.

Examples:

```

// Draw an arc of points
fl_begin_points();
fl_arc(100.0, 100.0, 50.0, 0.0, 180.0);
fl_end_points();
// Draw arc with a line
fl_begin_line();
fl_arc(200.0, 100.0, 50.0, 0.0, 180.0);
fl_end_line();
// Draw filled arc
fl_begin_polygon();
fl_arc(300.0, 100.0, 50.0, 0.0, 180.0);
fl_end_polygon();

```

30.7.4.4 fl_arc() [2/2]

```

void fl_arc (
    int x,
    int y,
    int w,
    int h,
    double a1,
    double a2 ) [inline]

```

Draw ellipse sections using integer coordinates.

These functions match the rather limited circle drawing code provided by X and WIN32. The advantage over using `fl_arc` with floating point coordinates is that they are faster because they often use the hardware, and they draw much nicer small circles, since the small sizes are often hard-coded bitmaps.

If a complete circle is drawn it will fit inside the passed bounding box. The two angles are measured in degrees counter-clockwise from 3 o'clock and are the starting and ending angle of the arc, `a2` must be greater or equal to `a1`.

`fl_arc()` draws a series of lines to approximate the arc. Notice that the integer version of `fl_arc()` has a different number of arguments than the double version `fl_arc(double x, double y, double r, double start, double end)`

Parameters

in	<i>x,y,w,h</i>	bounding box of complete circle
in	<i>a1,a2</i>	start and end angles of arc measured in degrees counter-clockwise from 3 o'clock. <code>a2</code> must be greater than or equal to <code>a1</code> .

30.7.4.5 fl_begin_complex_polygon()

```

void fl_begin_complex_polygon ( ) [inline]

```

Starts drawing a complex filled polygon.

The polygon may be concave, may have holes in it, or may be several disconnected pieces. Call `fl_gap()` to separate loops of the path.

To outline the polygon, use `fl_begin_loop()` and replace each `fl_gap()` with `fl_end_loop();fl_begin_loop()` pairs.

Note

For portability, you should only draw polygons that appear the same whether "even/odd" or "non-zero" winding rules are used to fill them. Holes should be drawn in the opposite direction to the outside loop.

30.7.4.6 fl_begin_offscreen()

```
void fl_begin_offscreen (
    Fl_Offscreen ctx )
```

Send all subsequent drawing commands to this offscreen buffer.

Parameters

<code>ctx</code>	the offscreen buffer.
------------------	-----------------------

30.7.4.7 fl_begin_points()

```
void fl_begin_points ( ) [inline]
```

Starts drawing a list of points.

Points are added to the list with [fl_vertex\(\)](#)

30.7.4.8 fl_can_do_alpha_blending()

```
FL_EXPORT char fl_can_do_alpha_blending ( )
```

Checks whether platform supports true alpha blending for RGBA images.

Returns

1 if true alpha blending supported by platform

0 not supported so FLTK will use screen door transparency

30.7.4.9 fl_circle()

```
void fl_circle (
    double x,
    double y,
    double r ) [inline]
```

[fl_circle\(\)](#) is equivalent to [fl_arc\(x,y,r,0,360\)](#), but may be faster.

It must be the *only* thing in the path: if you want a circle as part of a complex polygon you must use [fl_arc\(\)](#)

Parameters

in	<code>x,y,r</code>	center and radius of circle
----	--------------------	-----------------------------

30.7.4.10 fl_clip_box()

```
int fl_clip_box (
    int x,
    int y,
    int w,
    int h,
    int & X,
```

```

    int & Y,
    int & W,
    int & H ) [inline]

```

Intersects the rectangle with the current clip region and returns the bounding box of the result.

Returns non-zero if the resulting rectangle is different to the original. This can be used to limit the necessary drawing to a rectangle. W and H are set to zero if the rectangle is completely outside the region.

Parameters

in	<i>x,y,w,h</i>	position and size of rectangle
out	<i>X,Y,W,H</i>	position and size of resulting bounding box.

Returns

Non-zero if the resulting rectangle is different to the original.

30.7.4.11 fl_clip_region()

```

void fl_clip_region (
    Fl_Region r ) [inline]

```

Replaces the top of the clipping stack with a clipping region of any shape.

Fl_Region is an operating system specific type.

Parameters

in	<i>r</i>	clipping region
----	----------	-----------------

30.7.4.12 fl_copy_offscreen()

```

void fl_copy_offscreen (
    int x,
    int y,
    int w,
    int h,
    Fl_Offscreen pixmap,
    int srcx,
    int srcy )

```

Copy a rectangular area of the given offscreen buffer into the current drawing destination.

Parameters

<i>x,y</i>	position where to draw the copied rectangle
<i>w,h</i>	size of the copied rectangle
<i>pixmap</i>	offscreen buffer containing the rectangle to copy
<i>srcx,srcy</i>	origin in offscreen buffer of rectangle to copy

30.7.4.13 fl_create_offscreen()

```

Fl_Offscreen fl_create_offscreen (
    int w,
    int h )

```

Creation of an offscreen graphics buffer.

Parameters

<code>w,h</code>	width and height in pixels of the buffer.
------------------	---

Returns

the created graphics buffer.

30.7.4.14 `fl_cursor()`

```
FL_EXPORT void fl_cursor (
    Fl_Cursor c )
```

Sets the cursor for the current window to the specified shape and colors. The cursors are defined in the [<FL/Enumerations.H>](#) header file.

30.7.4.15 `fl_curve()`

```
void fl_curve (
    double X0,
    double Y0,
    double X1,
    double Y1,
    double X2,
    double Y2,
    double X3,
    double Y3 ) [inline]
```

Adds a series of points on a Bezier curve to the path. The curve ends (and two of the points) are at X0,Y0 and X3,Y3.

Parameters

in	<code>X0,Y0</code>	curve start point
in	<code>X1,Y1</code>	curve control point
in	<code>X2,Y2</code>	curve control point
in	<code>X3,Y3</code>	curve end point

30.7.4.16 `fl_delete_offscreen()`

```
void fl_delete_offscreen (
    Fl_Offscreen ctx )
```

Deletion of an offscreen graphics buffer.

Parameters

<code>ctx</code>	the buffer to be deleted.
------------------	---------------------------

30.7.4.17 `fl_draw()` [1/4]

```
FL_EXPORT void fl_draw (
    const char * str,
```

```

    int x,
    int y )

```

Draws a nul-terminated UTF-8 string starting at the given `x`, `y` location.

Text is aligned to the left and to the baseline of the font. To align to the bottom, subtract `fl_descent()` from `y`. To align to the top, subtract `fl_descent()` and add `fl_height()`. This version of `fl_draw` provides direct access to the text drawing function of the underlying OS. It does not apply any special handling to control characters.

30.7.4.18 `fl_draw()` [2/4]

```

FL_EXPORT void fl_draw (
    const char * str,
    int x,
    int y,
    int w,
    int h,
    Fl_Align align,
    Fl_Image * img,
    int draw_symbols )

```

Fancy string drawing function which is used to draw all the labels.

The string is formatted and aligned inside the passed box. Handles `\t` and `\n`, expands all other control characters to `^X`, and aligns inside or against the edges of the box. See `Fl_Widget::align()` for values of `align`. The value `FL_ALIGN_INSIDE` is ignored, as this function always prints inside the box. If `img` is provided and is not `NULL`, the image is drawn above or below the text as specified by the `align` value. The `draw_symbols` argument specifies whether or not to look for symbol names starting with the `'@'` character

30.7.4.19 `fl_draw()` [3/4]

```

void fl_draw (
    int angle,
    const char * str,
    int n,
    int x,
    int y ) [inline]

```

Draws at the given `x`, `y` location a UTF-8 string of length `n` bytes rotating `angle` degrees counter-clockwise.

Note

When using X11 (Unix, Linux, Cygwin et al.) this needs Xft to work. Under plain X11 (w/o Xft) rotated text is not supported by FLTK. A warning will be issued to `stderr` at runtime (only once) if you use this method with an angle other than 0.

30.7.4.20 `fl_draw()` [4/4]

```

FL_EXPORT void fl_draw (
    int angle,
    const char * str,
    int x,
    int y )

```

Draws a nul-terminated UTF-8 string starting at the given `x`, `y` location and rotating `angle` degrees counter-clockwise.

This version of `fl_draw` provides direct access to the text drawing function of the underlying OS and is supported by Xft, Win32 and MacOS fltk subsets.

30.7.4.21 `fl_draw_box()`

```

FL_EXPORT void fl_draw_box (
    Fl_Boxtype t,
    int x,

```

```

    int y,
    int w,
    int h,
    Fl_Color c )

```

Draws a box using given type, position, size and color.

Parameters

in	<i>t</i>	box type
in	<i>x,y,w,h</i>	position and size
in	<i>c</i>	color

30.7.4.22 fl_draw_image() [1/2]

```

void fl_draw_image (
    const uchar * buf,
    int X,
    int Y,
    int W,
    int H,
    int D = 3,
    int L = 0 ) [inline]

```

Draws an 8-bit per color RGB or luminance image.

Parameters

in	<i>buf</i>	points at the "r" data of the top-left pixel. Color data must be in <i>r, g, b</i> order. Luminance data is only one <i>gray</i> byte.
in	<i>X,Y</i>	position where to put top-left corner of image
in	<i>W,H</i>	size of the image
in	<i>D</i>	delta to add to the pointer between pixels. It may be any value greater than or equal to 1, or it can be negative to flip the image horizontally
in	<i>L</i>	delta to add to the pointer between lines (if 0 is passed it uses $\bar{W} * D$), and may be larger than $\bar{W} * D$ to crop data, or negative to flip the image vertically

It is highly recommended that you put the following code before the first `show()` of any window in your program to get rid of the dithering if possible:

```
Fl::visual(FL_RGB);
```

Gray scale (1-channel) images may be drawn. This is done if `abs(D)` is less than 3, or by calling `fl_draw_image_mono()`. Only one 8-bit sample is used for each pixel, and on screens with different numbers of bits for red, green, and blue only gray colors are used. Setting `D` greater than 1 will let you display one channel of a color image.

Note:

The X version does not support all possible visuals. If FLTK cannot draw the image in the current visual it will abort. FLTK supports any visual of 8 bits or less, and all common TrueColor visuals up to 32 bits.

30.7.4.23 fl_draw_image() [2/2]

```

void fl_draw_image (
    Fl_Draw_Image_Cb cb,
    void * data,
    int X,

```

```

    int Y,
    int W,
    int H,
    int D = 3 ) [inline]

```

Draws an image using a callback function to generate image data.

You can generate the image as it is being drawn, or do arbitrary decompression of stored data, provided it can be decompressed to individual scan lines easily.

Parameters

in	<i>cb</i>	callback function to generate scan line data
in	<i>data</i>	user data passed to callback function
in	<i>X,Y</i>	screen position of top left pixel
in	<i>W,H</i>	image width and height
in	<i>D</i>	data size in bytes (must be greater than 0)

See also

[fl_draw_image\(const uchar* buf, int X,int Y,int W,int H, int D, int L\)](#)

The callback function *cb* is called with the `void*` *data* user data pointer to allow access to a structure of information about the image, and the *x*, *y*, and *w* of the scan line desired from the image. 0,0 is the upper-left corner of the image, not *x*, *y*. A pointer to a buffer to put the data into is passed. You must copy *w* pixels from scanline *y*, starting at pixel *x*, to this buffer.

Due to cropping, less than the whole image may be requested. So *x* may be greater than zero, the first *y* may be greater than zero, and *w* may be less than *W*. The buffer is long enough to store the entire *W* * *D* pixels, this is for convenience with some decompression schemes where you must decompress the entire line at once: decompress it into the buffer, and then if *x* is not zero, copy the data over so the *x*'th pixel is at the start of the buffer.

You can assume the *y*'s will be consecutive, except the first one may be greater than zero.

If *D* is 4 or more, you must fill in the unused bytes with zero.

30.7.4.24 fl_draw_image_mono() [1/2]

```

void fl_draw_image_mono (
    const uchar * buf,
    int X,
    int Y,
    int W,
    int H,
    int D = 1,
    int L = 0 ) [inline]

```

Draws a gray-scale (1 channel) image.

See also

[fl_draw_image\(const uchar* buf, int X,int Y,int W,int H, int D, int L\)](#)

30.7.4.25 fl_draw_image_mono() [2/2]

```

void fl_draw_image_mono (
    Fl_Draw_Image_Cb cb,
    void * data,
    int X,
    int Y,
    int W,
    int H,
    int D = 1 ) [inline]

```

Draws a gray-scale image using a callback function to generate image data.

See also

[fl_draw_image\(Fl_Draw_Image_Cb cb, void* data, int X,int Y,int W,int H, int D\)](#)

30.7.4.26 fl_draw_pixmap() [1/2]

```
FL_EXPORT int fl_draw_pixmap (
    char *const * data,
    int x,
    int y,
    Fl_Color bg )
```

Draw XPM image data, with the top-left corner at the given position.

The image is dithered on 8-bit displays so you won't lose color space for programs displaying both images and pixmaps.

Parameters

in	<i>data</i>	pointer to XPM image data
in	<i>x,y</i>	position of top-left corner
in	<i>bg</i>	background color

Returns

0 if there was any error decoding the XPM data.

30.7.4.27 fl_draw_pixmap() [2/2]

```
FL_EXPORT int fl_draw_pixmap (
    const char *const * cdata,
    int x,
    int y,
    Fl_Color bg )
```

Draw XPM image data, with the top-left corner at the given position.

See also

[fl_draw_pixmap\(char* const* data, int x, int y, Fl_Color bg\)](#)

30.7.4.28 fl_draw_symbol()

```
FL_EXPORT int fl_draw_symbol (
    const char * label,
    int x,
    int y,
    int w,
    int h,
    Fl_Color col )
```

Draw the named symbol in the given rectangle using the given color.

Parameters

in	<i>label</i>	name of symbol
in	<i>x,y</i>	position of symbol
in	<i>w,h</i>	size of symbol
in	<i>col</i>	color of symbol

Returns

1 on success, 0 on failure

30.7.4.29 fl_expand_text()

```
FL_EXPORT const char * fl_expand_text (
    const char * from,
    char * buf,
    int maxbuf,
    double maxw,
    int & n,
    double & width,
    int wrap,
    int draw_symbols )
```

Copy *from* to *buf*, replacing control characters with ^X.

Stop at a newline or if *maxbuf* characters written to buffer. Also word-wrap if *width* exceeds *maxw*. Returns a pointer to the start of the next line of characters. Sets *n* to the number of characters put into the buffer. Sets *width* to the width of the string in the [current font](#).

30.7.4.30 fl_frame()

```
FL_EXPORT void fl_frame (
    const char * s,
    int x,
    int y,
    int w,
    int h )
```

Draws a series of line segments around the given box.

The string *s* must contain groups of 4 letters which specify one of 24 standard grayscale values, where 'A' is black and 'X' is white. The order of each set of 4 characters is: top, left, bottom, right. The result of calling [fl_frame\(\)](#) with a string that is not a multiple of 4 characters in length is undefined. The only difference between this function and [fl_frame2\(\)](#) is the order of the line segments.

Parameters

in	<i>s</i>	sets of 4 grayscale values in top, left, bottom, right order
in	<i>x,y,w,h</i>	position and size

30.7.4.31 fl_frame2()

```
FL_EXPORT void fl_frame2 (
    const char * s,
    int x,
    int y,
    int w,
    int h )
```

Draws a series of line segments around the given box.

The string *s* must contain groups of 4 letters which specify one of 24 standard grayscale values, where 'A' is black and 'X' is white. The order of each set of 4 characters is: bottom, right, top, left. The result of calling [fl_frame2\(\)](#) with a string that is not a multiple of 4 characters in length is undefined. The only difference between this function and [fl_frame\(\)](#) is the order of the line segments.

Parameters

in	<i>s</i>	sets of 4 grayscale values in bottom, right, top, left order
----	----------	--

Parameters

in	<i>x,y,w,h</i>	position and size
----	----------------	-------------------

30.7.4.32 fl_gap()

```
void fl_gap ( ) [inline]
```

Call `fl_gap()` to separate loops of the path.

It is unnecessary but harmless to call `fl_gap()` before the first vertex, after the last vertex, or several times in a row.

30.7.4.33 fl_line_style()

```
void fl_line_style (
    int style,
    int width = 0,
    char * dashes = 0 ) [inline]
```

Sets how to draw lines (the "pen").

If you change this it is your responsibility to set it back to the default using `fl_line_style(0)`.

Parameters

in	<i>style</i>	A bitmask which is a bitwise-OR of a line style, a cap style, and a join style. If you don't specify a dash type you will get a solid line. If you don't specify a cap or join type you will get a system-defined default of whatever value is fastest.
in	<i>width</i>	The thickness of the lines in pixels. Zero results in the system defined default, which on both X and Windows is somewhat different and nicer than 1.
in	<i>dashes</i>	A pointer to an array of dash lengths, measured in pixels. The first location is how long to draw a solid portion, the next is how long to draw the gap, then the solid, etc. It is terminated with a zero-length entry. A NULL pointer or a zero-length array results in a solid line. Odd array sizes are not supported and result in undefined behavior.

Note

Because of how line styles are implemented on Win32 systems, you *must* set the line style *after* setting the drawing color. If you set the color after the line style you will lose the line style settings.

The *dashes* array does not work under Windows 95, 98 or Me, since those operating systems do not support complex line styles.

30.7.4.34 fl_measure()

```
FL_EXPORT void fl_measure (
    const char * str,
    int & w,
    int & h,
    int draw_symbols )
```

Measure how wide and tall the string will be when printed by the `fl_draw()` function with `align` parameter.

If the incoming `w` is non-zero it will wrap to that width.

The `current font` is used to do the width/height calculations, so unless its value is known at the time `fl_measure()` is called, it is advised to first set the current font with `fl_font()`. With event-driven GUI programming you can never be sure which widget was exposed and redrawn last, nor which font it used. If you have not called `fl_font()` explicitly in your own code, the width and height may be set to unexpected values, even zero!

Note: In the general use case, it's a common error to forget to set `w` to 0 before calling `fl_measure()` when wrap behavior isn't needed.

Parameters

in	<i>str</i>	nul-terminated string
out	<i>w,h</i>	width and height of string in current font
in	<i>draw_symbols</i>	non-zero to enable @symbol handling [default=1]

```
// Example: Common use case for fl_measure()
const char *s = "This is a test";
int wi=0, hi=0;           // initialize to zero before calling fl_measure()
fl_font(FL_HELVETICA, 14); // set current font face/size to be used for measuring
fl_measure(s, wi, hi);    // returns pixel width/height of string in current font
```

30.7.4.35 fl_measure_pixmap() [1/2]

```
FL_EXPORT int fl_measure_pixmap (
    char *const * data,
    int & w,
    int & h )
```

Get the dimensions of a pixmap.

An XPM image contains the dimensions in its data. This function returns the width and height.

Parameters

in	<i>data</i>	pointer to XPM image data.
out	<i>w,h</i>	width and height of image

Returns

non-zero if the dimensions were parsed OK

0 if there were any problems

30.7.4.36 fl_measure_pixmap() [2/2]

```
FL_EXPORT int fl_measure_pixmap (
    const char *const * cdata,
    int & w,
    int & h )
```

Get the dimensions of a pixmap.

See also

[fl_measure_pixmap\(char* const* data, int &w, int &h\)](#)

30.7.4.37 fl_mult_matrix()

```
void fl_mult_matrix (
    double a,
    double b,
    double c,
    double d,
    double x,
    double y ) [inline]
```

Concatenates another transformation onto the current one.

Parameters

in	a,b,c,d,x,y	transformation matrix elements such that $X' = aX + cY + x$ and $Y' = bX + dY + y$
----	---------------	--

30.7.4.38 fl_not_clipped()

```
int fl_not_clipped (
    int x,
    int y,
    int w,
    int h ) [inline]
```

Does the rectangle intersect the current clip region?

Parameters

in	x,y,w,h	position and size of rectangle
----	-----------	--------------------------------

Returns

non-zero if any of the rectangle intersects the current clip region. If this returns 0 you don't have to draw the object.

Note

Under X this returns 2 if the rectangle is partially clipped, and 1 if it is entirely inside the clip region.

30.7.4.39 fl_old_shortcut()

```
FL_EXPORT unsigned int fl_old_shortcut (
    const char * s )
```

Emulation of XForms named shortcuts.

Converts ascii shortcut specifications (eg. " $\wedge c$ ") into the FLTK integer equivalent (eg. FL_CTRL+'c')

These ascii characters are used to specify the various keyboard modifier keys:

```
# - Alt
+ - Shift
^ - Control
! - Meta
@ - Command (Ctrl on linux/win, Meta on OSX)
```

These special characters can be combined to form chords of modifier keys. (See 'Remarks' below)

After the optional modifier key prefixes listed above, one can either specify a single keyboard character to use as the shortcut, or a numeric sequence in hex, decimal or octal.

Examples:

```
"c"      -- Uses 'c' as the shortcut
"#^c"    -- Same as FL_ALT|FL_CTRL|'c'
"#^!c"   -- Same as FL_ALT|FL_CTRL|FL_META|'c'
"@c"     -- Same as FL_COMMAND|'c' (see FL_COMMAND for platform specific behavior)
"0x63"   -- Same as "c" (hex 63=='c')
"99"     -- Same as "c" (dec 99=='c')
"0143"   -- Same as "c" (octal 0143=='c')
"^0x63"  -- Same as (FL_CTRL|'c'), or (FL_CTRL|0x63)
"^99"    -- Same as (FL_CTRL|'c'), or (FL_CTRL|99)
"^0143"  -- Same as (FL_CTRL|'c'), or (FL_CTRL|0143)
```

Remarks

Due to XForms legacy, there are some odd things to consider when using the modifier characters.

(1) You can use the special modifier keys for chords *only* if the modifiers are provided in this order: #, +, ^, !, @. Other ordering can yield undefined results.

So for instance, Ctrl-Alt-c must be specified as "#^c" (and not "^#c"), due to the above ordering rule.

(2) If you want to make a shortcut that uses one of the special modifier characters (as the character being modified), then to avoid confusion, specify the numeric equivalent, e.g.

If you want..	Then use..
'#' as the shortcut..	"0x23" (instead of just "#").
'+' as the shortcut..	"0x2b" (instead of just "+").
'^' as the shortcut..	"0x5e" (instead of just "^").
Alt-+ as the shortcut..	"#0x2b" (instead of "#+").
Alt-^ as the shortcut..	"#0x5e" (instead of "#^").
..etc..	

As a general rule that's easy to remember, unless the shortcut key to be modified is a single alpha-numeric character [A-Z,a-z,0-9), it's probably best to use the numeric equivalents.

Todo Fix these silly legacy issues in a future release to support more predictable behavior for the modifier keys.

30.7.4.40 fl_pie()

```
void fl_pie (
    int x,
    int y,
    int w,
    int h,
    double a1,
    double a2 ) [inline]
```

Draw filled ellipse sections using integer coordinates.

Like [fl_arc\(\)](#), but [fl_pie\(\)](#) draws a filled-in pie slice. This slice may extend outside the line drawn by [fl_arc\(\)](#); to avoid this use $w - 1$ and $h - 1$.

Parameters

in	x,y,w,h	bounding box of complete circle
in	$a1,a2$	start and end angles of arc measured in degrees counter-clockwise from 3 o'clock. $a2$ must be greater than or equal to $a1$.

30.7.4.41 fl_polygon() [1/2]

```
void fl_polygon (
    int x,
    int y,
    int x1,
    int y1,
    int x2,
    int y2 ) [inline]
```

Fills a 3-sided polygon.

The polygon must be convex.

30.7.4.42 fl_polygon() [2/2]

```
void fl_polygon (
```

```

    int x,
    int y,
    int x1,
    int y1,
    int x2,
    int y2,
    int x3,
    int y3 ) [inline]

```

Fills a 4-sided polygon.

The polygon must be convex.

30.7.4.43 fl_pop_clip()

```
void fl_pop_clip ( ) [inline]
```

Restores the previous clip region.

You must call [fl_pop_clip\(\)](#) once for every time you call [fl_push_clip\(\)](#). Unpredictable results may occur if the clip stack is not empty when you return to FLTK.

30.7.4.44 fl_push_clip()

```
void fl_push_clip (
    int x,
    int y,
    int w,
    int h ) [inline]

```

Intersects the current clip region with a rectangle and pushes this new region onto the stack.

Parameters

in	<i>x,y,w,h</i>	position and size
----	----------------	-------------------

30.7.4.45 fl_push_matrix()

```
void fl_push_matrix ( ) [inline]
```

Saves the current transformation matrix on the stack.

The maximum depth of the stack is 32.

30.7.4.46 fl_read_image()

```
FL_EXPORT uchar * fl_read_image (
    uchar * p,
    int X,
    int Y,
    int W,
    int H,
    int alpha = 0 )

```

Reads an RGB(A) image from the current window or off-screen buffer.

Parameters

in	<i>p</i>	pixel buffer, or NULL to allocate one
in	<i>X,Y</i>	position of top-left of image to read
in	<i>W,H</i>	width and height of image to read
in	<i>alpha</i>	alpha value for image (0 for none)

Returns

pointer to pixel buffer, or NULL if allocation failed.

The `p` argument points to a buffer that can hold the image and must be at least $W*H*3$ bytes when reading RGB images, or $W*H*4$ bytes when reading RGBA images. If NULL, `fl_read_image()` will create an array of the proper size which can be freed using `delete[]`.

The `alpha` parameter controls whether an alpha channel is created and the value that is placed in the alpha channel. If 0, no alpha channel is generated.

30.7.4.47 fl_rect()

```
void fl_rect (
    int x,
    int y,
    int w,
    int h ) [inline]
```

Draws a 1-pixel border *inside* the given bounding box.

This function is meant for quick drawing of simple boxes. The behavior is undefined for line widths that are not 1.

30.7.4.48 fl_rectf()

```
FL_EXPORT void fl_rectf (
    int x,
    int y,
    int w,
    int h,
    uchar r,
    uchar g,
    uchar b )
```

Colors a rectangle with "exactly" the passed `r, g, b` color.

On screens with less than 24 bits of color this is done by drawing a solid-colored block using `fl_draw_image()` so that the correct color shade is produced.

30.7.4.49 fl_reset_spot()

```
FL_EXPORT void fl_reset_spot (
    void )
```

Todo provide user documentation for `fl_reset_spot` function

30.7.4.50 fl_rotate()

```
void fl_rotate (
    double d ) [inline]
```

Concatenates rotation transformation onto the current one.

Parameters

<code>in</code>	<code>d</code>	- rotation angle, counter-clockwise in degrees (not radians)
-----------------	----------------	--

30.7.4.51 fl_scale() [1/2]

```
void fl_scale (
    double x ) [inline]
```

Concatenates scaling transformation onto the current one.

Parameters

in	<i>x</i>	scale factor in both x-direction and y-direction
----	----------	--

30.7.4.52 fl_scale() [2/2]

```
void fl_scale (
    double x,
    double y ) [inline]
```

Concatenates scaling transformation onto the current one.

Parameters

in	<i>x,y</i>	scale factors in x-direction and y-direction
----	------------	--

30.7.4.53 fl_scroll()

```
FL_EXPORT void fl_scroll (
    int X,
    int Y,
    int W,
    int H,
    int dx,
    int dy,
    void(*) (void *, int, int, int, int) draw_area,
    void * data )
```

Scroll a rectangle and draw the newly exposed portions.

Parameters

in	<i>X,Y</i>	position of top-left of rectangle
in	<i>W,H</i>	size of rectangle
in	<i>dx,dy</i>	pixel offsets for shifting rectangle
in	<i>draw_area</i>	callback function to draw rectangular areas
in	<i>data</i>	pointer to user data for callback The contents of the rectangular area is first shifted by <i>dx</i> and <i>dy</i> pixels. The <i>draw_area</i> callback is then called for every newly exposed rectangular area.

30.7.4.54 fl_set_spot()

```
FL_EXPORT void fl_set_spot (
    int font,
    int size,
    int X,
    int Y,
    int W,
    int H,
    FL_Window * win = 0 )
```

Todo provide user documentation for `fl_set_spot` function

30.7.4.55 fl_set_status()

```
FL_EXPORT void fl_set_status (
    int X,
    int Y,
    int W,
    int H )
```

Todo provide user documentation for fl_set_status function

30.7.4.56 fl_shortcut_label() [1/2]

```
FL_EXPORT const char * fl_shortcut_label (
    unsigned int shortcut )
```

Get a human-readable string from a shortcut value.

Unparse a shortcut value as used by [Fl_Button](#) or [Fl_Menu_Item](#) into a human-readable string like "Alt+N". This only works if the shortcut is a character key or a numbered function key. If the shortcut is zero then an empty string is returned. The return value points at a static buffer that is overwritten with each call.

Since

FLTK 1.3.4 modifier key names can be localized, but key names can not yet be localized. This may be added to a future FLTK version.

Modifier key names (human-readable shortcut names) can be defined with the following global const char * pointer variables:

- fl_local_ctrl => name of FL_CTRL
- fl_local_alt => name of FL_ALT
- fl_local_shift => name of FL_SHIFT
- fl_local_meta => name of FL_META

```
fl_local_ctrl = "Strg"; // German for "Ctrl"
fl_local_shift = "Umschalt"; // German for "Shift"
```

The shortcut name will be constructed by adding all modifier names in the order defined above plus the name of the key. A '+' character is added to each modifier name unless it has a trailing '\' or a trailing '+'.

Example:

Ctrl+Alt+Shift+Meta+F12

The default values for modifier key names are as given above for all platforms except Mac OS X. Mac OS X uses graphical characters that represent the typical OS X modifier names in menus, e.g. cloverleaf, saucepan, etc. You may, however, redefine Mac OS X modifier names as well.

Parameters

in	<i>shortcut</i>	the integer value containing the ascii character or extended keystroke plus modifiers
----	-----------------	---

Returns

a pointer to a static buffer containing human readable text for the shortcut

30.7.4.57 fl_shortcut_label() [2/2]

```
FL_EXPORT const char * fl_shortcut_label (
    unsigned int shortcut,
    const char ** eom )
```

Get a human-readable string from a shortcut value.

Parameters

in	<i>shortcut</i>	the integer value containing the ascii character or extended keystroke plus modifiers
in	<i>eom</i>	if this pointer is set, it will receive a pointer to the end of the modifier text

Returns

a pointer to a static buffer containing human readable text for the shortcut

See also

[fl_shortcut_label\(unsigned int shortcut\)](#)

30.7.4.58 fl_transform_dx()

```
double fl_transform_dx (
    double x,
    double y ) [inline]
```

Transforms distance using current transformation matrix.

Parameters

in	<i>x,y</i>	coordinate
----	------------	------------

30.7.4.59 fl_transform_dy()

```
double fl_transform_dy (
    double x,
    double y ) [inline]
```

Transforms distance using current transformation matrix.

Parameters

in	<i>x,y</i>	coordinate
----	------------	------------

30.7.4.60 fl_transform_x()

```
double fl_transform_x (
    double x,
    double y ) [inline]
```

Transforms coordinate using the current transformation matrix.

Parameters

in	<i>x,y</i>	coordinate
----	------------	------------

30.7.4.61 fl_transform_y()

```
double fl_transform_y (
    double x,
```

```
double y ) [inline]
```

Transforms coordinate using the current transformation matrix.

Parameters

in	<i>x,y</i>	coordinate
----	------------	------------

30.7.4.62 fl_transformed_vertex()

```
void fl_transformed_vertex (
    double xf,
    double yf ) [inline]
```

Adds coordinate pair to the vertex list without further transformations.

Parameters

in	<i>xf,yf</i>	transformed coordinate
----	--------------	------------------------

30.7.4.63 fl_translate()

```
void fl_translate (
    double x,
    double y ) [inline]
```

Concatenates translation transformation onto the current one.

Parameters

in	<i>x,y</i>	translation factor in x-direction and y-direction
----	------------	---

30.7.4.64 fl_vertex()

```
void fl_vertex (
    double x,
    double y ) [inline]
```

Adds a single vertex to the current path.

Parameters

in	<i>x,y</i>	coordinate
----	------------	------------

30.8 Multithreading support functions

fl multithreading support functions declared in <FL/FI.H>

Functions

- static int [Fl::awake](#) ([Fl_Awake_Handler](#) cb, void *message=0)
See *void awake(void* message=0)*.
- static void [Fl::awake](#) (void *message=0)

Sends a message pointer to the main thread, causing any pending [Fl::wait\(\)](#) call to terminate so that the main thread can retrieve the message and any pending redraws can be processed.

- static int [Fl::lock](#) ()

The [lock\(\)](#) method blocks the current thread until it can safely access FLTK widgets and data.

- static void * [Fl::thread_message](#) ()

The [thread_message\(\)](#) method returns the last message that was sent from a child by the [awake\(\)](#) method.

- static void [Fl::unlock](#) ()

The [unlock\(\)](#) method releases the lock that was set using the [lock\(\)](#) method.

30.8.1 Detailed Description

fl multithreading support functions declared in [<FL/FL.H>](#)

30.8.2 Function Documentation

30.8.2.1 [awake\(\)](#) [1/2]

```
int Fl::awake (
    Fl_Awake_Handler func,
    void * data = 0 ) [static]
```

See void [awake\(void* message=0\)](#).

Let the main thread know an update is pending and have it call a specific function.

Registers a function that will be called by the main thread during the next message handling cycle. Returns 0 if the callback function was registered, and -1 if registration failed. Over a thousand [awake](#) callbacks can be registered simultaneously.

See also

[Fl::awake\(void* message=0\)](#)

30.8.2.2 [awake\(\)](#) [2/2]

```
void Fl::awake (
    void * msg = 0 ) [static]
```

Sends a message pointer to the main thread, causing any pending [Fl::wait\(\)](#) call to terminate so that the main thread can retrieve the message and any pending redraws can be processed.

Multiple calls to [Fl::awake\(\)](#) will queue multiple pointers for the main thread to process, up to a system-defined (typically several thousand) depth. The default message handler saves the last message which can be accessed using the [Fl::thread_message\(\)](#) function.

In the context of a threaded application, a call to [Fl::awake\(\)](#) with no argument will trigger event loop handling in the main thread. Since it is not possible to call [Fl::flush\(\)](#) from a subsidiary thread, [Fl::awake\(\)](#) is the best (and only, really) substitute.

See also: [Multithreading](#)

30.8.2.3 [lock\(\)](#)

```
int Fl::lock ( ) [static]
```

The [lock\(\)](#) method blocks the current thread until it can safely access FLTK widgets and data.

Child threads should call this method prior to updating any widgets or accessing data. The main thread must call [lock\(\)](#) to initialize the threading support in FLTK. [lock\(\)](#) will return non-zero if threading is not available on the platform.

Child threads must call [unlock\(\)](#) when they are done accessing FLTK.

When the [wait\(\)](#) method is waiting for input or timeouts, child threads are given access to FLTK. Similarly, when the main thread needs to do processing, it will wait until all child threads have called [unlock\(\)](#) before processing additional data.

Returns

0 if threading is available on the platform; non-zero otherwise.

See also: [Multithreading](#)

30.8.2.4 thread_message()

```
void * Fl::thread_message ( ) [static]
```

The [thread_message\(\)](#) method returns the last message that was sent from a child by the [awake\(\)](#) method.

See also: [Multithreading](#)

30.8.2.5 unlock()

```
void Fl::unlock ( ) [static]
```

The [unlock\(\)](#) method releases the lock that was set using the [lock\(\)](#) method.

Child threads should call this method as soon as they are finished accessing FLTK.

See also: [Multithreading](#)

30.9 Safe widget deletion support functions

These functions, declared in `<FL/Fl.H>`, support deletion of widgets inside callbacks.

Functions

- static void [Fl::clear_widget_pointer](#) ([Fl_Widget](#) const *w)
Clears a widget pointer in the watch list.
- static void [Fl::delete_widget](#) ([Fl_Widget](#) *w)
Schedules a widget for deletion at the next call to the event loop.
- static void [Fl::do_widget_deletion](#) ()
Deletes widgets previously scheduled for deletion.
- static void [Fl::release_widget_pointer](#) ([Fl_Widget](#) *&w)
Releases a widget pointer from the watch list.
- static void [Fl::watch_widget_pointer](#) ([Fl_Widget](#) *&w)
Adds a widget pointer to the widget watch list.

30.9.1 Detailed Description

These functions, declared in `<FL/Fl.H>`, support deletion of widgets inside callbacks.

[Fl::delete_widget\(\)](#) should be called when deleting widgets or complete widget trees ([Fl_Group](#), [Fl_Window](#), ...)
inside callbacks.

The other functions are intended for internal use. The preferred way to use them is by using the helper class [Fl_Widget_Tracker](#).

The following is to show how it works ...

There are three groups of related methods:

1. scheduled widget deletion
 - [Fl::delete_widget\(\)](#) schedules widgets for deletion
 - [Fl::do_widget_deletion\(\)](#) deletes all scheduled widgets
2. widget watch list ("smart pointers")
 - [Fl::watch_widget_pointer\(\)](#) adds a widget pointer to the watch list
 - [Fl::release_widget_pointer\(\)](#) removes a widget pointer from the watch list
 - [Fl::clear_widget_pointer\(\)](#) clears a widget pointer *in* the watch list
3. the class [Fl_Widget_Tracker](#):

- the constructor calls [Fl::watch_widget_pointer\(\)](#)
- the destructor calls [Fl::release_widget_pointer\(\)](#)
- the access methods can be used to test, if a widget has been deleted

See also

[Fl_Widget_Tracker](#).

30.9.2 Function Documentation

30.9.2.1 clear_widget_pointer()

```
void Fl::clear_widget_pointer (
    Fl_Widget const * w ) [static]
```

Clears a widget pointer *in* the watch list.

This is called when a widget is destroyed (by its destructor). You should never call this directly.

Note

Internal use only !

This method searches the widget watch list for pointers to the widget and clears each pointer that points to it. Widget pointers can be added to the widget watch list by calling [Fl::watch_widget_pointer\(\)](#) or by using the helper class [Fl_Widget_Tracker](#) (recommended).

See also

[Fl::watch_widget_pointer\(\)](#)

class [Fl_Widget_Tracker](#)

30.9.2.2 delete_widget()

```
void Fl::delete_widget (
    Fl_Widget * wi ) [static]
```

Schedules a widget for deletion at the next call to the event loop.

Use this method to delete a widget inside a callback function.

To avoid early deletion of widgets, this function should be called toward the end of a callback and only after any call to the event loop ([Fl::wait\(\)](#), [Fl::flush\(\)](#), [Fl::check\(\)](#), [fl_ask\(\)](#), etc.).

When deleting groups or windows, you must only delete the group or window widget and not the individual child widgets.

Since

FLTK 1.3.4 the widget will be hidden immediately, but the actual destruction will be delayed until the event loop is finished. Up to FLTK 1.3.3 windows wouldn't be hidden before the event loop was done, hence you had to `hide()` a window in your window close callback if you called [Fl::delete_widget\(\)](#) to destroy (and hide) the window.

FLTK 1.3.0 it is not necessary to remove widgets from their parent groups or windows before calling this, because it will be done in the widget's destructor, but it is not a failure to do this nevertheless.

Note

In FLTK 1.1 you **must** remove widgets from their parent group (or window) before deleting them.

See also

[Fl_Widget::~~Fl_Widget\(\)](#)

30.9.2.3 do_widget_deletion()

```
void Fl::do_widget_deletion ( ) [static]
```

Deletes widgets previously scheduled for deletion.

This is for internal use only. You should never call this directly.

[Fl::do_widget_deletion\(\)](#) is called from the FLTK event loop or whenever you call [Fl::wait\(\)](#). The previously scheduled widgets are deleted in the same order they were scheduled by calling [Fl::delete_widget\(\)](#).

See also

[Fl::delete_widget\(Fl_Widget *wi\)](#)

30.9.2.4 release_widget_pointer()

```
void Fl::release_widget_pointer (
    Fl_Widget *& w ) [static]
```

Releases a widget pointer from the watch list.

This is used to remove a widget pointer that has been added to the watch list with [Fl::watch_widget_pointer\(\)](#), when it is not needed anymore.

Note

Internal use only, please use class [Fl_Widget_Tracker](#) instead.

See also

[Fl::watch_widget_pointer\(\)](#)

30.9.2.5 watch_widget_pointer()

```
void Fl::watch_widget_pointer (
    Fl_Widget *& w ) [static]
```

Adds a widget pointer to the widget watch list.

Note

Internal use only, please use class [Fl_Widget_Tracker](#) instead.

This can be used, if it is possible that a widget might be deleted during a callback or similar function. The widget pointer must be added to the watch list before calling the callback. After the callback the widget pointer can be queried, if it is NULL. If it is NULL, then the widget has been deleted during the callback and must not be accessed anymore. If the widget pointer is *not* NULL, then the widget has not been deleted and can be accessed safely.

After accessing the widget, the widget pointer must be released from the watch list by calling [Fl::release_widget_pointer\(\)](#).

Example for a button that is clicked (from its [handle\(\)](#) method):

```
Fl_Widget *wp = this;           // save 'this' in a pointer variable
Fl::watch_widget_pointer(wp);  // add the pointer to the watch list
set_changed();                 // set the changed flag
do_callback();                 // call the callback
if (!wp) {                     // the widget has been deleted
    // DO NOT ACCESS THE DELETED WIDGET !
} else {                       // the widget still exists
    clear_changed();           // reset the changed flag
}
Fl::release_widget_pointer(wp); // remove the pointer from the watch list
```

This works, because all widgets call [Fl::clear_widget_pointer\(\)](#) in their destructors.

See also

[Fl::release_widget_pointer\(\)](#)

[Fl::clear_widget_pointer\(\)](#)

An easier and more convenient method to control widget deletion during callbacks is to use the class [Fl_Widget_Tracker](#) with a local (automatic) variable.

See also

class [Fl_Widget_Tracker](#)

30.10 Cairo Support Functions and Classes

Classes

- class [Fl_Cairo_State](#)
Contains all the necessary info on the current cairo context.
- class [Fl_Cairo_Window](#)
This defines a pre-configured cairo fltk window.

Functions

- static bool [Fl::cairo_autolink_context](#) ()
Gets the current autolink mode for cairo support.
- static void [Fl::cairo_autolink_context](#) (bool alink)
when `FLTK_HAVE_CAIRO` is defined and `cairo_autolink_context()` is true, any current window `dc` is linked to a current cairo context.
- static cairo_t* [Fl::cairo_cc](#) ()
Gets the current cairo context linked with a fltk window.
- static void [Fl::cairo_cc](#) (cairo_t *c, bool own=false)
Sets the current cairo context to `c`.
- static cairo_t* [Fl::cairo_make_current](#) ([Fl_Window](#) *w)
Provides a corresponding cairo context for window `wi`.

30.10.1 Detailed Description

30.10.2 Function Documentation

30.10.2.1 `cairo_autolink_context()` [1/2]

```
static bool Fl::cairo_autolink_context ( ) [inline], [static]
```

Gets the current autolink mode for cairo support.

Return values

<i>false</i>	if no cairo context autolink is made for each window.
<i>true</i>	if any fltk window is attached a cairo context when it is current.

See also

void [cairo_autolink_context](#)(bool alink)

Note

Only available when configure has the `--enable-cairo` option

30.10.2.2 `cairo_autolink_context()` [2/2]

```
static void Fl::cairo_autolink_context (
    bool alink ) [inline], [static]
```

when `FLTK_HAVE_CAIRO` is defined and `cairo_autolink_context()` is true, any current window `dc` is linked to a current cairo context.

This is not the default, because it may not be necessary to add cairo support to all fltk supported windows. When you wish to associate a cairo context in this mode, you need to call explicitly in your `draw()` overridden method, `Fl::cairo_make_current(Fl_Window*)`. This will create a cairo context but only for this Window. Still in custom cairo application it is possible to handle completely this process automatically by setting `alink` to true. In this last case, you don't need anymore to call `Fl::cairo_make_current()`. You can use `Fl::cairo_cc()` to get the current cairo context anytime.

Note

Only available when configure has the `--enable-cairo` option

30.10.2.3 `cairo_cc()`

```
static void Fl::cairo_cc (
    cairo_t * c,
    bool own = false ) [inline], [static]
```

Sets the current cairo context to `c`.

Set `own` to true if you want fltk to handle this cc deletion.

Note

Only available when configure has the `--enable-cairo` option

30.10.2.4 `cairo_make_current()`

```
cairo_t * Fl::cairo_make_current (
    Fl_Window * wi ) [static]
```

Provides a corresponding cairo context for window `wi`.

This is needed in a `draw()` override if `Fl::cairo_autolink_context()` returns false, which is the default. The `cairo_↔context()` does not need to be freed as it is freed every time a new cairo context is created. When the program terminates, a call to `Fl::cairo_make_current(0)` will destroy any residual context.

Note

A new cairo context is not always re-created when this method is used. In particular, if the current graphical context and the current window didn't change between two calls, the previous gc is internally kept, thus optimizing the drawing performances. Also, after this call, `Fl::cairo_cc()` is adequately updated with this cairo context.

Only available when configure has the `--enable-cairo` option

Returns

the valid `cairo_t*` cairo context associated to this window.

30.11 Unicode and UTF-8 functions

fl global Unicode and UTF-8 handling functions declared in `<FL/fl_utf8.h>`

Macros

- `#define ERRORS_TO_CP1252 1`
- `#define ERRORS_TO_ISO8859_1 1`
- `#define NBC 0xFFFF + 1`
- `#define STRICT_RFC3629 0`

Functions

- FL_EXPORT int [fl_access](#) (const char *f, int mode)
Cross-platform function to test a files access() with a UTF-8 encoded name or value.
- FL_EXPORT int [fl_chmod](#) (const char *f, int mode)
Cross-platform function to set a files mode() with a UTF-8 encoded name or value.
- FL_EXPORT int [fl_execvp](#) (const char *file, char *const *argv)
- FL_EXPORT FILE * [fl_fopen](#) (const char *f, const char *mode)
Cross-platform function to open files with a UTF-8 encoded name.
- FL_EXPORT char * [fl_getcwd](#) (char *b, int l)
Cross-platform function to get the current working directory as a UTF-8 encoded value.
- FL_EXPORT char * [fl_getenv](#) (const char *v)
Cross-platform function to get environment variables with a UTF-8 encoded name or value.
- FL_EXPORT char [fl_make_path](#) (const char *path)
Cross-platform function to recursively create a path in the file system.
- FL_EXPORT void [fl_make_path_for_file](#) (const char *path)
Cross-platform function to create a path for the file in the file system.
- FL_EXPORT int [fl_mkdir](#) (const char *f, int mode)
Cross-platform function to create a directory with a UTF-8 encoded name.
- FL_EXPORT unsigned int [fl_nonspacing](#) (unsigned int ucs)
Returns true if the Unicode character ucs is non-spacing.
- FL_EXPORT int [fl_open](#) (const char *f, int oflags,...)
Cross-platform function to open files with a UTF-8 encoded name.
- FL_EXPORT int [fl_rename](#) (const char *f, const char *n)
Cross-platform function to rename a filesystem object using UTF-8 encoded names.
- FL_EXPORT int [fl_rmdir](#) (const char *f)
Cross-platform function to remove a directory with a UTF-8 encoded name.
- FL_EXPORT int [fl_stat](#) (const char *f, struct stat *b)
Cross-platform function to stat() a file using a UTF-8 encoded name or value.
- FL_EXPORT int [fl_system](#) (const char *cmd)
Cross-platform function to run a system command with a UTF-8 encoded string.
- FL_EXPORT int [fl_tolower](#) (unsigned int ucs)
Returns the Unicode lower case value of ucs.
- FL_EXPORT int [fl_toupper](#) (unsigned int ucs)
Returns the Unicode upper case value of ucs.
- FL_EXPORT unsigned [fl_ucs_to_Utf16](#) (const unsigned ucs, unsigned short *dst, const unsigned dstlen)
- FL_EXPORT int [fl_unlink](#) (const char *f)
Cross-platform function to unlink() (that is, delete) a file using a UTF-8 encoded filename.
- FL_EXPORT char * [fl_utf2mbs](#) (const char *s)
Converts UTF-8 string s to a local multi-byte character string.
- FL_EXPORT const char * [fl_utf8back](#) (const char *p, const char *start, const char *end)
- FL_EXPORT int [fl_utf8bytes](#) (unsigned ucs)
Return the number of bytes needed to encode the given UCS4 character in UTF-8.
- FL_EXPORT unsigned [fl_utf8decode](#) (const char *p, const char *end, int *len)
- FL_EXPORT int [fl_utf8encode](#) (unsigned ucs, char *buf)
- FL_EXPORT unsigned [fl_utf8from_mb](#) (char *dst, unsigned dstlen, const char *src, unsigned srclen)
- FL_EXPORT unsigned [fl_utf8froma](#) (char *dst, unsigned dstlen, const char *src, unsigned srclen)
- FL_EXPORT unsigned [fl_utf8fromwbc](#) (char *dst, unsigned dstlen, const wchar_t *src, unsigned srclen)
- FL_EXPORT const char * [fl_utf8fwd](#) (const char *p, const char *start, const char *end)
- FL_EXPORT int [fl_utf8len](#) (char c)
Returns the byte length of the UTF-8 sequence with first byte c, or -1 if c is not valid.
- FL_EXPORT int [fl_utf8len1](#) (char c)

Returns the byte length of the UTF-8 sequence with first byte *c*, or 1 if *c* is not valid.

- FL_EXPORT int [fl_utf8locale](#) (void)
- FL_EXPORT int [fl_utf8test](#) (const char *src, unsigned len)
- FL_EXPORT unsigned [fl_utf8to_mb](#) (const char *src, unsigned srclen, char *dst, unsigned dstlen)
- FL_EXPORT unsigned [fl_utf8toa](#) (const char *src, unsigned srclen, char *dst, unsigned dstlen)
- FL_EXPORT unsigned [fl_utf8toUtf16](#) (const char *src, unsigned srclen, unsigned short *dst, unsigned dstlen)
- FL_EXPORT unsigned [fl_utf8towc](#) (const char *src, unsigned srclen, wchar_t *dst, unsigned dstlen)

Converts a UTF-8 string into a wide character string.

- FL_EXPORT int [fl_utf_nb_char](#) (const unsigned char *buf, int len)

Returns the number of Unicode chars in the UTF-8 string.

- FL_EXPORT int [fl_utf_strcasecmp](#) (const char *s1, const char *s2)

UTF-8 aware strcasecmp - converts to Unicode and tests.

- FL_EXPORT int [fl_utf_strncasecmp](#) (const char *s1, const char *s2, int n)

UTF-8 aware strncasecmp - converts to lower case Unicode and tests.

- FL_EXPORT int [fl_utf_tolower](#) (const unsigned char *str, int len, char *buf)

Converts the string *str* to its lower case equivalent into *buf*.

- FL_EXPORT int [fl_utf_toupper](#) (const unsigned char *str, int len, char *buf)

Converts the string *str* to its upper case equivalent into *buf*.

- FL_EXPORT int [fl_wcwidth](#) (const char *src)

extended wrapper around [fl_wcwidth_\(unsigned int ucs\)](#) function.

- FL_EXPORT int [fl_wcwidth_](#) (unsigned int ucs)

wrapper to adapt Markus Kuhn's implementation of `wcwidth()` for FLTK

30.11.1 Detailed Description

fl global Unicode and UTF-8 handling functions declared in [<FL/fl_utf8.h>](#)

30.11.2 Macro Definition Documentation

30.11.2.1 ERRORS_TO_CP1252

```
#define ERRORS_TO_CP1252 1
```

Set to 1 to turn bad UTF-8 bytes in the 0x80-0x9f range into the Unicode index for Microsoft's CP1252 character set. You should also set `ERRORS_TO_ISO8859_1`. With this a huge amount of more available text (such as all web pages) are correctly converted to Unicode.

30.11.2.2 ERRORS_TO_ISO8859_1

```
#define ERRORS_TO_ISO8859_1 1
```

Set to 1 to turn bad UTF-8 bytes into ISO-8859-1. If this is zero they are instead turned into the Unicode REPLACEMENT CHARACTER, of value 0xfffd. If this is on [fl_utf8decode\(\)](#) will correctly map most (perhaps all) human-readable text that is in ISO-8859-1. This may allow you to completely ignore character sets in your code because virtually everything is either ISO-8859-1 or UTF-8.

30.11.2.3 STRICT_RFC3629

```
#define STRICT_RFC3629 0
```

A number of Unicode code points are in fact illegal and should not be produced by a UTF-8 converter. Turn this on will replace the bytes in those encodings with errors. If you do this then converting arbitrary 16-bit data to UTF-8 and then back is not an identity, which will probably break a lot of software.

30.11.3 Function Documentation

30.11.3.1 fl_access()

```
int fl_access (
    const char * f,
    int mode )
```

Cross-platform function to test a files access() with a UTF-8 encoded name or value.

This function is especially useful under the MSWindows platform where the standard access() function fails with UTF-8 encoded non-ASCII filenames.

Parameters

in	<i>f</i>	the UTF-8 encoded filename
in	<i>mode</i>	the mode to test

Returns

the return value of _waccess() on Windows or access() on other platforms.

30.11.3.2 fl_chmod()

```
int fl_chmod (
    const char * f,
    int mode )
```

Cross-platform function to set a files mode() with a UTF-8 encoded name or value.

This function is especially useful under the MSWindows platform where the standard chmod() function fails with UTF-8 encoded non-ASCII filenames.

Parameters

in	<i>f</i>	the UTF-8 encoded filename
in	<i>mode</i>	the mode to set

Returns

the return value of _wchmod() on Windows or chmod() on other platforms.

30.11.3.3 fl_fopen()

```
FILE * fl_fopen (
    const char * f,
    const char * mode )
```

Cross-platform function to open files with a UTF-8 encoded name.

This function is especially useful under the MSWindows platform where the standard fopen() function fails with UTF-8 encoded non-ASCII filenames.

Parameters

<i>f</i>	the UTF-8 encoded filename
<i>mode</i>	same as the second argument of the standard fopen() function

Returns

a FILE pointer upon successful completion, or NULL in case of error.

See also

[fl_open\(\)](#).

30.11.3.4 fl_getcwd()

```
char * fl_getcwd (
    char * b,
    int l )
```

Cross-platform function to get the current working directory as a UTF-8 encoded value.

This function is especially useful under the MSWindows platform where the standard `_wgetcwd()` function returns UTF-16 encoded non-ASCII filenames.

Parameters

<i>b</i>	the buffer to populate
<i>l</i>	the length of the buffer

Returns

the CWD encoded as UTF-8.

30.11.3.5 fl_getenv()

```
char * fl_getenv (
    const char * v )
```

Cross-platform function to get environment variables with a UTF-8 encoded name or value.

This function is especially useful under the MSWindows platform where non-ASCII environment variables are encoded as wide characters. The returned value of the variable is encoded in UTF-8 as well.

On platforms other than MSWindows this function calls `getenv` directly. The return value is returned as-is.

Parameters

<i>in</i>	<i>v</i>	the UTF-8 encoded environment variable
-----------	----------	--

Returns

the environment variable in UTF-8 encoding, or NULL in case of error.

30.11.3.6 fl_make_path()

```
char fl_make_path (
    const char * path )
```

Cross-platform function to recursively create a path in the file system.

This function creates a `path` in the file system by recursively creating all directories.

30.11.3.7 fl_make_path_for_file()

```
void fl_make_path_for_file (
    const char * path )
```

Cross-platform function to create a path for the file in the file system.

This function strips the filename from the given `path` and creates a path in the file system by recursively creating all directories.

30.11.3.8 fl_mkdir()

```
int fl_mkdir (
    const char * f,
    int mode )
```

Cross-platform function to create a directory with a UTF-8 encoded name.

This function is especially useful on the MSWindows platform where the standard `_wmkdir()` function expects UTF-16 encoded non-ASCII filenames.

Parameters

in	<i>f</i>	the UTF-8 encoded filename
in	<i>mode</i>	the mode of the directory

Returns

the return value of `_wmkdir()` on Windows or `mkdir()` on other platforms.

30.11.3.9 fl_nonspacing()

```
unsigned int fl_nonspacing (
    unsigned int ucs )
```

Returns true if the Unicode character `ucs` is non-spacing.

Non-spacing characters in Unicode are typically combining marks like tilde (~), diaeresis (¨), or other marks that are added to a base character, for instance 'a' (base character) + '¨' (combining mark) = 'ä' (German Umlaut).

- http://unicode.org/glossary/#base_character
- http://unicode.org/glossary/#nonspacing_mark
- http://unicode.org/glossary/#combining_character

30.11.3.10 fl_open()

```
int fl_open (
    const char * f,
    int oflags,
    ... )
```

Cross-platform function to open files with a UTF-8 encoded name.

This function is especially useful under the MSWindows platform where the standard `open()` function fails with UTF-8 encoded non-ASCII filenames.

Parameters

<i>f</i>	the UTF-8 encoded filename
<i>oflags</i>	other arguments are as in the standard <code>open()</code> function

Returns

a file descriptor upon successful completion, or -1 in case of error.

See also

[fl_fopen\(\)](#).

30.11.3.11 fl_rename()

```
int fl_rename (
    const char * f,
    const char * n )
```

Cross-platform function to rename a filesystem object using UTF-8 encoded names.

This function is especially useful on the MSWindows platform where the standard `_wrename()` function expects UTF-16 encoded non-ASCII filenames.

Parameters

in	<i>f</i>	the UTF-8 encoded filename to change
in	<i>n</i>	the new UTF-8 encoded filename to set

Returns

the return value of `_wrename()` on Windows or `rename()` on other platforms.

30.11.3.12 fl_rmdir()

```
int fl_rmdir (
    const char * f )
```

Cross-platform function to remove a directory with a UTF-8 encoded name.

This function is especially useful on the MSWindows platform where the standard `_wrmdir()` function expects UTF-16 encoded non-ASCII filenames.

Parameters

in	<i>f</i>	the UTF-8 encoded filename to remove
----	----------	--------------------------------------

Returns

the return value of `_wrmdir()` on Windows or `rmdir()` on other platforms.

30.11.3.13 fl_stat()

```
int fl_stat (
    const char * f,
    struct stat * b )
```

Cross-platform function to `stat()` a file using a UTF-8 encoded name or value.

This function is especially useful under the MSWindows platform where the standard `stat()` function fails with UTF-8 encoded non-ASCII filenames.

Parameters

in	<i>f</i>	the UTF-8 encoded filename
	<i>b</i>	the stat struct to populate

Returns

the return value of `_wstat()` on Windows or `stat()` on other platforms.

30.11.3.14 `fl_system()`

```
int fl_system (
    const char * cmd )
```

Cross-platform function to run a system command with a UTF-8 encoded string.

This function is especially useful under the MSWindows platform where non-ASCII program (file) names must be encoded as wide characters.

On platforms other than MSWindows this function calls `system()` directly.

Parameters

<code>in</code>	<code>cmd</code>	the UTF-8 encoded command string
-----------------	------------------	----------------------------------

Returns

the return value of `_wsystem()` on Windows or `system()` on other platforms.

30.11.3.15 `fl_ucs_to_Utf16()`

```
unsigned fl_ucs_to_Utf16 (
    const unsigned ucs,
    unsigned short * dst,
    const unsigned dstlen )
```

Convert a single 32-bit Unicode codepoint into an array of 16-bit characters. These are used by some system calls, especially on Windows.

`ucs` is the value to convert.

`dst` points at an array to write, and `dstlen` is the number of locations in this array. At most `dstlen` words will be written, and a 0 terminating word will be added if `dstlen` is large enough. Thus this function will never overwrite the buffer and will attempt return a zero-terminated string if space permits. If `dstlen` is zero then `dst` can be set to NULL and no data is written, but the length is returned.

The return value is the number of 16-bit words that *would* be written to `dst` if it is large enough, not counting any terminating zero.

If the return value is greater than `dstlen` it indicates truncation, you should then allocate a new array of size `return+1` and call this again.

Unicode characters in the range 0x10000 to 0x10ffff are converted to "surrogate pairs" which take two words each (in UTF-16 encoding). Typically, setting `dstlen` to 2 will ensure that any valid Unicode value can be converted, and setting `dstlen` to 3 or more will allow a NULL terminated sequence to be returned.

30.11.3.16 `fl_unlink()`

```
int fl_unlink (
    const char * f )
```

Cross-platform function to `unlink()` (that is, delete) a file using a UTF-8 encoded filename.

This function is especially useful under the MSWindows platform where the standard function expects UTF-16 encoded non-ASCII filenames.

Parameters

<code>f</code>	the filename to unlink
----------------	------------------------

Returns

the return value of `_wunlink()` on Windows or `unlink()` on other platforms.

30.11.3.17 fl_utf8back()

```
const char * fl_utf8back (
    const char * p,
    const char * start,
    const char * end )
```

Move `p` backward until it points to the start of a UTF-8 character. If it already points at the start of one then it is returned unchanged. Any UTF-8 errors are treated as though each byte of the error is an individual character.

`start` is the start of the string and is used to limit the backwards search for the start of a UTF-8 character.

`end` is the end of the string and is assumed to be a break between characters. It is assumed to be greater than `p`.

If you wish to decrement a UTF-8 pointer, pass `p-1` to this.

30.11.3.18 fl_utf8bytes()

```
int fl_utf8bytes (
    unsigned ucs )
```

Return the number of bytes needed to encode the given UCS4 character in UTF-8.

Parameters

in	<i>ucs</i>	UCS4 encoded character
----	------------	------------------------

Returns

number of bytes required

Returns number of bytes that `utf8encode()` will use to encode the character `ucs`.

30.11.3.19 fl_utf8decode()

```
unsigned fl_utf8decode (
    const char * p,
    const char * end,
    int * len )
```

Decode a single UTF-8 encoded character starting at `p`. The resulting Unicode value (in the range 0-0x10ffff) is returned, and `len` is set to the number of bytes in the UTF-8 encoding (adding `len` to `p` will point at the next character).

If `p` points at an illegal UTF-8 encoding, including one that would go past `end`, or where a code uses more bytes than necessary, then `*(unsigned char*)p` is translated as though it is in the Microsoft CP1252 character set and `len` is set to 1. Treating errors this way allows this to decode almost any ISO-8859-1 or CP1252 text that has been mistakenly placed where UTF-8 is expected, and has proven very useful.

If you want errors to be converted to error characters (as the standards recommend), adding a test to see if the length is unexpectedly 1 will work:

```
if (*p & 0x80) { // what should be a multibyte encoding
    code = fl_utf8decode(p,end,&len);
    if (len<2) code = 0xFFFD; // Turn errors into REPLACEMENT CHARACTER
} else { // handle the 1-byte UTF-8 encoding:
    code = *p;
    len = 1;
}
```

Direct testing for the 1-byte case (as shown above) will also speed up the scanning of strings where the majority of characters are ASCII.

30.11.3.20 fl_utf8encode()

```
int fl_utf8encode (
    unsigned ucs,
    char * buf )
```


Write the UTF-8 encoding of *ucs* into *buf* and return the number of bytes written. Up to 4 bytes may be written. If you know that *ucs* is less than 0x10000 then at most 3 bytes will be written. If you wish to speed this up, remember that anything less than 0x80 is written as a single byte.

If *ucs* is greater than 0x10ffff this is an illegal character according to RFC 3629. These are converted as though they are 0xFFFF (REPLACEMENT CHARACTER).

RFC 3629 also says many other values for *ucs* are illegal (in the range 0xd800 to 0xdfff, or ending with 0xfffe or 0xffff). However I encode these as though they are legal, so that `utf8encode/fl_utf8decode` will be the identity for all codes between 0 and 0x10ffff.

30.11.3.21 `fl_utf8from_mb()`

```
unsigned fl_utf8from_mb (
    char * dst,
    unsigned dstlen,
    const char * src,
    unsigned srclen )
```

Convert a filename from the locale-specific multibyte encoding used by Windows to UTF-8 as used by FLTK.

Up to `dstlen` bytes are written to `dst`, including a null terminator. The return value is the number of bytes that would be written, not counting the null terminator. If greater or equal to `dstlen` then if you malloc a new array of size `n+1` you will have the space needed for the entire string. If `dstlen` is zero then nothing is written and this call just measures the storage space needed.

On Unix or on Windows when a UTF-8 locale is in effect, this does not change the data. You may also want to check if `fl_utf8test()` returns non-zero, so that the filesystem can store filenames in UTF-8 encoding regardless of the locale.

30.11.3.22 `fl_utf8froma()`

```
unsigned fl_utf8froma (
    char * dst,
    unsigned dstlen,
    const char * src,
    unsigned srclen )
```

Convert an ISO-8859-1 (ie normal c-string) byte stream to UTF-8.

It is possible this should convert Microsoft's CP1252 to UTF-8 instead. This would translate the codes in the range 0x80-0x9f to different characters. Currently it does not do this.

Up to `dstlen` bytes are written to `dst`, including a null terminator. The return value is the number of bytes that would be written, not counting the null terminator. If greater or equal to `dstlen` then if you malloc a new array of size `n+1` you will have the space needed for the entire string. If `dstlen` is zero then nothing is written and this call just measures the storage space needed.

`srclen` is the number of bytes in `src` to convert.

If the return value equals `srclen` then this indicates that no conversion is necessary, as only ASCII characters are in the string.

30.11.3.23 `fl_utf8fromwc()`

```
unsigned fl_utf8fromwc (
    char * dst,
    unsigned dstlen,
    const wchar_t * src,
    unsigned srclen )
```

Turn "wide characters" as returned by some system calls (especially on Windows) into UTF-8.

Up to `dstlen` bytes are written to `dst`, including a null terminator. The return value is the number of bytes that would be written, not counting the null terminator. If greater or equal to `dstlen` then if you malloc a new array of size `n+1` you will have the space needed for the entire string. If `dstlen` is zero then nothing is written and this call just measures the storage space needed.

`srclen` is the number of words in `src` to convert. On Windows this is not necessarily the number of characters, due to there possibly being "surrogate pairs" in the UTF-16 encoding used. On Unix `wchar_t` is 32 bits and each location is a character.

On Unix if a `src` word is greater than 0x10fff then this is an illegal character according to RFC 3629. These are converted as though they are 0xFFFFD (REPLACEMENT CHARACTER). Characters in the range 0xd800 to 0xdfff, or ending with 0xfffe or 0xffff are also illegal according to RFC 3629. However I encode these as though they are legal, so that `fl_utf8towc` will return the original data.

On Windows "surrogate pairs" are converted to a single character and UTF-8 encoded (as 4 bytes). Mismatched halves of surrogate pairs are converted as though they are individual characters.

30.11.3.24 `fl_utf8fwd()`

```
const char * fl_utf8fwd (
    const char * p,
    const char * start,
    const char * end )
```

Move `p` forward until it points to the start of a UTF-8 character. If it already points at the start of one then it is returned unchanged. Any UTF-8 errors are treated as though each byte of the error is an individual character.

`start` is the start of the string and is used to limit the backwards search for the start of a UTF-8 character.

`end` is the end of the string and is assumed to be a break between characters. It is assumed to be greater than `p`.

This function is for moving a pointer that was jumped to the middle of a string, such as when doing a binary search for a position. You should use either this or `fl_utf8back()` depending on which direction your algorithm can handle the pointer moving. Do not use this to scan strings, use `fl_utf8decode()` instead.

30.11.3.25 `fl_utf8len()`

```
int fl_utf8len (
    char c )
```

Returns the byte length of the UTF-8 sequence with first byte `c`, or -1 if `c` is not valid.

This function is helpful for finding faulty UTF-8 sequences.

See also

[fl_utf8len1](#)

30.11.3.26 `fl_utf8len1()`

```
int fl_utf8len1 (
    char c )
```

Returns the byte length of the UTF-8 sequence with first byte `c`, or 1 if `c` is not valid.

This function can be used to scan faulty UTF-8 sequences, albeit ignoring invalid codes.

See also

[fl_utf8len](#)

30.11.3.27 `fl_utf8locale()`

```
int fl_utf8locale (
    void )
```

Return true if the "locale" seems to indicate that UTF-8 encoding is used. If true the `fl_utf8to_mb` and `fl_utf8from_mb` don't do anything useful.

It is highly recommended that you change your system so this does return true. On Windows this is done by setting the "codepage" to CP_UTF8. On Unix this is done by setting `$LC_CTYPE` to a string containing the letters "utf" or "UTF" in it, or by deleting all `$LC*` and `$LANG` environment variables. In the future it is likely that all non-Asian Unix systems will return true, due to the compatibility of UTF-8 with ISO-8859-1.

30.11.3.28 fl_utf8test()

```
int fl_utf8test (
    const char * src,
    unsigned srclen )
```

Examines the first `srclen` bytes in `src` and returns a verdict on whether it is UTF-8 or not.

- Returns 0 if there is any illegal UTF-8 sequences, using the same rules as [fl_utf8decode\(\)](#). Note that some UCS values considered illegal by RFC 3629, such as 0xffff, are considered legal by this.
- Returns 1 if there are only single-byte characters (ie no bytes have the high bit set). This is legal UTF-8, but also indicates plain ASCII. It also returns 1 if `srclen` is zero.
- Returns 2 if there are only characters less than 0x800.
- Returns 3 if there are only characters less than 0x10000.
- Returns 4 if there are characters in the 0x10000 to 0x10ffff range.

Because there are many illegal sequences in UTF-8, it is almost impossible for a string in another encoding to be confused with UTF-8. This is very useful for transitioning Unix to UTF-8 filenames, you can simply test each filename with this to decide if it is UTF-8 or in the locale encoding. My hope is that if this is done we will be able to cleanly transition to a locale-less encoding.

30.11.3.29 fl_utf8to_mb()

```
unsigned fl_utf8to_mb (
    const char * src,
    unsigned srclen,
    char * dst,
    unsigned dstlen )
```

Convert the UTF-8 used by FLTK to the locale-specific encoding used for filenames (and sometimes used for data in files). Unfortunately due to stupid design you will have to do this as needed for filenames. This is a bug on both Unix and Windows.

Up to `dstlen` bytes are written to `dst`, including a null terminator. The return value is the number of bytes that would be written, not counting the null terminator. If greater or equal to `dstlen` then if you malloc a new array of size `n+1` you will have the space needed for the entire string. If `dstlen` is zero then nothing is written and this call just measures the storage space needed.

If [fl_utf8locale\(\)](#) returns true then this does not change the data.

30.11.3.30 fl_utf8toa()

```
unsigned fl_utf8toa (
    const char * src,
    unsigned srclen,
    char * dst,
    unsigned dstlen )
```

Convert a UTF-8 sequence into an array of 1-byte characters.

If the UTF-8 decodes to a character greater than 0xff then it is replaced with '?'.

Errors in the UTF-8 sequence are converted as individual bytes, same as [fl_utf8decode\(\)](#) does. This allows ISO-8859-1 text mistakenly identified as UTF-8 to be printed correctly (and possibly CP1252 on Windows).

`src` points at the UTF-8 sequence, and `srclen` is the number of bytes to convert.

Up to `dstlen` bytes are written to `dst`, including a null terminator. The return value is the number of bytes that would be written, not counting the null terminator. If greater or equal to `dstlen` then if you malloc a new array of size `n+1` you will have the space needed for the entire string. If `dstlen` is zero then nothing is written and this call just measures the storage space needed.

30.11.3.31 fl_utf8toUtf16()

```
unsigned fl_utf8toUtf16 (
    const char * src,
```

```

    unsigned srclen,
    unsigned short * dst,
    unsigned dstlen )

```

Convert a UTF-8 sequence into an array of 16-bit characters. These are used by some system calls, especially on Windows.

`src` points at the UTF-8, and `srclen` is the number of bytes to convert.

`dst` points at an array to write, and `dstlen` is the number of locations in this array. At most `dstlen-1` words will be written there, plus a 0 terminating word. Thus this function will never overwrite the buffer and will always return a zero-terminated string. If `dstlen` is zero then `dst` can be null and no data is written, but the length is returned. The return value is the number of 16-bit words that *would* be written to `dst` if it were long enough, not counting the terminating zero. If the return value is greater or equal to `dstlen` it indicates truncation, you can then allocate a new array of size `return+1` and call this again.

Errors in the UTF-8 are converted as though each byte in the erroneous string is in the Microsoft CP1252 encoding. This allows ISO-8859-1 text mistakenly identified as UTF-8 to be printed correctly.

Unicode characters in the range 0x10000 to 0x10ffff are converted to "surrogate pairs" which take two words each (this is called UTF-16 encoding).

30.11.3.32 fl_utf8towc()

```

unsigned fl_utf8towc (
    const char * src,
    unsigned srclen,
    wchar_t * dst,
    unsigned dstlen )

```

Converts a UTF-8 string into a wide character string.

This function generates 32-bit `wchar_t` (e.g. "ucs4" as it were) except on Windows where it is equivalent to `fl_utf8toUtf16` and returns UTF-16.

`src` points at the UTF-8, and `srclen` is the number of bytes to convert.

`dst` points at an array to write, and `dstlen` is the number of locations in this array. At most `dstlen-1` `wchar_t` will be written there, plus a 0 terminating `wchar_t`.

The return value is the number of `wchar_t` that *would* be written to `dst` if it were long enough, not counting the terminating zero. If the return value is greater or equal to `dstlen` it indicates truncation, you can then allocate a new array of size `return+1` and call this again.

Notice that `sizeof(wchar_t)` is 2 on Windows and is 4 on Linux and most other systems. Where `wchar_t` is 16 bits, Unicode characters in the range 0x10000 to 0x10ffff are converted to "surrogate pairs" which take two words each (this is called UTF-16 encoding). If `wchar_t` is 32 bits this rather nasty problem is avoided.

Note that Windows includes Cygwin, i.e. compiled with Cygwin's POSIX layer (`cygwin1.dll`, `-enable-cygwin`), either native (GDI) or X11.

30.11.3.33 fl_utf_strcasecmp()

```

int fl_utf_strcasecmp (
    const char * s1,
    const char * s2 )

```

UTF-8 aware `strcasecmp` - converts to Unicode and tests.

Returns

result of comparison

Return values

0	if the strings are equal
1	if s1 is greater than s2
-1	if s1 is less than s2

30.11.3.34 fl_utf_strncasecmp()

```
int fl_utf_strncasecmp (
    const char * s1,
    const char * s2,
    int n )
```

UTF-8 aware strncasecmp - converts to lower case Unicode and tests.

Parameters

<i>s1,s2</i>	the UTF-8 strings to compare
<i>n</i>	the maximum number of UTF-8 characters to compare

Returns

result of comparison

Return values

<i>0</i>	if the strings are equal
<i>>0</i>	if s1 is greater than s2
<i><0</i>	if s1 is less than s2

30.11.3.35 fl_utf_tolower()

```
int fl_utf_tolower (
    const unsigned char * str,
    int len,
    char * buf )
```

Converts the string *str* to its lower case equivalent into *buf*.

Warning: to be safe *buf* length must be at least $3 * len$ [for 16-bit Unicode]

30.11.3.36 fl_utf_toupper()

```
int fl_utf_toupper (
    const unsigned char * str,
    int len,
    char * buf )
```

Converts the string *str* to its upper case equivalent into *buf*.

Warning: to be safe *buf* length must be at least $3 * len$ [for 16-bit Unicode]

30.11.3.37 fl_wcwidth()

```
int fl_wcwidth (
    const char * src )
```

extended wrapper around [fl_wcwidth_\(unsigned int ucs\)](#) function.

Parameters

<i>in</i>	<i>src</i>	pointer to start of UTF-8 byte sequence
-----------	------------	---

Returns

width of character in columns

Depending on build options, this function may map C1 control characters (0x80 to 0x9f) to CP1252, and return the width of that character instead. This is not the same behaviour as `fl_wcwidth_(unsigned int ucs)`.

Note that other control characters and DEL will still return -1, so if you want different behaviour, you need to test for those characters before calling `fl_wcwidth_()`, and handle them separately.

30.11.3.38 fl_wcwidth_()

```
int fl_wcwidth_ (
    unsigned int ucs )
```

wrapper to adapt Markus Kuhn's implementation of `wcwidth()` for FLTK

Parameters

in	<i>ucs</i>	Unicode character value
----	------------	-------------------------

Returns

width of character in columns

See <http://www.cl.cam.ac.uk/~mgk25/ucs/wcwidth.c> for Markus Kuhn's original implementation of `wcwidth()` and `wcswidth()` (defined in IEEE Std 1002.1-2001) for Unicode.

WARNING: this function returns widths for "raw" Unicode characters. It does not even try to map C1 control characters (0x80 to 0x9F) to CP1252, and C0/C1 control characters and DEL will return -1. You are advised to use `fl_width(const char* src)` instead.

30.12 Mac OS X-specific symbols

Mac OS X-specific symbols declared in `<FL/x.H>` or `<FL/gl.h>`

Classes

- class `Fl_Mac_App_Menu`
Mac OS-specific class allowing to customize and localize the application menu.

Functions

- void `fl_mac_set_about` (`Fl_Callback *`*cb*, `void *`*user_data*, `int` *shortcut=0*)
Attaches a callback to the "About myprog" item of the system application menu.
- void `fl_open_callback` (`void(*cb)(const char *)`)
Register a function called for each file dropped onto an application icon.
- void `gl_texture_pile_height` (`int` *max*)
Changes the height of the pile of pre-computed string textures.
- `int` `gl_texture_pile_height` (`void`)
Returns the current height of the pile of pre-computed string textures.

Variables

- `int` `fl_mac_os_version`
The version number of the running Mac OS X (e.g., 100604 for 10.6.4)
- `int` `fl_mac_quit_early`
Determines whether cmd-Q or the "Quit xxx" item of application menu terminates the app or only the event loop.
- class `Fl_Sys_Menu_Bar` * `fl_sys_menu_bar`
The system menu bar.

30.12.1 Detailed Description

Mac OS X-specific symbols declared in `<FL/x.H>` or `<FL/gl.h>`

See also

[The Apple OS X Interface](#)

30.12.2 Function Documentation

30.12.2.1 `fl_mac_set_about()`

```
void fl_mac_set_about (
    Fl_Callback * cb,
    void * user_data,
    int shortcut = 0 )
```

Attaches a callback to the "About myprog" item of the system application menu.

Parameters

<code>cb</code>	a callback that will be called by "About myprog" menu item with NULL 1st argument.
<code>user_data</code>	a pointer transmitted as 2nd argument to the callback.
<code>shortcut</code>	optional shortcut to attach to the "About myprog" menu item (e.g., <code>FL_META+'a'</code>)

30.12.2.2 `fl_open_callback()`

```
void fl_open_callback (
    void(*) (const char *) cb )
```

Register a function called for each file dropped onto an application icon.

`cb` will be called with a single Unix-style file name and path. If multiple files were dropped, `cb` will be called multiple times.

30.12.2.3 `gl_texture_pile_height()` [1/2]

```
void gl_texture_pile_height (
    int max )
```

Changes the height of the pile of pre-computed string textures.

Strings that are often re-displayed can be processed much faster if this pile is set high enough to hold all of them.

Parameters

<code>max</code>	Height of the texture pile
------------------	----------------------------

30.12.2.4 `gl_texture_pile_height()` [2/2]

```
int gl_texture_pile_height (
    void )
```

Returns the current height of the pile of pre-computed string textures.

The default value is 100

30.12.3 Variable Documentation

30.12.3.1 fl_mac_quit_early

```
int fl_mac_quit_early [extern]
```

Determines whether cmd-Q or the "Quit xxx" item of application menu terminates the app or only the event loop. By default, `fl_mac_quit_early = 1`, and cmd-Q or "Quit xxx" terminate the app when all windows are closed without `Fl::run()` returning. If `fl_mac_quit_early` is set to 0, cmd-Q or "Quit xxx" terminate only the event loop when all windows are closed, and `Fl::run()` returns.

Note

This OS-specific variable will not be part of the API of FLTK 1.4.

30.13 Common Dialogs classes and functions

Classes

- class `Fl_Color_Chooser`
The `Fl_Color_Chooser` widget provides a standard RGB color chooser.
- class `Fl_File_Chooser`
The `Fl_File_Chooser` widget displays a standard file selection dialog that supports various selection modes.

Functions

- void `fl_alert` (const char *fmt,...)
Shows an alert message dialog box.
- int `fl_ask` (const char *fmt,...)
Shows a dialog displaying the `fmt` message, this dialog features 2 yes/no buttons.
- void `fl_beep` (int type)
Emits a system beep message.
- int `fl_choice` (const char *fmt, const char *b0, const char *b1, const char *b2,...)
Shows a dialog displaying the printf style `fmt` message, this dialog features up to 3 customizable choice buttons.
- int `fl_choice_n` (const char *fmt, const char *b0, const char *b1, const char *b2,...)
Like `fl_choice()` but with extended (negative) return values.
- int `fl_color_chooser` (const char *name, double &r, double &g, double &b, int cmode)
Pops up a window to let the user pick an arbitrary RGB color.
- int `fl_color_chooser` (const char *name, uchar &r, uchar &g, uchar &b, int cmode)
Pops up a window to let the user pick an arbitrary RGB color.
- char * `fl_dir_chooser` (const char *message, const char *fname, int relative)
Shows a file chooser dialog and gets a directory.
- char * `fl_file_chooser` (const char *message, const char *pat, const char *fname, int relative)
Shows a file chooser dialog and gets a filename.
- void `fl_file_chooser_callback` (void(*cb)(const char *))
Set the file chooser callback.
- void `fl_file_chooser_ok_label` (const char *l)
Set the "OK" button label.
- const char * `fl_input` (const char *fmt, const char *defstr,...)
Shows an input dialog displaying the `fmt` message.
- void `fl_message` (const char *fmt,...)
Shows an information message dialog box.
- void `fl_message_hotspot` (int enable)
Sets whether or not to move the common message box used in many common dialogs like `fl_message()`, `fl_alert()`, `fl_ask()`, `fl_choice()`, `fl_input()`, `fl_password()` to follow the mouse pointer.
- int `fl_message_hotspot` (void)

Gets whether or not to move the common message box used in many common dialogs like `fl_message()`, `fl_alert()`, `fl_ask()`, `fl_choice()`, `fl_input()`, `fl_password()` to follow the mouse pointer.

- `Fl_Widget * fl_message_icon ()`

Gets the `Fl_Box` icon container of the current default dialog used in many common dialogs like `fl_message()`, `fl_alert()`, `fl_ask()`, `fl_choice()`, `fl_input()`, `fl_password()`

- `void fl_message_title (const char *title)`

Sets the title of the dialog window used in many common dialogs.

- `void fl_message_title_default (const char *title)`

Sets the default title of the dialog window used in many common dialogs.

- `const char * fl_password (const char *fmt, const char *defstr,...)`

Shows an input dialog displaying the `fmt` message.

Variables

- `static void(* Fl::error)(const char *,...) = ::error`

FLTK calls `Fl::error()` to output a normal error message.

- `static void(* Fl::fatal)(const char *,...) = ::fatal`

FLTK calls `Fl::fatal()` to output a fatal error message.

- `const char * fl_cancel = "Cancel"`

string pointer used in common dialogs, you can change it to another language

- `const char * fl_close = "Close"`

string pointer used in common dialogs, you can change it to another language

- `const char * fl_no = "No"`

string pointer used in common dialogs, you can change it to another language

- `const char * fl_ok = "OK"`

string pointer used in common dialogs, you can change it to another language

- `const char * fl_yes = "Yes"`

string pointer used in common dialogs, you can change it to another language

- `static void(* Fl::warning)(const char *,...) = ::warning`

FLTK calls `Fl::warning()` to output a warning message.

30.13.1 Detailed Description

30.13.2 Function Documentation

30.13.2.1 `fl_alert()`

```
void fl_alert (
    const char * fmt,
    ... )
```

Shows an alert message dialog box.

Note

Common dialog boxes are application modal. No more than one common dialog box can be open at any time. Requests for additional dialog boxes are ignored.

```
#include <FL/fl_ask.H>
```

Parameters

<code>in</code>	<code>fmt</code>	can be used as an sprintf-like format and variables for the message text
-----------------	------------------	--

30.13.2.2 fl_ask()

```
int fl_ask (
    const char * fmt,
    ... )
```

Shows a dialog displaying the `fmt` message, this dialog features 2 yes/no buttons.

Note

Common dialog boxes are application modal. No more than one common dialog box can be open at any time. Requests for additional dialog boxes are ignored.

```
#include <FL/fl_ask.H>
```

Parameters

in	<i>fmt</i>	can be used as an sprintf-like format and variables for the message text
----	------------	--

Return values

0	if the no button is selected or another dialog box is still open
1	if yes is selected

Deprecated `fl_ask()` is deprecated since it uses "Yes" and "No" for the buttons which does not conform to the current FLTK Human Interface Guidelines. Use `fl_choice()` with the appropriate verbs instead.

30.13.2.3 fl_beep()

```
void fl_beep (
    int type )
```

Emits a system beep message.

Parameters

in	<i>type</i>	The beep type from the <code>FL_Beep</code> enumeration.
----	-------------	--

Note

```
#include <FL/fl_ask.H>
```

30.13.2.4 fl_choice()

```
int fl_choice (
    const char * fmt,
    const char * b0,
    const char * b1,
    const char * b2,
    ... )
```

Shows a dialog displaying the printf style `fmt` message, this dialog features up to 3 customizable choice buttons.

Note

Common dialog boxes are application modal. No more than one common dialog box can be open at any time. Requests for additional dialog boxes are ignored.

```
#include <FL/fl_ask.H>
```

Three choices with printf() style formatting:

```
int num_msgs = GetNumberOfMessages();
switch ( fl_choice("What to do with %d messages?", "Send", "Save", "Delete", num_msgs) ) {
  case 0: .. // Send
  case 1: .. // Save (default)
  case 2: .. // Delete
  ..
}
```

Three choice example:

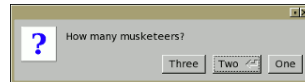


Figure 30.2 fl_choice() three choices

```
switch ( fl_choice("How many musketeers?", "One", "Two", "Three") ) {
  case 0: .. // One
  case 1: .. // Two (default)
  case 2: .. // Three
}
```

Two choice example:

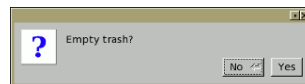


Figure 30.3 fl_choice() two choices

```
switch ( fl_choice("Empty trash?", "Yes", "No", 0) ) {
  case 0: .. // Yes
  case 1: .. // No (default)
}
```

One choice example:

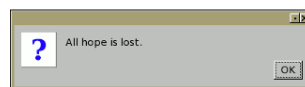


Figure 30.4 fl_choice() one choice

```
fl_choice("All hope is lost.", "OK", 0, 0); // "OK" default
```

Parameters

in	<i>fmt</i>	can be used as an sprintf-like format and variables for the message text
in	<i>b0</i>	text label of button 0
in	<i>b1</i>	text label of button 1 (can be 0)
in	<i>b2</i>	text label of button 2 (can be 0)

Return values

0	if the first button with <i>b0</i> text is pushed or another dialog box is still open
1	if the second button with <i>b1</i> text is pushed
2	if the third button with <i>b2</i> text is pushed

30.13.2.5 fl_choice_n()

```
int fl_choice_n (
    const char * fmt,
    const char * b0,
    const char * b1,
    const char * b2,
    ... )
```

Like [fl_choice\(\)](#) but with extended (negative) return values.

This function can return negative values as described below whereas [fl_choice\(\)](#) only returns "button values" (0, 1, 2).

With [fl_choice_n\(\)](#) you can arrange the buttons in a way that any button can be the standard "cancel" button because Escape and closing the window with the close button can be distinguished from button return codes.

Negative values are always "special" and should be considered like "cancel".

The special value `-3` means that the dialog was blocked (not executed).

Other than that both functions are the same.

See also

[fl_choice\(\)](#)

Since

1.3.8

Parameters

in	<i>fmt</i>	can be used as an sprintf-like format and variables for the message text
in	<i>b0</i>	text label of button 0
in	<i>b1</i>	text label of button 1 (can be 0)
in	<i>b2</i>	text label of button 2 (can be 0)

Return values

-3	if another dialog box is still open (the dialog was blocked)
-2	if the dialog window was closed by clicking the close button
-1	if the dialog was closed by hitting Escape
0	if the first button with <i>b0</i> text is pushed
1	if the second button with <i>b1</i> text is pushed
2	if the third button with <i>b2</i> text is pushed

30.13.2.6 fl_color_chooser() [1/2]

```
int fl_color_chooser (
    const char * name,
    double & r,
    double & g,
    double & b,
    int cmode ) [related]
```

Pops up a window to let the user pick an arbitrary RGB color.

Note

```
#include <FL/Fl_Color_Chooser.H>
```

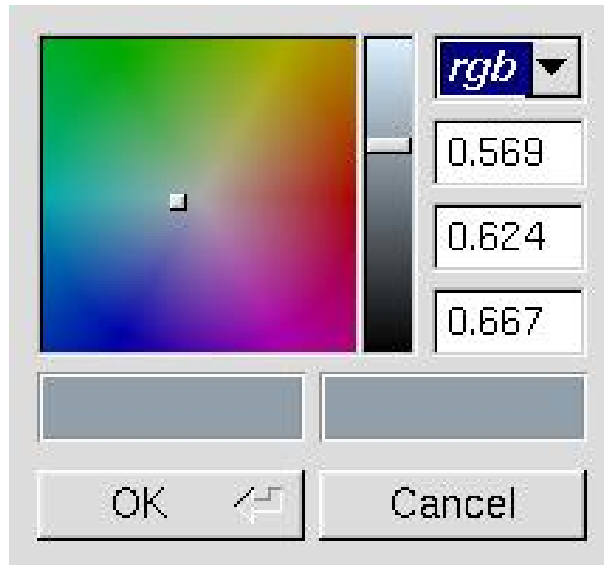


Figure 30.5 fl_color_chooser

Parameters

in	<i>name</i>	Title label for the window
in, out	<i>r,g,b</i>	Color components in the range 0.0 to 1.0.
in	<i>cmode</i>	Optional mode for color chooser. See mode(int) . Default -1 if none (rgb mode).

Return values

1	if user confirms the selection
0	if user cancels the dialog

30.13.2.7 fl_color_chooser() [2/2]

```
int fl_color_chooser (
    const char * name,
    uchar & r,
    uchar & g,
    uchar & b,
    int cmode ) [related]
```

Pops up a window to let the user pick an arbitrary RGB color.

Note

```
#include <FL/Fl_Color_Chooser.H>
```

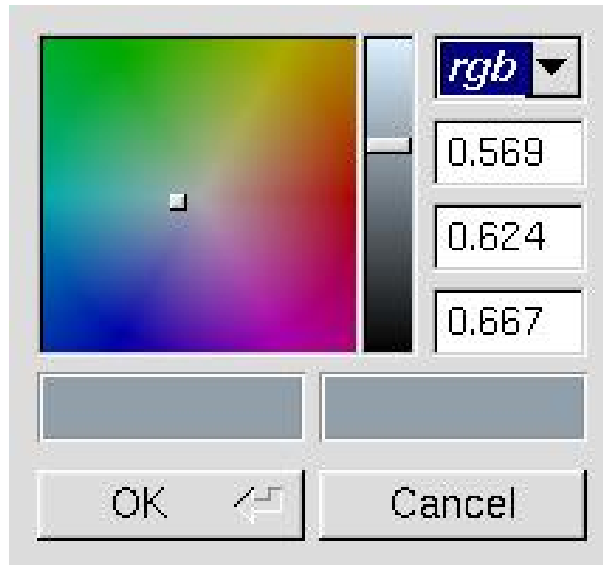


Figure 30.6 fl_color_chooser

Parameters

in	<i>name</i>	Title label for the window
in, out	<i>r,g,b</i>	Color components in the range 0 to 255.
in	<i>cmode</i>	Optional mode for color chooser. See mode(int) . Default -1 if none (rgb mode).

Return values

1	if user confirms the selection
0	if user cancels the dialog

30.13.2.8 fl_dir_chooser()

```
char * fl_dir_chooser (
    const char * message,
    const char * fname,
    int relative ) [related]
```

Shows a file chooser dialog and gets a directory.

Note

```
#include <FL/Fl_File_Chooser.H>
```

Parameters

in	<i>message</i>	title bar text
in	<i>fname</i>	initial/default directory name
in	<i>relative</i>	0 for absolute path return, relative otherwise

Returns

the directory path string chosen by the user or NULL if user cancels

30.13.2.9 fl_file_chooser()

```
char * fl_file_chooser (
    const char * message,
    const char * pat,
    const char * fname,
    int relative ) [related]
```

Shows a file chooser dialog and gets a filename.

Note

```
#include <FL/Fl_File_Chooser.H>
```

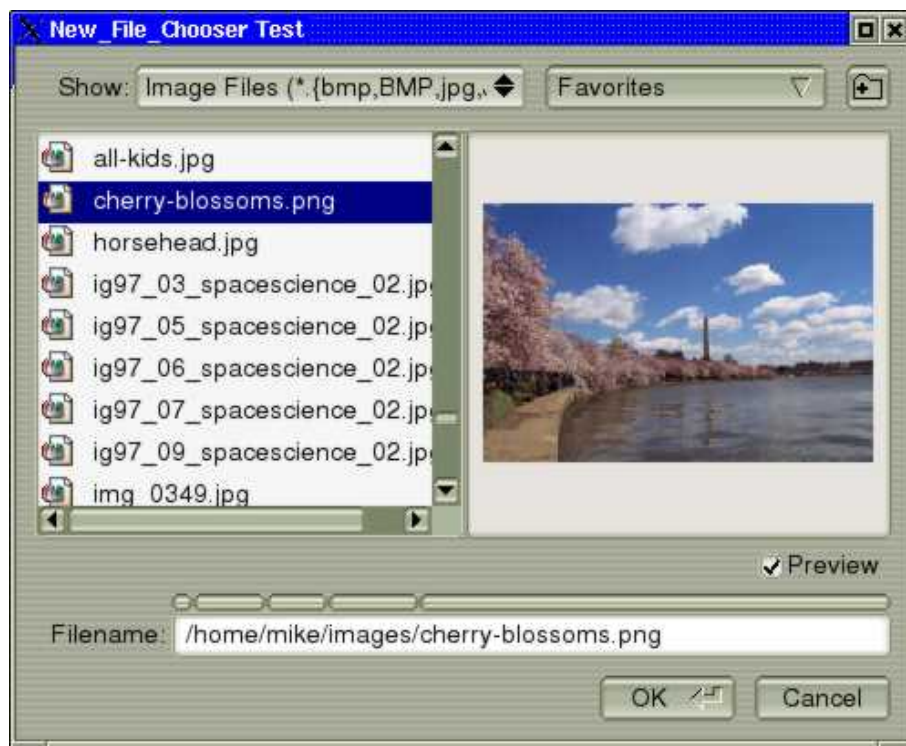


Figure 30.7 Fl_File_Chooser

Parameters

in	<i>message</i>	text in title bar
in	<i>pat</i>	filename pattern filter
in	<i>fname</i>	initial/default filename selection
in	<i>relative</i>	0 for absolute path name, relative path name otherwise

Returns

the user selected filename, in absolute or relative format or NULL if user cancels

30.13.2.10 fl_file_chooser_callback()

```
void fl_file_chooser_callback (
    void(*) (const char *) cb ) [related]
```

Set the file chooser callback.

Note

```
#include <FL/Fl_File_Chooser.H>
```

30.13.2.11 fl_file_chooser_ok_label()

```
void fl_file_chooser_ok_label (
    const char * l ) [related]
```

Set the "OK" button label.

Note

```
#include <FL/Fl_File_Chooser.H>
```

30.13.2.12 fl_input()

```
const char * fl_input (
    const char * fmt,
    const char * defstr,
    ... )
```

Shows an input dialog displaying the *fmt* message.

Note

Common dialog boxes are application modal. No more than one common dialog box can be open at any time. Requests for additional dialog boxes are ignored.

```
#include <FL/fl_ask.H>
```

Parameters

in	<i>fmt</i>	can be used as an printf-like format and variables for the message text
in	<i>defstr</i>	defines the default returned string if no text is entered

Returns

the user string input if OK was pushed, NULL if Cancel was pushed or another dialog box was still open

30.13.2.13 fl_message()

```
void fl_message (
    const char * fmt,
    ... )
```

Shows an information message dialog box.

Note

Common dialog boxes are application modal. No more than one common dialog box can be open at any time. Requests for additional dialog boxes are ignored.

```
#include <FL/fl_ask.H>
```

Parameters

<code>in</code>	<code>fmt</code>	can be used as an sprintf-like format and variables for the message text
-----------------	------------------	--

30.13.2.14 fl_message_hotspot() [1/2]

```
void fl_message_hotspot (
    int enable )
```

Sets whether or not to move the common message box used in many common dialogs like [fl_message\(\)](#), [fl_alert\(\)](#), [fl_ask\(\)](#), [fl_choice\(\)](#), [fl_input\(\)](#), [fl_password\(\)](#) to follow the mouse pointer.

The default is *enabled*, so that the default button is the hotspot and appears at the mouse position.

Note

```
#include <FL/fl_ask.H>
```

Parameters

<code>in</code>	<code>enable</code>	non-zero enables hotspot behavior, 0 disables hotspot
-----------------	---------------------	---

30.13.2.15 fl_message_hotspot() [2/2]

```
int fl_message_hotspot (
    void )
```

Gets whether or not to move the common message box used in many common dialogs like [fl_message\(\)](#), [fl_alert\(\)](#), [fl_ask\(\)](#), [fl_choice\(\)](#), [fl_input\(\)](#), [fl_password\(\)](#) to follow the mouse pointer.

Note

```
#include <FL/fl_ask.H>
```

Returns

0 if disable, non-zero otherwise

See also

[fl_message_hotspot\(int\)](#)

30.13.2.16 fl_message_icon()

```
Fl_Widget * fl_message_icon ( )
```

Gets the [Fl_Box](#) icon container of the current default dialog used in many common dialogs like [fl_message\(\)](#), [fl_alert\(\)](#), [fl_ask\(\)](#), [fl_choice\(\)](#), [fl_input\(\)](#), [fl_password\(\)](#)

Note

```
#include <FL/fl_ask.H>
```

30.13.2.17 fl_message_title()

```
void fl_message_title (
    const char * title )
```

Sets the title of the dialog window used in many common dialogs.

This window `title` will be used in the next call of one of the common dialogs like [fl_message\(\)](#), [fl_alert\(\)](#), [fl_ask\(\)](#), [fl_choice\(\)](#), [fl_input\(\)](#), [fl_password\(\)](#).

The `title` string is copied internally, so that you can use a local variable or free the string immediately after this call. It applies only to the **next** call of one of the common dialogs and will be reset to an empty title (the default for all dialogs) after that call.

Note

```
#include <FL/fl_ask.H>
```

Parameters

in	<i>title</i>	window label, string copied internally
----	--------------	--

30.13.2.18 fl_message_title_default()

```
void fl_message_title_default (
    const char * title )
```

Sets the default title of the dialog window used in many common dialogs.

This window `title` will be used in all subsequent calls of one of the common dialogs like [fl_message\(\)](#), [fl_alert\(\)](#), [fl_ask\(\)](#), [fl_choice\(\)](#), [fl_input\(\)](#), [fl_password\(\)](#), unless a specific title has been set with [fl_message_title\(const char *title\)](#).

The default is no title. You can override the default title for a single dialog with [fl_message_title\(const char *title\)](#).

The `title` string is copied internally, so that you can use a local variable or free the string immediately after this call.

Note

```
#include <FL/fl_ask.H>
```

Parameters

in	<i>title</i>	default window label, string copied internally
----	--------------	--

30.13.2.19 fl_password()

```
const char * fl_password (
    const char * fmt,
    const char * defstr,
    ... )
```

Shows an input dialog displaying the `fmt` message.

Like [fl_input\(\)](#) except the input text is not shown, '*' characters are displayed instead.

Note

Common dialog boxes are application modal. No more than one common dialog box can be open at any time. Requests for additional dialog boxes are ignored.

```
#include <FL/fl_ask.H>
```

Parameters

in	<i>fmt</i>	can be used as an sprintf-like format and variables for the message text
in	<i>defstr</i>	defines the default returned string if no text is entered

Returns

the user string input if OK was pushed, NULL if Cancel was pushed or another dialog box was still open

30.13.3 Variable Documentation

30.13.3.1 error

```
void(* Fl::error)(const char *format,...) = ::error [static]
```

FLTK calls [Fl::error\(\)](#) to output a normal error message.

The default version on Windows displays the error message in a MessageBox window.

The default version on all other platforms prints the error message to stderr.

You can override the behavior by setting the function pointer to your own routine.

[Fl::error\(\)](#) means there is a recoverable error such as the inability to read an image file. The default implementation returns after displaying the message.

Note

```
#include <FL/Fl.H>
```

30.13.3.2 fatal

```
void(* Fl::fatal)(const char *format,...) = ::fatal [static]
```

FLTK calls [Fl::fatal\(\)](#) to output a fatal error message.

The default version on Windows displays the error message in a MessageBox window.

The default version on all other platforms prints the error message to stderr.

You can override the behavior by setting the function pointer to your own routine.

[Fl::fatal\(\)](#) must not return, as FLTK is in an unusable state, however your version may be able to use longjmp or an exception to continue, as long as it does not call FLTK again. The default implementation exits with status 1 after displaying the message.

Note

```
#include <FL/Fl.H>
```

30.13.3.3 warning

```
void(* Fl::warning)(const char *format,...) = ::warning [static]
```

FLTK calls [Fl::warning\(\)](#) to output a warning message.

The default version on Windows returns *without* printing a warning message, because Windows programs normally don't have stderr (a console window) enabled.

The default version on all other platforms prints the warning message to stderr.

You can override the behavior by setting the function pointer to your own routine.

[Fl::warning\(\)](#) means that there was a recoverable problem, the display may be messed up, but the user can probably keep working - all X protocol errors call this, for example. The default implementation returns after displaying the message.

Note

```
#include <FL/Fl.H>
```

30.14 File names and URI utility functions

File names and URI functions defined in [<FL/filename.H>](#)

Macros

- `#define fl_dirent_h_cyclic_include`
- `#define FL_PATH_MAX 2048`
all path buffers should use this length

Typedefs

- typedef int() [Fl_File_Sort_F](#)(struct dirent **, struct dirent **)
File sorting function.

Functions

- FL_EXPORT void [fl_decode_uri](#) (char *uri)
Decodes a URL-encoded string.
- FL_EXPORT int [fl_filename_absolute](#) (char *to, int tolen, const char *from)
Makes a filename absolute from a relative filename.
- FL_EXPORT int [fl_filename_expand](#) (char *to, int tolen, const char *from)
Expands a filename containing shell variables and tilde (~).
- FL_EXPORT const char * [fl_filename_ext](#) (const char *buf)
Gets the extensions of a filename.
- FL_EXPORT void [fl_filename_free_list](#) (struct dirent ***l, int n)
Free the list of filenames that is generated by [fl_filename_list\(\)](#).
- FL_EXPORT int [fl_filename_isdir](#) (const char *name)
Determines if a file exists and is a directory from its filename.
- FL_EXPORT int [fl_filename_list](#) (const char *d, struct dirent ***l, [Fl_File_Sort_F](#) *s=fl_numericsort)
Portable and const-correct wrapper for the `scandir()` function.
- FL_EXPORT int [fl_filename_match](#) (const char *name, const char *pattern)
*Checks if a string *s* matches a pattern *p*.*
- FL_EXPORT const char * [fl_filename_name](#) (const char *filename)
Gets the file name from a path.
- FL_EXPORT int [fl_filename_relative](#) (char *to, int tolen, const char *from)
Makes a filename relative to the current working directory.
- FL_EXPORT char * [fl_filename_setext](#) (char *to, int tolen, const char *ext)
*Replaces the extension in *buf* of *max*.*
- FL_EXPORT int [fl_open_uri](#) (const char *uri, char *msg, int msglen)
Opens the specified Uniform Resource Identifier (URI).

30.14.1 Detailed Description

File names and URI functions defined in [<FL/filename.H>](#)

30.14.2 Typedef Documentation

30.14.2.1 Fl_File_Sort_F

```
typedef int() Fl_File_Sort_F(struct dirent **, struct dirent **)
File sorting function.
```

See also

[fl_filename_list\(\)](#)

30.14.3 Function Documentation

30.14.3.1 fl_decode_uri()

```
void fl_decode_uri (
    char * uri )
```

Decodes a URL-encoded string.

In a Uniform Resource Identifier (URI), all non-ASCII bytes and several others (e.g., '<', '"', ''') are URL-encoded using 3 bytes by "%XY" where XY is the hexadecimal value of the byte. This function decodes the URI restoring its original UTF-8 encoded content. Decoding is done in-place.

30.14.3.2 fl_filename_absolute()

```
FL_EXPORT int fl_filename_absolute (
    char * to,
    int tolen,
    const char * from )
```

Makes a filename absolute from a relative filename.

```
#include <FL/filename.H>
[.]
chdir("/var/tmp");
fl_filename_absolute(out, sizeof(out), "foo.txt");           // out="/var/tmp/foo.txt"
fl_filename_absolute(out, sizeof(out), "./foo.txt");        // out="/var/tmp/foo.txt"
fl_filename_absolute(out, sizeof(out), "../log/messages"); // out="/var/log/messages"
```

Parameters

out	to	resulting absolute filename
in	tolen	size of the absolute filename buffer
in	from	relative filename

Returns

0 if no change, non zero otherwise

30.14.3.3 fl_filename_expand()

```
FL_EXPORT int fl_filename_expand (
    char * to,
    int tolen,
    const char * from )
```

Expands a filename containing shell variables and tilde (~).

Currently handles these variants:

```
"~username"           // if 'username' does not exist, result will be unchanged
"~/file"
"$VARNAME"            // does NOT handle ${VARNAME}
```

Examples:

```
#include <FL/filename.H>
[.]
putenv("TMPDIR=/var/tmp");
fl_filename_expand(out, sizeof(out), "~/fred.cshrc");      // out="/usr/fred.cshrc"
fl_filename_expand(out, sizeof(out), "~/cshrc");           // out="/usr/<yourname>/cshrc"
```

```
fl_filename_expand(out, sizeof(out), "$TMPDIR/foo.txt"); // out="/var/tmp/foo.txt"
```

Parameters

out	<i>to</i>	resulting expanded filename
in	<i>toLen</i>	size of the expanded filename buffer
in	<i>from</i>	filename containing shell variables

Returns

0 if no change, non zero otherwise

30.14.3.4 `fl_filename_ext()`

```
FL_EXPORT const char * fl_filename_ext (
    const char * buf )
```

Gets the extensions of a filename.

```
#include <FL/filename.H>
[...
const char *out;
out = fl_filename_ext("/some/path/foo.txt"); // result: ".txt"
out = fl_filename_ext("/some/path/foo"); // result: NULL
```

Parameters

in	<i>buf</i>	the filename to be parsed
----	------------	---------------------------

Returns

a pointer to the extension (including '.') if any or NULL otherwise

30.14.3.5 `fl_filename_free_list()`

```
FL_EXPORT void fl_filename_free_list (
    struct dirent *** list,
    int n )
```

Free the list of filenames that is generated by `fl_filename_list()`.

Free everything that was allocated by a previous call to `fl_filename_list()`. Use the return values as parameters for this function.

Parameters

in, out	<i>list</i>	table containing the resulting directory listing
in	<i>n</i>	number of entries in the list

30.14.3.6 `fl_filename_isdir()`

```
FL_EXPORT int fl_filename_isdir (
    const char * n )
```

Determines if a file exists and is a directory from its filename.

```
#include <FL/filename.H>
[...
fl_filename_isdir("/etc"); // returns non-zero
fl_filename_isdir("/etc/hosts"); // returns 0
```

Parameters

in	<i>n</i>	the filename to parse
----	----------	-----------------------

Returns

non zero if file exists and is a directory, zero otherwise

30.14.3.7 fl_filename_list()

```
FL_EXPORT int fl_filename_list (
    const char * d,
    dirent *** list,
    Fl_File_Sort_F * sort )
```

Portable and const-correct wrapper for the scandir() function.

For each file in that directory a "dirent" structure is created. The only portable thing about a dirent is that dirent->d_name is the nul-terminated file name. An pointers array to these dirent's is created and a pointer to the array is returned in *list. The number of entries is given as a return value. If there is an error reading the directory a number less than zero is returned, and errno has the reason; errno does not work under WIN32.

Include:

```
#include <FL/filename.H>
```

Parameters

in	<i>d</i>	the name of the directory to list. It does not matter if it has a trailing slash.
out	<i>list</i>	table containing the resulting directory listing
in	<i>sort</i>	sorting functor: <ul style="list-style-type: none"> • <code>fl_alphasort</code>: The files are sorted in ascending alphabetical order; upper and lowercase letters are compared according to their ASCII ordering uppercase before lowercase. • <code>fl_casealphasort</code>: The files are sorted in ascending alphabetical order; upper and lowercase letters are compared equally case is not significant. • <code>fl_casenumERICsort</code>: The files are sorted in ascending "alphanumeric" order, where an attempt is made to put unpadding numbers in consecutive order; upper and lowercase letters are compared equally case is not significant. • <code>fl_numericSORT</code>: The files are sorted in ascending "alphanumeric" order, where an attempt is made to put unpadding numbers in consecutive order; upper and lowercase letters are compared according to their ASCII ordering - uppercase before lowercase.

Returns

the number of entries if no error, a negative value otherwise.

30.14.3.8 fl_filename_match()

```
FL_EXPORT int fl_filename_match (
    const char * s,
    const char * p )
```

Checks if a string *s* matches a pattern *p*.

The following syntax is used for the pattern:

- `*` matches any sequence of 0 or more characters.
- `?` matches any single character.
- `[set]` matches any character in the set. Set can contain any single characters, or a-z to represent a range. To match `]` or `-` they must be the first characters. To match `^` or `!` they must not be the first characters.
- `[^set]` or `[!set]` matches any character not in the set.

- {X|Y|Z} or {X,Y,Z} matches any one of the subexpressions literally.
- \x quotes the character x so it has no special meaning.
- x all other characters must be matched exactly.

Include:

```
#include <FL/filename.H>
```

Parameters

in	<i>s</i>	the string to check for a match
in	<i>p</i>	the string pattern

Returns

non zero if the string matches the pattern

30.14.3.9 fl_filename_name()

```
FL_EXPORT const char * fl_filename_name (
    const char * filename )
```

Gets the file name from a path.

Similar to `basename(3)`, exceptions shown below.

```
#include <FL/filename.H>
[.]
const char *out;
out = fl_filename_name("/usr/lib");      // out="lib"
out = fl_filename_name("/usr/");        // out=""      (basename(3) returns "usr" instead)
out = fl_filename_name("/usr");        // out="usr"
out = fl_filename_name("/");           // out=""      (basename(3) returns "/" instead)
out = fl_filename_name(".");           // out="."
out = fl_filename_name("..");          // out=".."
```

Returns

a pointer to the char after the last slash, or to `filename` if there is none.

30.14.3.10 fl_filename_relative()

```
FL_EXPORT int fl_filename_relative (
    char * to,
    int tolen,
    const char * from )
```

Makes a filename relative to the current working directory.

```
#include <FL/filename.H>
[.]
chdir("/var/tmp/somedir");           // set cwd to /var/tmp/somedir
[.]
char out[FL_PATH_MAX];
fl_filename_relative(out, sizeof(out), "/var/tmp/somedir/foo.txt"); // out="foo.txt", return=1
fl_filename_relative(out, sizeof(out), "/var/tmp/foo.txt");         // out="../foo.txt", return=1
fl_filename_relative(out, sizeof(out), "foo.txt");                  // out="foo.txt", return=0 (no
change)
fl_filename_relative(out, sizeof(out), "../foo.txt");              // out="../foo.txt", return=0 (no
change)
fl_filename_relative(out, sizeof(out), "../foo.txt");              // out="../foo.txt", return=0 (no
change)
```

Parameters

out	<i>to</i>	resulting relative filename
in	<i>tolen</i>	size of the relative filename buffer
in	<i>from</i>	absolute filename

Returns

0 if no change, non zero otherwise

30.14.3.11 fl_filename_setext()

```
FL_EXPORT char * fl_filename_setext (
    char * buf,
    int buflen,
    const char * ext )
```

Replaces the extension in *buf* of max.

size *buflen* with the extension in *ext*.

If there's no '.' in *buf*, *ext* is appended.

If *ext* is NULL, behaves as if it were an empty string ("").

Example

```
#include <FL/filename.H>
[.]
char buf[FL_PATH_MAX] = "/path/myfile.cxx";
fl_filename_setext(buf, sizeof(buf), ".txt"); // buf[] becomes "/path/myfile.txt"
```

Returns

buf itself for calling convenience.

30.14.3.12 fl_open_uri()

```
int fl_open_uri (
    const char * uri,
    char * msg,
    int msglen )
```

Opens the specified Uniform Resource Identifier (URI).

Uses an operating-system dependent program or interface. For URIs using the "ftp", "http", or "https" schemes, the system default web browser is used to open the URI, while "mailto" and "news" URIs are typically opened using the system default mail reader and "file" URIs are opened using the file system navigator.

On success, the (optional) *msg* buffer is filled with the command that was run to open the URI; on Windows, this will always be "open uri".

On failure, the *msg* buffer is filled with an English error message.

Note**Platform Specific Issues: Windows**

With "file:" based URIs on Windows, you may encounter issues with anchors being ignored. Example: "file://c:/some/index.html#anchor" may open in the browser without the "#anchor" suffix. The behavior seems to vary across different Windows versions. Workaround: open a link to a separate html file that redirects to the desired "file:" URI.

Example

```
#include <FL/filename.H>
[.]
char errmsg[512];
if ( !fl_open_uri("http://google.com/", errmsg, sizeof(errmsg)) ) {
    char warnmsg[768];
    sprintf(warnmsg, "Error: %s", errmsg);
    fl_alert(warnmsg);
}
```

Parameters

<i>uri</i>	The URI to open
<i>msg</i>	Optional buffer which contains the command or error message
<i>msglen</i>	Length of optional buffer

Returns

1 on success, 0 on failure

Chapter 31

Class Documentation

31.1 FI_Preferences::Entry Struct Reference

Public Attributes

- char * **name**
- char * **value**

The documentation for this struct was generated from the following file:

- FI_Preferences.H

31.2 FI Class Reference

The **FI** is the FLTK global (static) class containing state information and global methods for the current application.
`#include <Fl.H>`

Public Types

- enum **FI_Option** {
 OPTION_ARROW_FOCUS = 0, **OPTION_VISIBLE_FOCUS**, **OPTION_DND_TEXT**, **OPTION_SHOW_TOOLTIPS**,
 OPTION_FNFC_USES_GTK, **OPTION_LAST** }

Enumerator for global FLTK options.

Static Public Member Functions

- static int **abi_check** (const int val=**FL_ABI_VERSION**)
Returns whether the runtime library ABI version is correct.
- static int **abi_version** ()
*Returns the compiled-in value of the **FL_ABI_VERSION** constant.*
- static int **add_awake_handler_** (**FI_Awake_Handler**, void *)
*Adds an awake handler for use in **awake()**.*
- static void **add_check** (**FI_Timeout_Handler**, void *=0)
FLTK will call this callback just before it flushes the display and waits for events.
- static void **add_clipboard_notify** (**FI_Clipboard_Notify_Handler** h, void *data=0)
FLTK will call the registered callback whenever there is a change to the selection buffer or the clipboard.
- static void **add_fd** (int fd, **FI_FD_Handler** cb, void *=0)
*See void **add_fd**(int fd, int when, **FI_FD_Handler** cb, void* = 0)*
- static void **add_fd** (int fd, int when, **FI_FD_Handler** cb, void *=0)
Adds file descriptor fd to listen to.
- static void **add_handler** (**FI_Event_Handler** h)

- Install a function to parse unrecognized events.*

 - static void `add_idle` (`FI_Idle_Handler` cb, void *data=0)

Adds a callback function that is called every time by `Fl::wait()` and also makes it act as though the timeout is zero (this makes `Fl::wait()` return immediately, so if it is in a loop it is called repeatedly, and thus the idle function is called repeatedly).
 - static void `add_system_handler` (`FI_System_Handler` h, void *data)

Install a function to intercept system events.
 - static void `add_timeout` (double t, `FI_Timeout_Handler`, void *=0)

Adds a one-shot timeout callback.
 - static int `api_version` ()

Returns the compiled-in value of the `FL_API_VERSION` constant.
 - static int `arg` (int argc, char **argv, int &i)

Parse a single switch from `argv`, starting at word `i`.
 - static void `args` (int argc, char **argv)

Parse all command line switches matching standard FLTK options only.
 - static int `args` (int argc, char **argv, int &i, `FI_Args_Handler` cb=0)

Parse command line switches using the `cb` argument handler.
 - static int `awake` (`FI_Awake_Handler` cb, void *message=0)

See void `awake(void message=0)`.*
 - static void `awake` (void *message=0)

Sends a message pointer to the main thread, causing any pending `Fl::wait()` call to terminate so that the main thread can retrieve the message and any pending redraws can be processed.
 - static void `background` (`uchar`, `uchar`, `uchar`)

Changes `fl_color(FL_BACKGROUND_COLOR)` to the given color, and changes the gray ramp from 32 to 56 to black to white.
 - static void `background2` (`uchar`, `uchar`, `uchar`)

Changes the alternative background color.
 - static `Fl_Widget *` `belowmouse` ()

Gets the widget that is below the mouse.
 - static void `belowmouse` (`Fl_Widget *`)

Sets the widget that is below the mouse.
 - static `Fl_Color` `box_color` (`Fl_Color`)

Gets the drawing color to be used for the background of a box.
 - static int `box_dh` (`Fl_Boxtype`)

Returns the height offset for the given boxtype.
 - static int `box_dw` (`Fl_Boxtype`)

Returns the width offset for the given boxtype.
 - static int `box_dx` (`Fl_Boxtype`)

Returns the X offset for the given boxtype.
 - static int `box_dy` (`Fl_Boxtype`)

Returns the Y offset for the given boxtype.
 - static bool `cairo_autolink_context` ()

Gets the current autolink mode for cairo support.
 - static void `cairo_autolink_context` (bool alink)

when `FLTK_HAVE_CAIRO` is defined and `cairo_autolink_context()` is true, any current window dc is linked to a current cairo context.
 - static `cairo_t *` `cairo_cc` ()

Gets the current cairo context linked with a fltk window.
 - static void `cairo_cc` (`cairo_t *`c, bool own=false)

Sets the current cairo context to `c`.
 - static `cairo_t *` `cairo_make_current` (`Fl_Window *`w)

Provides a corresponding cairo context for window `wi`.

- static int `check` ()
Same as `Fl::wait(0)`.
- static void `clear_widget_pointer` (`Fl_Widget` const *w)
Clears a widget pointer in the watch list.
- static int `clipboard_contains` (const char *type)
Returns non 0 if the clipboard contains data matching `type`.
- static int `compose` (int &del)
Any text editing widget should call this for each `FL_KEYBOARD` event.
- static void `compose_reset` ()
If the user moves the cursor, be sure to call `Fl::compose_reset()`.
- static void `copy` (const char *stuff, int len, int destination=0, const char *type=`Fl::clipboard_plain_text`)
Copies the data pointed to by `stuff` to the selection buffer (`destination` is 0), the clipboard (`destination` is 1), or both (`destination` is 2).
- static int `damage` ()
If true then `flush()` will do something.
- static void `damage` (int d)
If true then `flush()` will do something.
- static void `default_atclose` (`Fl_Window` *, void *)
Default callback for window widgets.
- static void `delete_widget` (`Fl_Widget` *w)
Schedules a widget for deletion at the next call to the event loop.
- static void `disable_im` ()
Disables the system input methods facilities.
- static void `display` (const char *)
Sets the X display to use for all windows.
- static int `dnd` ()
Initiate a Drag And Drop operation.
- static int `dnd_text_ops` ()
Gets or sets whether drag and drop text operations are supported.
- static void `dnd_text_ops` (int v)
Gets or sets whether drag and drop text operations are supported.
- static void `do_widget_deletion` ()
Deletes widgets previously scheduled for deletion.
- static int `draw_box_active` ()
Determines if the currently drawn box is active or inactive.
- static void `enable_im` ()
Enables the system input methods facilities.
- static int `event` ()
Returns the last event that was processed.
- static int `event_alt` ()
Returns non-zero if the Alt key is pressed.
- static int `event_button` ()
Gets which particular mouse button caused the current event.
- static int `event_button1` ()
Returns non-zero if mouse button 1 is currently held down.
- static int `event_button2` ()
Returns non-zero if button 2 is currently held down.
- static int `event_button3` ()
Returns non-zero if button 3 is currently held down.
- static int `event_buttons` ()
Returns the mouse buttons state bits; if non-zero, then at least one button is pressed now.

- static int [event_clicks](#) ()
Returns non zero if we had a double click event.
- static void [event_clicks](#) (int i)
Manually sets the number returned by [Fl::event_clicks\(\)](#).
- static void * [event_clipboard](#) ()
During an [FL_PASTE](#) event of non-textual data, returns a pointer to the pasted data.
- static const char * [event_clipboard_type](#) ()
Returns the type of the pasted data during an [FL_PASTE](#) event.
- static int [event_command](#) ()
Returns non-zero if the [FL_COMMAND](#) key is pressed, either [FL_CTRL](#) or on OSX [FL_META](#).
- static int [event_ctrl](#) ()
Returns non-zero if the Control key is pressed.
- static [Fl_Event_Dispatch](#) [event_dispatch](#) ()
Return the current event dispatch function.
- static void [event_dispatch](#) ([Fl_Event_Dispatch](#) d)
Set a new event dispatch function.
- static int [event_dx](#) ()
Returns the current horizontal mouse scrolling associated with the [FL_MOUSEWHEEL](#) event.
- static int [event_dy](#) ()
Returns the current vertical mouse scrolling associated with the [FL_MOUSEWHEEL](#) event.
- static int [event_inside](#) (const [Fl_Widget](#) *)
Returns whether or not the mouse event is inside a given child widget.
- static int [event_inside](#) (int, int, int, int)
Returns whether or not the mouse event is inside the given rectangle.
- static int [event_is_click](#) ()
Returns non-zero if the mouse has not moved far enough and not enough time has passed since the last [FL_PUSH](#) or [FL_KEYBOARD](#) event for it to be considered a "drag" rather than a "click".
- static void [event_is_click](#) (int i)
Clears the value returned by [Fl::event_is_click\(\)](#).
- static int [event_key](#) ()
Gets which key on the keyboard was last pushed.
- static int [event_key](#) (int key)
Returns true if the given `key` was held down (or pressed) during the last event.
- static int [event_length](#) ()
Returns the length of the text in [Fl::event_text\(\)](#).
- static int [event_original_key](#) ()
Returns the keycode of the last key event, regardless of the NumLock state.
- static int [event_shift](#) ()
Returns non-zero if the Shift key is pressed.
- static int [event_state](#) ()
Returns the keyboard and mouse button states of the last event.
- static int [event_state](#) (int mask)
Returns non-zero if any of the passed event state bits are turned on.
- static const char * [event_text](#) ()
Returns the text associated with the current event, including [FL_PASTE](#) or [FL_DND_RELEASE](#) events.
- static int [event_x](#) ()
Returns the mouse position of the event relative to the [Fl_Window](#) it was passed to.
- static int [event_x_root](#) ()
Returns the mouse position on the screen of the event.
- static int [event_y](#) ()
Returns the mouse position of the event relative to the [Fl_Window](#) it was passed to.

- static int `event_y_root` ()
Returns the mouse position on the screen of the event.
- static `FI_Window * first_window` ()
Returns the first top-level window in the list of shown() windows.
- static void `first_window` (`FI_Window *`)
Sets the window that is returned by `first_window()`.
- static void `flush` ()
Causes all the windows that need it to be redrawn and graphics forced out through the pipes.
- static `FI_Widget * focus` ()
Gets the current `FI::focus()` widget.
- static void `focus` (`FI_Widget *`)
Sets the widget that will receive `FL_KEYBOARD` events.
- static void `foreground` (`uchar, uchar, uchar`)
Changes `fl_color(FL_FOREGROUND_COLOR)`.
- static void `free_color` (`FI_Color i, int overlay=0`)
Frees the specified color from the colormap, if applicable.
- static int `get_awesome_handler_` (`FI_Awake_Handler &, void *&`)
Gets the last stored awake handler for use in `awake()`.
- static `FI_Box_Draw_F * get_boxtype` (`FI_Boxtype`)
Gets the current box drawing function for the specified box type.
- static unsigned `get_color` (`FI_Color i`)
Returns the RGB value(s) for the given FLTK color index.
- static void `get_color` (`FI_Color i, uchar &red, uchar &green, uchar &blue`)
Returns the RGB value(s) for the given FLTK color index.
- static const char * `get_font` (`FI_Font`)
Gets the string for this face.
- static const char * `get_font_name` (`FI_Font, int *attributes=0`)
Get a human-readable string describing the family of this face.
- static int `get_font_sizes` (`FI_Font, int *&sizep`)
Return an array of sizes in `sizep`.
- static int `get_key` (int `key`)
Returns true if the given `key` is held down now.
- static void `get_mouse` (int &, int &)
Return where the mouse is on the screen by doing a round-trip query to the server.
- static void `get_system_colors` ()
Read the user preference colors from the system and use them to call `FI::foreground()`, `FI::background()`, and `FI::background2()`.
- static int `gl_visual` (int, int *`alist=0`)
This does the same thing as `FI::visual(int)` but also requires OpenGL drawing to work.
- static `FI_Window * grab` ()
Returns the window that currently receives all events.
- static void `grab` (`FI_Window &win`)
See `grab(FI_Window)`*
- static void `grab` (`FI_Window *`)
Selects the window to grab.
- static int `h` ()
Returns the height in pixels of the main screen work area.
- static int `handle` (int, `FI_Window *`)
Handle events from the window system.
- static int `handle_` (int, `FI_Window *`)
Handle events from the window system.

- static int **has_check** ([FI_Timeout_Handler](#), void *=0)
Returns 1 if the check exists and has not been called yet, 0 otherwise.
- static int **has_idle** ([FI_Idle_Handler](#) cb, void *data=0)
Returns true if the specified idle callback is currently installed.
- static int **has_timeout** ([FI_Timeout_Handler](#), void *=0)
Returns true if the timeout exists and has not been called yet.
- static int **is_scheme** (const char *name)
Returns whether the current scheme is the given name.
- static int **lock** ()
The `lock()` method blocks the current thread until it can safely access FLTK widgets and data.
- static [FI_Window](#) * **modal** ()
Returns the top-most `modal()` window currently shown.
- static [FI_Window](#) * **next_window** (const [FI_Window](#) *)
Returns the next top-level window in the list of `shown()` windows.
- static bool **option** ([FI_Option](#) opt)
FLTK library options management.
- static void **option** ([FI_Option](#) opt, bool val)
Override an option while the application is running.
- static void **own_colormap** ()
Makes FLTK use its `own colormap`.
- static void **paste** ([FI_Widget](#) &receiver)
Backward compatibility only.
- static void **paste** ([FI_Widget](#) &receiver, int source, const char *type=[FI::clipboard_plain_text](#))
Pastes the data from the selection buffer (`source` is 0) or the clipboard (`source` is 1) into receiver.
- static [FI_Widget](#) * **pushed** ()
Gets the widget that is being pushed.
- static void **pushed** ([FI_Widget](#) *)
Sets the widget that is being pushed.
- static [FI_Widget](#) * **readqueue** ()
Reads the default callback queue and returns the first widget.
- static int **ready** ()
This is similar to `FI::check()` except this does not call `FI::flush()` or any callbacks, which is useful if your program is in a state where such callbacks are illegal.
- static void **redraw** ()
Redraws all widgets.
- static void **release** ()
Releases the current grabbed window, equals `grab(0)`.
- static void **release_widget_pointer** ([FI_Widget](#) *&w)
Releases a widget pointer from the watch list.
- static int **reload_scheme** ()
Called by scheme according to scheme name.
- static void **remove_check** ([FI_Timeout_Handler](#), void *=0)
Removes a check callback.
- static void **remove_clipboard_notify** ([FI_Clipboard_Notify_Handler](#) h)
Stop calling the specified callback when there are changes to the selection buffer or the clipboard.
- static void **remove_fd** (int)
Removes a file descriptor handler.
- static void **remove_fd** (int, int when)
Removes a file descriptor handler.
- static void **remove_handler** ([FI_Event_Handler](#) h)
Removes a previously added event handler.

- static void **remove_idle** ([FI_Idle_Handler](#) cb, void *data=0)
Removes the specified idle callback, if it is installed.
- static void **remove_system_handler** ([FI_System_Handler](#) h)
Removes a previously added system event handler.
- static void **remove_timeout** ([FI_Timeout_Handler](#), void *=0)
Removes a timeout callback.
- static void **repeat_timeout** (double t, [FI_Timeout_Handler](#), void *=0)
Repeats a timeout callback from the expiration of the previous timeout, allowing for more accurate timing.
- static int **run** ()
As long as any windows are displayed this calls [FI::wait\(\)](#) repeatedly.
- static const char * **scheme** ()
*See void [scheme\(const char *name\)](#)*
- static int **scheme** (const char *name)
Sets the current widget scheme.
- static int **screen_count** ()
Gets the number of available screens.
- static void **screen_dpi** (float &h, float &v, int n=0)
Gets the screen resolution in dots-per-inch for the given screen.
- static int **screen_num** (int x, int y)
Gets the screen number of a screen that contains the specified screen position x, y.
- static int **screen_num** (int x, int y, int w, int h)
Gets the screen number for the screen which intersects the most with the rectangle defined by x, y, w, h.
- static void **screen_work_area** (int &X, int &Y, int &W, int &H)
Gets the bounding box of the work area of the screen that contains the mouse pointer.
- static void **screen_work_area** (int &X, int &Y, int &W, int &H, int mx, int my)
Gets the bounding box of the work area of a screen that contains the specified screen position mx, my.
- static void **screen_work_area** (int &X, int &Y, int &W, int &H, int n)
Gets the bounding box of the work area of the given screen.
- static void **screen_xywh** (int &X, int &Y, int &W, int &H)
Gets the bounding box of a screen that contains the mouse pointer.
- static void **screen_xywh** (int &X, int &Y, int &W, int &H, int mx, int my)
Gets the bounding box of a screen that contains the specified screen position mx, my.
- static void **screen_xywh** (int &X, int &Y, int &W, int &H, int mx, int my, int mw, int mh)
Gets the screen bounding rect for the screen which intersects the most with the rectangle defined by mx, my, mw, mh.
- static void **screen_xywh** (int &X, int &Y, int &W, int &H, int n)
Gets the screen bounding rect for the given screen.
- static int **scrollbar_size** ()
Gets the default scrollbar size used by [FI_Browser_](#), [FI_Help_View](#), [FI_Scroll](#), and [FI_Text_Display](#) widgets.
- static void **scrollbar_size** (int W)
Sets the default scrollbar size that is used by the [FI_Browser_](#), [FI_Help_View](#), [FI_Scroll](#), and [FI_Text_Display](#) widgets.
- static void **selection** ([FI_Widget](#) &owner, const char *, int len)
Changes the current selection.
- static [FI_Widget](#) * **selection_owner** ()
back-compatibility only: Gets the widget owning the current selection
- static void **selection_owner** ([FI_Widget](#) *)
Back-compatibility only: The single-argument call can be used to move the selection to another widget or to set the owner to NULL, without changing the actual text of the selection.
- static void **set_abort** ([FI_Abort_Handler](#) f)
For back compatibility, sets the void [FI::fatal](#) handler callback.
- static void **set_atclose** ([FI_Atclose_Handler](#) f)

- For back compatibility, sets the `Fl::atclose` handler callback.*

 - static void `set_box_color` (`Fl_Color`)
 - Sets the drawing color for the box that is currently drawn.*
 - static void `set_boxtype` (`Fl_Boxtype`, `Fl_Box_Draw_F *`, `uchar`, `uchar`, `uchar`, `uchar`)
 - Sets the function to call to draw a specific boxtype.*
 - static void `set_boxtype` (`Fl_Boxtype`, `Fl_Boxtype` from)
 - Copies the from boxtype.*
 - static void `set_color` (`Fl_Color` i, unsigned c)
 - Sets an entry in the `fl_color` index table.*
 - static void `set_color` (`Fl_Color`, `uchar`, `uchar`, `uchar`)
 - Sets an entry in the `fl_color` index table.*
 - static void `set_font` (`Fl_Font`, const char *)
 - Changes a face.*
 - static void `set_font` (`Fl_Font`, `Fl_Font`)
 - Copies one face to another.*
 - static `Fl_Font` `set_fonts` (const char *=0)
 - FLTK will open the display, and add every fonts on the server to the face table.*
 - static void `set_idle` (`Fl_Old_Idle_Handler` cb)
 - Sets an idle callback.*
 - static void `set_labeltype` (`Fl_Labeltype`, `Fl_Label_Draw_F *`, `Fl_Label_Measure_F *`)
 - Sets the functions to call to draw and measure a specific labeltype.*
 - static void `set_labeltype` (`Fl_Labeltype`, `Fl_Labeltype` from)
 - Sets the functions to call to draw and measure a specific labeltype.*
 - static int `test_shortcut` (`Fl_Shortcut`)
 - Tests the current event, which must be an `FL_KEYBOARD` or `FL_SHORTCUT`, against a shortcut value (described in `Fl_Button`).*
 - static void * `thread_message` ()
 - The `thread_message()` method returns the last message that was sent from a child by the `awake()` method.*
 - static void `unlock` ()
 - The `unlock()` method releases the lock that was set using the `lock()` method.*
 - static int `use_high_res_GL` ()
 - returns whether GL windows should be drawn at high resolution on Apple computers with retina displays.*
 - static void `use_high_res_GL` (int val)
 - sets whether GL windows should be drawn at high resolution on Apple computers with retina displays*
 - static double `version` ()
 - Returns the compiled-in value of the `FL_VERSION` constant.*
 - static int `visible_focus` ()
 - Gets or sets the visible keyboard focus on buttons and other non-text widgets.*
 - static void `visible_focus` (int v)
 - Gets or sets the visible keyboard focus on buttons and other non-text widgets.*
 - static int `visual` (int)
 - Selects a visual so that your graphics are drawn correctly.*
 - static int `w` ()
 - Returns the width in pixels of the main screen work area.*
 - static int `wait` ()
 - Waits until "something happens" and then returns.*
 - static double `wait` (double time)
 - See int `Fl::wait()`*
 - static void `watch_widget_pointer` (`Fl_Widget *&w`)
 - Adds a widget pointer to the widget watch list.*
 - static int `x` ()
 - Returns the leftmost x coordinate of the main screen work area.*
 - static int `y` ()
 - Returns the topmost y coordinate of the main screen work area.*

Static Public Attributes

- static void(* [atclose](#))(FI_Window *, void *)
Back compatibility: default window callback handler.
- static char const *const [clipboard_image](#) = "image"
Denotes image data.
- static char const *const [clipboard_plain_text](#) = "text/plain"
Denotes plain textual data.
- static void(* [error](#))(const char *,...) = ::error
FLTK calls [Fl::error\(\)](#) to output a normal error message.
- static void(* [fatal](#))(const char *,...) = ::fatal
FLTK calls [Fl::fatal\(\)](#) to output a fatal error message.
- static const char *const [help](#) = helpmsg+13
Usage string displayed if [Fl::args\(\)](#) detects an invalid argument.
- static void(* [idle](#))()
The currently executing idle callback function: DO NOT USE THIS DIRECTLY!
- static void(* [warning](#))(const char *,...) = ::warning
FLTK calls [Fl::warning\(\)](#) to output a warning message.

31.2.1 Detailed Description

The [Fl](#) is the FLTK global (static) class containing state information and global methods for the current application.

31.2.2 Member Enumeration Documentation

31.2.2.1 Fl_Option

enum [Fl::Fl_Option](#)

Enumerator for global FLTK options.

These options can be set system wide, per user, or for the running application only.

See also

[Fl::option\(Fl_Option, bool\)](#)

[Fl::option\(Fl_Option\)](#)

Enumerator

OPTION_ARROW_FOCUS	When switched on, moving the text cursor beyond the start or end of a text in a text widget will change focus to the next text widget. (This is considered 'old' behavior) When switched off (default), the cursor will stop at the end of the text. Pressing Tab or Ctrl-Tab will advance the keyboard focus. See also: Fl_Input_::tab_nav()
OPTION_VISIBLE_FOCUS	If visible focus is switched on (default), FLTK will draw a dotted rectangle inside the widget that will receive the next keystroke. If switched off, no such indicator will be drawn and keyboard navigation is disabled.
OPTION_DND_TEXT	If text drag-and-drop is enabled (default), the user can select and drag text from any text widget. If disabled, no dragging is possible, however dropping text from other applications still works.
OPTION_SHOW_TOOLTIPS	If tooltips are enabled (default), hovering the mouse over a widget with a tooltip text will open a little tooltip window until the mouse leaves the widget. If disabled, no tooltip is shown.

Enumerator

OPTION_FNFC_USES_GTK	When switched on (default), Fl_Native_File_Chooser runs GTK file dialogs if the GTK library is available on the platform (linux/unix only). When switched off, GTK file dialogs aren't used even if the GTK library is available.
OPTION_LAST	For internal use only.

31.2.3 Member Function Documentation

31.2.3.1 `abi_check()`

```
static int Fl::abi_check (
    const int val = FL_ABI_VERSION ) [inline], [static]
```

Returns whether the runtime library ABI version is correct.

This enables you to check the ABI version of the linked FLTK library at runtime.

Returns 1 (true) if the compiled ABI version (in the header files) and the linked library ABI version (used at runtime) are the same, 0 (false) otherwise.

Argument `val` can be used to query a particular library ABI version. Use for instance 10303 to query if the runtime library is compatible with FLTK ABI version 1.3.3. This is rarely useful.

The default `val` argument is `FL_ABI_VERSION`, which checks the version defined at configure time (i.e. in the header files at program compilation time) against the linked library version used at runtime. This is particularly useful if you linked with a shared object library, but it also concerns static linking.

See also

[Fl::abi_version\(\)](#)

31.2.3.2 `abi_version()`

```
int Fl::abi_version ( ) [static]
```

Returns the compiled-in value of the `FL_ABI_VERSION` constant.

This is useful for checking the version of a shared library.

31.2.3.3 `add_check()`

```
void Fl::add_check (
    Fl_Timeout_Handler cb,
    void * argp = 0 ) [static]
```

FLTK will call this callback just before it flushes the display and waits for events.

This is different than an idle callback because it is only called once, then FLTK calls the system and tells it not to return until an event happens.

This can be used by code that wants to monitor the application's state, such as to keep a display up to date. The advantage of using a check callback is that it is called only when no events are pending. If events are coming in quickly, whole blocks of them will be processed before this is called once. This can save significant time and avoid the application falling behind the events.

Sample code:

```
bool state_changed; // anything that changes the display turns this on
void callback(void*) {
    if (!state_changed) return;
    state_changed = false;
    do_expensive_calculation();
    widget->redraw();
}
main() {
    Fl::add_check(callback);
    return Fl::run();
}
```

31.2.3.4 add_fd()

```
static void Fl::add_fd (
    int fd,
    int when,
    Fl_FD_Handler cb,
    void * = 0 ) [static]
```

Adds file descriptor `fd` to listen to.

When the `fd` becomes ready for reading `Fl::wait()` will call the callback and then return. The callback is passed the `fd` and the arbitrary `void*` argument.

The second version takes a `when` bitfield, with the bits `FL_READ`, `FL_WRITE`, and `FL_EXCEPT` defined, to indicate when the callback should be done.

There can only be one callback of each type for a file descriptor. `Fl::remove_fd()` gets rid of *all* the callbacks for a given file descriptor.

Under UNIX *any* file descriptor can be monitored (files, devices, pipes, sockets, etc.). Due to limitations in Microsoft Windows, WIN32 applications can only monitor sockets.

31.2.3.5 add_idle()

```
void Fl::add_idle (
    Fl_Idle_Handler cb,
    void * data = 0 ) [static]
```

Adds a callback function that is called every time by `Fl::wait()` and also makes it act as though the timeout is zero (this makes `Fl::wait()` return immediately, so if it is in a loop it is called repeatedly, and thus the idle function is called repeatedly).

The idle function can be used to get background processing done.

You can have multiple idle callbacks. To remove an idle callback use `Fl::remove_idle()`.

`Fl::wait()` and `Fl::check()` call idle callbacks, but `Fl::ready()` does not.

The idle callback can call any FLTK functions, including `Fl::wait()`, `Fl::check()`, and `Fl::ready()`.

FLTK will not recursively call the idle callback.

31.2.3.6 add_timeout()

```
void Fl::add_timeout (
    double t,
    Fl_Timeout_Handler cb,
    void * argp = 0 ) [static]
```

Adds a one-shot timeout callback.

The function will be called by `Fl::wait()` at `t` seconds after this function is called. The optional `void*` argument is passed to the callback.

You can have multiple timeout callbacks. To remove a timeout callback use `Fl::remove_timeout()`.

If you need more accurate, repeated timeouts, use `Fl::repeat_timeout()` to reschedule the subsequent timeouts.

The following code will print "TICK" each second on stdout with a fair degree of accuracy:

```
#include <stdio.h>
#include <FL/Fl.H>
#include <FL/Fl_Window.H>
void callback(void*) {
    printf("TICK\n");
    Fl::repeat_timeout(1.0, callback);    // retrigger timeout
}
int main() {
    Fl_Window win(100,100);
    win.show();
    Fl::add_timeout(1.0, callback);    // set up first timeout
    return Fl::run();
}
```

31.2.3.7 api_version()

```
int Fl::api_version ( ) [static]
```

Returns the compiled-in value of the `FL_API_VERSION` constant.

This is useful for checking the version of a shared library.

31.2.3.8 `arg()`

```
int Fl::arg (
    int argc,
    char ** argv,
    int & i ) [static]
```

Parse a single switch from `argv`, starting at word `i`.

Returns the number of words eaten (1 or 2, or 0 if it is not recognized) and adds the same value to `i`.

This is the default argument handler used internally by `Fl::args(...)`, but you can use this function if you prefer to step through the standard FLTK switches yourself.

All standard FLTK switches except `-bg2` may be abbreviated to just one letter and case is ignored:

- `-bg color` or `-background color`
Sets the background color using [Fl::background\(\)](#).
- `-bg2 color` or `-background2 color`
Sets the secondary background color using [Fl::background2\(\)](#).
- `-display host:n.n`
Sets the X display to use; this option is silently ignored under WIN32 and MacOS.
- `-dnd` and `-nodnd`
Enables or disables drag and drop text operations using [Fl::dnd_text_ops\(\)](#).
- `-fg color` or `-foreground color`
Sets the foreground color using [Fl::foreground\(\)](#).
- `-geometry WxH+X+Y`
Sets the initial window position and size according to the standard X geometry string.
- `-iconic`
Iconifies the window using [Fl_Window::iconize\(\)](#).
- `-kbd` and `-nokbd`
Enables or disables visible keyboard focus for non-text widgets using [Fl::visible_focus\(\)](#).
- `-name string`
Sets the window class using [Fl_Window::xclass\(\)](#).
- `-scheme string`
Sets the widget scheme using [Fl::scheme\(\)](#).
- `-title string`
Sets the window title using [Fl_Window::label\(\)](#).
- `-tooltips` and `-notooltips`
Enables or disables tooltips using [Fl_Tooltip::enable\(\)](#).

If your program requires other switches in addition to the standard FLTK options, you will need to pass your own argument handler to [Fl::args\(int,char**,int&,Fl_Args_Handler\)](#) explicitly.

31.2.3.9 `args()` [1/2]

```
void Fl::args (
    int argc,
    char ** argv ) [static]
```

Parse all command line switches matching standard FLTK options only.

It parses all the switches, and if any are not recognized it calls `Fl::abort(Fl::help)`, i.e. unlike the long form, an unrecognized switch generates an error message and causes the program to exit.

31.2.3.10 args() [2/2]

```
int Fl::args (
    int argc,
    char ** argv,
    int & i,
    Fl_Args_Handler cb = 0 ) [static]
```

Parse command line switches using the `cb` argument handler.

Returns 0 on error, or the number of words processed.

FLTK provides this as an *entirely optional* command line switch parser. You don't have to call it if you don't want to. Everything it can do can be done with other calls to FLTK.

To use the switch parser, call `Fl::args(...)` near the start of your program. This does **not** open the display, instead switches that need the display open are stashed into static variables. Then you **must** display your first window by calling `window->show(argc, argv)`, which will do anything stored in the static variables.

Providing an argument handler callback `cb` lets you define your own switches. It is called with the same `argc` and `argv`, and with `i` set to the index of the switch to be processed. The `cb` handler should return zero if the switch is unrecognized, and not change `i`. It should return non-zero to indicate the number of words processed if the switch is recognized, i.e. 1 for just the switch, and more than 1 for the switch plus associated parameters. `i` should be incremented by the same amount.

The `cb` handler is called **before** any other tests, so *you can also override any standard FLTK switch* (this is why FLTK can use very short switches instead of the long ones all other toolkits force you to use). See [Fl::arg\(\)](#) for descriptions of the standard switches.

On return `i` is set to the index of the first non-switch. This is either:

- The first word that does not start with '-'.
- The word '-' (used by many programs to name stdin as a file)
- The first unrecognized switch (return value is 0).
- `argc`

The return value is `i` unless an unrecognized switch is found, in which case it is zero. If your program takes no arguments other than switches you should produce an error if the return value is less than `argc`.

A usage string is displayed if `Fl::args()` detects an invalid argument on the command-line. You can change the message by setting the `Fl::help` pointer.

A very simple command line parser can be found in `examples/howto-parse-args.cxx`

The simpler `Fl::args(int argc, char **argv)` form is useful if your program does not have command line switches of its own.

31.2.3.11 background()

```
void Fl::background (
    uchar r,
    uchar g,
    uchar b ) [static]
```

Changes `fl_color(FL_BACKGROUND_COLOR)` to the given color, and changes the gray ramp from 32 to 56 to black to white.

These are the colors used as backgrounds by almost all widgets and used to draw the edges of all the boxtypes.

31.2.3.12 background2()

```
void Fl::background2 (
    uchar r,
    uchar g,
    uchar b ) [static]
```

Changes the alternative background color.

This color is used as a background by [Fl_Input](#) and other text widgets.

This call may change `fl_color(FL_FOREGROUND_COLOR)` if it does not provide sufficient contrast to `FL_BACKGROUND2_COLOR`.

31.2.3.13 box_color()

```
Fl_Color Fl::box_color (
    Fl_Color c ) [static]
```

Gets the drawing color to be used for the background of a box.

This method is only useful inside box drawing code. It returns the color to be used, either `fl_inactive(c)` if the widget is `inactive_r()` or `c` otherwise.

31.2.3.14 box_dh()

```
int Fl::box_dh (
    Fl_Boxtype t ) [static]
```

Returns the height offset for the given boxtype.

See also

[box_dy\(\)](#).

31.2.3.15 box_dw()

```
int Fl::box_dw (
    Fl_Boxtype t ) [static]
```

Returns the width offset for the given boxtype.

See also

[box_dy\(\)](#).

31.2.3.16 box_dx()

```
int Fl::box_dx (
    Fl_Boxtype t ) [static]
```

Returns the X offset for the given boxtype.

See also

[box_dy\(\)](#)

31.2.3.17 box_dy()

```
int Fl::box_dy (
    Fl_Boxtype t ) [static]
```

Returns the Y offset for the given boxtype.

These functions return the offset values necessary for a given boxtype, useful for computing the area inside a box's borders, to prevent overdrawing the borders.

For instance, in the case of a boxtype like `FL_DOWN_BOX` where the border width might be 2 pixels all around, the above functions would return 2, 2, 4, and 4 for `box_dx`, `box_dy`, `box_dw`, and `box_dh` respectively.

An example to compute the area inside a widget's box():

```
int X = yourwidget->x() + Fl::box_dx(yourwidget->box());
int Y = yourwidget->y() + Fl::box_dy(yourwidget->box());
int W = yourwidget->w() - Fl::box_dw(yourwidget->box());
int H = yourwidget->h() - Fl::box_dh(yourwidget->box());
```

These functions are mainly useful in the `draw()` code for deriving custom widgets, where one wants to avoid drawing over the widget's own border `box()`.

31.2.3.18 check()

```
int Fl::check ( ) [static]
```

Same as `Fl::wait(0)`.

Calling this during a big calculation will keep the screen up to date and the interface responsive:

```
while (!calculation_done()) {
  calculate();
  Fl::check();
  if (user_hit_abort_button()) break;
}
```

This returns non-zero if any windows are displayed, and 0 if no windows are displayed (this is likely to change in future versions of FLTK).

31.2.3.19 display()

```
void Fl::display (
    const char * d ) [static]
```

Sets the X display to use for all windows.

Actually this just sets the environment variable `$DISPLAY` to the passed string, so this only works before you `show()` the first window or otherwise open the display, and does nothing useful under WIN32.

31.2.3.20 dnd_text_ops() [1/2]

```
static int Fl::dnd_text_ops ( ) [inline], [static]
```

Gets or sets whether drag and drop text operations are supported.

This specifically affects whether selected text can be dragged from text fields or dragged within a text field as a cut/paste shortcut.

31.2.3.21 dnd_text_ops() [2/2]

```
static void Fl::dnd_text_ops (
    int v ) [inline], [static]
```

Gets or sets whether drag and drop text operations are supported.

This specifically affects whether selected text can be dragged from text fields or dragged within a text field as a cut/paste shortcut.

31.2.3.22 draw_box_active()

```
int Fl::draw_box_active ( ) [static]
```

Determines if the currently drawn box is active or inactive.

If inactive, the box color should be changed to the inactive color.

See also

[Fl::box_color\(Fl_Color c\)](#)

31.2.3.23 flush()

```
void Fl::flush ( ) [static]
```

Causes all the windows that need it to be redrawn and graphics forced out through the pipes.

This is what `wait()` does before looking for events.

Note: in multi-threaded applications you should only call `Fl::flush()` from the main thread. If a child thread needs to trigger a redraw event, it should instead call `Fl::awake()` to get the main thread to process the event queue.

31.2.3.24 get_system_colors()

```
void Fl::get_system_colors ( ) [static]
```

Read the user preference colors from the system and use them to call `Fl::foreground()`, `Fl::background()`, and `Fl::background2()`.

This is done by `Fl_Window::show(argc,argv)` before applying the `-fg` and `-bg` switches.

On X this reads some common values from the Xdefaults database. KDE users can set these values by running the "krdb" program, and newer versions of KDE set this automatically if you check the "apply style to other X programs" switch in their control panel.

31.2.3.25 gl_visual()

```
int Fl::gl_visual (
    int mode,
    int * alist = 0 ) [static]
```

This does the same thing as [Fl::visual\(int\)](#) but also requires OpenGL drawing to work.

This *must* be done if you want to draw in normal windows with OpenGL with [gl_start\(\)](#) and [gl_end\(\)](#). It may be useful to call this so your X windows use the same visual as an [Fl_Gl_Window](#), which on some servers will reduce colormap flashing.

See [Fl_Gl_Window](#) for a list of additional values for the argument.

31.2.3.26 is_scheme()

```
static int Fl::is_scheme (
    const char * name ) [inline], [static]
```

Returns whether the current scheme is the given name.

This is a fast inline convenience function to support scheme-specific code in widgets, e.g. in their [draw\(\)](#) methods, if required.

Use a valid scheme name, not NULL (although NULL is allowed, this is not a useful argument - see below).

If [Fl::scheme\(\)](#) has not been set or has been set to the default scheme ("none" or "base"), then this will always return 0 regardless of the argument, because [Fl::scheme\(\)](#) is NULL in this case.

Note

The stored scheme name is always lowercase, and this method will do a case-sensitive compare, so you **must** provide a lowercase string to return the correct value. This is intentional for performance reasons.

Example:

```
if ( Fl::is_scheme("gtk+") ) { your_code_here(); }
```

Parameters

in	<i>name</i>	lowercase string of requested scheme name.
----	-------------	--

Returns

1 if the given scheme is active, 0 otherwise.

See also

[Fl::scheme\(const char *name\)](#)

31.2.3.27 option() [1/2]

```
bool Fl::option (
    Fl_Option opt ) [static]
```

FLTK library options management.

This function needs to be documented in more detail. It can be used for more optional settings, such as using a native file chooser instead of the FLTK one wherever possible, disabling tooltips, disabling visible focus, disabling FLTK file chooser preview, etc. .

There should be a command line option interface.

There should be an application that manages options system wide, per user, and per application.

Example:

```
if ( Fl::option(Fl::OPTION_ARROW_FOCUS) )
    { ..on.. }
else
    { ..off.. }
```

Note

As of FLTK 1.3.0, options can be managed within fluid, using the menu *Edit/Global FLTK Settings*.

Parameters

<i>opt</i>	which option
------------	--------------

Returns

true or false

See also

enum [Fl::Fl_Option](#)
[Fl::option\(Fl_Option, bool\)](#)

Since

FLTK 1.3.0

31.2.3.28 option() [2/2]

```
void Fl::option (
    Fl_Option opt,
    bool val ) [static]
```

Override an option while the application is running.
 This function does not change any system or user settings.

Example:

```
Fl::option(Fl::OPTION_ARROW_FOCUS, true);    // on
Fl::option(Fl::OPTION_ARROW_FOCUS, false);  // off
```

Parameters

<i>opt</i>	which option
<i>val</i>	set to true or false

See also

enum [Fl::Fl_Option](#)
 bool [Fl::option\(Fl_Option\)](#)

31.2.3.29 own_colormap()

```
void Fl::own_colormap ( ) [static]
```

Makes FLTK use its [own colormap](#).

This may make FLTK display better and will reduce conflicts with other programs that want lots of colors. However the colors may flash as you move the cursor between windows.

This does nothing if the current visual is not colormapped.

31.2.3.30 readqueue()

```
Fl_Widget * Fl::readqueue ( ) [static]
```

Reads the default callback queue and returns the first widget.

All `Fl_Widgets` that don't have a callback defined use the default callback `static Fl_Widget::default_callback()` that puts a pointer to the widget in a queue. This method reads the oldest widget out of this queue. The queue (FIFO) is limited (currently 20 items). If the queue overflows, the oldest entry (`Fl_Widget *`) is discarded. Relying on the default callback and reading the callback queue with `Fl::readqueue()` is not recommended. If you need a callback, you should set one with `Fl_Widget::callback(Fl_Callback *cb, void *data)` or one of its variants.

See also

[Fl_Widget::callback\(\)](#)
[Fl_Widget::callback\(Fl_Callback *cb, void *data\)](#)
[Fl_Widget::default_callback\(\)](#)

31.2.3.31 ready()

```
int Fl::ready ( ) [static]
```

This is similar to `Fl::check()` except this does *not* call `Fl::flush()` or any callbacks, which is useful if your program is in a state where such callbacks are illegal.

This returns true if `Fl::check()` would do anything (it will continue to return true until you call `Fl::check()` or `Fl::wait()`).

```
while (!calculation_done()) {
    calculate();
    if (Fl::ready()) {
        do_expensive_cleanup();
        Fl::check();
        if (user_hit_abort_button()) break;
    }
}
```

31.2.3.32 release()

```
static void Fl::release ( ) [inline], [static]
```

Releases the current grabbed window, equals `grab(0)`.

Deprecated Use `Fl::grab(0)` instead.

See also

[grab\(Fl_Window*\)](#)

31.2.3.33 reload_scheme()

```
int Fl::reload_scheme ( ) [static]
```

Called by scheme according to scheme name.

Loads or reloads the current scheme selection. See void [scheme\(const char *name\)](#)

31.2.3.34 remove_check()

```
void Fl::remove_check (
    Fl_Timeout_Handler cb,
    void * argp = 0 ) [static]
```

Removes a check callback.

It is harmless to remove a check callback that no longer exists.

31.2.3.35 remove_timeout()

```
void Fl::remove_timeout (
    Fl_Timeout_Handler cb,
    void * argp = 0 ) [static]
```

Removes a timeout callback.

It is harmless to remove a timeout callback that no longer exists.

Note

This version removes all matching timeouts, not just the first one. This may change in the future.

31.2.3.36 repeat_timeout()

```
void Fl::repeat_timeout (
    double t,
    Fl_Timeout_Handler cb,
    void * argp = 0 ) [static]
```

Repeats a timeout callback from the expiration of the previous timeout, allowing for more accurate timing.

You may only call this method inside a timeout callback.

The following code will print "TICK" each second on stdout with a fair degree of accuracy:

```
void callback(void*) {
    puts("TICK");
    Fl::repeat_timeout(1.0, callback);
}
int main() {
    Fl::add_timeout(1.0, callback);
    return Fl::run();
}
```

31.2.3.37 run()

```
int Fl::run ( ) [static]
```

As long as any windows are displayed this calls [Fl::wait\(\)](#) repeatedly.

When all the windows are closed it returns zero (supposedly it would return non-zero on any errors, but FLTK calls exit directly for these). A normal program will end main() with return [Fl::run\(\)](#);

31.2.3.38 scheme()

```
int Fl::scheme (
    const char * s ) [static]
```

Sets the current widget scheme.

NULL will use the scheme defined in the FLTK_SCHEME environment variable or the scheme resource under X11.

Otherwise, any of the following schemes can be used:

- "none" - This is the default look-n-feel which resembles old Windows (95/98/Me/NT/2000) and old GTK/KDE
- "base" - This is an alias for "none"
- "plastic" - This scheme is inspired by the Aqua user interface on Mac OS X
- "gtk+" - This scheme is inspired by the Red Hat Bluecurve theme
- "gleam" - This scheme is inspired by the Clearlooks Glossy scheme. (Colin Jones and Edmanuel Torres).

Uppercase scheme names are equivalent, but the stored scheme name will always be lowercase and [Fl::scheme\(\)](#) will return this lowercase name.

If the resulting scheme name is not defined, the default scheme will be used and [Fl::scheme\(\)](#) will return NULL.

See also

[Fl::is_scheme\(\)](#)

31.2.3.39 scrollbar_size() [1/2]

```
int Fl::scrollbar_size ( ) [static]
```

Gets the default scrollbar size used by [Fl_Browser_](#), [Fl_Help_View](#), [Fl_Scroll](#), and [Fl_Text_Display](#) widgets.

Returns

The default size for widget scrollbars, in pixels.

31.2.3.40 scrollbar_size() [2/2]

```
void Fl::scrollbar_size (
    int W ) [static]
```

Sets the default scrollbar size that is used by the [Fl_Browser_](#), [Fl_Help_View](#), [Fl_Scroll](#), and [Fl_Text_Display](#) widgets.

Parameters

in	<i>W</i>	The new default size for widget scrollbars, in pixels.
----	----------	--

31.2.3.41 set_box_color()

```
void Fl::set_box_color (
    Fl_Color c ) [static]
```

Sets the drawing color for the box that is currently drawn.

This method sets the current drawing color [fl_color\(\)](#) depending on the widget's state to either *c* or [fl_inactive\(c\)](#).

It should be used whenever a box background is drawn in the box (type) drawing code instead of calling [fl_color\(Fl_Color bg\)](#) with the background color *bg*, usually [Fl_Widget::color\(\)](#).

This method is only useful inside box drawing code. Whenever a box is drawn with one of the standard box drawing methods, a static variable is set depending on the widget's current state - if the widget is [inactive_r\(\)](#) then the internal variable is false (0), otherwise it is true (1). This is faster than calling [Fl_Widget::active_r\(\)](#) because the state is cached.

See also

[Fl::draw_box_active\(\)](#)

[Fl::box_color\(Fl_Color\)](#)

31.2.3.42 set_idle()

```
static void Fl::set_idle (
    Fl_Old_Idle_Handler cb ) [inline], [static]
```

Sets an idle callback.

Deprecated This method is obsolete - use the [add_idle\(\)](#) method instead.

31.2.3.43 use_high_res_GL() [1/2]

```
static int Fl::use_high_res_GL ( ) [inline], [static]
```

returns whether GL windows should be drawn at high resolution on Apple computers with retina displays.

Default is no.

Version

1.3.4

31.2.3.44 use_high_res_GL() [2/2]

```
static void Fl::use_high_res_GL (
    int val ) [inline], [static]
```

sets whether GL windows should be drawn at high resolution on Apple computers with retina displays

Version

1.3.4

31.2.3.45 version()

```
double Fl::version ( ) [static]
```

Returns the compiled-in value of the FL_VERSION constant.

This is useful for checking the version of a shared library.

Deprecated Use int [Fl::api_version\(\)](#) instead.

31.2.3.46 visible_focus() [1/2]

```
static int Fl::visible_focus ( ) [inline], [static]
```

Gets or sets the visible keyboard focus on buttons and other non-text widgets.

The default mode is to enable keyboard focus for all widgets.

31.2.3.47 visible_focus() [2/2]

```
static void Fl::visible_focus (
    int v ) [inline], [static]
```

Gets or sets the visible keyboard focus on buttons and other non-text widgets.

The default mode is to enable keyboard focus for all widgets.

31.2.3.48 visual()

```
int Fl::visual (
    int flags ) [static]
```

Selects a visual so that your graphics are drawn correctly.

This is only allowed before you call `show()` on any windows. This does nothing if the default visual satisfies the capabilities, or if no visual satisfies the capabilities, or on systems that don't have such brain-dead notions.

Only the following combinations do anything useful:

- `Fl::visual(FL_RGB)`
Full/true color (if there are several depths FLTK chooses the largest). Do this if you use `fl_draw_image` for much better (non-dithered) output.
- `Fl::visual(FL_RGB8)`
Full color with at least 24 bits of color. `FL_RGB` will always pick this if available, but if not it will happily return a less-than-24 bit deep visual. This call fails if 24 bits are not available.
- `Fl::visual(FL_DOUBLE|FL_INDEX)`
Hardware double buffering. Call this if you are going to use [Fl_Double_Window](#).
- `Fl::visual(FL_DOUBLE|FL_RGB)`
- `Fl::visual(FL_DOUBLE|FL_RGB8)`
Hardware double buffering and full color.

This returns true if the system has the capabilities by default or FLTK succeeded in turning them on. Your program will still work even if this returns false (it just won't look as good).

31.2.3.49 wait()

```
int Fl::wait ( ) [static]
```

Waits until "something happens" and then returns.

Call this repeatedly to "run" your program. You can also check what happened each time after this returns, which is quite useful for managing program state.

What this really does is call all idle callbacks, all elapsed timeouts, call [Fl::flush\(\)](#) to get the screen to update, and then wait some time (zero if there are idle callbacks, the shortest of all pending timeouts, or infinity), for any events from the user or any [Fl::add_fd\(\)](#) callbacks. It then handles the events and calls the callbacks and then returns.

The return value of [Fl::wait\(\)](#) is non-zero if there are any visible windows - this may change in future versions of FLTK.

[Fl::wait\(time\)](#) waits a maximum of *time* seconds. *It can return much sooner if something happens.*

The return value is positive if an event or fd happens before the time elapsed. It is zero if nothing happens (on Win32 this will only return zero if *time* is zero). It is negative if an error occurs (this will happen on UNIX if a signal happens).

31.2.4 Member Data Documentation

31.2.4.1 help

```
const char *const Fl::help = helpmsg+13 [static]
```

Usage string displayed if [Fl::args\(\)](#) detects an invalid argument.

This may be changed to point to customized text at run-time.

31.2.4.2 idle

```
void(* Fl::idle)() [static]
```

The currently executing idle callback function: DO NOT USE THIS DIRECTLY!

This is now used as part of a higher level system allowing multiple idle callback functions to be called.

See also

[add_idle\(\)](#), [remove_idle\(\)](#)

The documentation for this class was generated from the following files:

- [Fl.H](#)
- [Fl.cxx](#)
- [Fl_abort.cxx](#)
- [Fl_add_idle.cxx](#)
- [Fl_arg.cxx](#)
- [fl_boxtype.cxx](#)
- [fl_color.cxx](#)
- [fl_color_mac.cxx](#)
- [fl_color_win32.cxx](#)
- [Fl_compose.cxx](#)
- [Fl_display.cxx](#)
- [fl_dnd_win32.cxx](#)
- [fl_dnd_x.cxx](#)
- [Fl_get_key.cxx](#)
- [Fl_get_key_mac.cxx](#)
- [Fl_get_key_win32.cxx](#)
- [Fl_get_system_colors.cxx](#)
- [Fl_grab.cxx](#)
- [fl_labeltype.cxx](#)
- [Fl_lock.cxx](#)
- [Fl_own_colormap.cxx](#)

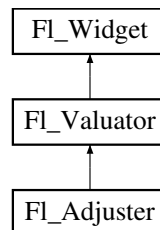
- fl_set_font.cxx
- fl_set_fonts_mac.cxx
- fl_set_fonts_win32.cxx
- fl_set_fonts_x.cxx
- fl_set_fonts_xft.cxx
- fl_shortcut.cxx
- FL_visual.cxx
- FL_Widget.cxx
- FL_Window.cxx
- gl_start.cxx
- screen_xywh.cxx
- FL_Cairo.cxx

31.3 FL_Adjuster Class Reference

The [FL_Adjuster](#) widget was stolen from Prisms, and has proven to be very useful for values that need a large dynamic range.

```
#include <FL_Adjuster.H>
```

Inheritance diagram for FL_Adjuster:



Public Member Functions

- [FL_Adjuster](#) (int X, int Y, int W, int H, const char *l=0)
Creates a new [FL_Adjuster](#) widget using the given position, size, and label string.
- int [soft](#) () const
If "soft" is turned on, the user is allowed to drag the value outside the range.
- void [soft](#) (int s)
If "soft" is turned on, the user is allowed to drag the value outside the range.

Protected Member Functions

- void [draw](#) ()
Draws the widget.
- int [handle](#) (int)
Handles the specified event.
- void [value_damage](#) ()
Asks for partial redraw.

Additional Inherited Members

31.3.1 Detailed Description

The [FL_Adjuster](#) widget was stolen from Prisms, and has proven to be very useful for values that need a large dynamic range.

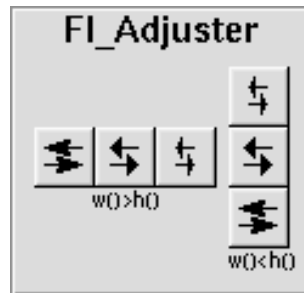


Figure 31.1 FL_Adjuster

When you press a button and drag to the right the value increases. When you drag to the left it decreases. The largest button adjusts by $100 * \text{step}()$, the next by $10 * \text{step}()$ and that smallest button by $\text{step}()$. Clicking on the buttons increments by 10 times the amount dragging by a pixel does. Shift + click decrements by 10 times the amount.

31.3.2 Constructor & Destructor Documentation

31.3.2.1 FL_Adjuster()

```
Fl_Adjuster::Fl_Adjuster (
    int X,
    int Y,
    int W,
    int H,
    const char * L = 0 )
```

Creates a new [FL_Adjuster](#) widget using the given position, size, and label string. It looks best if one of the dimensions is 3 times the other. Inherited destructor destroys the Valuator.

31.3.3 Member Function Documentation

31.3.3.1 draw()

```
void Fl_Adjuster::draw ( ) [protected], [virtual]
```

Draws the widget.

Never call this function directly. FLTK will schedule redrawing whenever needed. If your widget must be redrawn as soon as possible, call [redraw\(\)](#) instead.

Override this function to draw your own widgets.

If you ever need to call another widget's draw method *from within your own draw() method*, e.g. for an embedded scrollbar, you can do it (because [draw\(\)](#) is virtual) like this:

```
Fl_Widget *s = &scroll;           // scroll is an embedded Fl_Scrollbar
s->draw();                         // calls Fl_Scrollbar::draw()
```

Implements [FL_Widget](#).

31.3.3.2 handle()

```
int Fl_Adjuster::handle (
    int event ) [protected], [virtual]
```

Handles the specified event.

You normally don't call this method directly, but instead let FLTK do it when the user interacts with the widget.

When implemented in a widget, this function must return 0 if the widget does not use the event or 1 otherwise.

Most of the time, you want to call the inherited [handle\(\)](#) method in your overridden method so that you don't short-circuit events that you don't handle. In this last case you should return the callee retval.

Parameters

<code>in</code>	<code>event</code>	the kind of event received
-----------------	--------------------	----------------------------

Return values

<code>0</code>	if the event was not used or understood
<code>1</code>	if the event was used and can be deleted

See also

[Fl_Event](#)

Reimplemented from [Fl_Widget](#).

31.3.3.3 soft() [1/2]

```
int Fl_Adjuster::soft ( ) const [inline]
```

If "soft" is turned on, the user is allowed to drag the value outside the range.

If they drag the value to one of the ends, let go, then grab again and continue to drag, they can get to any value. Default is one.

31.3.3.4 soft() [2/2]

```
void Fl_Adjuster::soft (
    int s ) [inline]
```

If "soft" is turned on, the user is allowed to drag the value outside the range.

If they drag the value to one of the ends, let go, then grab again and continue to drag, they can get to any value. Default is one.

31.3.3.5 value_damage()

```
void Fl_Adjuster::value_damage ( ) [protected], [virtual]
```

Asks for partial redraw.

Reimplemented from [Fl_Valuator](#).

The documentation for this class was generated from the following files:

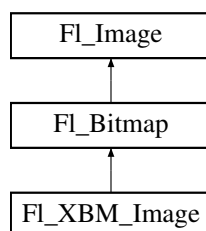
- `Fl_Adjuster.H`
- `Fl_Adjuster.cxx`

31.4 Fl_Bitmap Class Reference

The [Fl_Bitmap](#) class supports caching and drawing of mono-color (bitmap) images.

```
#include <Fl_Bitmap.H>
```

Inheritance diagram for [Fl_Bitmap](#):



Public Member Functions

- [Fl_Image](#) * **copy** ()
- virtual [Fl_Image](#) * **copy** (int W, int H)

The copy() method creates a copy of the specified image.
- void **draw** (int X, int Y)
- virtual void **draw** (int X, int Y, int W, int H, int cx=0, int cy=0)

Draws the image with a bounding box.
- **Fl_Bitmap** (const char *bits, int W, int H)

The constructors create a new bitmap from the specified bitmap data.
- **Fl_Bitmap** (const [uchar](#) *bits, int W, int H)

The constructors create a new bitmap from the specified bitmap data.
- virtual void **label** ([Fl_Menu_Item](#) *m)

The label() methods are an obsolete way to set the image attribute of a widget or menu item.
- virtual void **label** ([Fl_Widget](#) *w)

The label() methods are an obsolete way to set the image attribute of a widget or menu item.
- virtual void **uncache** ()

If the image has been cached for display, delete the cache data.
- virtual ~**Fl_Bitmap** ()

The destructor frees all memory and server resources that are used by the bitmap.

Public Attributes

- int **alloc_array**

Non-zero if array points to bitmap data allocated internally.
- const [uchar](#) * **array**

pointer to raw bitmap data

Friends

- class [Fl_GDI_Graphics_Driver](#)
- class [Fl_GDI_Printer_Graphics_Driver](#)
- class [Fl_Quartz_Graphics_Driver](#)
- class [Fl_Xlib_Graphics_Driver](#)

Additional Inherited Members

31.4.1 Detailed Description

The [Fl_Bitmap](#) class supports caching and drawing of mono-color (bitmap) images. Images are drawn using the current color.

31.4.2 Member Function Documentation

31.4.2.1 copy()

```
Fl_Image * Fl_Bitmap::copy (
    int W,
    int H ) [virtual]
```

The copy() method creates a copy of the specified image.

If the width and height are provided, the image is resized to the specified size. The image should be deleted (or in the case of [Fl_Shared_Image](#), released) when you are done with it.

Reimplemented from [Fl_Image](#).

31.4.2.2 draw()

```
void Fl_Bitmap::draw (
    int X,
    int Y,
    int W,
    int H,
    int cx = 0,
    int cy = 0 ) [virtual]
```

Draws the image with a bounding box.

Arguments *X*, *Y*, *W*, *H* specify a bounding box for the image, with the origin (upper-left corner) of the image offset by the *cx* and *cy* arguments.

In other words: `fl_push_clip(X, Y, W, H)` is applied, the image is drawn with its upper-left corner at *X-cx*, *Y-cy* and its own width and height, `fl_pop_clip()` is applied.

Reimplemented from [Fl_Image](#).

31.4.2.3 label() [1/2]

```
void Fl_Bitmap::label (
    Fl_Menu_Item * m ) [virtual]
```

The `label()` methods are an obsolete way to set the image attribute of a widget or menu item.

Use the `image()` or `deimage()` methods of the [Fl_Widget](#) and [Fl_Menu_Item](#) classes instead.

Reimplemented from [Fl_Image](#).

31.4.2.4 label() [2/2]

```
void Fl_Bitmap::label (
    Fl_Widget * widget ) [virtual]
```

The `label()` methods are an obsolete way to set the image attribute of a widget or menu item.

Use the `image()` or `deimage()` methods of the [Fl_Widget](#) and [Fl_Menu_Item](#) classes instead.

Reimplemented from [Fl_Image](#).

31.4.2.5 uncache()

```
void Fl_Bitmap::uncache ( ) [virtual]
```

If the image has been cached for display, delete the cache data.

This allows you to change the data used for the image and then redraw it without recreating an image object.

Reimplemented from [Fl_Image](#).

The documentation for this class was generated from the following files:

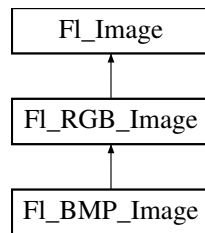
- [Fl_Bitmap.H](#)
- [Fl_Bitmap.cxx](#)

31.5 FI_BMP_Image Class Reference

The [FI_BMP_Image](#) class supports loading, caching, and drawing of Windows Bitmap (BMP) image files.

```
#include <Fl_BMP_Image.H>
```

Inheritance diagram for [FI_BMP_Image](#):



Public Member Functions

- [FI_BMP_Image](#) (const char *filename)
The constructor loads the named BMP image from the given bmp filename.

Additional Inherited Members

31.5.1 Detailed Description

The [FI_BMP_Image](#) class supports loading, caching, and drawing of Windows Bitmap (BMP) image files.

31.5.2 Constructor & Destructor Documentation

31.5.2.1 FI_BMP_Image()

```

FI_BMP_Image::FI_BMP_Image (
    const char * bmp )
  
```

The constructor loads the named BMP image from the given bmp filename.

The destructor frees all memory and server resources that are used by the image.

Use [FI_Image::fail\(\)](#) to check if [FI_BMP_Image](#) failed to load. [fail\(\)](#) returns `ERR_FILE_ACCESS` if the file could not be opened or read, `ERR_FORMAT` if the BMP format could not be decoded, and `ERR_NO_IMAGE` if the image could not be loaded for another reason.

The documentation for this class was generated from the following files:

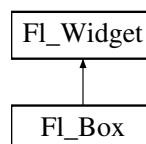
- FI_BMP_Image.H
- FI_BMP_Image.cxx

31.6 FI_Box Class Reference

This widget simply draws its box, and possibly its label.

```
#include <Fl_Box.H>
```

Inheritance diagram for FI_Box:



Public Member Functions

- [FI_Box](#) ([FI_Boxtype](#) b, int X, int Y, int W, int H, const char *)
See [FI_Box::FI_Box\(int x, int y, int w, int h, const char * = 0\)](#)
- [FI_Box](#) (int X, int Y, int W, int H, const char *|=0)
- virtual int [handle](#) (int)
Handles the specified event.

Protected Member Functions

- void `draw()`
Draws the widget.

Additional Inherited Members

31.6.1 Detailed Description

This widget simply draws its box, and possibly its label.

Putting it before some other widgets and making it big enough to surround them will let you draw a frame around them.

31.6.2 Constructor & Destructor Documentation

31.6.2.1 Fl_Box()

```
Fl_Box::Fl_Box (
    int X,
    int Y,
    int W,
    int H,
    const char * l = 0 )
```

- The first constructor sets `box()` to `FL_NO_BOX`, which means it is invisible. However such widgets are useful as placeholders or `Fl_Group::resizable()` values. To change the box to something visible, use `box(n)`.
- The second form of the constructor sets the box to the specified box type.

The destructor removes the box.

31.6.3 Member Function Documentation

31.6.3.1 draw()

```
void Fl_Box::draw ( ) [protected], [virtual]
```

Draws the widget.

Never call this function directly. FLTK will schedule redrawing whenever needed. If your widget must be redrawn as soon as possible, call `redraw()` instead.

Override this function to draw your own widgets.

If you ever need to call another widget's draw method *from within your own draw() method*, e.g. for an embedded scrollbar, you can do it (because `draw()` is virtual) like this:

```
Fl_Widget *s = &scroll;           // scroll is an embedded Fl_Scrollbar
s->draw();                          // calls Fl_Scrollbar::draw()
```

Implements `Fl_Widget`.

31.6.3.2 handle()

```
int Fl_Box::handle (
    int event ) [virtual]
```

Handles the specified event.

You normally don't call this method directly, but instead let FLTK do it when the user interacts with the widget.

When implemented in a widget, this function must return 0 if the widget does not use the event or 1 otherwise.

Most of the time, you want to call the inherited `handle()` method in your overridden method so that you don't short-circuit events that you don't handle. In this last case you should return the callee retval.

Parameters

in	<i>event</i>	the kind of event received
----	--------------	----------------------------

Return values

0	if the event was not used or understood
1	if the event was used and can be deleted

See also

[Fl_Event](#)

Reimplemented from [Fl_Widget](#).

The documentation for this class was generated from the following files:

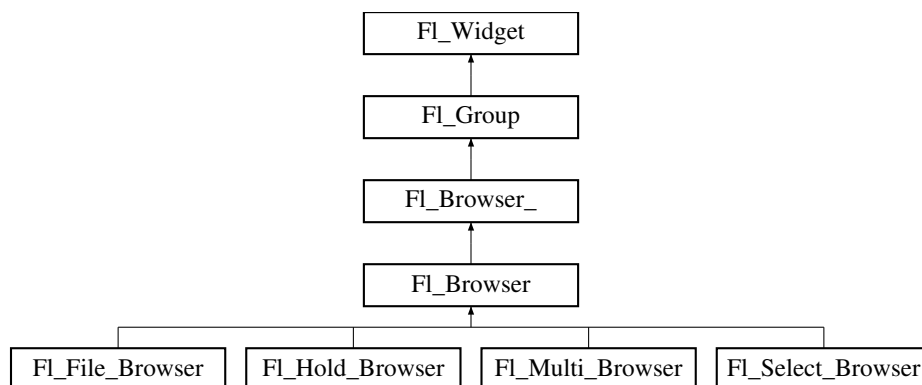
- [Fl_Box.H](#)
- [Fl_Box.cxx](#)

31.7 Fl_Browser Class Reference

The [Fl_Browser](#) widget displays a scrolling list of text lines, and manages all the storage for the text.

```
#include <Fl_Browser.H>
```

Inheritance diagram for [Fl_Browser](#):



Public Types

- enum [Fl_Line_Position](#) { **TOP** , **BOTTOM** , **MIDDLE** }

For internal use only?

Public Member Functions

- void [add](#) (const char *newtext, void *d=0)
Adds a new line to the end of the browser.
- void [bottomline](#) (int line)
Scrolls the browser so the bottom item in the browser is showing the specified line.
- void [clear](#) ()
Removes all the lines in the browser.
- char [column_char](#) () const
Gets the current column separator character.
- void [column_char](#) (char c)

- Sets the column separator to `c`.*

 - `const int * column_widths () const`
- Gets the current column width array.*

 - `void column_widths (const int *arr)`
- Sets the current array to `arr`.*

 - `void * data (int line) const`
- Returns the user `data()` for specified `line`.*

 - `void data (int line, void *d)`
- Sets the user data for specified `line` to `d`.*

 - `void display (int line, int val=1)`
- For back compatibility.*

 - `int displayed (int line) const`
- Returns non-zero if `line` has been scrolled to a position where it is being displayed.*

 - `FI_Browser (int X, int Y, int W, int H, const char *L=0)`
- The constructor makes an empty browser.*

 - `char format_char () const`
- Gets the current format code prefix character, which by default is '@'.*

 - `void format_char (char c)`
- Sets the current format code prefix character to `c`.*

 - `void hide ()`
- Hides the entire `FI_Browser` widget – opposite of `show()`.*

 - `void hide (int line)`
- Makes `line` invisible, preventing selection by the user.*

 - `FI_Image * icon (int line) const`
- Returns the icon currently defined for `line`.*

 - `void icon (int line, FI_Image *icon)`
- Set the image icon for `line` to the value `icon`.*

 - `void insert (int line, const char *newtext, void *d=0)`
- Insert a new entry whose label is `newtext` above given `line`, optional data `d`.*

 - `void lineposition (int line, FI_Line_Position pos)`
- Updates the browser so that `line` is shown at position `pos`.*

 - `int load (const char *filename)`
- Clears the browser and reads the file, adding each line from the file to the browser.*

 - `void make_visible (int line)`
- Make the item at the specified `line` `visible()`.*

 - `void middleline (int line)`
- Scrolls the browser so the middle item in the browser is showing the specified `line`.*

 - `void move (int to, int from)`
- Line `from` is removed and reinserted at `to`.*

 - `void remove (int line)`
- Remove entry for given `line` number, making the browser one line shorter.*

 - `void remove_icon (int line)`
- Removes the icon for `line`.*

 - `void replace (int a, const char *b)`
- For back compatibility only.*

 - `int select (int line, int val=1)`
- Sets the selection state of the item at `line` to the value `val`.*

 - `int selected (int line) const`
- Returns 1 if specified `line` is selected, 0 if not.*

 - `void show ()`
- Shows the entire `FI_Browser` widget – opposite of `hide()`.*

- void **show** (int line)
 - Makes `line` visible, and available for selection by user.*
- int **size** () const
 - Returns how many lines are in the browser.*
- void **size** (int W, int H)
- void **swap** (int a, int b)
 - Swaps two browser lines `a` and `b`.*
- const char * **text** (int line) const
 - Returns the label text for the specified `line`.*
- void **text** (int line, const char *newtext)
 - Sets the text for the specified `line` to `newtext`.*
- **FI_Fontsize textsize** () const
 - Gets the default text size (in pixels) for the lines in the browser.*
- void **textsize** (**FI_Fontsize** newSize)
 - Sets the default text size (in pixels) for the lines in the browser to `newSize`.*
- int **topline** () const
 - Returns the line that is currently visible at the top of the browser.*
- void **topline** (int line)
 - Scrolls the browser so the top item in the browser is showing the specified `line`.*
- int **value** () const
 - Returns the line number of the currently selected line, or 0 if none selected.*
- void **value** (int line)
 - Sets the browser's `value()`, which selects the specified `line`.*
- int **visible** (int line) const
 - Returns non-zero if the specified `line` is visible, 0 if hidden.*
- ~**FI_Browser** ()
 - The destructor deletes all list items and destroys the browser.*

Protected Member Functions

- **FL_BLINE** * **_remove** (int line)
 - Removes the item at the specified `line`.*
- **FL_BLINE** * **find_line** (int line) const
 - Returns the item for specified `line`.*
- int **full_height** () const
 - The height of the entire list of all `visible()` items in pixels.*
- int **incr_height** () const
 - The default 'average' item height (including inter-item spacing) in pixels.*
- void **insert** (int line, **FL_BLINE** *item)
 - Insert specified `item` above `line`.*
- void * **item_at** (int line) const
 - Return the item at specified `line`.*
- void **item_draw** (void *item, int X, int Y, int W, int H) const
 - Draws `item` at the position specified by `X Y W H`.*
- void * **item_first** () const
 - Returns the very first item in the list.*
- int **item_height** (void *item) const
 - Returns height of `item` in pixels.*
- void * **item_last** () const
 - Returns the very last item in the list.*
- void * **item_next** (void *item) const

- Returns the next item after `item`.*

 - void * [item_prev](#) (void *item) const
- Returns the previous item before `item`.*

 - void [item_select](#) (void *item, int val)
- Change the selection state of `item` to the value `val`.*

 - int [item_selected](#) (void *item) const
- See if `item` is selected.*

 - void [item_swap](#) (void *a, void *b)
- Swap the items `a` and `b`.*

 - const char * [item_text](#) (void *item) const
- Returns the label text for `item`.*

 - int [item_width](#) (void *item) const
- Returns width of `item` in pixels.*

 - int [lineno](#) (void *item) const
- Returns line number corresponding to `item`, or zero if not found.*

 - void [swap](#) (FL_BLINE *a, FL_BLINE *b)
- Swap the two items `a` and `b`.*

Additional Inherited Members

31.7.1 Detailed Description

The [FI_Browser](#) widget displays a scrolling list of text lines, and manages all the storage for the text.

This is not a text editor or spreadsheet! But it is useful for showing a vertical list of named objects to the user.

Each line in the browser is identified by number. *The numbers start at one* (this is so that zero can be reserved for "no line" in the selective browsers). *Unless otherwise noted, the methods do not check to see if the passed line number is in range and legal. It must always be greater than zero and \leq [size\(\)](#).*

Each line contains a null-terminated string of text and a void * data pointer. The text string is displayed, the void * pointer can be used by the callbacks to reference the object the text describes.

The base class does nothing when the user clicks on it. The subclasses [FI_Select_Browser](#), [FI_Hold_Browser](#), and [FI_Multi_Browser](#) react to user clicks to select lines in the browser and do callbacks.

The base class [FI_Browser_](#) provides the scrolling and selection mechanisms of this and all the subclasses, but the dimensions and appearance of each item are determined by the subclass. You can use [FI_Browser_](#) to display information other than text, or text that is dynamically produced from your own data structures. If you find that loading the browser is a lot of work or is inefficient, you may want to make a subclass of [FI_Browser_](#).

Some common coding patterns used for working with [FI_Browser](#):

```
// How to loop through all the items in the browser
for ( int t=1; t<=browser->size(); t++ ) { // index 1 based..!
    printf("item #%d, label='%s'\n", t, browser->text(t));
}
```

Note: If you are *subclassing* [FI_Browser](#), it's more efficient to use the protected methods [item_first\(\)](#) and [item_next\(\)](#), since [FI_Browser](#) internally uses linked lists to manage the browser's items. For more info, see [find_item\(int\)](#).

31.7.2 Constructor & Destructor Documentation

31.7.2.1 FI_Browser()

```
Fl_Browser::Fl_Browser (
    int X,
    int Y,
    int W,
    int H,
    const char * L = 0 )
```

The constructor makes an empty browser.

Parameters

in	<i>X,Y,W,H</i>	position and size.
in	<i>L</i>	label string, may be NULL.

31.7.3 Member Function Documentation

31.7.3.1 `_remove()`

```
FL_BLINE * Fl_Browser::_remove (
    int line ) [protected]
```

Removes the item at the specified `line`.

Caveat: See efficiency note in [find_line\(\)](#). You must call [redraw\(\)](#) to make any changes visible.

Parameters

in	<i>line</i>	The line number to be removed. (1 based) Must be in range!
----	-------------	--

Returns

Pointer to browser item that was removed (and is no longer valid).

See also

[add\(\)](#), [insert\(\)](#), [remove\(\)](#), [swap\(int,int\)](#), [clear\(\)](#)

31.7.3.2 `add()`

```
void Fl_Browser::add (
    const char * newtext,
    void * d = 0 )
```

Adds a new line to the end of the browser.

The text string `newtext` may contain format characters; see [format_char\(\)](#) for details. `newtext` is copied using the `strdup()` function, and can be NULL to make a blank line.

The optional `void*` argument `d` will be the [data\(\)](#) for the new item.

Parameters

in	<i>newtext</i>	The label text used for the added item
in	<i>d</i>	Optional user data() for the item (0 if unspecified)

See also

[add\(\)](#), [insert\(\)](#), [remove\(\)](#), [swap\(int,int\)](#), [clear\(\)](#)

31.7.3.3 `bottomline()`

```
void Fl_Browser::bottomline (
    int line ) [inline]
```

Scrolls the browser so the bottom item in the browser is showing the specified `line`.

Parameters

<code>in</code>	<i>line</i>	The line to be displayed at the bottom.
-----------------	-------------	---

See also

[topline\(\)](#), [middleline\(\)](#), [bottomline\(\)](#), [displayed\(\)](#), [lineposition\(\)](#)

31.7.3.4 clear()

```
void Fl_Browser::clear ( )
```

Removes all the lines in the browser.

See also

[add\(\)](#), [insert\(\)](#), [remove\(\)](#), [swap\(int,int\)](#), [clear\(\)](#)

31.7.3.5 column_char() [1/2]

```
char Fl_Browser::column_char ( ) const [inline]
```

Gets the current column separator character.

The default is '\t' (tab).

See also

[column_char\(\)](#), [column_widths\(\)](#)

31.7.3.6 column_char() [2/2]

```
void Fl_Browser::column_char (
    char c ) [inline]
```

Sets the column separator to c.

This will only have an effect if you also set [column_widths\(\)](#). The default is '\t' (tab).

See also

[column_char\(\)](#), [column_widths\(\)](#)

31.7.3.7 column_widths() [1/2]

```
const int * Fl_Browser::column_widths ( ) const [inline]
```

Gets the current column width array.

This array is zero-terminated and specifies the widths in pixels of each column. The text is split at each [column_char\(\)](#) and each part is formatted into it's own column. After the last column any remaining text is formatted into the space between the last column and the right edge of the browser, even if the text contains instances of [column_char\(\)](#). The default value is a one-element array of just a zero, which means there are no columns.

Example:

```
Fl_Browser *b = new Fl_Browser(..);
static int widths[] = { 50, 50, 50, 70, 70, 40, 40, 70, 70, 50, 0 }; // widths for each column
b->column_widths(widths); // assign array to widget
b->column_char('\t'); // use tab as the column character
b->add("USER\tPID\tCPU\tMEM\tVSZ\tRSS\tTTY\tSTAT\tSTART\tTIME\tCOMMAND");
b->add("root\t2888\t0.0\t0.0\t1352\t0\ttty3\tSW\tAug15\t0:00\t@b@f/sbin/mingetty tty3");
b->add("root\t13115\t0.0\t0.0\t1352\t0\ttty2\tSW\tAug30\t0:00\t@b@f/sbin/mingetty tty2");
[...]
```

See also

[column_char\(\)](#), [column_widths\(\)](#)

31.7.3.8 column_widths() [2/2]

```
void Fl_Browser::column_widths (
    const int * arr ) [inline]
```

Sets the current array to `arr`.
Make sure the last entry is zero.

See also

[column_char\(\)](#), [column_widths\(\)](#)

31.7.3.9 data() [1/2]

```
void * Fl_Browser::data (
    int line ) const
```

Returns the user [data\(\)](#) for specified `line`.

Return value can be NULL if `line` is out of range or no user [data\(\)](#) was defined. The parameter `line` is 1 based (1 will be the first item in the list).

Parameters

in	<i>line</i>	The line number of the item whose data() is returned. (1 based)
----	-------------	---

Returns

The user data pointer (can be NULL)

31.7.3.10 data() [2/2]

```
void Fl_Browser::data (
    int line,
    void * d )
```

Sets the user data for specified `line` to `d`.
Does nothing if `line` is out of range.

Parameters

in	<i>line</i>	The line of the item whose data() is to be changed. (1 based)
in	<i>d</i>	The new data to be assigned to the item. (can be NULL)

31.7.3.11 display()

```
void Fl_Browser::display (
    int line,
    int val = 1 )
```

For back compatibility.

This calls `show(line)` if `val` is true, and `hide(line)` otherwise. If `val` is not specified, the default is 1 (makes the line visible).

See also

[show\(int\)](#), [hide\(int\)](#), [display\(\)](#), [visible\(\)](#), [make_visible\(\)](#)

31.7.3.12 displayed()

```
int Fl_Browser::displayed (
    int line ) const [inline]
```

Returns non-zero if `line` has been scrolled to a position where it is being displayed.

Checks to see if the item's vertical position is within the top and bottom edges of the display window. This does NOT take into account the [hide\(\)/show\(\)](#) status of the widget or item.

Parameters

in	<i>line</i>	The line to be checked
----	-------------	------------------------

Returns

1 if visible, 0 if not visible.

See also

[topline\(\)](#), [middleline\(\)](#), [bottomline\(\)](#), [displayed\(\)](#), [lineposition\(\)](#)

31.7.3.13 find_line()

```
FL_BLINE * Fl_Browser::find_line (
    int line ) const [protected]
```

Returns the item for specified `line`.

Note: This call is slow. It's fine for e.g. responding to user clicks, but slow if called often, such as in a tight sorting loop. Finding an item 'by line' involves a linear lookup on the internal linked list. The performance hit can be significant if the browser's contents is large, and the method is called often (e.g. during a sort). If you're writing a subclass, use the protected methods [item_first\(\)](#), [item_next\(\)](#), etc. to access the internal linked list more efficiently.

Parameters

in	<i>line</i>	The line number of the item to return. (1 based)
----	-------------	--

Return values

<i>item</i>	that was found.
<i>NULL</i>	if line is out of range.

See also

[item_at\(\)](#), [find_line\(\)](#), [lineno\(\)](#)

31.7.3.14 format_char() [1/2]

```
char Fl_Browser::format_char ( ) const [inline]
```

Gets the current format code prefix character, which by default is '@'.

A string of formatting codes at the start of each column are stripped off and used to modify how the rest of the line is printed:

- '@.' Print rest of line, don't look for more '@' signs
- '@@' Print rest of line starting with '@'
- '@1' Use a LARGE (24 point) font

- '@m' Use a medium large (18 point) font
- '@s' Use a small (11 point) font
- '@b' Use a **bold** font (adds FL_BOLD to font)
- '@i' Use an *italic* font (adds FL_ITALIC to font)
- '@f' or '@t' Use a fixed-pitch font (sets font to FL_COURIER)
- '@c' Center the line horizontally
- '@r' Right-justify the text
- '@B0', '@B1', ... '@B255' Fill the background with fl_color(n)
- '@C0', '@C1', ... '@C255' Use fl_color(n) to draw the text
- '@F0', '@F1', ... Use fl_font(n) to draw the text
- '@S1', '@S2', ... Use point size n to draw the text
- '@u' or '@_' Underline the text.
- '@-' draw an engraved line through the middle.

Notice that the '@.' command can be used to reliably terminate the parsing. To print a random string in a random color, use `printf("@C%d@.%s", color, string)` and it will work even if the string starts with a digit or has the format character in it.

31.7.3.15 format_char() [2/2]

```
void Fl_Browser::format_char (
    char c ) [inline]
```

Sets the current format code prefix character to c.

The default prefix is '@'. Set the prefix to 0 to disable formatting.

See also

[format_char\(\)](#) for list of '@' codes

31.7.3.16 full_height()

```
int Fl_Browser::full_height ( ) const [protected], [virtual]
```

The height of the entire list of all [visible\(\)](#) items in pixels.

This returns the accumulated height of *all* the items in the browser that are not hidden with [hide\(\)](#), including items scrolled off screen.

Returns

The accumulated size of all the visible items in pixels.

See also

[item_height\(\)](#), [item_width\(\)](#),
[incr_height\(\)](#), [full_height\(\)](#)

Reimplemented from [Fl_Browser_.](#)

31.7.3.17 hide() [1/2]

```
void Fl_Browser::hide ( ) [inline], [virtual]
```

Hides the entire [Fl_Browser](#) widget – opposite of [show\(\)](#).

Reimplemented from [Fl_Widget.](#)

31.7.3.18 hide() [2/2]

```
void Fl_Browser::hide (
    int line )
```

Makes `line` invisible, preventing selection by the user.

The line can still be selected under program control. This changes the `full_height()` if the state was changed. When a line is made invisible, lines below it are moved up in the display. `redraw()` is called automatically if a change occurred.

Parameters

in	<i>line</i>	The line to be hidden. (1 based)
----	-------------	----------------------------------

See also

[show\(int\)](#), [hide\(int\)](#), [display\(\)](#), [visible\(\)](#), [make_visible\(\)](#)

31.7.3.19 icon() [1/2]

```
Fl_Image * Fl_Browser::icon (
    int line ) const
```

Returns the icon currently defined for `line`.

If no icon is defined, NULL is returned.

Parameters

in	<i>line</i>	The line whose icon is returned.
----	-------------	----------------------------------

Returns

The icon defined, or NULL if none.

31.7.3.20 icon() [2/2]

```
void Fl_Browser::icon (
    int line,
    Fl_Image * icon )
```

Set the image icon for `line` to the value `icon`.

Caller is responsible for keeping the icon allocated. The `line` is automatically redrawn.

Parameters

in	<i>line</i>	The line to be modified. If out of range, nothing is done.
in	<i>icon</i>	The image icon to be assigned to the <code>line</code> . If NULL, any previous icon is removed.

31.7.3.21 incr_height()

```
int Fl_Browser::incr_height ( ) const [protected], [virtual]
```

The default 'average' item height (including inter-item spacing) in pixels.

This currently returns `textsize() + 2`.

Returns

The value in pixels.

See also

[item_height\(\)](#), [item_width\(\)](#),
[incr_height\(\)](#), [full_height\(\)](#)

Reimplemented from [Fl_Browser_](#).

31.7.3.22 insert() [1/2]

```
void Fl_Browser::insert (
    int line,
    const char * newtext,
    void * d = 0 )
```

Insert a new entry whose label is *newtext* *above* given *line*, optional data *d*.

Text may contain format characters; see [format_char\(\)](#) for details. *newtext* is copied using the `strdup()` function, and can be NULL to make a blank line.

The optional `void *` argument *d* will be the [data\(\)](#) of the new item.

Parameters

in	<i>line</i>	Line position for insert. (1 based) If <code>line > size()</code> , the entry will be added at the end.
in	<i>newtext</i>	The label text for the new line.
in	<i>d</i>	Optional pointer to user data to be associated with the new line.

31.7.3.23 insert() [2/2]

```
void Fl_Browser::insert (
    int line,
    FL_BLINE * item ) [protected]
```

Insert specified *item* above *line*.

If `line > size()` then the line is added to the end.

Caveat: See efficiency note in [find_line\(\)](#).

Parameters

in	<i>line</i>	The new line will be inserted above this line (1 based).
in	<i>item</i>	The item to be added.

31.7.3.24 item_at()

```
void * Fl_Browser::item_at (
    int line ) const [inline], [protected], [virtual]
```

Return the item at specified *line*.

Parameters

in	<i>line</i>	The line of the item to return. (1 based)
----	-------------	---

Returns

The item, or NULL if line out of range.

See also

[item_at\(\)](#), [find_line\(\)](#), [lineno\(\)](#)

Reimplemented from [Fl_Browser_.](#)

31.7.3.25 item_draw()

```
void Fl_Browser::item_draw (
    void * item,
    int X,
    int Y,
    int W,
    int H ) const [protected], [virtual]
```

Draws *item* at the position specified by X Y W H.

The W and H values are used for clipping. Should only be called within the context of an FLTK [draw\(\)](#).

Parameters

in	<i>item</i>	The item to be drawn
in	X,Y,W,H	position and size.

Implements [Fl_Browser_.](#)

31.7.3.26 item_first()

```
void * Fl_Browser::item_first ( ) const [protected], [virtual]
```

Returns the very first item in the list.

Example of use:

```
// Walk the browser from beginning to end
for ( void *i=item_first(); i; i=item_next(i) ) {
    printf("item label='%s'\n", item_text(i));
}
```

Returns

The first item, or NULL if list is empty.

See also

[item_first\(\)](#), [item_last\(\)](#), [item_next\(\)](#), [item_prev\(\)](#)

Implements [Fl_Browser_.](#)

31.7.3.27 item_height()

```
int Fl_Browser::item_height (
    void * item ) const [protected], [virtual]
```

Returns height of *item* in pixels.

This takes into account embedded @ codes within the [text\(\)](#) label.

Parameters

in	<i>item</i>	The item whose height is returned.
----	-------------	------------------------------------

Returns

The height of the item in pixels.

See also

[item_height\(\)](#), [item_width\(\)](#),
[incr_height\(\)](#), [full_height\(\)](#)

Implements [Fl_Browser_](#).

31.7.3.28 item_last()

```
void * Fl_Browser::item_last ( ) const [protected], [virtual]
```

Returns the very last item in the list.

Example of use:

```
// Walk the browser in reverse, from end to start
for ( void *i=item_last(); i; i=item_prev(i) ) {
    printf("item label='%s'\n", item_text(i));
}
```

Returns

The last item, or NULL if list is empty.

See also

[item_first\(\)](#), [item_last\(\)](#), [item_next\(\)](#), [item_prev\(\)](#)

Reimplemented from [Fl_Browser_](#).

31.7.3.29 item_next()

```
void * Fl_Browser::item_next (
    void * item ) const [protected], [virtual]
```

Returns the next item after *item*.

Parameters

in	<i>item</i>	The 'current' item
----	-------------	--------------------

Returns

The next item after *item*, or NULL if there are none after this one.

See also

[item_first\(\)](#), [item_last\(\)](#), [item_next\(\)](#), [item_prev\(\)](#)

Implements [Fl_Browser_](#).

31.7.3.30 item_prev()

```
void * Fl_Browser::item_prev (
    void * item ) const [protected], [virtual]
```

Returns the previous item before *item*.

Parameters

in	<i>item</i>	The 'current' item
----	-------------	--------------------

Returns

The previous item before `item`, or NULL if there are none before this one.

See also

[item_first\(\)](#), [item_last\(\)](#), [item_next\(\)](#), [item_prev\(\)](#)

Implements [Fl_Browser_](#).

31.7.3.31 item_select()

```
void Fl_Browser::item_select (
    void * item,
    int val ) [protected], [virtual]
```

Change the selection state of `item` to the value `val`.

Parameters

in	<i>item</i>	The item to be changed.
in	<i>val</i>	The new selection state: 1 selects, 0 de-selects.

See also

[select\(\)](#), [selected\(\)](#), [value\(\)](#), [item_select\(\)](#), [item_selected\(\)](#)

Reimplemented from [Fl_Browser_](#).

31.7.3.32 item_selected()

```
int Fl_Browser::item_selected (
    void * item ) const [protected], [virtual]
```

See if `item` is selected.

Parameters

in	<i>item</i>	The item whose selection state is to be checked.
----	-------------	--

Returns

1 if selected, 0 if not.

See also

[select\(\)](#), [selected\(\)](#), [value\(\)](#), [item_select\(\)](#), [item_selected\(\)](#)

Reimplemented from [Fl_Browser_](#).

31.7.3.33 item_swap()

```
void Fl_Browser::item_swap (
    void * a,
    void * b ) [inline], [protected], [virtual]
```

Swap the items `a` and `b`.

You must call [redraw\(\)](#) to make any changes visible.

Parameters

in	<i>a,b</i>	the items to be swapped.
----	------------	--------------------------

See also

[swap\(int,int\), item_swap\(\)](#)

Reimplemented from [Fl_Browser_](#).

31.7.3.34 item_text()

```
const char * Fl_Browser::item_text (
    void * item ) const [protected], [virtual]
```

Returns the label text for *item*.

Parameters

in	<i>item</i>	The item whose label text is returned.
----	-------------	--

Returns

The item's text string. (Can be NULL)

Reimplemented from [Fl_Browser_](#).

31.7.3.35 item_width()

```
int Fl_Browser::item_width (
    void * item ) const [protected], [virtual]
```

Returns width of *item* in pixels.

This takes into account embedded @ codes within the [text\(\)](#) label.

Parameters

in	<i>item</i>	The item whose width is returned.
----	-------------	-----------------------------------

Returns

The width of the item in pixels.

See also

[item_height\(\)](#), [item_width\(\)](#),
[incr_height\(\)](#), [full_height\(\)](#)

Implements [Fl_Browser_](#).

31.7.3.36 lineno()

```
int Fl_Browser::lineno (
    void * item ) const [protected]
```

Returns line number corresponding to *item*, or zero if not found.

Caveat: See efficiency note in [find_line\(\)](#).

Parameters

in	<i>item</i>	The item to be found
----	-------------	----------------------

Returns

The line number of the item, or 0 if not found.

See also

[item_at\(\)](#), [find_line\(\)](#), [lineno\(\)](#)

31.7.3.37 lineposition()

```
void Fl_Browser::lineposition (
    int line,
    Fl_Line_Position pos )
```

Updates the browser so that `line` is shown at position `pos`.

Parameters

in	<i>line</i>	line number. (1 based)
in	<i>pos</i>	position.

See also

[topline\(\)](#), [middleline\(\)](#), [bottomline\(\)](#)

31.7.3.38 load()

```
int Fl_Browser::load (
    const char * filename )
```

Clears the browser and reads the file, adding each line from the file to the browser.

If the filename is NULL or a zero-length string then this just clears the browser. This returns zero if there was any error in opening or reading the file, in which case `errno` is set to the system error. The [data\(\)](#) of each line is set to NULL.

Parameters

in	<i>filename</i>	The filename to load
----	-----------------	----------------------

Returns

1 if OK, 0 on error (`errno` has reason)

See also

[add\(\)](#)

31.7.3.39 make_visible()

```
void Fl_Browser::make_visible (
    int line ) [inline]
```

Make the item at the specified `line` [visible\(\)](#).
Functionally similar to [show\(int line\)](#). If `line` is out of range, redisplay top or bottom of list as appropriate.

Parameters

<code>in</code>	<code>line</code>	The line to be made visible.
-----------------	-------------------	------------------------------

See also

[show\(int\)](#), [hide\(int\)](#), [display\(\)](#), [visible\(\)](#), [make_visible\(\)](#)

31.7.3.40 `middleline()`

```
void Fl_Browser::middleline (
    int line ) [inline]
```

Scrolls the browser so the middle item in the browser is showing the specified `line`.

Parameters

<code>in</code>	<code>line</code>	The line to be displayed in the middle.
-----------------	-------------------	---

See also

[topline\(\)](#), [middleline\(\)](#), [bottomline\(\)](#), [displayed\(\)](#), [lineposition\(\)](#)

31.7.3.41 `move()`

```
void Fl_Browser::move (
    int to,
    int from )
```

Line `from` is removed and reinserted at `to`.

Note: `to` is calculated *after* line `from` gets removed.

Parameters

<code>in</code>	<code>to</code>	Destination line number (calculated <i>after</i> line <code>from</code> is removed)
<code>in</code>	<code>from</code>	Line number of item to be moved

31.7.3.42 `remove()`

```
void Fl_Browser::remove (
    int line )
```

Remove entry for given `line` number, making the browser one line shorter.

You must call [redraw\(\)](#) to make any changes visible.

Parameters

<code>in</code>	<code>line</code>	Line to be removed. (1 based) If <code>line</code> is out of range, no action is taken.
-----------------	-------------------	--

See also

[add\(\)](#), [insert\(\)](#), [remove\(\)](#), [swap\(int,int\)](#), [clear\(\)](#)

31.7.3.43 remove_icon()

```
void Fl_Browser::remove_icon (
    int line )
```

Removes the icon for `line`.

It's ok to remove an icon if none has been defined.

Parameters

in	<i>line</i>	The line whose icon is to be removed.
----	-------------	---------------------------------------

31.7.3.44 select()

```
int Fl_Browser::select (
    int line,
    int val = 1 )
```

Sets the selection state of the item at `line` to the value `val`.

If `val` is not specified, the default is 1 (selects the item).

Parameters

in	<i>line</i>	The line number of the item to be changed. (1 based)
in	<i>val</i>	The new selection state (1=select, 0=de-select).

Returns

1 if the state changed, 0 if not.

See also

[select\(\)](#), [selected\(\)](#), [value\(\)](#), [item_select\(\)](#), [item_selected\(\)](#)

31.7.3.45 selected()

```
int Fl_Browser::selected (
    int line ) const
```

Returns 1 if specified `line` is selected, 0 if not.

Parameters

in	<i>line</i>	The line being checked (1 based)
----	-------------	----------------------------------

Returns

1 if item selected, 0 if not.

See also

[select\(\)](#), [selected\(\)](#), [value\(\)](#), [item_select\(\)](#), [item_selected\(\)](#)

31.7.3.46 show() [1/2]

```
void Fl_Browser::show ( ) [inline], [virtual]
```

Shows the entire [Fl_Browser](#) widget – opposite of [hide\(\)](#).
Reimplemented from [Fl_Widget](#).

31.7.3.47 show() [2/2]

```
void Fl_Browser::show (
    int line )
```

Makes `line` visible, and available for selection by user.
Opposite of [hide\(int\)](#). This changes the [full_height\(\)](#) if the state was changed. [redraw\(\)](#) is called automatically if a change occurred.

Parameters

in	<i>line</i>	The line to be shown. (1 based)
----	-------------	---------------------------------

See also

[show\(int\)](#), [hide\(int\)](#), [display\(\)](#), [visible\(\)](#), [make_visible\(\)](#)

31.7.3.48 size()

```
int Fl_Browser::size ( ) const [inline]
```

Returns how many lines are in the browser.
The last line number is equal to this. Returns 0 if browser is empty.

31.7.3.49 swap() [1/2]

```
void Fl_Browser::swap (
    FL_BLINE * a,
    FL_BLINE * b ) [protected]
```

Swap the two items `a` and `b`.
Uses [swapping\(\)](#) to ensure list updates correctly.

Parameters

in	<i>a,b</i>	The two items to be swapped.
----	------------	------------------------------

See also

[swap\(int,int\)](#), [item_swap\(\)](#)

31.7.3.50 swap() [2/2]

```
void Fl_Browser::swap (
    int a,
    int b )
```

Swaps two browser lines `a` and `b`.
You must call [redraw\(\)](#) to make any changes visible.

Parameters

in	<i>a,b</i>	The two lines to be swapped. (both 1 based)
----	------------	---

See also

[swap\(int,int\), item_swap\(\)](#)

31.7.3.51 text() [1/2]

```
const char * Fl_Browser::text (
    int line ) const
```

Returns the label text for the specified `line`.

Return value can be NULL if `line` is out of range or unset. The parameter `line` is 1 based.

Parameters

in	<i>line</i>	The line number of the item whose text is returned. (1 based)
----	-------------	---

Returns

The text string (can be NULL)

31.7.3.52 text() [2/2]

```
void Fl_Browser::text (
    int line,
    const char * newtext )
```

Sets the text for the specified `line` to `newtext`.

Text may contain format characters; see [format_char\(\)](#) for details. `newtext` is copied using the `strdup()` function, and can be NULL to make a blank line.

Does nothing if `line` is out of range.

Parameters

in	<i>line</i>	The line of the item whose text will be changed. (1 based)
in	<i>newtext</i>	The new string to be assigned to the item.

31.7.3.53 textsize()

```
void Fl_Browser::textsize (
    Fl_Fontsize newSize )
```

Sets the default text size (in pixels) for the lines in the browser to `newSize`.

This method recalculates all item heights and caches the total height internally for optimization of later item changes. This can be slow if there are many items in the browser.

It returns immediately (w/o recalculation) if `newSize` equals the current [textsize\(\)](#).

You *may* need to call [redraw\(\)](#) to see the effect and to have the scrollbar positions recalculated.

You should set the text size *before* populating the browser with items unless you really need to *change* the size later.

31.7.3.54 topline() [1/2]

```
int Fl_Browser::topline ( ) const
```

Returns the line that is currently visible at the top of the browser.

If there is no vertical scrollbar then this will always return 1.

Returns

The [lineno\(\)](#) of the [top\(\)](#) of the browser.

31.7.3.55 topline() [2/2]

```
void Fl_Browser::topline (
    int line ) [inline]
```

Scrolls the browser so the top item in the browser is showing the specified `line`.

Parameters

<code>in</code>	<code>line</code>	The line to be displayed at the top.
-----------------	-------------------	--------------------------------------

See also

[topline\(\)](#), [middleline\(\)](#), [bottomline\(\)](#), [displayed\(\)](#), [lineposition\(\)](#)

31.7.3.56 value() [1/2]

```
int Fl_Browser::value ( ) const
```

Returns the line number of the currently selected line, or 0 if none selected.

Returns

The line number of current selection, or 0 if none selected.

See also

[select\(\)](#), [selected\(\)](#), [value\(\)](#), [item_select\(\)](#), [item_selected\(\)](#)

31.7.3.57 value() [2/2]

```
void Fl_Browser::value (
    int line ) [inline]
```

Sets the browser's [value\(\)](#), which selects the specified `line`. This is the same as calling `select(line)`.

See also

[select\(\)](#), [selected\(\)](#), [value\(\)](#), [item_select\(\)](#), [item_selected\(\)](#)

31.7.3.58 visible()

```
int Fl_Browser::visible (
    int line ) const
```

Returns non-zero if the specified `line` is visible, 0 if hidden. Use [show\(int\)](#), [hide\(int\)](#), or [make_visible\(int\)](#) to change an item's visible state.

Parameters

<code>in</code>	<code>line</code>	The line in the browser to be tested. (1 based)
-----------------	-------------------	---

See also

[show\(int\)](#), [hide\(int\)](#), [display\(\)](#), [visible\(\)](#), [make_visible\(\)](#)

The documentation for this class was generated from the following files:

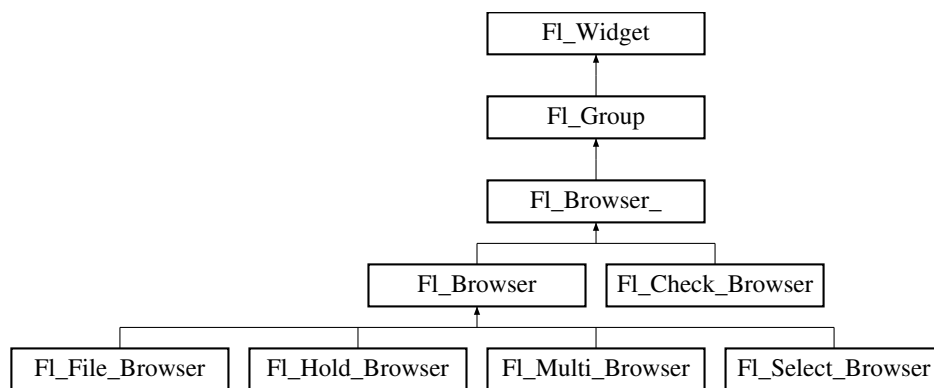
- `Fl_Browser.H`
- `Fl_Browser.cxx`
- `Fl_Browser_load.cxx`

31.8 `Fl_Browser_` Class Reference

This is the base class for browsers.

```
#include <Fl_Browser_.H>
```

Inheritance diagram for `Fl_Browser_`:



Public Types

- enum {
[HORIZONTAL](#) = 1 , [VERTICAL](#) = 2 , [BOTH](#) = 3 , [ALWAYS_ON](#) = 4 ,
[HORIZONTAL_ALWAYS](#) = 5 , [VERTICAL_ALWAYS](#) = 6 , [BOTH_ALWAYS](#) = 7 }
 Values for [has_scrollbar\(\)](#).

Public Member Functions

- int [deselect](#) (int docallbacks=0)
 Deselects all items in the list and returns 1 if the state changed or 0 if it did not.
- void [display](#) (void *item)
 Displays the *item*, scrolling the list as necessary.
- int [handle](#) (int event)
 Handles the *event* within the normal widget bounding box.
- uchar [has_scrollbar](#) () const
 Returns the current scrollbar mode, see [Fl_Browser_::has_scrollbar\(uchar\)](#)
- void [has_scrollbar](#) (uchar mode)
 Sets whether the widget should have scrollbars or not (default [Fl_Browser_::BOTH](#)).
- int [hposition](#) () const
 Gets the horizontal scroll position of the list as a pixel position *pos*.
- void [hposition](#) (int)
 Sets the horizontal scroll position of the list to pixel position *pos*.
- int [position](#) () const
 Gets the vertical scroll position of the list as a pixel position *pos*.
- void [position](#) (int pos)

- Sets the vertical scroll position of the list to pixel position `pos`.*

 - void `resize` (int X, int Y, int W, int H)

Repositions and/or resizes the browser.
- void `scrollbar_left` ()
- Moves the vertical scrollbar to the lefthand side of the list.*
- void `scrollbar_right` ()
- Moves the vertical scrollbar to the righthand side of the list.*
- int `scrollbar_size` () const
- Gets the current size of the scrollbars' troughs, in pixels.*
- void `scrollbar_size` (int newSize)
- Sets the pixel size of the scrollbars' troughs to `newSize`, in pixels.*
- int `scrollbar_width` () const
- This method has been deprecated, existing for backwards compatibility only.*
- void `scrollbar_width` (int width)
- This method has been deprecated, existing for backwards compatibility only.*
- int `select` (void *item, int val=1, int docallbacks=0)
- Sets the selection state of `item` to `val`, and returns 1 if the state changed or 0 if it did not.*
- int `select_only` (void *item, int docallbacks=0)
- Selects `item` and returns 1 if the state changed or 0 if it did not.*
- void `sort` (int flags=0)
- Sort the items in the browser based on `flags`.*
- `FL_Color` `textcolor` () const
- Gets the default text color for the lines in the browser.*
- void `textcolor` (`FL_Color` col)
- Sets the default text color for the lines in the browser to color `col`.*
- `FL_Font` `textfont` () const
- Gets the default text font for the lines in the browser.*
- void `textfont` (`FL_Font` font)
- Sets the default text font for the lines in the browser to `font`.*
- `FL_Fontsize` `textsize` () const
- Gets the default text size (in pixels) for the lines in the browser.*
- void `textsize` (`FL_Fontsize` newSize)
- Sets the default text size (in pixels) for the lines in the browser to `size`.*

Public Attributes

- `FL_Scrollbar` `hscrollbar`
- Horizontal scrollbar.*
- `FL_Scrollbar` `scrollbar`
- Vertical scrollbar.*

Protected Member Functions

- void `bbox` (int &X, int &Y, int &W, int &H) const
- Returns the bounding box for the interior of the list's display window, inside the scrollbars.*
- void `deleting` (void *item)
- This method should be used when `item` is being deleted from the list.*
- int `displayed` (void *item) const
- Returns non-zero if `item` has been scrolled to a position where it is being displayed.*
- void `draw` ()
- Draws the list within the normal widget bounding box.*
- void * `find_item` (int ypos)

- This method returns the item under mouse y position `ypos`.*

 - `FI_Browser_` (int X, int Y, int W, int H, const char *L=0)

The constructor makes an empty browser.
- virtual int `full_height` () const

This method may be provided by the subclass to indicate the full height of the item list, in pixels.
- virtual int `full_width` () const

This method may be provided by the subclass to indicate the full width of the item list, in pixels.
- virtual int `incr_height` () const

This method may be provided to return the average height of all items to be used for scrolling.
- void `inserting` (void *a, void *b)

This method should be used when an item is in the process of being inserted into the list.
- virtual void * `item_at` (int index) const

This method must be provided by the subclass to return the item for the specified `index`.
- virtual void `item_draw` (void *item, int X, int Y, int W, int H) const =0

This method must be provided by the subclass to draw the `item` in the area indicated by X, Y, W, H.
- virtual void * `item_first` () const =0

This method must be provided by the subclass to return the first item in the list.
- virtual int `item_height` (void *item) const =0

This method must be provided by the subclass to return the height of `item` in pixels.
- virtual void * `item_last` () const

This method must be provided by the subclass to return the last item in the list.
- virtual void * `item_next` (void *item) const =0

This method must be provided by the subclass to return the item in the list after `item`.
- virtual void * `item_prev` (void *item) const =0

This method must be provided by the subclass to return the item in the list before `item`.
- virtual int `item_quick_height` (void *item) const

This method may be provided by the subclass to return the height of the `item`, in pixels.
- virtual void `item_select` (void *item, int val=1)

This method must be implemented by the subclass if it supports multiple selections; sets the selection state to `val` for the `item`.
- virtual int `item_selected` (void *item) const

This method must be implemented by the subclass if it supports multiple selections; returns the selection state for `item`.
- virtual void `item_swap` (void *a, void *b)

This optional method should be provided by the subclass to efficiently swap browser items `a` and `b`, such as for sorting.
- virtual const char * `item_text` (void *item) const

This optional method returns a string (label) that may be used for sorting.
- virtual int `item_width` (void *item) const =0

This method must be provided by the subclass to return the width of the `item` in pixels.
- int `leftedge` () const

This method returns the X position of the left edge of the list area after adjusting for the scrollbar and border, if any.
- void `new_list` ()

This method should be called when the list data is completely replaced or cleared.
- void `redraw_line` (void *item)

This method should be called when the contents of `item` has changed, but not its height.
- void `redraw_lines` ()

This method will cause the entire list to be redrawn.
- void `replacing` (void *a, void *b)

This method should be used when item `a` is being replaced by item `b`.
- void * `selection` () const

Returns the item currently selected, or NULL if there is no selection.

- void [swapping](#) (void *a, void *b)

This method should be used when two items *a* and *b* are being swapped.

- void * [top](#) () const

Returns the item that appears at the top of the list.

Additional Inherited Members

31.8.1 Detailed Description

This is the base class for browsers.

To be useful it must be subclassed and several virtual functions defined. The Forms-compatible browser and the file chooser's browser are subclassed off of this.

This has been designed so that the subclass has complete control over the storage of the data, although because `next()` and `prev()` functions are used to index, it works best as a linked list or as a large block of characters in which the line breaks must be searched for.

A great deal of work has been done so that the "height" of a data object does not need to be determined until it is drawn. This is useful if actually figuring out the size of an object requires accessing image data or doing `stat()` on a file or doing some other slow operation.

31.8.1.1 Keyboard navigation of browser items

The keyboard navigation of browser items is only possible if `visible_focus()` is enabled. If disabled, the widget rejects keyboard focus; Tab and Shift-Tab focus navigation will skip the widget.

In 'Select' and 'Normal' mode, the widget rejects keyboard focus; no navigation keys are supported (other than scrollbar positioning).

In 'Hold' mode, the widget accepts keyboard focus, and Up/Down arrow keys can navigate the selected item.

In 'Multi' mode, the widget accepts keyboard focus, and Up/Down arrow keys navigate the focus box; Space toggles the current item's selection, Enter selects only the current item (deselects all others). If Shift (or Ctrl) is combined with Up/Down arrow keys, the current item's selection state is extended to the next item. In this way one can extend a selection or de-selection.

31.8.2 Member Enumeration Documentation

31.8.2.1 anonymous enum

anonymous enum

Values for `has_scrollbar()`.

Anonymous enum bit flags for `has_scrollbar()`.

- bit 0: horizontal
- bit 1: vertical
- bit 2: 'always' (to be combined with bits 0 and 1)
- bit 3-31: reserved for future use

Enumerator

HORIZONTAL	Only show horizontal scrollbar.
VERTICAL	Only show vertical scrollbar.
BOTH	Show both scrollbars. (default)
ALWAYS_ON	Specified scrollbar(s) should 'always' be shown (to be used with HORIZONTAL/VERTICAL)
HORIZONTAL_ALWAYS	Horizontal scrollbar always on.
VERTICAL_ALWAYS	Vertical scrollbar always on.
BOTH_ALWAYS	Both scrollbars always on.

31.8.3 Constructor & Destructor Documentation

31.8.3.1 Fl_Browser_()

```
Fl_Browser_::Fl_Browser_ (
    int X,
    int Y,
    int W,
    int H,
    const char * L = 0 ) [protected]
```

The constructor makes an empty browser.

Parameters

in	<i>X,Y,W,H</i>	position and size.
in	<i>L</i>	The label string, may be NULL.

31.8.4 Member Function Documentation

31.8.4.1 bbox()

```
void Fl_Browser_::bbox (
    int & X,
    int & Y,
    int & W,
    int & H ) const [protected]
```

Returns the bounding box for the interior of the list's display window, inside the scrollbars.

Parameters

out	<i>X,Y,W,H</i>	The returned bounding box. (The original contents of these parameters are overwritten)
-----	----------------	---

31.8.4.2 deleting()

```
void Fl_Browser_::deleting (
    void * item ) [protected]
```

This method should be used when *item* is being deleted from the list.

It allows the `Fl_Browser_` to discard any cached data it has on the item. This method does not actually delete the item, but handles the follow up bookkeeping after the item has just been deleted.

Parameters

in	<i>item</i>	The item being deleted.
----	-------------	-------------------------

31.8.4.3 deselect()

```
int Fl_Browser_::deselect (
    int docallbacks = 0 )
```

Deselects all items in the list and returns 1 if the state changed or 0 if it did not.
 If the optional `docalbacks` parameter is non-zero, `deselect` tries to call the callback function for the widget.

Parameters

<code>in</code>	<code>docalbacks</code>	If 1, invokes widget callback if item changed. If 0, doesn't do callback (default).
-----------------	-------------------------	--

31.8.4.4 `display()`

```
void Fl_Browser_::display (
    void * item )
```

Displays the `item`, scrolling the list as necessary.

Parameters

<code>in</code>	<code>item</code>	The item to be displayed.
-----------------	-------------------	---------------------------

See also

[display\(\)](#), [displayed\(\)](#)

31.8.4.5 `displayed()`

```
int Fl_Browser_::displayed (
    void * item ) const [protected]
```

Returns non-zero if `item` has been scrolled to a position where it is being displayed.
 Checks to see if the item's vertical position is within the top and bottom edges of the display window. This does NOT take into account the [hide\(\)/show\(\)](#) status of the widget or item.

Parameters

<code>in</code>	<code>item</code>	The item to check
-----------------	-------------------	-------------------

Returns

1 if visible, 0 if not visible.

See also

[display\(\)](#), [displayed\(\)](#)

31.8.4.6 `draw()`

```
void Fl_Browser_::draw (
    void ) [protected], [virtual]
```

Draws the list within the normal widget bounding box.
 Implements [Fl_Widget](#).

31.8.4.7 `find_item()`

```
void * Fl_Browser_::find_item (
    int ypos ) [protected]
```

This method returns the item under mouse y position `ypos`.
 NULL is returned if no item is displayed at that position.

Parameters

<code>in</code>	<code>ypos</code>	The y position (eg. Fl::event_y()) to find an item under.
-----------------	-------------------	--

Returns

The item, or NULL if not found

31.8.4.8 full_height()

```
int Fl_Browser_::full_height ( ) const [protected], [virtual]
```

This method may be provided by the subclass to indicate the full height of the item list, in pixels.

The default implementation computes the full height from the item heights. Includes the items that are scrolled off screen.

Returns

The height of the entire list, in pixels.

Reimplemented in [Fl_Browser](#).

31.8.4.9 full_width()

```
int Fl_Browser_::full_width ( ) const [protected], [virtual]
```

This method may be provided by the subclass to indicate the full width of the item list, in pixels.

The default implementation computes the full width from the item widths.

Returns

The maximum width of all the items, in pixels.

31.8.4.10 handle()

```
int Fl_Browser_::handle (
    int event ) [virtual]
```

Handles the `event` within the normal widget bounding box.

Parameters

<code>in</code>	<code>event</code>	The event to process.
-----------------	--------------------	-----------------------

Returns

1 if event was processed, 0 if not.

Reimplemented from [Fl_Widget](#).

Reimplemented in [Fl_Check_Browser](#).

31.8.4.11 has_scrollbar()

```
void Fl_Browser_::has_scrollbar (
    uchar mode ) [inline]
```

Sets whether the widget should have scrollbars or not (default `Fl_Browser_::BOTH`).

By default you can scroll in both directions, and the scrollbars disappear if the data will fit in the widget.

`has_scrollbar()` changes this based on the value of `mode` :

- 0 - No scrollbars.
- `Fl_Browser_::HORIZONTAL` - Only a horizontal scrollbar.
- `Fl_Browser_::VERTICAL` - Only a vertical scrollbar.
- `Fl_Browser_::BOTH` - The default is both scrollbars.
- `Fl_Browser_::HORIZONTAL_ALWAYS` - Horizontal scrollbar always on, vertical always off.
- `Fl_Browser_::VERTICAL_ALWAYS` - Vertical scrollbar always on, horizontal always off.
- `Fl_Browser_::BOTH_ALWAYS` - Both always on.

31.8.4.12 `hposition()` [1/2]

```
int Fl_Browser_::hposition ( ) const [inline]
```

Gets the horizontal scroll position of the list as a pixel position `pos`.

The position returned is how many pixels of the list are scrolled off the left edge of the screen. Example: A position of '18' indicates the left 18 pixels of the list are scrolled off the left edge of the screen.

See also

[position\(\)](#), [hposition\(\)](#)

31.8.4.13 `hposition()` [2/2]

```
void Fl_Browser_::hposition (
    int pos )
```

Sets the horizontal scroll position of the list to pixel position `pos`.

The position is how many pixels of the list are scrolled off the left edge of the screen. Example: A position of '18' scrolls the left 18 pixels of the list off the left edge of the screen.

Parameters

<code>in</code>	<code>pos</code>	The horizontal position (in pixels) to scroll the browser to.
-----------------	------------------	---

See also

[position\(\)](#), [hposition\(\)](#)

31.8.4.14 `incr_height()`

```
int Fl_Browser_::incr_height ( ) const [protected], [virtual]
```

This method may be provided to return the average height of all items to be used for scrolling.

The default implementation uses the height of the first item.

Returns

The average height of items, in pixels.

Reimplemented in [Fl_Browser](#).

31.8.4.15 inserting()

```
void Fl_Browser_::inserting (
    void * a,
    void * b ) [protected]
```

This method should be used when an item is in the process of being inserted into the list. It allows the [Fl_Browser_](#) to update its cache data as needed, scheduling a redraw for the affected lines. This method does not actually insert items, but handles the follow up bookkeeping after items have been inserted.

Parameters

in	<i>a</i>	The starting item position
in	<i>b</i>	The new item being inserted

31.8.4.16 item_at()

```
virtual void * Fl_Browser_::item_at (
    int index ) const [inline], [protected], [virtual]
```

This method must be provided by the subclass to return the item for the specified *index*.

Parameters

in	<i>index</i>	The <i>index</i> of the item to be returned
----	--------------	---

Returns

The item at the specified *index*.

Reimplemented in [Fl_Browser](#).

31.8.4.17 item_draw()

```
virtual void Fl_Browser_::item_draw (
    void * item,
    int X,
    int Y,
    int W,
    int H ) const [protected], [pure virtual]
```

This method must be provided by the subclass to draw the *item* in the area indicated by *X*, *Y*, *W*, *H*. Implemented in [Fl_Browser](#).

31.8.4.18 item_first()

```
virtual void * Fl_Browser_::item_first ( ) const [protected], [pure virtual]
```

This method must be provided by the subclass to return the first item in the list.

See also

[item_first\(\)](#), [item_next\(\)](#), [item_last\(\)](#), [item_prev\(\)](#)

Implemented in [Fl_Browser](#).

31.8.4.19 item_height()

```
virtual int Fl_Browser_::item_height (
    void * item ) const [protected], [pure virtual]
```

This method must be provided by the subclass to return the height of `item` in pixels. Allow for two additional pixels for the list selection box.

Parameters

<code>in</code>	<code>item</code>	The item whose height is returned.
-----------------	-------------------	------------------------------------

Returns

The height of the specified `item` in pixels.

See also

[item_height\(\)](#), [item_width\(\)](#), [item_quick_height\(\)](#)

Implemented in [FI_Browser](#).

31.8.4.20 item_last()

```
virtual void * Fl_Browser_::item_last ( ) const [inline], [protected], [virtual]
```

This method must be provided by the subclass to return the last item in the list.

See also

[item_first\(\)](#), [item_next\(\)](#), [item_last\(\)](#), [item_prev\(\)](#)

Reimplemented in [FI_Browser](#).

31.8.4.21 item_next()

```
virtual void * Fl_Browser_::item_next (
    void * item ) const [protected], [pure virtual]
```

This method must be provided by the subclass to return the item in the list after `item`.

See also

[item_first\(\)](#), [item_next\(\)](#), [item_last\(\)](#), [item_prev\(\)](#)

Implemented in [FI_Browser](#).

31.8.4.22 item_prev()

```
virtual void * Fl_Browser_::item_prev (
    void * item ) const [protected], [pure virtual]
```

This method must be provided by the subclass to return the item in the list before `item`.

See also

[item_first\(\)](#), [item_next\(\)](#), [item_last\(\)](#), [item_prev\(\)](#)

Implemented in [FI_Browser](#).

31.8.4.23 item_quick_height()

```
int Fl_Browser_::item_quick_height (
    void * item ) const [protected], [virtual]
```

This method may be provided by the subclass to return the height of the `item`, in pixels.

Allow for two additional pixels for the list selection box. This method differs from `item_height` in that it is only called for selection and scrolling operations. The default implementation calls `item_height`.

Parameters

in	<i>item</i>	The item whose height to return.
----	-------------	----------------------------------

Returns

The height, in pixels.

31.8.4.24 item_select()

```
void Fl_Browser_::item_select (
    void * item,
    int val = 1 ) [protected], [virtual]
```

This method must be implemented by the subclass if it supports multiple selections; sets the selection state to *val* for the *item*.

Sets the selection state for *item*, where optional *val* is 1 (select, the default) or 0 (de-select).

Parameters

in	<i>item</i>	The item to be selected
in	<i>val</i>	The optional selection state; 1=select, 0=de-select. The default is to select the item (1).

Reimplemented in [Fl_Browser](#).

31.8.4.25 item_selected()

```
int Fl_Browser_::item_selected (
    void * item ) const [protected], [virtual]
```

This method must be implemented by the subclass if it supports multiple selections; returns the selection state for *item*.

The method should return 1 if *item* is selected, or 0 otherwise.

Parameters

in	<i>item</i>	The item to test.
----	-------------	-------------------

Reimplemented in [Fl_Browser](#).

31.8.4.26 item_swap()

```
virtual void Fl_Browser_::item_swap (
    void * a,
    void * b ) [inline], [protected], [virtual]
```

This optional method should be provided by the subclass to efficiently swap browser items *a* and *b*, such as for sorting.

Parameters

in	<i>a,b</i>	The two items to be swapped.
----	------------	------------------------------

Reimplemented in [Fl_Browser](#).

31.8.4.27 item_text()

```
virtual const char * Fl_Browser_::item_text (
    void * item ) const [inline], [protected], [virtual]
```

This optional method returns a string (label) that may be used for sorting.

Parameters

in	<i>item</i>	The item whose label text is returned.
----	-------------	--

Returns

The item's text label. (Can be NULL if blank)

Reimplemented in [Fl_Browser](#).

31.8.4.28 item_width()

```
virtual int Fl_Browser_::item_width (
    void * item ) const [protected], [pure virtual]
```

This method must be provided by the subclass to return the width of the *item* in pixels. Allow for two additional pixels for the list selection box.

Parameters

in	<i>item</i>	The item whose width is returned.
----	-------------	-----------------------------------

Returns

The width of the item in pixels.

Implemented in [Fl_Browser](#).

31.8.4.29 leftedge()

```
int Fl_Browser_::leftedge ( ) const [protected]
```

This method returns the X position of the left edge of the list area after adjusting for the scrollbar and border, if any.

Returns

The X position of the left edge of the list, in pixels.

See also

[Fl_Browser_::bbox\(\)](#)

31.8.4.30 new_list()

```
void Fl_Browser_::new_list ( ) [protected]
```

This method should be called when the list data is completely replaced or cleared.

It informs the [Fl_Browser_](#) widget that any cached information it has concerning the items is invalid. This method does not clear the list, it just handles the follow up bookkeeping after the list has been cleared.

31.8.4.31 position() [1/2]

```
int Fl_Browser_::position ( ) const [inline]
```

Gets the vertical scroll position of the list as a pixel position *pos*.

The position returned is how many pixels of the list are scrolled off the top edge of the screen. Example: A position of '3' indicates the top 3 pixels of the list are scrolled off the top edge of the screen.

See also

[position\(\)](#), [hposition\(\)](#)

31.8.4.32 position() [2/2]

```
void Fl_Browser_::position (
    int pos )
```

Sets the vertical scroll position of the list to pixel position `pos`.

The position is how many pixels of the list are scrolled off the top edge of the screen. Example: A position of '3' scrolls the top three pixels of the list off the top edge of the screen.

Parameters

<code>in</code>	<code>pos</code>	The vertical position (in pixels) to scroll the browser to.
-----------------	------------------	---

See also

[position\(\)](#), [hposition\(\)](#)

31.8.4.33 redraw_line()

```
void Fl_Browser_::redraw_line (
    void * item ) [protected]
```

This method should be called when the contents of `item` has changed, but not its height.

Parameters

<code>in</code>	<code>item</code>	The item that needs to be redrawn.
-----------------	-------------------	------------------------------------

See also

[redraw_lines\(\)](#), [redraw_line\(\)](#)

31.8.4.34 redraw_lines()

```
void Fl_Browser_::redraw_lines ( ) [inline], [protected]
```

This method will cause the entire list to be redrawn.

See also

[redraw_lines\(\)](#), [redraw_line\(\)](#)

31.8.4.35 replacing()

```
void Fl_Browser_::replacing (
    void * a,
    void * b ) [protected]
```

This method should be used when item `a` is being replaced by item `b`.

It allows the `Fl_Browser_` to update its cache data as needed, schedules a redraw for the item being changed, and tries to maintain the selection. This method does not actually replace the item, but handles the follow up bookkeeping after the item has just been replaced.

Parameters

in	<i>a</i>	Item being replaced
in	<i>b</i>	Item to replace 'a'

31.8.4.36 `resize()`

```
void Fl_Browser_::resize (
    int X,
    int Y,
    int W,
    int H ) [virtual]
```

Repositions and/or resizes the browser.

Parameters

in	<i>X,Y,W,H</i>	The new position and size for the browser, in pixels.
----	----------------	---

Reimplemented from [Fl_Widget](#).

31.8.4.37 `scrollbar_left()`

```
void Fl_Browser_::scrollbar_left ( ) [inline]
```

Moves the vertical scrollbar to the lefthand side of the list.
For back compatibility.

31.8.4.38 `scrollbar_right()`

```
void Fl_Browser_::scrollbar_right ( ) [inline]
```

Moves the vertical scrollbar to the righthand side of the list.
For back compatibility.

31.8.4.39 `scrollbar_size()` [1/2]

```
int Fl_Browser_::scrollbar_size ( ) const [inline]
```

Gets the current size of the scrollbars' troughs, in pixels.
If this value is zero (default), this widget will use the [Fl::scrollbar_size\(\)](#) value as the scrollbar's width.

Returns

Scrollbar size in pixels, or 0 if the global [Fl::scrollbar_size\(\)](#) is being used.

See also

[Fl::scrollbar_size\(int\)](#)

31.8.4.40 `scrollbar_size()` [2/2]

```
void Fl_Browser_::scrollbar_size (
    int newSize ) [inline]
```

Sets the pixel size of the scrollbars' troughs to `newSize`, in pixels.
Normally you should not need this method, and should use [Fl::scrollbar_size\(int\)](#) instead to manage the size of ALL your widgets' scrollbars. This ensures your application has a consistent UI, is the default behavior, and is normally what you want.

Only use THIS method if you really need to override the global scrollbar size. The need for this should be rare. Setting `newSize` to the special value of 0 causes the widget to track the global `Fl::scrollbar_size()`, which is the default.

Parameters

in	<i>newSize</i>	Sets the scrollbar size in pixels. If 0 (default), scrollbar size tracks the global <code>Fl::scrollbar_size()</code>
----	----------------	--

See also

[Fl::scrollbar_size\(\)](#)

31.8.4.41 scrollbar_width() [1/2]

```
int Fl_Browser_::scrollbar_width ( ) const [inline]
```

This method has been deprecated, existing for backwards compatibility only.

Use `scrollbar_size()` instead. This method always returns the global value `Fl::scrollbar_size()`.

Returns

Always returns the global value `Fl::scrollbar_size()`.

Todo This method should eventually be removed in 1.4+

31.8.4.42 scrollbar_width() [2/2]

```
void Fl_Browser_::scrollbar_width (
    int width ) [inline]
```

This method has been deprecated, existing for backwards compatibility only.

Use `scrollbar_size(int)` instead. This method sets the global `Fl::scrollbar_size()`, and forces this instance of the widget to use it.

Todo This method should eventually be removed in 1.4+

31.8.4.43 select()

```
int Fl_Browser_::select (
    void * item,
    int val = 1,
    int docallbacks = 0 )
```

Sets the selection state of `item` to `val`, and returns 1 if the state changed or 0 if it did not.

If `docallbacks` is non-zero, `select` tries to call the callback function for the widget.

Parameters

in	<i>item</i>	The item whose selection state is to be changed
in	<i>val</i>	The new selection state (1=select, 0=de-select)
in	<i>docallbacks</i>	If 1, invokes widget callback if item changed. If 0, doesn't do callback (default).

Returns

1 if state was changed, 0 if not.

31.8.4.44 select_only()

```
int Fl_Browser_::select_only (
    void * item,
    int docallbacks = 0 )
```

Selects *item* and returns 1 if the state changed or 0 if it did not.
Any other items in the list are deselected.

Parameters

in	<i>item</i>	The <i>item</i> to select.
in	<i>docallbacks</i>	If 1, invokes widget callback if item changed. If 0, doesn't do callback (default).

31.8.4.45 selection()

```
void * Fl_Browser_::selection ( ) const [inline], [protected]
```

Returns the item currently selected, or NULL if there is no selection.

For multiple selection browsers this call returns the currently focused item, even if it is not selected. To find all selected items, call [Fl_Multi_Browser::selected\(\)](#) for every item in question.

31.8.4.46 sort()

```
void Fl_Browser_::sort (
    int flags = 0 )
```

Sort the items in the browser based on *flags*.

[item_swap\(void*, void*\)](#) and [item_text\(void*\)](#) must be implemented for this call.

Parameters

in	<i>flags</i>	FL_SORT_ASCENDING – sort in ascending order FL_SORT_DESCENDING – sort in descending order Values other than the above will cause undefined behavior Other flags may appear in the future.
----	--------------	--

Todo Add a flag to ignore case

31.8.4.47 swapping()

```
void Fl_Browser_::swapping (
    void * a,
    void * b ) [protected]
```

This method should be used when two items *a* and *b* are being swapped.

It allows the [Fl_Browser_](#) to update its cache data as needed, schedules a redraw for the two items, and tries to maintain the current selection. This method does not actually swap items, but handles the follow up bookkeeping after items have been swapped.

Parameters

in	<i>a,b</i>	Items being swapped.
----	------------	----------------------

31.8.4.48 textfont()

```
Fl_Font Fl_Browser_::textfont ( ) const [inline]
```

Gets the default text font for the lines in the browser.

See also

[textfont\(\)](#), [textsize\(\)](#), [textcolor\(\)](#)

31.8.5 Member Data Documentation**31.8.5.1 hscrollbar**

```
Fl_Scrollbar Fl_Browser_::hscrollbar
```

Horizontal scrollbar.

Public, so that it can be accessed directly.

31.8.5.2 scrollbar

```
Fl_Scrollbar Fl_Browser_::scrollbar
```

Vertical scrollbar.

Public, so that it can be accessed directly.

The documentation for this class was generated from the following files:

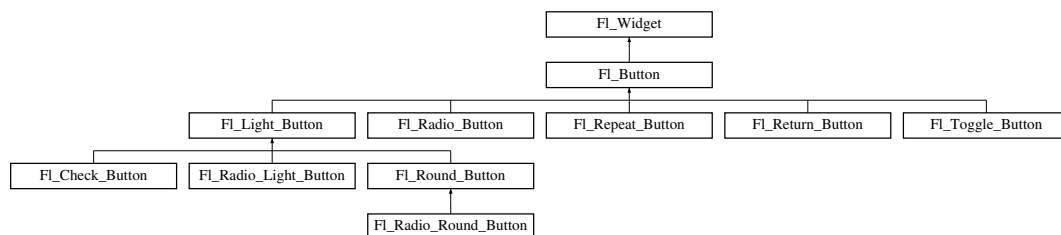
- Fl_Browser_.H
- Fl_Browser_.cxx

31.9 Fl_Button Class Reference

Buttons generate callbacks when they are clicked by the user.

```
#include <Fl_Button.H>
```

Inheritance diagram for Fl_Button:

**Public Member Functions**

- int [clear](#) ()
Same as `value(0)`.
- [Fl_Boxtype down_box](#) () const
Returns the current down box type, which is drawn when `value()` is non-zero.
- void [down_box](#) ([Fl_Boxtype](#) b)
Sets the down box type.

- [Fl_Color](#) **down_color** () const
(for backwards compatibility)
- void **down_color** (unsigned c)
(for backwards compatibility)
- [Fl_Button](#) (int X, int Y, int W, int H, const char *L=0)
The constructor creates the button using the given position, size, and label.
- virtual int **handle** (int)
Handles the specified event.
- int **set** ()
Same as `value(1)`.
- void **setonly** ()
Turns on this button and turns off all other radio buttons in the group (calling `value(1)` or `set()` does not do this).
- int **shortcut** () const
Returns the current shortcut key for the button.
- void **shortcut** (const char *s)
(for backwards compatibility)
- void **shortcut** (int s)
Sets the shortcut key to `s`.
- char **value** () const
Returns the current value of the button (0 or 1).
- int **value** (int v)
Sets the current value of the button.

Protected Member Functions

- virtual void **draw** ()
Draws the widget.
- void **simulate_key_action** ()

Static Protected Member Functions

- static void **key_release_timeout** (void *)

Static Protected Attributes

- static [Fl_Widget_Tracker](#) * **key_release_tracker** = 0

Additional Inherited Members

31.9.1 Detailed Description

Buttons generate callbacks when they are clicked by the user.

You control exactly when and how by changing the values for [type\(\)](#) and [when\(\)](#). Buttons can also generate callbacks in response to `FL_SHORTCUT` events. The button can either have an explicit [shortcut\(int s\)](#) value or a letter shortcut can be indicated in the [label\(\)](#) with an `'&'` character before it. For the label shortcut it does not matter if `Alt` is held down, but if you have an input field in the same window, the user will have to hold down the `Alt` key so that the input field does not eat the event first as an `FL_KEYBOARD` event.

Todo Refactor the doxygen comments for [Fl_Button type\(\)](#) documentation.

For an [Fl_Button](#) object, the [type\(\)](#) call returns one of:

- `FL_NORMAL_BUTTON` (0): [value\(\)](#) remains unchanged after button press.
- `FL_TOGGLE_BUTTON`: [value\(\)](#) is inverted after button press.

- `FL_RADIO_BUTTON`: `value()` is set to 1 after button press, and all other buttons in the current group with `type() == FL_RADIO_BUTTON` are set to zero.

Todo Refactor the doxygen comments for `Fl_Button when()` documentation.

For an `Fl_Button` object, the following `when()` values are useful, the default being `FL_WHEN_RELEASE`:

- 0: The callback is not done, instead `changed()` is turned on.
- `FL_WHEN_RELEASE`: The callback is done after the user successfully clicks the button, or when a shortcut is typed.
- `FL_WHEN_CHANGED`: The callback is done each time the `value()` changes (when the user pushes and releases the button, and as the mouse is dragged around in and out of the button).

31.9.2 Constructor & Destructor Documentation

31.9.2.1 Fl_Button()

```
Fl_Button::Fl_Button (
    int X,
    int Y,
    int W,
    int H,
    const char * L = 0 )
```

The constructor creates the button using the given position, size, and label.

The default box type is `box(FL_UP_BOX)`.

You can control how the button is drawn when ON by setting `down_box()`. The default is `FL_NO_BOX` (0) which will select an appropriate box type using the normal (OFF) box type by using `fl_down(box())`.

Derived classes may handle this differently.

Parameters

in	<i>X,Y,W,H</i>	position and size of the widget
in	<i>L</i>	widget label, default is no label

31.9.3 Member Function Documentation

31.9.3.1 clear()

```
int Fl_Button::clear ( ) [inline]
```

Same as `value(0)`.

See also

[value\(int v\)](#)

31.9.3.2 down_box() [1/2]

```
Fl_Boxtype Fl_Button::down_box ( ) const [inline]
```

Returns the current down box type, which is drawn when `value()` is non-zero.

Return values

<code>Fl_Boxtype</code>	
-------------------------	--

31.9.3.3 down_box() [2/2]

```
void Fl_Button::down_box (
    Fl_Boxtype b ) [inline]
```

Sets the down box type.

The default value of 0 causes FLTK to figure out the correct matching down version of `box()`.

Some derived classes (e.g. `Fl_Round_Button` and `Fl_Light_Button` use `down_box()` for special purposes. See docs of these classes.

Parameters

<code>in</code>	<code>b</code>	down box type
-----------------	----------------	---------------

31.9.3.4 draw()

```
void Fl_Button::draw ( ) [protected], [virtual]
```

Draws the widget.

Never call this function directly. FLTK will schedule redrawing whenever needed. If your widget must be redrawn as soon as possible, call `redraw()` instead.

Override this function to draw your own widgets.

If you ever need to call another widget's draw method *from within your own `draw()` method*, e.g. for an embedded scrollbar, you can do it (because `draw()` is virtual) like this:

```
Fl_Widget *s = &scroll; // scroll is an embedded Fl_Scrollbar
s->draw(); // calls Fl_Scrollbar::draw()
```

Implements `Fl_Widget`.

Reimplemented in `Fl_Light_Button`, and `Fl_Return_Button`.

31.9.3.5 handle()

```
int Fl_Button::handle (
    int event ) [virtual]
```

Handles the specified event.

You normally don't call this method directly, but instead let FLTK do it when the user interacts with the widget.

When implemented in a widget, this function must return 0 if the widget does not use the event or 1 otherwise.

Most of the time, you want to call the inherited `handle()` method in your overridden method so that you don't short-circuit events that you don't handle. In this last case you should return the callee retval.

Parameters

<code>in</code>	<code>event</code>	the kind of event received
-----------------	--------------------	----------------------------

Return values

<code>0</code>	if the event was not used or understood
<code>1</code>	if the event was used and can be deleted

See also

[FI_Event](#)

Reimplemented from [FI_Widget](#).

Reimplemented in [FI_Light_Button](#), [FI_Repeat_Button](#), and [FI_Return_Button](#).

31.9.3.6 set()

```
int FI_Button::set ( ) [inline]
```

Same as `value(1)`.

See also

[value\(int v\)](#)

31.9.3.7 shortcut() [1/2]

```
int FI_Button::shortcut ( ) const [inline]
```

Returns the current shortcut key for the button.

Return values

<i>int</i>	
------------	--

31.9.3.8 shortcut() [2/2]

```
void FI_Button::shortcut (
    int s ) [inline]
```

Sets the shortcut key to `s`.

Setting this overrides the use of '&' in the [label\(\)](#). The value is a bitwise OR of a key and a set of shift flags, for example: `FL_ALT | 'a'`, or `FL_ALT | (FL_F + 10)`, or just `'a'`. A value of 0 disables the shortcut.

The key can be any value returned by [FI::event_key\(\)](#), but will usually be an ASCII letter. Use a lower-case letter unless you require the shift key to be held down.

The shift flags can be any set of values accepted by [FI::event_state\(\)](#). If the bit is on, that shift key must be pushed. Meta, Alt, Ctrl, and Shift must be off if they are not in the shift flags (zero for the other bits indicates a "don't care" setting).

Parameters

<i>in</i>	<i>s</i>	bitwise OR of key and shift flags
-----------	----------	-----------------------------------

31.9.3.9 value()

```
int FI_Button::value (
    int v )
```

Sets the current value of the button.

A non-zero value sets the button to 1 (ON), and zero sets it to 0 (OFF).

Parameters

<i>in</i>	<i>v</i>	button value.
-----------	----------	---------------

See also

[set\(\)](#), [clear\(\)](#)

The documentation for this class was generated from the following files:

- [Fl_Button.H](#)
- [Fl_Button.cxx](#)

31.10 Fl_Cairo_State Class Reference

Contains all the necessary info on the current cairo context.

```
#include <Fl_Cairo.H>
```

Public Member Functions

- `bool autolink () const`
Gets the autolink option. See [Fl::cairo_autolink_context\(bool\)](#)
- `void autolink (bool b)`
Sets the autolink option, only available with `--enable-cairoext`.
- `cairo_t * cc () const`
Gets the current cairo context.
- `void cc (cairo_t *c, bool own=true)`
Sets the current cairo context.
- `void * gc () const`
Gets the last gc attached to a cc.
- `void gc (void *c)`
Sets the gc c to keep track on.
- `void * window () const`
Gets the last window attached to a cc.
- `void window (void *w)`
Sets the window w to keep track on.

31.10.1 Detailed Description

Contains all the necessary info on the current cairo context.

A private internal & unique corresponding object is created to permit cairo context state handling while keeping it opaque. For internal use only.

Note

Only available when configure has the `--enable-cairo` option

31.10.2 Member Function Documentation

31.10.2.1 cc()

```
void Fl_Cairo_State::cc (
    cairo_t * c,
    bool own = true ) [inline]
```

Sets the current cairo context.

`own == true` (the default) indicates that the cairo context `c` will be deleted by FLTK internally when another `cc` is set later.

`own == false` indicates `cc` deletion is handled externally by the user program.

The documentation for this class was generated from the following files:

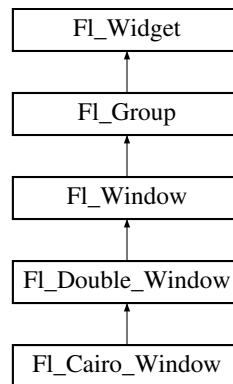
- [Fl_Cairo.H](#)
- [Fl_Cairo.cxx](#)

31.11 Fl_Cairo_Window Class Reference

This defines a pre-configured cairo fltk window.

```
#include <Fl_Cairo_Window.H>
```

Inheritance diagram for Fl_Cairo_Window:



Public Types

- typedef void(* [cairo_draw_cb](#)) ([Fl_Cairo_Window](#) *self, cairo_t *def)
This defines the cairo draw callback prototype that you must further.

Public Member Functions

- [Fl_Cairo_Window](#) (int w, int h)
- void [set_draw_cb](#) ([cairo_draw_cb](#) cb)
You must provide a draw callback which will implement your cairo rendering.

Protected Member Functions

- void [draw](#) ()
Overloaded to provide cairo callback support.

Additional Inherited Members

31.11.1 Detailed Description

This defines a pre-configured cairo fltk window.

This class overloads the virtual [draw\(\)](#) method for you, so that the only thing you have to do is to provide your cairo code. All cairo context handling is achieved transparently.

Note

You can alternatively define your custom cairo fltk window, and thus at least override the [draw\(\)](#) method to provide custom cairo support. In this case you will probably use [Fl::cairo_make_current\(Fl_Window*\)](#) to attach a context to your window. You should do it only when your window is the current window.

See also

[Fl_Window::current\(\)](#)

31.11.2 Member Function Documentation

31.11.2.1 draw()

```
void Fl_Cairo_Window::draw (
    void ) [inline], [protected], [virtual]
```

Overloaded to provide cairo callback support.

Reimplemented from [Fl_Window](#).

31.11.2.2 set_draw_cb()

```
void Fl_Cairo_Window::set_draw_cb (
    cairo_draw_cb cb ) [inline]
```

You must provide a draw callback which will implement your cairo rendering.

This method will permit you to set your cairo callback to `cb`.

The documentation for this class was generated from the following file:

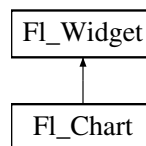
- [Fl_Cairo_Window.H](#)

31.12 Fl_Chart Class Reference

[Fl_Chart](#) displays simple charts.

```
#include <Fl_Chart.H>
```

Inheritance diagram for [Fl_Chart](#):



Public Member Functions

- void [add](#) (double val, const char *str=0, unsigned col=0)
 - Add the data value val with optional label str and color col to the chart.*
- [uchar autosize](#) () const
 - Get whether the chart will automatically adjust the bounds of the chart.*
- void [autosize](#) (uchar n)
 - Set whether the chart will automatically adjust the bounds of the chart.*
- void [bounds](#) (double *a, double *b) const
 - Gets the lower and upper bounds of the chart values.*
- void [bounds](#) (double a, double b)
 - Sets the lower and upper bounds of the chart values.*
- void [clear](#) ()
 - Removes all values from the chart.*
- [Fl_Chart](#) (int X, int Y, int W, int H, const char *L=0)
 - Create a new Fl_Chart widget using the given position, size and label string.*
- void [insert](#) (int ind, double val, const char *str=0, unsigned col=0)
 - Inserts a data value val at the given position ind.*
- int [maxsize](#) () const
 - Gets the maximum number of data values for a chart.*
- void [maxsize](#) (int m)
 - Set the maximum number of data values for a chart.*
- void [replace](#) (int ind, double val, const char *str=0, unsigned col=0)
 - Replace a data value val at the given position ind.*

- `int size () const`
Returns the number of data values in the chart.
- `void size (int W, int H)`
- `Fl_Color textcolor () const`
Gets the chart's text color.
- `void textcolor (Fl_Color n)`
gets the chart's text color to n.
- `Fl_Font textfont () const`
Gets the chart's text font.
- `void textfont (Fl_Font s)`
Sets the chart's text font to s.
- `Fl_Fontsize textsize () const`
Gets the chart's text size.
- `void textsize (Fl_Fontsize s)`
gets the chart's text size to s.
- `~Fl_Chart ()`
Destroys the `Fl_Chart` widget and all of its data.

Protected Member Functions

- `void draw ()`
Draws the widget.

Additional Inherited Members

31.12.1 Detailed Description

`Fl_Chart` displays simple charts.

It is provided for Forms compatibility.

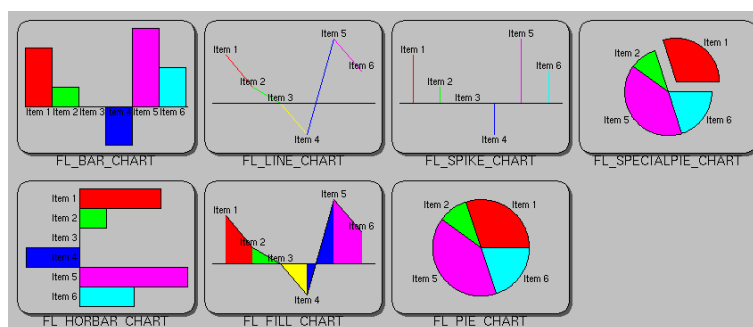


Figure 31.2 `Fl_Chart`

Todo Refactor `Fl_Chart::type()` information.

The type of an `Fl_Chart` object can be set using `type(uchar t)` to:

- `FL_BAR_CHART`: Each sample value is drawn as a vertical bar.
- `FL_FILLED_CHART`: The chart is filled from the bottom of the graph to the sample values.
- `FL_HORBAR_CHART`: Each sample value is drawn as a horizontal bar.
- `FL_LINE_CHART`: The chart is drawn as a polyline with vertices at each sample value.
- `FL_PIE_CHART`: A pie chart is drawn with each sample value being drawn as a proportionate slice in the circle.
- `FL_SPECIALPIE_CHART`: Like `FL_PIE_CHART`, but the first slice is separated from the pie.
- `FL_SPIKE_CHART`: Each sample value is drawn as a vertical line.

31.12.2 Constructor & Destructor Documentation

31.12.2.1 Fl_Chart()

```
Fl_Chart::Fl_Chart (
    int X,
    int Y,
    int W,
    int H,
    const char * L = 0 )
```

Create a new [Fl_Chart](#) widget using the given position, size and label string. The default boxstyle is `FL_NO_BOX`.

Parameters

in	<i>X,Y,W,H</i>	position and size of the widget
in	<i>L</i>	widget label, default is no label

31.12.3 Member Function Documentation

31.12.3.1 add()

```
void Fl_Chart::add (
    double val,
    const char * str = 0,
    unsigned col = 0 )
```

Add the data value `val` with optional label `str` and color `col` to the chart.

Parameters

in	<i>val</i>	data value
in	<i>str</i>	optional data label
in	<i>col</i>	optional data color

31.12.3.2 autosize() [1/2]

```
uchar Fl_Chart::autosize ( ) const [inline]
```

Get whether the chart will automatically adjust the bounds of the chart.

Returns

non-zero if auto-sizing is enabled and zero if disabled.

31.12.3.3 autosize() [2/2]

```
void Fl_Chart::autosize (
    uchar n ) [inline]
```

Set whether the chart will automatically adjust the bounds of the chart.

Parameters

in	<i>n</i>	non-zero to enable automatic resizing, zero to disable.
----	----------	---

31.12.3.4 bounds() [1/2]

```
void Fl_Chart::bounds (
    double * a,
    double * b ) const [inline]
```

Gets the lower and upper bounds of the chart values.

Parameters

out	<i>a,b</i>	are set to lower, upper
-----	------------	-------------------------

31.12.3.5 bounds() [2/2]

```
void Fl_Chart::bounds (
    double a,
    double b )
```

Sets the lower and upper bounds of the chart values.

Parameters

in	<i>a,b</i>	are used to set lower, upper
----	------------	------------------------------

31.12.3.6 draw()

```
void Fl_Chart::draw ( ) [protected], [virtual]
```

Draws the widget.

Never call this function directly. FLTK will schedule redrawing whenever needed. If your widget must be redrawn as soon as possible, call [redraw\(\)](#) instead.

Override this function to draw your own widgets.

If you ever need to call another widget's draw method *from within your own draw() method*, e.g. for an embedded scrollbar, you can do it (because [draw\(\)](#) is virtual) like this:

```
Fl_Widget *s = &scroll; // scroll is an embedded Fl_Scrollbar
s->draw(); // calls Fl_Scrollbar::draw()
```

Implements [Fl_Widget](#).

31.12.3.7 insert()

```
void Fl_Chart::insert (
    int ind,
    double val,
    const char * str = 0,
    unsigned col = 0 )
```

Inserts a data value *val* at the given position *ind*.

Position 1 is the first data value.

Parameters

in	<i>ind</i>	insertion position
----	------------	--------------------

Parameters

in	<i>val</i>	data value
in	<i>str</i>	optional data label
in	<i>col</i>	optional data color

31.12.3.8 maxsize()

```
void Fl_Chart::maxsize (
    int m )
```

Set the maximum number of data values for a chart.

If you do not call this method then the chart will be allowed to grow to any size depending on available memory.

Parameters

in	<i>m</i>	maximum number of data values allowed.
----	----------	--

31.12.3.9 replace()

```
void Fl_Chart::replace (
    int ind,
    double val,
    const char * str = 0,
    unsigned col = 0 )
```

Replace a data value *val* at the given position *ind*.

Position 1 is the first data value.

Parameters

in	<i>ind</i>	insertion position
in	<i>val</i>	data value
in	<i>str</i>	optional data label
in	<i>col</i>	optional data color

The documentation for this class was generated from the following files:

- Fl_Chart.H
- Fl_Chart.cxx

31.13 FL_CHART_ENTRY Struct Reference

For internal use only.

```
#include <Fl_Chart.H>
```

Public Attributes

- unsigned **col**
For internal use only.
- char **str** [FL_CHART_LABEL_MAX+1]
For internal use only.
- float **val**
For internal use only.

31.13.1 Detailed Description

For internal use only.

The documentation for this struct was generated from the following file:

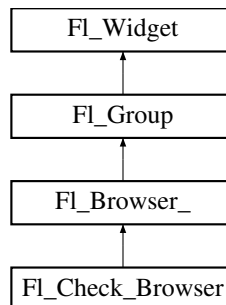
- FI_Chart.H

31.14 FI_Check_Browser Class Reference

The [FI_Check_Browser](#) widget displays a scrolling list of text lines that may be selected and/or checked by the user.

```
#include <FI_Check_Browser.H>
```

Inheritance diagram for FI_Check_Browser:



Public Member Functions

- int **add** (char *s)
Add a new unchecked line to the end of the browser.
- int **add** (char *s, int b)
Add a new line to the end of the browser.
- int **add** (const char *s)
*See int FI_Check_Browser::add(char *s)*
- int **add** (const char *s, int b)
*See int FI_Check_Browser::add(char *s)*
- void **check_all** ()
Sets all the items checked.
- void **check_none** ()
Sets all the items unchecked.
- int **checked** (int item) const
Gets the current status of item item.
- void **checked** (int item, int b)
Sets the check status of item item to b.
- void **clear** ()
Remove every item from the browser.
- **FI_Check_Browser** (int x, int y, int w, int h, const char *l=0)
The constructor makes an empty browser.
- int **nchecked** () const
Returns how many items are currently checked.
- int **nitems** () const
Returns how many lines are in the browser.
- int **remove** (int item)
Remove line n and make the browser one line shorter.
- void **set_checked** (int item)
Equivalent to FI_Check_Browser::checked(item, 1).

- char * **text** (int item) const
Return a pointer to an internal buffer holding item item's text.
- int **value** () const
Returns the index of the currently selected item.
- ~**Fl_Check_Browser** ()
The destructor deletes all list items and destroys the browser.

Protected Member Functions

- int **handle** (int)
Handles the event within the normal widget bounding box.

Additional Inherited Members

31.14.1 Detailed Description

The [Fl_Check_Browser](#) widget displays a scrolling list of text lines that may be selected and/or checked by the user.

31.14.2 Member Function Documentation

31.14.2.1 add() [1/2]

```
int Fl_Check_Browser::add (
    char * s )
```

Add a new unchecked line to the end of the browser.

See also

[add\(char *s, int b\)](#)

31.14.2.2 add() [2/2]

```
int Fl_Check_Browser::add (
    char * s,
    int b )
```

Add a new line to the end of the browser.

The text is copied using the `strdup()` function. It may also be NULL to make a blank line. It can set the item checked if `b` is not 0.

31.14.2.3 handle()

```
int Fl_Check_Browser::handle (
    int event ) [protected], [virtual]
```

Handles the `event` within the normal widget bounding box.

Parameters

in	<i>event</i>	The event to process.
----	--------------	-----------------------

Returns

1 if event was processed, 0 if not.

Reimplemented from [Fl_Browser_](#).

31.14.2.4 nchecked()

```
int Fl_Check_Browser::nchecked ( ) const [inline]
```

Returns how many items are currently checked.

31.14.2.5 nitems()

```
int Fl_Check_Browser::nitems ( ) const [inline]
```

Returns how many lines are in the browser.

The last line number is equal to this.

31.14.2.6 remove()

```
int Fl_Check_Browser::remove (
    int item )
```

Remove line n and make the browser one line shorter.

Returns the number of lines left in the browser.

31.14.2.7 set_checked()

```
void Fl_Check_Browser::set_checked (
    int item ) [inline]
```

Equivalent to `Fl_Check_Browser::checked(item, 1)`.

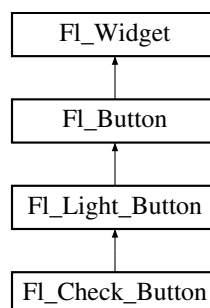
The documentation for this class was generated from the following files:

- `Fl_Check_Browser.H`
- `Fl_Check_Browser.cxx`

31.15 Fl_Check_Button Class Reference

A button with a "checkmark" to show its status.

Inheritance diagram for `Fl_Check_Button`:

**Public Member Functions**

- [Fl_Check_Button](#) (int X, int Y, int W, int H, const char *L=0)

Creates a new [Fl_Check_Button](#) widget using the given position, size, and label string.

Additional Inherited Members

31.15.1 Detailed Description

A button with a "checkmark" to show its status.

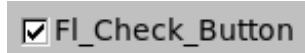


Figure 31.3 FI_Check_Button

Buttons generate callbacks when they are clicked by the user. You control exactly when and how by changing the values for `type()` and `when()`.

The `FI_Check_Button` subclass displays its "ON" state by showing a "checkmark" rather than drawing itself pushed in.

31.15.2 Constructor & Destructor Documentation

31.15.2.1 FI_Check_Button()

```
Fl_Check_Button::Fl_Check_Button (
    int X,
    int Y,
    int W,
    int H,
    const char * L = 0 )
```

Creates a new `FI_Check_Button` widget using the given position, size, and label string.

The default box type is `FL_NO_BOX`, which draws the label w/o a box right of the checkmark.

The `selection_color()` sets the color of the checkmark. Default is `FL_FOREGROUND_COLOR` (usually black).

You can use `down_box()` to change the box type of the checkmark. Default is `FL_DOWN_BOX`.

Parameters

in	<i>X,Y,W,H</i>	position and size of the widget
in	<i>L</i>	widget label, default is no label

The documentation for this class was generated from the following files:

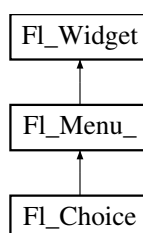
- `FI_Check_Button.H`
- `FI_Check_Button.cxx`

31.16 FI_Choice Class Reference

A button that is used to pop up a menu.

```
#include <Fl_Choice.H>
```

Inheritance diagram for `FI_Choice`:



Public Member Functions

- [Fl_Choice](#) (int X, int Y, int W, int H, const char *L=0)
Create a new [Fl_Choice](#) widget using the given position, size and label string.
- int [handle](#) (int)
Handles the specified event.
- int [value](#) () const
Gets the index of the last item chosen by the user.
- int [value](#) (const [Fl_Menu_Item](#) *v)
Sets the currently selected value using a pointer to menu item.
- int [value](#) (int v)
Sets the currently selected value using the index into the menu item array.

Protected Member Functions

- void [draw](#) ()
Draws the widget.

Additional Inherited Members

31.16.1 Detailed Description

A button that is used to pop up a menu.

This is a button that, when pushed, pops up a menu (or hierarchy of menus) defined by an array of [Fl_Menu_Item](#) objects. Motif calls this an `OptionButton`.

The only difference between this and a [Fl_Menu_Button](#) is that the name of the most recent chosen menu item is displayed inside the box, while the label is displayed outside the box. However, since the use of this is most often to control a single variable rather than do individual callbacks, some of the [Fl_Menu_Button](#) methods are redescribed here in those terms.

When the user clicks a menu item, [value\(\)](#) is set to that item and then:

- The item's callback is done if one has been set; the `Fl_Choice` is passed as the `Fl_Widget*` argument, along with any userdata configured for the callback.
- If the item does not have a callback, the `Fl_Choice` widget's callback is done instead, along with any userdata configured for it. The callback can determine which item was picked using [value\(\)](#), [mvalue\(\)](#), [item_pathname\(\)](#), etc.

All three mouse buttons pop up the menu. The Forms behavior of the first two buttons to increment/decrement the choice is not implemented. This could be added with a subclass, however.

The menu will also pop up in response to shortcuts indicated by putting a '&' character in the [label\(\)](#). See [Fl_Button::shortcut\(int s\)](#) for a description of this.

Typing the [shortcut\(\)](#) of any of the items will do exactly the same as when you pick the item with the mouse. The '&' character in item names are only looked at when the menu is popped up, however.

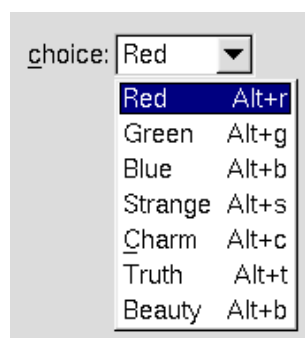


Figure 31.4 [Fl_Choice](#)

Todo Refactor the doxygen comments for `Fl_Choice changed()` documentation.

- `int Fl_Widget::changed() const` This value is true the user picks a different value. *It is turned off by `value()` and just before doing a callback (the callback can turn it back on if desired).*
- `void Fl_Widget::set_changed()` This method sets the `changed()` flag.
- `void Fl_Widget::clear_changed()` This method clears the `changed()` flag.
- `Fl_Boxtype Fl_Choice::down_box() const` Gets the current down box, which is used when the menu is popped up. The default down box type is `FL_DOWN_BOX`.
- `void Fl_Choice::down_box(Fl_Boxtype b)` Sets the current down box type to `b`.

31.16.2 Constructor & Destructor Documentation

31.16.2.1 Fl_Choice()

```
Fl_Choice::Fl_Choice (
    int X,
    int Y,
    int W,
    int H,
    const char * L = 0 )
```

Create a new `Fl_Choice` widget using the given position, size and label string.

The default boxtype is `FL_UP_BOX`.

The constructor sets `menu()` to `NULL`. See `Fl_Menu_` for the methods to set or change the menu.

Parameters

in	<code>X,Y,W,H</code>	position and size of the widget
in	<code>L</code>	widget label, default is no label

31.16.3 Member Function Documentation

31.16.3.1 draw()

```
void Fl_Choice::draw ( ) [protected], [virtual]
```

Draws the widget.

Never call this function directly. FLTK will schedule redrawing whenever needed. If your widget must be redrawn as soon as possible, call `redraw()` instead.

Override this function to draw your own widgets.

If you ever need to call another widget's draw method *from within your own `draw()` method*, e.g. for an embedded scrollbar, you can do it (because `draw()` is virtual) like this:

```
Fl_Widget *s = &scroll;           // scroll is an embedded Fl_Scrollbar
s->draw();                          // calls Fl_Scrollbar::draw()
```

Implements `Fl_Widget`.

31.16.3.2 handle()

```
int Fl_Choice::handle (
    int event ) [virtual]
```

Handles the specified event.

You normally don't call this method directly, but instead let FLTK do it when the user interacts with the widget.

When implemented in a widget, this function must return 0 if the widget does not use the event or 1 otherwise. Most of the time, you want to call the inherited [handle\(\)](#) method in your overridden method so that you don't short-circuit events that you don't handle. In this last case you should return the callee retval.

Parameters

<code>in</code>	<code>event</code>	the kind of event received
-----------------	--------------------	----------------------------

Return values

<code>0</code>	if the event was not used or understood
<code>1</code>	if the event was used and can be deleted

See also

[Fl_Event](#)

Reimplemented from [Fl_Widget](#).

31.16.3.3 value() [1/3]

```
int Fl_Choice::value ( ) const [inline]
```

Gets the index of the last item chosen by the user. The index is zero initially.

31.16.3.4 value() [2/3]

```
int Fl_Choice::value (
    const Fl_Menu_Item * v )
```

Sets the currently selected value using a pointer to menu item. Changing the selected value causes a [redraw\(\)](#).

Parameters

<code>in</code>	<code>v</code>	pointer to menu item in the menu item array.
-----------------	----------------	--

Returns

non-zero if the new value is different to the old one.

31.16.3.5 value() [3/3]

```
int Fl_Choice::value (
    int v )
```

Sets the currently selected value using the index into the menu item array. Changing the selected value causes a [redraw\(\)](#).

Parameters

<code>in</code>	<code>v</code>	index of value in the menu item array.
-----------------	----------------	--

Returns

non-zero if the new value is different to the old one.

The documentation for this class was generated from the following files:

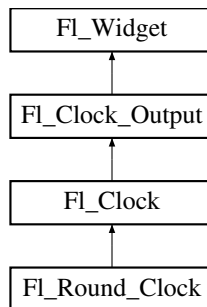
- `Fl_Choice.H`
- `Fl_Choice.cxx`

31.17 `Fl_Clock` Class Reference

This widget provides a round analog clock display.

```
#include <Fl_Clock.H>
```

Inheritance diagram for `Fl_Clock`:



Public Member Functions

- `Fl_Clock` (int X, int Y, int W, int H, const char *L=0)
Create an `Fl_Clock` widget using the given position, size, and label string.
- `Fl_Clock` (uchar t, int X, int Y, int W, int H, const char *L)
Create an `Fl_Clock` widget using the given boxtype, position, size, and label string.
- int `handle` (int)
Handles the specified event.
- `~Fl_Clock` ()
The destructor removes the clock.

Additional Inherited Members

31.17.1 Detailed Description

This widget provides a round analog clock display.

`Fl_Clock` is provided for Forms compatibility. It installs a 1-second timeout callback using `Fl::add_timeout()`. You can choose the rounded or square type of the clock with `type()`, see below.

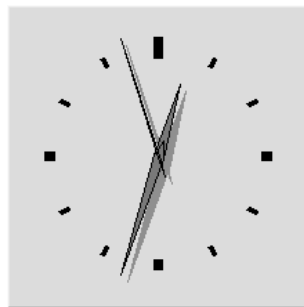


Figure 31.5 `FL_SQUARE_CLOCK` type

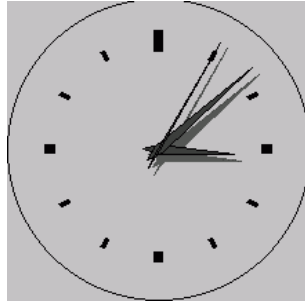


Figure 31.6 FL_ROUND_CLOCK type

31.17.2 Constructor & Destructor Documentation

31.17.2.1 Fl_Clock() [1/2]

```
Fl_Clock::Fl_Clock (
    int X,
    int Y,
    int W,
    int H,
    const char * L = 0 )
```

Create an [Fl_Clock](#) widget using the given position, size, and label string. The default boxtype is FL_NO_BOX.

Parameters

in	<i>X,Y,W,H</i>	position and size of the widget
in	<i>L</i>	widget label, default is no label

31.17.2.2 Fl_Clock() [2/2]

```
Fl_Clock::Fl_Clock (
    uchar t,
    int X,
    int Y,
    int W,
    int H,
    const char * L )
```

Create an [Fl_Clock](#) widget using the given boxtype, position, size, and label string.

Parameters

in	<i>t</i>	boxtype
in	<i>X,Y,W,H</i>	position and size of the widget
in	<i>L</i>	widget label, default is no label

31.17.3 Member Function Documentation

31.17.3.1 handle()

```
int Fl_Clock::handle (
    int event ) [virtual]
```

Handles the specified event.

You normally don't call this method directly, but instead let FLTK do it when the user interacts with the widget. When implemented in a widget, this function must return 0 if the widget does not use the event or 1 otherwise. Most of the time, you want to call the inherited [handle\(\)](#) method in your overridden method so that you don't short-circuit events that you don't handle. In this last case you should return the callee retval.

Parameters

in	<i>event</i>	the kind of event received
----	--------------	----------------------------

Return values

0	if the event was not used or understood
1	if the event was used and can be deleted

See also

[Fl_Event](#)

Reimplemented from [Fl_Widget](#).

The documentation for this class was generated from the following files:

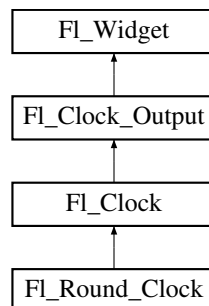
- Fl_Clock.H
- Fl_Clock.cxx

31.18 Fl_Clock_Output Class Reference

This widget can be used to display a program-supplied time.

```
#include <Fl_Clock.H>
```

Inheritance diagram for Fl_Clock_Output:



Public Member Functions

- [Fl_Clock_Output](#) (int X, int Y, int W, int H, const char *L=0)
Create a new [Fl_Clock_Output](#) widget with the given position, size and label.
- int [hour](#) () const
Returns the displayed hour (0 to 23).
- int [minute](#) () const
Returns the displayed minute (0 to 59).
- int [second](#) () const
Returns the displayed second (0 to 60, 60=leap second).

- `ulong value () const`
Returns the displayed time.
- `void value (int H, int m, int s)`
Set the displayed time.
- `void value (ulong v)`
Set the displayed time.

Protected Member Functions

- `void draw ()`
Draw clock with current position and size.
- `void draw (int X, int Y, int W, int H)`
Draw clock with the given position and size.

Additional Inherited Members

31.18.1 Detailed Description

This widget can be used to display a program-supplied time. The time shown on the clock is not updated. To display the current time, use [Fl_Clock](#) instead.

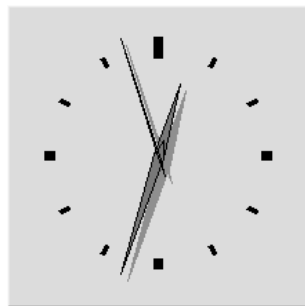


Figure 31.7 FL_SQUARE_CLOCK type

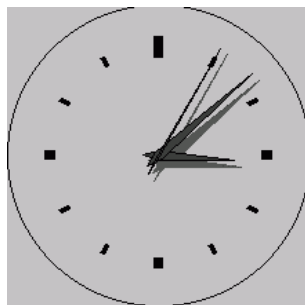


Figure 31.8 FL_ROUND_CLOCK type

31.18.2 Constructor & Destructor Documentation

31.18.2.1 Fl_Clock_Output()

```
Fl_Clock_Output::Fl_Clock_Output (
    int X,
    int Y,
    int W,
```

```

    int H,
    const char * L = 0 )

```

Create a new [Fl_Clock_Output](#) widget with the given position, size and label.
The default boxtype is `FL_NO_BOX`.

Parameters

in	<i>X,Y,W,H</i>	position and size of the widget
in	<i>L</i>	widget label, default is no label

31.18.3 Member Function Documentation

31.18.3.1 `draw()` [1/2]

```

void Fl_Clock_Output::draw (
    void ) [protected], [virtual]

```

Draw clock with current position and size.

Implements [Fl_Widget](#).

31.18.3.2 `draw()` [2/2]

```

void Fl_Clock_Output::draw (
    int X,
    int Y,
    int W,
    int H ) [protected]

```

Draw clock with the given position and size.

Parameters

in	<i>X,Y,W,H</i>	position and size
----	----------------	-------------------

31.18.3.3 `hour()`

```

int Fl_Clock_Output::hour ( ) const [inline]

```

Returns the displayed hour (0 to 23).

See also

[value\(\)](#), [minute\(\)](#), [second\(\)](#)

31.18.3.4 `minute()`

```

int Fl_Clock_Output::minute ( ) const [inline]

```

Returns the displayed minute (0 to 59).

See also

[value\(\)](#), [hour\(\)](#), [second\(\)](#)

31.18.3.5 second()

```
int Fl_Clock_Output::second ( ) const [inline]
```

Returns the displayed second (0 to 60, 60=leap second).

See also

[value\(\)](#), [hour\(\)](#), [minute\(\)](#)

31.18.3.6 value() [1/3]

```
ulong Fl_Clock_Output::value ( ) const [inline]
```

Returns the displayed time.

Returns the time in seconds since the UNIX epoch (January 1, 1970).

See also

[value\(ulong\)](#)

31.18.3.7 value() [2/3]

```
void Fl_Clock_Output::value (
    int H,
    int m,
    int s )
```

Set the displayed time.

Set the time in hours, minutes, and seconds.

Parameters

in	<i>H,m,s</i>	displayed time
----	--------------	----------------

See also

[hour\(\)](#), [minute\(\)](#), [second\(\)](#)

31.18.3.8 value() [3/3]

```
void Fl_Clock_Output::value (
    ulong v )
```

Set the displayed time.

Set the time in seconds since the UNIX epoch (January 1, 1970).

Parameters

in	<i>v</i>	seconds since epoch
----	----------	---------------------

See also

[value\(\)](#)

The documentation for this class was generated from the following files:

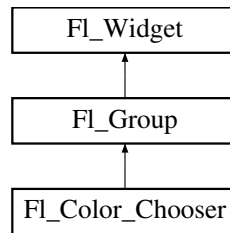
- Fl_Clock.H
- Fl_Clock.cxx

31.19 FI_Color_Chooser Class Reference

The `FI_Color_Chooser` widget provides a standard RGB color chooser.

```
#include <FI_Color_Chooser.H>
```

Inheritance diagram for `FI_Color_Chooser`:



Public Member Functions

- double `b` () const
Returns the current blue value.
- `FI_Color_Chooser` (int X, int Y, int W, int H, const char *L=0)
Creates a new FI_Color_Chooser widget using the given position, size, and label string.
- double `g` () const
Returns the current green value.
- int `hsv` (double H, double S, double V)
Set the hsv values.
- double `hue` () const
Returns the current hue.
- int `mode` ()
Returns which FI_Color_Chooser variant is currently active.
- void `mode` (int newMode)
Set which FI_Color_Chooser variant is currently active.
- double `r` () const
Returns the current red value.
- int `rgb` (double R, double G, double B)
Sets the current rgb color values.
- double `saturation` () const
Returns the saturation.
- double `value` () const
Returns the value/brightness.

Static Public Member Functions

- static void `hsv2rgb` (double H, double S, double V, double &R, double &G, double &B)
This static method converts HSV colors to RGB colorspace.
- static void `rgb2hsv` (double R, double G, double B, double &H, double &S, double &V)
This static method converts RGB colors to HSV colorspace.

Related Functions

(Note that these are not member functions.)

- int `fl_color_chooser` (const char *name, double &r, double &g, double &b, int cmode)
Pops up a window to let the user pick an arbitrary RGB color.
- int `fl_color_chooser` (const char *name, uchar &r, uchar &g, uchar &b, int cmode)
Pops up a window to let the user pick an arbitrary RGB color.

Additional Inherited Members

31.19.1 Detailed Description

The `Fl_Color_Chooser` widget provides a standard RGB color chooser.

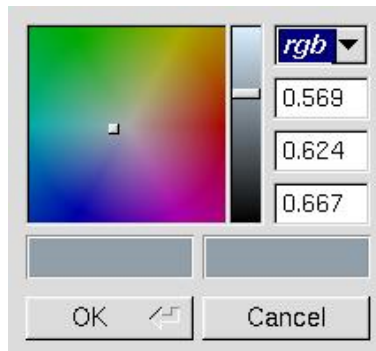


Figure 31.9 `fl_color_chooser()`

You can place any number of the widgets into a panel of your own design. The diagram shows the widget as part of a color chooser dialog created by the `fl_color_chooser()` function. The `Fl_Color_Chooser` widget contains the hue box, value slider, and rgb input fields from the above diagram (it does not have the color chips or the Cancel or OK buttons). The callback is done every time the user changes the rgb value. It is not done if they move the hue control in a way that produces the *same* rgb value, such as when saturation or value is zero.

The `fl_color_chooser()` function pops up a window to let the user pick an arbitrary RGB color. They can pick the hue and saturation in the "hue box" on the left (hold down CTRL to just change the saturation), and the brightness using the vertical slider. Or they can type the 8-bit numbers into the RGB `Fl_Value_Input` fields, or drag the mouse across them to adjust them. The pull-down menu lets the user set the input fields to show RGB, HSV, or 8-bit RGB (0 to 255).

`fl_color_chooser()` returns non-zero if the user picks ok, and updates the RGB values. If the user picks cancel or closes the window this returns zero and leaves RGB unchanged.

If you use the color chooser on an 8-bit screen, it will allocate all the available colors, leaving you no space to exactly represent the color the user picks! You can however use `fl_rectf()` to fill a region with a simulated color using dithering.

31.19.2 Constructor & Destructor Documentation

31.19.2.1 `Fl_Color_Chooser()`

```
Fl_Color_Chooser::Fl_Color_Chooser (
    int X,
    int Y,
    int W,
    int H,
    const char * L = 0 )
```

Creates a new `Fl_Color_Chooser` widget using the given position, size, and label string. The recommended dimensions are 200x95. The color is initialized to black.

Parameters

in	<code>X,Y,W,H</code>	position and size of the widget
in	<code>L</code>	widget label, default is no label

31.19.3 Member Function Documentation

31.19.3.1 b()

```
double Fl_Color_Chooser::b ( ) const [inline]
```

Returns the current blue value.

$0 \leq b \leq 1$.

31.19.3.2 g()

```
double Fl_Color_Chooser::g ( ) const [inline]
```

Returns the current green value.

$0 \leq g \leq 1$.

31.19.3.3 hsv()

```
int Fl_Color_Chooser::hsv (
    double H,
    double S,
    double V )
```

Set the hsv values.

The passed values are clamped (or for hue, modulus 6 is used) to get legal values. Does not do the callback.

Parameters

in	H, S, V	color components.
----	-----------	-------------------

Returns

1 if a new hsv value was set, 0 if the hsv value was the previous one.

31.19.3.4 hsv2rgb()

```
void Fl_Color_Chooser::hsv2rgb (
    double H,
    double S,
    double V,
    double & R,
    double & G,
    double & B ) [static]
```

This *static* method converts HSV colors to RGB colorspace.

Parameters

in	H, S, V	color components
out	R, G, B	color components

31.19.3.5 hue()

```
double Fl_Color_Chooser::hue ( ) const [inline]
```

Returns the current hue.

$0 \leq \text{hue} < 6$. Zero is red, one is yellow, two is green, etc. *This value is convenient for the internal calculations - some other systems consider hue to run from zero to one, or from 0 to 360.*

31.19.3.6 mode() [1/2]

```
int Fl_Color_Chooser::mode ( ) [inline]
```

Returns which [Fl_Color_Chooser](#) variant is currently active.

Returns

color modes are rgb(0), byte(1), hex(2), or hsv(3)

31.19.3.7 mode() [2/2]

```
void Fl_Color_Chooser::mode (
    int newMode )
```

Set which [Fl_Color_Chooser](#) variant is currently active.

Parameters

in	<i>newMode</i>	color modes are rgb(0), byte(1), hex(2), or hsv(3)
----	----------------	--

31.19.3.8 r()

```
double Fl_Color_Chooser::r ( ) const [inline]
```

Returns the current red value.

$0 \leq r \leq 1$.

31.19.3.9 rgb()

```
int Fl_Color_Chooser::rgb (
    double R,
    double G,
    double B )
```

Sets the current rgb color values.

Does not do the callback. Does not clamp (but out of range values will produce psychedelic effects in the hue selector).

Parameters

in	<i>R,G,B</i>	color components.
----	--------------	-------------------

Returns

1 if a new rgb value was set, 0 if the rgb value was the previous one.

31.19.3.10 rgb2hsv()

```
void Fl_Color_Chooser::rgb2hsv (
    double R,
    double G,
    double B,
    double & H,
    double & S,
    double & V ) [static]
```

This *static* method converts RGB colors to HSV colorspace.

Parameters

in	<i>R,G,B</i>	color components
out	<i>H,S,V</i>	color components

31.19.3.11 saturation()

```
double Fl_Color_Chooser::saturation ( ) const [inline]
```

Returns the saturation.

0 <= saturation <= 1.

31.19.3.12 value()

```
double Fl_Color_Chooser::value ( ) const [inline]
```

Returns the value/brightness.

0 <= value <= 1.

The documentation for this class was generated from the following files:

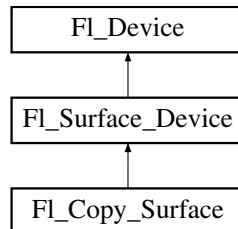
- [Fl_Color_Chooser.H](#)
- [Fl_Color_Chooser.cxx](#)

31.20 Fl_Copy_Surface Class Reference

Supports copying of graphical data to the clipboard.

```
#include <Fl_Copy_Surface.H>
```

Inheritance diagram for Fl_Copy_Surface:

**Public Member Functions**

- `const char * class_name ()`
Returns the name of the class of this object.
- `void draw (Fl_Widget *widget, int delta_x=0, int delta_y=0)`
Copies a widget in the clipboard.
- `void draw_decorated_window (Fl_Window *win, int delta_x=0, int delta_y=0)`
Copies a window and its borders and title bar to the clipboard.
- `Fl_Copy_Surface (int w, int h)`
Constructor.
- `int h ()`
Returns the pixel height of the copy surface.
- `void set_current ()`
Make this surface the current drawing surface.
- `int w ()`
Returns the pixel width of the copy surface.
- `~Fl_Copy_Surface ()`
Destructor.

Static Public Attributes

- static const char * `class_id` = "Fl_Copy_Surface"

Additional Inherited Members

31.20.1 Detailed Description

Supports copying of graphical data to the clipboard.

After creation of an [Fl_Copy_Surface](#) object, call [set_current\(\)](#) on it, and all subsequent graphics requests will be recorded in the clipboard. It's possible to draw widgets (using [Fl_Copy_Surface::draw\(\)](#)) or to use any of the [Drawing functions](#) or the [Color & Font functions](#). Finally, delete the [Fl_Copy_Surface](#) object to load the clipboard with the graphical data.

Fl_GL_Window 's can be copied to the clipboard as well.

Usage example:

```
Fl_Widget *g = ...; // a widget you want to copy to the clipboard
Fl_Copy_Surface *copy_surf = new Fl_Copy_Surface(g->w(), g->h()); // create an Fl_Copy_Surface object
copy_surf->set_current(); // direct graphics requests to the clipboard
fl_color(FL_WHITE); fl_rectf(0, 0, g->w(), g->h()); // draw a white background
copy_surf->draw(g); // draw the g widget in the clipboard
delete copy_surf; // after this, the clipboard is loaded
Fl_Display_Device::display_device()->set_current(); // direct graphics requests back to the display
```

Platform details:

- MSWindows: Transparent RGB images copy without transparency. The graphical data are copied to the clipboard as an 'enhanced metafile'.
- Mac OS: The graphical data are copied to the clipboard (a.k.a. pasteboard) in two 'flavors': 1) in vectorial form as PDF data; 2) in bitmap form as a TIFF image. Applications to which the clipboard content is pasted can use the flavor that suits them best.
- X11: the graphical data are copied to the clipboard as an image in BMP format.

31.20.2 Constructor & Destructor Documentation

31.20.2.1 Fl_Copy_Surface()

```
Fl_Copy_Surface::Fl_Copy_Surface (
    int w,
    int h )
```

Constructor.

Parameters

<code>w</code>	and
<code>h</code>	are the width and height of the clipboard surface in pixels where drawing will occur.

31.20.3 Member Function Documentation

31.20.3.1 class_name()

```
const char * Fl_Copy_Surface::class_name ( ) [inline], [virtual]
```

Returns the name of the class of this object.

Use of the [class_name\(\)](#) function is discouraged because it will be removed from future FLTK versions.

The class of an instance of an [Fl_Device](#) subclass can be checked with code such as:

```
if ( instance->class_name() == Fl_Printer::class_id ) { ... }
```

Reimplemented from [Fl_Device](#).

31.20.3.2 draw()

```
void Fl_Copy_Surface::draw (
    Fl_Widget * widget,
    int delta_x = 0,
    int delta_y = 0 )
```

Copies a widget in the clipboard.

Parameters

<i>widget</i>	any FLTK widget (e.g., standard, custom, window, GL view) to copy
<i>delta</i> _↔ <i>_x</i>	and
<i>delta</i> _↔ <i>_y</i>	give the position in the clipboard of the top-left corner of the widget

31.20.3.3 draw_decorated_window()

```
void Fl_Copy_Surface::draw_decorated_window (
    Fl_Window * win,
    int delta_x = 0,
    int delta_y = 0 )
```

Copies a window and its borders and title bar to the clipboard.

Parameters

<i>win</i>	an FLTK window to copy
<i>delta</i> _↔ <i>_x</i>	and
<i>delta</i> _↔ <i>_y</i>	give the position in the clipboard of the top-left corner of the window's title bar

31.20.3.4 set_current()

```
void Fl_Copy_Surface::set_current (
    void ) [virtual]
```

Make this surface the current drawing surface.

This surface will receive all future graphics requests.

Reimplemented from [Fl_Surface_Device](#).

The documentation for this class was generated from the following files:

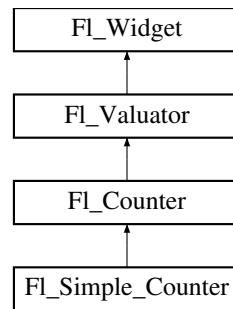
- `Fl_Copy_Surface.H`
- `Fl_Copy_Surface.cxx`

31.21 Fl_Counter Class Reference

Controls a single floating point value with button (or keyboard) arrows.

```
#include <Fl_Counter.H>
```

Inheritance diagram for `Fl_Counter`:



Public Member Functions

- [FI_Counter](#) (int X, int Y, int W, int H, const char *L=0)
Creates a new [FI_Counter](#) widget using the given position, size, and label string.
- int [handle](#) (int)
Handles the specified event.
- void [lstep](#) (double a)
Sets the increment for the large step buttons.
- double [step](#) () const
Returns the increment for normal step buttons.
- void [step](#) (double a)
Sets the increment for the normal step buttons.
- void [step](#) (double a, double b)
Sets the increments for the normal and large step buttons.
- [FI_Color](#) [textcolor](#) () const
Gets the font color.
- void [textcolor](#) ([FI_Color](#) s)
Sets the font color to s.
- [FI_Font](#) [textfont](#) () const
Gets the text font.
- void [textfont](#) ([FI_Font](#) s)
Sets the text font to s.
- [FI_Fontsize](#) [textsize](#) () const
Gets the font size.
- void [textsize](#) ([FI_Fontsize](#) s)
Sets the font size to s.
- [~FI_Counter](#) ()
Destroys the valuator.

Protected Member Functions

- void [draw](#) ()
Draws the widget.

Additional Inherited Members

31.21.1 Detailed Description

Controls a single floating point value with button (or keyboard) arrows. Double arrows buttons achieve larger steps than simple arrows.

See also

[Fl_Spinner](#) for `value` input with vertical `step` arrows.

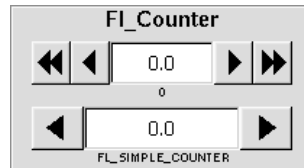


Figure 31.10 Fl_Counter

Todo Refactor the doxygen comments for [Fl_Counter type\(\)](#) documentation.

The type of an [Fl_Counter](#) object can be set using `type(uchar t)` to:

- `FL_NORMAL_COUNTER`: Displays a counter with 4 arrow buttons.
- `FL_SIMPLE_COUNTER`: Displays a counter with only 2 arrow buttons.

31.21.2 Constructor & Destructor Documentation

31.21.2.1 Fl_Counter()

```
Fl_Counter::Fl_Counter (
    int X,
    int Y,
    int W,
    int H,
    const char * L = 0 )
```

Creates a new [Fl_Counter](#) widget using the given position, size, and label string. The default type is `FL_NORMAL_COUNTER`.

Parameters

in	<code>X,Y,W,H</code>	position and size of the widget
in	<code>L</code>	widget label, default is no label

31.21.3 Member Function Documentation

31.21.3.1 draw()

```
void Fl_Counter::draw ( ) [protected], [virtual]
```

Draws the widget.

Never call this function directly. FLTK will schedule redrawing whenever needed. If your widget must be redrawn as soon as possible, call [redraw\(\)](#) instead.

Override this function to draw your own widgets.

If you ever need to call another widget's draw method *from within your own [draw\(\)](#) method*, e.g. for an embedded scrollbar, you can do it (because [draw\(\)](#) is virtual) like this:

```
Fl_Widget *s = &scroll; // scroll is an embedded Fl_Scrollbar
s->draw(); // calls Fl_Scrollbar::draw()
```

Implements [Fl_Widget](#).

31.21.3.2 handle()

```
int Fl_Counter::handle (
    int event ) [virtual]
```

Handles the specified event.

You normally don't call this method directly, but instead let FLTK do it when the user interacts with the widget.

When implemented in a widget, this function must return 0 if the widget does not use the event or 1 otherwise.

Most of the time, you want to call the inherited [handle\(\)](#) method in your overridden method so that you don't short-circuit events that you don't handle. In this last case you should return the callee retval.

Parameters

in	<i>event</i>	the kind of event received
----	--------------	----------------------------

Return values

0	if the event was not used or understood
1	if the event was used and can be deleted

See also

[Fl_Event](#)

Reimplemented from [Fl_Widget](#).

31.21.3.3 lstep()

```
void Fl_Counter::lstep (
    double a ) [inline]
```

Sets the increment for the large step buttons.

The default value is 1.0.

Parameters

in	<i>a</i>	large step increment.
----	----------	-----------------------

31.21.3.4 step() [1/2]

```
void Fl_Counter::step (
    double a ) [inline]
```

Sets the increment for the normal step buttons.

Parameters

in	<i>a</i>	normal step increment.
----	----------	------------------------

31.21.3.5 step() [2/2]

```
void Fl_Counter::step (
    double a,
    double b ) [inline]
```

Sets the increments for the normal and large step buttons.

Parameters

in	a,b	normal and large step increments.
----	-----	-----------------------------------

The documentation for this class was generated from the following files:

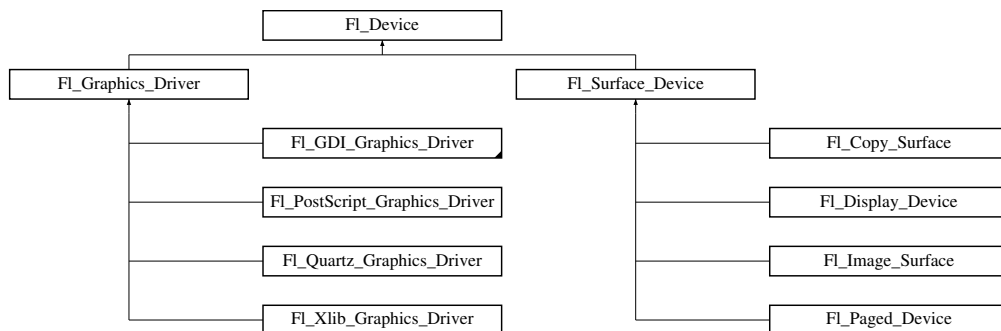
- FI_Counter.H
- FI_Counter.cxx

31.22 FI_Device Class Reference

All graphical output devices and all graphics systems.

```
#include <FI_Device.H>
```

Inheritance diagram for FI_Device:



Public Member Functions

- virtual const char * [class_name](#) ()
Returns the name of the class of this object.
- virtual [~FI_Device](#) ()
Virtual destructor.

Static Public Attributes

- static const char * [class_id](#) = "FI_Device"
A string that identifies each subclass of FI_Device.

31.22.1 Detailed Description

All graphical output devices and all graphics systems.

This class supports a rudimentary system of run-time type information.

31.22.2 Constructor & Destructor Documentation

31.22.2.1 [~FI_Device](#)()

```
virtual FI_Device::~FI_Device ( ) [inline], [virtual]
```

Virtual destructor.

The destructor of [FI_Device](#) must be virtual to make the destructors of derived classes being called correctly on destruction.

31.22.3 Member Function Documentation

31.22.3.1 class_name()

```
virtual const char * Fl_Device::class_name ( ) [inline], [virtual]
```

Returns the name of the class of this object.

Use of the `class_name()` function is discouraged because it will be removed from future FLTK versions.

The class of an instance of an `Fl_Device` subclass can be checked with code such as:

```
if ( instance->class_name() == Fl_Printer::class_id ) { ... }
```

Reimplemented in `Fl_Copy_Surface`, `Fl_Graphics_Driver`, `Fl_Quartz_Graphics_Driver`, `Fl_GDI_Graphics_Driver`, `Fl_GDI_Printer_Graphics_Driver`, `Fl_Xlib_Graphics_Driver`, `Fl_Surface_Device`, `Fl_Display_Device`, `Fl_Image_Surface`, `Fl_Paged_Device`, `Fl_PostScript_Graphics_Driver`, `Fl_PostScript_File_Device`, `Fl_System_Printer`, `Fl_PostScript_Printer`, and `Fl_Printer`.

31.22.4 Member Data Documentation

31.22.4.1 class_id

```
const char * Fl_Device::class_id = "Fl_Device" [static]
```

A string that identifies each subclass of `Fl_Device`.

Function `class_name()` applied to a device of this class returns this string.

The documentation for this class was generated from the following files:

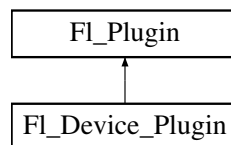
- [Fl_Device.H](#)
- [Fl_Device.cxx](#)

31.23 Fl_Device_Plugin Class Reference

This plugin socket allows the integration of new device drivers for special window or screen types.

```
#include <Fl_Device.H>
```

Inheritance diagram for `Fl_Device_Plugin`:



Public Member Functions

- `Fl_Device_Plugin` (const char *pluginName)
The constructor.
- virtual const char * **klass** ()
Returns the class name.
- virtual const char * **name** ()=0
Returns the plugin name.
- virtual int **print** (`Fl_Widget` *w, int x, int y, int height)=0
Prints a widget.
- `Fl_RGB_Image` * **rectangle_capture** (`Fl_Widget` *widget, int x, int y, int w, int h)
captures a rectangle of a widget as an image

31.23.1 Detailed Description

This plugin socket allows the integration of new device drivers for special window or screen types. This class is not intended for use outside the FLTK library. It is currently used to provide an automated printing service and screen capture for OpenGL windows, if linked with `fltk_gl`.

31.23.2 Member Function Documentation

31.23.2.1 `print()`

```
virtual int Fl_Device_Plugin::print (
    Fl_Widget * w,
    int x,
    int y,
    int height ) [pure virtual]
```

Prints a widget.

Parameters

<i>w</i>	the widget
<i>x,y</i>	offsets where to print relatively to coordinates origin
<i>height</i>	height of the current drawing area

31.23.2.2 `rectangle_capture()`

```
Fl_RGB_Image * Fl_Device_Plugin::rectangle_capture (
    Fl_Widget * widget,
    int x,
    int y,
    int w,
    int h ) [inline]
```

captures a rectangle of a widget as an image

Returns

The captured pixels as an RGB image

The documentation for this class was generated from the following file:

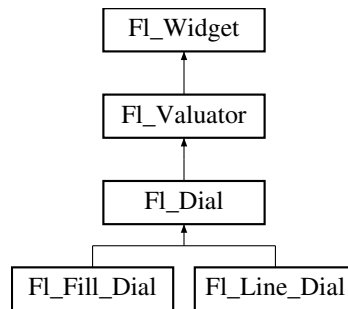
- [Fl_Device.H](#)

31.24 Fl_Dial Class Reference

The [Fl_Dial](#) widget provides a circular dial to control a single floating point value.

```
#include <Fl_Dial.H>
```

Inheritance diagram for [Fl_Dial](#):



Public Member Functions

- short [angle1](#) () const
Sets Or gets the angles used for the minimum and maximum values.
- void [angle1](#) (short a)
See short [angle1\(\)](#) const.
- short [angle2](#) () const
See short [angle1\(\)](#) const.
- void [angle2](#) (short a)
See short [angle1\(\)](#) const.
- void [angles](#) (short a, short b)
See short [angle1\(\)](#) const.
- [Fl_Dial](#) (int x, int y, int w, int h, const char *l=0)
Creates a new [Fl_Dial](#) widget using the given position, size, and label string.
- int [handle](#) (int)
Allow subclasses to handle event based on current position and size.

Protected Member Functions

- void [draw](#) ()
Draws dial at current position and size.
- void [draw](#) (int X, int Y, int W, int H)
Draws dial at given position and size.
- int [handle](#) (int event, int X, int Y, int W, int H)
Allows subclasses to handle event based on given position and size.

Additional Inherited Members

31.24.1 Detailed Description

The [Fl_Dial](#) widget provides a circular dial to control a single floating point value.

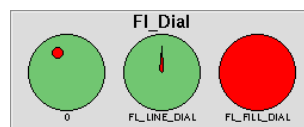


Figure 31.11 [Fl_Dial](#)

Use [type\(\)](#) to set the type of the dial to:

- `FL_NORMAL_DIAL` - Draws a normal dial with a knob.
- `FL_LINE_DIAL` - Draws a dial with a line.
- `FL_FILL_DIAL` - Draws a dial with a filled arc.

31.24.2 Constructor & Destructor Documentation

31.24.2.1 Fl_Dial()

```
Fl_Dial::Fl_Dial (
    int X,
    int Y,
    int W,
    int H,
    const char * l = 0 )
```

Creates a new [Fl_Dial](#) widget using the given position, size, and label string. The default type is FL_NORMAL_DIAL.

31.24.3 Member Function Documentation

31.24.3.1 angle1()

```
short Fl_Dial::angle1 ( ) const [inline]
```

Sets Or gets the angles used for the minimum and maximum values.

The default values are 45 and 315 (0 degrees is straight down and the angles progress clockwise). Normally angle1 is less than angle2, but if you reverse them the dial moves counter-clockwise.

31.24.3.2 draw() [1/2]

```
void Fl_Dial::draw (
    void ) [protected], [virtual]
```

Draws dial at current position and size.

Implements [Fl_Widget](#).

31.24.3.3 draw() [2/2]

```
void Fl_Dial::draw (
    int X,
    int Y,
    int W,
    int H ) [protected]
```

Draws dial at given position and size.

Parameters

in	X,Y,W,H	position and size
----	---------	-------------------

31.24.3.4 handle() [1/2]

```
int Fl_Dial::handle (
    int event,
    int X,
    int Y,
    int W,
    int H ) [protected]
```

Allows subclasses to handle event based on given position and size.

Parameters

in	<i>event,X,Y,W,H</i>	event to handle, related position and size.
----	----------------------	---

31.24.3.5 handle() [2/2]

```
int Fl_Dial::handle (
    int e ) [virtual]
```

Allow subclasses to handle event based on current position and size.

Reimplemented from [Fl_Widget](#).

The documentation for this class was generated from the following files:

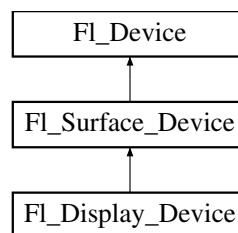
- [Fl_Dial.H](#)
- [Fl_Dial.cxx](#)

31.25 Fl_Display_Device Class Reference

A display to which the computer can draw.

```
#include <Fl_Device.H>
```

Inheritance diagram for [Fl_Display_Device](#):

**Public Member Functions**

- `const char * class_name ()`
Returns the name of the class of this object.
- `Fl_Display_Device (Fl_Graphics_Driver *graphics_driver)`
A constructor that sets the graphics driver used by the display.

Static Public Member Functions

- `static Fl_Display_Device * display_device ()`
Returns the platform display device.

Static Public Attributes

- `static const char * class_id = "Fl_Display_Device"`

Additional Inherited Members**31.25.1 Detailed Description**

A display to which the computer can draw.

When the program begins running, an [Fl_Display_Device](#) instance has been created and made the current drawing surface. There is no need to create any other object of this class.

31.25.2 Member Function Documentation

31.25.2.1 class_name()

```
const char * Fl_Display_Device::class_name ( ) [inline], [virtual]
```

Returns the name of the class of this object.

Use of the `class_name()` function is discouraged because it will be removed from future FLTK versions.

The class of an instance of an `Fl_Device` subclass can be checked with code such as:

```
if ( instance->class_name() == Fl_Printer::class_id ) { ... }
```

Reimplemented from `Fl_Surface_Device`.

The documentation for this class was generated from the following files:

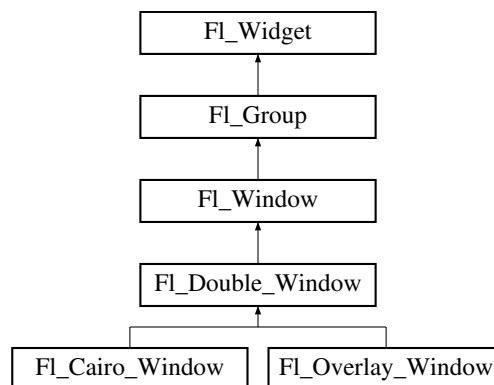
- [Fl_Device.H](#)
- [Fl_Device.cxx](#)

31.26 Fl_Double_Window Class Reference

The `Fl_Double_Window` provides a double-buffered window.

```
#include <Fl_Double_Window.H>
```

Inheritance diagram for `Fl_Double_Window`:



Public Member Functions

- `Fl_Double_Window` (int W, int H, const char *l=0)
Creates a new `Fl_Double_Window` widget using the given position, size, and label (title) string.
- `Fl_Double_Window` (int X, int Y, int W, int H, const char *l=0)
*See `Fl_Double_Window::Fl_Double_Window(int w, int h, const char *label = 0)`*
- void `flush` ()
Forces the window to be redrawn.
- void `hide` ()
Removes the window from the screen.
- void `resize` (int, int, int, int)
Changes the size and position of the window.
- void `show` ()
Puts the window on the screen.
- void `show` (int a, char **b)
- `~Fl_Double_Window` ()
The destructor also deletes all the children.

Protected Member Functions

- void [flush](#) (int eraseoverlay)
Forces the window to be redrawn.

Protected Attributes

- char [force_doublebuffering_](#)
Force double buffering, even if the OS already buffers windows (overlays need that on MacOS and Windows2000)

Additional Inherited Members

31.26.1 Detailed Description

The [Fl_Double_Window](#) provides a double-buffered window.

If possible this will use the X double buffering extension (Xdbe). If not, it will draw the window data into an off-screen pixmap, and then copy it to the on-screen window.

It is highly recommended that you put the following code before the first [show\(\)](#) of *any* window in your program:

```
Fl::visual (FL_DOUBLE|FL_INDEX)
```

This makes sure you can use Xdbe on servers where double buffering does not exist for every visual.

31.26.2 Constructor & Destructor Documentation

31.26.2.1 ~Fl_Double_Window()

```
Fl_Double_Window::~Fl_Double_Window ( )
```

The destructor *also deletes all the children*.

This allows a whole tree to be deleted at once, without having to keep a pointer to all the children in the user code.

31.26.3 Member Function Documentation

31.26.3.1 flush() [1/2]

```
void Fl_Double_Window::flush ( ) [virtual]
```

Forces the window to be redrawn.

Reimplemented from [Fl_Window](#).

Reimplemented in [Fl_Overlay_Window](#).

31.26.3.2 flush() [2/2]

```
void Fl_Double_Window::flush (
    int eraseoverlay ) [protected]
```

Forces the window to be redrawn.

Parameters

in	<i>eraseoverlay</i>	non-zero to erase overlay, zero to ignore
----	---------------------	---

[Fl_Overlay_Window](#) relies on flush(1) copying the back buffer to the front everywhere, even if [damage\(\)](#) == 0, thus erasing the overlay, and leaving the clip region set to the entire window.

31.26.3.3 hide()

```
void Fl_Double_Window::hide ( ) [virtual]
```

Removes the window from the screen.

If the window is already hidden or has not been shown then this does nothing and is harmless.

Reimplemented from [Fl_Window](#).

Reimplemented in [Fl_Overlay_Window](#).

31.26.3.4 `resize()`

```
void Fl_Double_Window::resize (
    int X,
    int Y,
    int W,
    int H ) [virtual]
```

Changes the size and position of the window.

If `shown()` is true, these changes are communicated to the window server (which may refuse that size and cause a further resize). If `shown()` is false, the size and position are used when `show()` is called. See [Fl_Group](#) for the effect of resizing on the child widgets.

You can also call the [Fl_Widget](#) methods `size(x,y)` and `position(w,h)`, which are inline wrappers for this virtual function.

A top-level window can not force, but merely suggest a position and size to the operating system. The window manager may not be willing or able to display a window at the desired position or with the given dimensions. It is up to the application developer to verify window parameters after the resize request.

Reimplemented from [Fl_Window](#).

Reimplemented in [Fl_Overlay_Window](#).

31.26.3.5 `show()`

```
void Fl_Double_Window::show ( ) [virtual]
```

Puts the window on the screen.

Usually (on X) this has the side effect of opening the display.

If the window is already shown then it is restored and raised to the top. This is really convenient because your program can call `show()` at any time, even if the window is already up. It also means that `show()` serves the purpose of `raise()` in other toolkits.

`Fl_Window::show(int argc, char **argv)` is used for top-level windows and allows standard arguments to be parsed from the command-line.

Note

For some obscure reasons `Fl_Window::show()` resets the current group by calling `Fl_Group::current(0)`. The comments in the code say "get rid of very common user bug: forgot end()". Although this is true it may have unwanted side effects if you `show()` an unrelated window (maybe for an error message or warning) while building a window or any other group widget.

Todo Check if we can remove resetting the current group in a later FLTK version (after 1.3.x). This may break "already broken" programs though if they rely on this "feature".

See also

[Fl_Window::show\(int argc, char **argv\)](#)

Reimplemented from [Fl_Window](#).

Reimplemented in [Fl_Overlay_Window](#).

The documentation for this class was generated from the following files:

- [Fl_Double_Window.H](#)
- [Fl_Double_Window.cxx](#)

31.27 Fl_End Class Reference

This is a dummy class that allows you to end a [Fl_Group](#) in a constructor list of a class:

```
#include <Fl_Group.H>
```

Public Member Functions

- `Fl_End ()`

All it does is calling `Fl_Group::current()->end()`

31.27.1 Detailed Description

This is a dummy class that allows you to end a `Fl_Group` in a constructor list of a class:

```
class MyClass {
  Fl_Group group;
  Fl_Button button_in_group;
  Fl_End end;
  Fl_Button button_outside_group;
  MyClass();
};
MyClass::MyClass() :
  group(10,10,100,100),
  button_in_group(20,20,60,30),
  end(),
  button_outside_group(10,120,60,30)
{ }
```

The documentation for this class was generated from the following file:

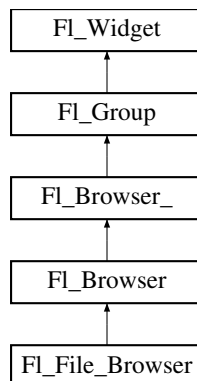
- `Fl_Group.H`

31.28 Fl_File_Browser Class Reference

The `Fl_File_Browser` widget displays a list of filenames, optionally with file-specific icons.

```
#include <Fl_File_Browser.H>
```

Inheritance diagram for `Fl_File_Browser`:



Public Types

- enum { `FILES` , `DIRECTORIES` }

Public Member Functions

- int `filetype ()` const

Sets or gets the file browser type, `FILES` or `DIRECTORIES`.

- void `filetype (int t)`

Sets or gets the file browser type, `FILES` or `DIRECTORIES`.

- const char * `filter ()` const

Sets or gets the filename filter.

- void `filter (const char *pattern)`

Sets or gets the filename filter.

- `Fl_File_Browser (int, int, int, int, const char * =0)`

The constructor creates the `Fl_File_Browser` widget at the specified position and size.

- `uchar iconsize () const`
Sets or gets the size of the icons.
- `void iconsize (uchar s)`
Sets or gets the size of the icons.
- `int load (const char *directory, Fl_File_Sort_F *sort=fl_numericsort)`
Loads the specified directory into the browser.
- `Fl_Fontsize textsize () const`
- `void textsize (Fl_Fontsize s)`

Additional Inherited Members

31.28.1 Detailed Description

The `Fl_File_Browser` widget displays a list of filenames, optionally with file-specific icons.

31.28.2 Constructor & Destructor Documentation

31.28.2.1 Fl_File_Browser()

```
Fl_File_Browser::Fl_File_Browser (
    int X,
    int Y,
    int W,
    int H,
    const char * l = 0 )
```

The constructor creates the `Fl_File_Browser` widget at the specified position and size. The destructor destroys the widget and frees all memory that has been allocated.

31.28.3 Member Function Documentation

31.28.3.1 filetype() [1/2]

```
int Fl_File_Browser::filetype ( ) const [inline]
```

Sets or gets the file browser type, FILES or DIRECTORIES.

When set to FILES, both files and directories are shown. Otherwise only directories are shown.

31.28.3.2 filetype() [2/2]

```
void Fl_File_Browser::filetype (
    int t ) [inline]
```

Sets or gets the file browser type, FILES or DIRECTORIES.

When set to FILES, both files and directories are shown. Otherwise only directories are shown.

31.28.3.3 filter() [1/2]

```
const char * Fl_File_Browser::filter ( ) const [inline]
```

Sets or gets the filename filter.

The pattern matching uses the `fl_filename_match()` function in FLTK.

31.28.3.4 filter() [2/2]

```
void Fl_File_Browser::filter (
    const char * pattern )
```

Sets or gets the filename filter.

The pattern matching uses the `fl_filename_match()` function in FLTK.

31.28.3.5 iconsize() [1/2]

```
uchar Fl_File_Browser::iconsize ( ) const [inline]
```

Sets or gets the size of the icons.

The default size is 20 pixels.

31.28.3.6 iconsize() [2/2]

```
void Fl_File_Browser::iconsize (
    uchar s ) [inline]
```

Sets or gets the size of the icons.

The default size is 20 pixels.

31.28.3.7 load()

```
int Fl_File_Browser::load (
    const char * directory,
    Fl_File_Sort_F * sort = fl_numeric_sort )
```

Loads the specified directory into the browser.

If icons have been loaded then the correct icon is associated with each file in the list.

The sort argument specifies a sort function to be used with [fl_filename_list\(\)](#).

The documentation for this class was generated from the following files:

- [Fl_File_Browser.H](#)
- [Fl_File_Browser.cxx](#)

31.29 FI_File_Chooser Class Reference

The [Fl_File_Chooser](#) widget displays a standard file selection dialog that supports various selection modes.

Public Types

- enum { **SINGLE** = 0 , **MULTI** = 1 , **CREATE** = 2 , **DIRECTORY** = 4 }

Public Member Functions

- [Fl_Widget](#) * **add_extra** ([Fl_Widget](#) *gr)
 - Adds extra widget at the bottom of [Fl_File_Chooser](#) window.*
- void **callback** (void(*cb)([Fl_File_Chooser](#) *, void *), void *d=0)
 - Sets the file chooser callback cb and associated data d.*
- [Fl_Color](#) **color** ()
 - Gets the background color of the [Fl_File_Browser](#) list.*
- void **color** ([Fl_Color](#) c)
 - Sets the background color of the [Fl_File_Browser](#) list.*
- int **count** ()
 - Returns the number of selected files.*
- char * **directory** ()
 - Gets the current directory.*
- void **directory** (const char *d)
 - Sets the current directory.*
- const char * **filter** ()
 - See void [filter\(const char *pattern\)](#)*
- void **filter** (const char *p)
 - Sets or gets the current filename filter patterns.*

- int **filter_value** ()
Gets the current filename filter selection.
- void **filter_value** (int f)
Sets the current filename filter selection.
- [FI_File_Chooser](#) (const char *d, const char *p, int t, const char *title)
The constructor creates the [FI_File_Chooser](#) dialog shown.
- void **hide** ()
Hides the [FI_File_Chooser](#) window.
- [uchar](#) **iconsize** ()
Gets the size of the icons in the [FI_File_Browser](#).
- void **iconsize** ([uchar](#) s)
Sets the size of the icons in the [FI_File_Browser](#).
- const char * **label** ()
Gets the title bar text for the [FI_File_Chooser](#).
- void **label** (const char *l)
Sets the title bar text for the [FI_File_Chooser](#).
- const char * **ok_label** ()
Gets the label for the "ok" button in the [FI_File_Chooser](#).
- void **ok_label** (const char *l)
Sets the label for the "ok" button in the [FI_File_Chooser](#).
- int **preview** () const
Returns the current state of the preview box.
- void **preview** (int e)
Enable or disable the preview tile.
- void **rescan** ()
Reloads the current directory in the [FI_File_Browser](#).
- void **rescan_keep_filename** ()
Rescan the current directory without clearing the filename, then select the file if it is in the list.
- void **show** ()
Shows the [FI_File_Chooser](#) window.
- int **shown** ()
Returns non-zero if the file chooser main window [show\(\)](#) has been called (but not [hide\(\)](#) see [FI_Window::shown\(\)](#))
- [FI_Color](#) **textcolor** ()
Gets the current [FI_File_Browser](#) text color.
- void **textcolor** ([FI_Color](#) c)
Sets the current [FI_File_Browser](#) text color.
- [FI_Font](#) **textfont** ()
Gets the current [FI_File_Browser](#) text font.
- void **textfont** ([FI_Font](#) f)
Sets the current [FI_File_Browser](#) text font.
- [FI_Fontsize](#) **textsize** ()
Gets the current [FI_File_Browser](#) text size.
- void **textsize** ([FI_Fontsize](#) s)
Sets the current [FI_File_Browser](#) text size.
- int **type** ()
Gets the current type of [FI_File_Chooser](#).
- void **type** (int t)
Sets the current type of [FI_File_Chooser](#).
- void * **user_data** () const
Gets the file chooser user data.
- void **user_data** (void *d)

- Sets the file chooser user data d.*
- void **value** (const char *filename)
 - Sets the current value of the selected file.*
- const char * **value** (int f=1)
 - Gets the current value of the selected file(s).*
- int **visible** ()
 - Returns 1 if the [Fl_File_Chooser](#) window is visible.*
- ~**Fl_File_Chooser** ()
 - Destroys the widget and frees all memory used by it.*

Public Attributes

- [Fl_Button](#) * **newButton**
 - The "new directory" button is exported so that application developers can control the appearance and use.*
- [Fl_Check_Button](#) * **previewButton**
 - The "preview" button is exported so that application developers can control the appearance and use.*
- [Fl_Check_Button](#) * **showHiddenButton**
 - When checked, hidden files (i.e., filename begins with dot) are displayed.*

Static Public Attributes

- static const char * **add_favorites_label** = "Add to Favorites"
 - [standard text may be customized at run-time]*
- static const char * **all_files_label** = "All Files (*)"
 - [standard text may be customized at run-time]*
- static const char * **custom_filter_label** = "Custom Filter"
 - [standard text may be customized at run-time]*
- static const char * **existing_file_label** = "Please choose an existing file!"
 - [standard text may be customized at run-time]*
- static const char * **favorites_label** = "Favorites"
 - [standard text may be customized at run-time]*
- static const char * **filename_label** = "Filename:"
 - [standard text may be customized at run-time]*
- static const char * **filesystems_label** = "File Systems"
 - [standard text may be customized at run-time]*
- static const char * **hidden_label** = "Show hidden files"
 - [standard text may be customized at run-time]*
- static const char * **manage_favorites_label** = "Manage Favorites"
 - [standard text may be customized at run-time]*
- static const char * **new_directory_label** = "New Directory?"
 - [standard text may be customized at run-time]*
- static const char * **new_directory_tooltip** = "Create a new directory."
 - [standard text may be customized at run-time]*
- static const char * **preview_label** = "Preview"
 - [standard text may be customized at run-time]*
- static const char * **save_label** = "Save"
 - [standard text may be customized at run-time]*
- static const char * **show_label** = "Show:"
 - [standard text may be customized at run-time]*
- static [Fl_File_Sort_F](#) * **sort** = fl_numericsort
 - the sort function that is used when loading the contents of a directory.*

Related Functions

(Note that these are not member functions.)

- char * [fl_dir_chooser](#) (const char *message, const char *fname, int relative)
Shows a file chooser dialog and gets a directory.
- char * [fl_file_chooser](#) (const char *message, const char *pat, const char *fname, int relative)
Shows a file chooser dialog and gets a filename.
- void [fl_file_chooser_callback](#) (void(*cb)(const char *))
Set the file chooser callback.
- void [fl_file_chooser_ok_label](#) (const char *)
Set the "OK" button label.

31.29.1 Detailed Description

The [FI_File_Chooser](#) widget displays a standard file selection dialog that supports various selection modes.

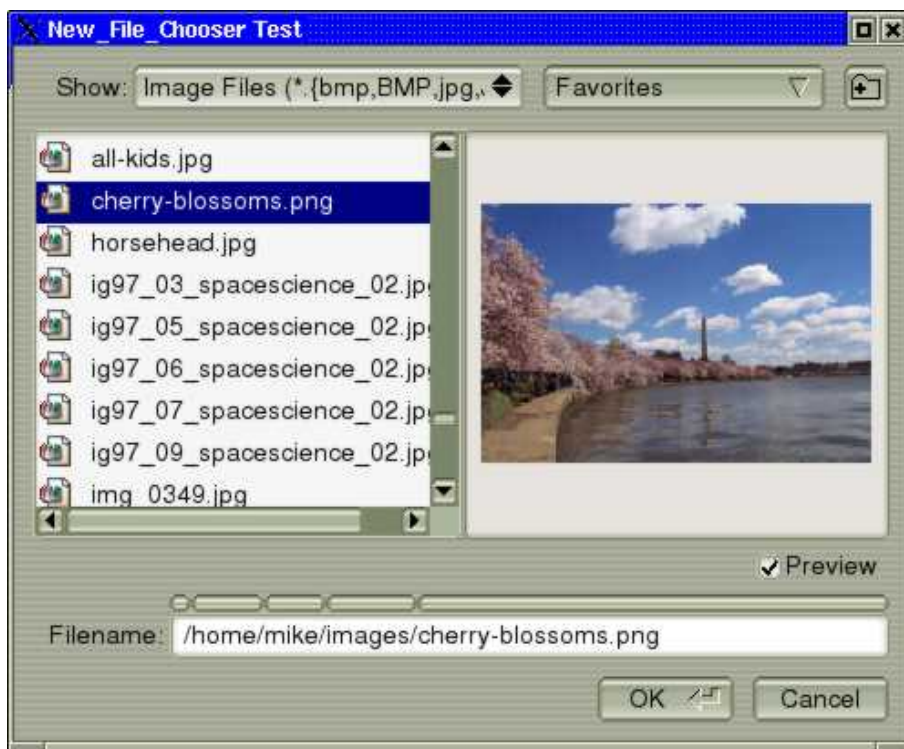


Figure 31.12 FI_File_Chooser

The [FI_File_Chooser](#) widget transmits UTF-8 encoded filenames to its user. It is recommended to open files that may have non-ASCII names with the [fl_fopen\(\)](#) or [fl_open\(\)](#) utility functions that handle these names in a cross-platform way (whereas the standard [fopen\(\)/open\(\)](#) functions fail on the MSWindows platform to open files with a non-ASCII name).

The [FI_File_Chooser](#) class also exports several static values that may be used to localize or customize the appearance of all file chooser dialogs:

Member	Default value
<code>add_favorites_label</code>	"Add to Favorites"
<code>all_files_label</code>	"All Files (*)"
<code>custom_filter_label</code>	"Custom Filter"
<code>existing_file_label</code>	"Please choose an existing file!"
<code>favorites_label</code>	"Favorites"

Member	Default value
filename_label	"Filename:"
filesystems_label	"My Computer" (WIN32) "File Systems" (all others)
hidden_label	"Show hidden files:"
manage_favorites_label	"Manage Favorites"
new_directory_label	"New Directory?"
new_directory_tooltip	"Create a new directory."
preview_label	"Preview"
save_label	"Save"
show_label	"Show:"
sort	fl_numericsort

The [Fl_File_Chooser::sort](#) member specifies the sort function that is used when loading the contents of a directory and can be customized at run-time.

The [Fl_File_Chooser](#) class also exports the [Fl_File_Chooser::newButton](#) and [Fl_File_Chooser::previewButton](#) widgets so that application developers can control their appearance and use. For more complex customization, consider copying the FLTK file chooser code and changing it accordingly.

31.29.2 Constructor & Destructor Documentation

31.29.2.1 Fl_File_Chooser()

```
Fl_File_Chooser::Fl_File_Chooser (
    const char * pathname,
    const char * pattern,
    int type,
    const char * title )
```

The constructor creates the [Fl_File_Chooser](#) dialog shown.

The pathname argument can be a directory name or a complete file name (in which case the corresponding file is highlighted in the list and in the filename input field.)

The pattern argument can be a NULL string or "*" to list all files, or it can be a series of descriptions and filter strings separated by tab characters (\t). The format of filters is either "Description text (patterns)" or just "patterns". A file chooser that provides filters for HTML and image files might look like:

```
"HTML Files (*.html)\tImage Files (*.bmp,gif,jpg,png)"
```

The file chooser will automatically add the "All Files (*)" pattern to the end of the string you pass if you do not provide one. The first filter in the string is the default filter.

See the FLTK documentation on [fl_filename_match\(\)](#) for the kinds of pattern strings that are supported.

The type argument can be one of the following:

- SINGLE - allows the user to select a single, existing file.
- MULTI - allows the user to select one or more existing files.
- CREATE - allows the user to select a single, existing file or specify a new filename.
- DIRECTORY - allows the user to select a single, existing directory.

The title argument is used to set the title bar text for the [Fl_File_Chooser](#) window.

31.29.3 Member Function Documentation

31.29.3.1 add_extra()

```
Fl_Widget * Fl_File_Chooser::add_extra (
    Fl_Widget * gr )
```

Adds extra widget at the bottom of [Fl_File_Chooser](#) window.

Returns pointer for previous extra widget or NULL if not set previously. If argument is NULL only remove previous extra widget.

Note

[Fl_File_Chooser](#) does **not** delete extra widget in destructor! To prevent memory leakage, don't forget to delete unused extra widgets

31.29.3.2 filter()

```
void Fl_File_Chooser::filter (
    const char * pattern )
```

Sets or gets the current filename filter patterns.

The filter patterns use [fl_filename_match\(\)](#). Multiple patterns can be used by separating them with tabs, like "*.jpg\t*.png\t*.gif\t* ". In addition, you can provide human-readable labels with the patterns inside parenthesis, like "JPEG Files (*.jpg)\tPNG Files (*.png)\tGIF Files (*.gif)\tAll Files (*)" .

Use filter(NULL) to show all files.

31.29.3.3 iconsize() [1/2]

```
uchar Fl_File_Chooser::iconsize ( )
```

Gets the size of the icons in the [Fl_File_Browser](#).

By default the icon size is set to 1.5 times the [textsize\(\)](#).

31.29.3.4 iconsize() [2/2]

```
void Fl_File_Chooser::iconsize (
    uchar s )
```

Sets the size of the icons in the [Fl_File_Browser](#).

By default the icon size is set to 1.5 times the [textsize\(\)](#).

31.29.3.5 preview()

```
void Fl_File_Chooser::preview (
    int e )
```

Enable or disable the preview tile.

1 = enable preview, 0 = disable preview.

31.29.3.6 value()

```
const char * Fl_File_Chooser::value (
    int f = 1 )
```

Gets the current value of the selected file(s).

f is a 1-based index into a list of file names. The number of selected files is returned by [Fl_File_Chooser::count\(\)](#).

This sample code loops through all selected files:

```
// Get list of filenames user selected from a MULTI chooser
for ( int t=1; t<=chooser->count(); t++ ) {
const char *filename = chooser->value(t);
...
}
```

31.29.4 Member Data Documentation

31.29.4.1 showHiddenButton

`Fl_File_Chooser::showHiddenButton`

When checked, hidden files (i.e., filename begins with dot) are displayed.

The "showHiddenButton" button is exported so that application developers can control its appearance.

The documentation for this class was generated from the following files:

- `Fl_File_Chooser.H`
- `Fl_File_Chooser.cxx`
- `Fl_File_Chooser2.cxx`
- `fl_file_dir.cxx`

31.30 FI_File_Icon Class Reference

The `Fl_File_Icon` class manages icon images that can be used as labels in other widgets and as icons in the `FileBrowser` widget.

```
#include <Fl_File_Icon.H>
```

Public Types

- enum {
 ANY , **PLAIN** , **FIFO** , **DEVICE** ,
 LINK , **DIRECTORY** }
- enum {
 END , **COLOR** , **LINE** , **CLOSEDLINE** ,
 POLYGON , **OUTLINEPOLYGON** , **VERTEX** }

Public Member Functions

- short * `add` (short d)
 Adds a keyword value to the icon array, returning a pointer to it.
- short * `add_color` (`Fl_Color` c)
 Adds a color value to the icon array, returning a pointer to it.
- short * `add_vertex` (float x, float y)
 Adds a vertex value to the icon array, returning a pointer to it.
- short * `add_vertex` (int x, int y)
 Adds a vertex value to the icon array, returning a pointer to it.
- void `clear` ()
 Clears all icon data from the icon.
- void `draw` (int x, int y, int w, int h, `Fl_Color` ic, int active=1)
 Draws an icon in the indicated area.
- `Fl_File_Icon` (const char *p, int t, int nd=0, short *d=0)
 Creates a new `Fl_File_Icon` with the specified information.
- void `label` (`Fl_Widget` *w)
 Applies the icon to the widget, registering the `Fl_File_Icon` label type as needed.
- void `load` (const char *f)
 Loads the specified icon image.
- int `load_fti` (const char *fti)
 Loads an SGI icon file.
- int `load_image` (const char *i)
 Load an image icon file from an image filename.
- `Fl_File_Icon` * `next` ()
 Returns next file icon object.
- const char * `pattern` ()

- Returns the filename matching pattern for the icon.*

 - int **size** ()

Returns the number of words of data used by the icon.
 - int **type** ()

Returns the filetype associated with the icon, which can be one of the following:
 - short * **value** ()

Returns the data array for the icon.
- ~**Fl_File_Icon** ()

The destructor destroys the icon and frees all memory that has been allocated for it.

Static Public Member Functions

- static **Fl_File_Icon** * **find** (const char *filename, int filetype=ANY)

Finds an icon that matches the given filename and file type.
- static **Fl_File_Icon** * **first** ()

Returns a pointer to the first icon in the list.
- static void **labeltype** (const **Fl_Label** *o, int x, int y, int w, int h, **Fl_Align** a)

Draw the icon label.
- static void **load_system_icons** (void)

Loads all system-defined icons.

31.30.1 Detailed Description

The **Fl_File_Icon** class manages icon images that can be used as labels in other widgets and as icons in the FileBrowser widget.

31.30.2 Constructor & Destructor Documentation

31.30.2.1 Fl_File_Icon()

```
Fl_File_Icon::Fl_File_Icon (
    const char * p,
    int t,
    int nd = 0,
    short * d = 0 )
```

Creates a new **Fl_File_Icon** with the specified information.

Parameters

in	<i>p</i>	filename pattern
in	<i>t</i>	file type
in	<i>nd</i>	number of data values
in	<i>d</i>	data values

31.30.3 Member Function Documentation

31.30.3.1 add()

```
short * Fl_File_Icon::add (
    short d )
```

Adds a keyword value to the icon array, returning a pointer to it.

Parameters

in	<i>d</i>	data value
----	----------	------------

31.30.3.2 add_color()

```
short * Fl_File_Icon::add_color (
    Fl_Color c ) [inline]
```

Adds a color value to the icon array, returning a pointer to it.

Parameters

in	<i>c</i>	color value
----	----------	-------------

31.30.3.3 add_vertex() [1/2]

```
short * Fl_File_Icon::add_vertex (
    float x,
    float y ) [inline]
```

Adds a vertex value to the icon array, returning a pointer to it.

The floating point version goes from 0.0 to 1.0. The origin (0.0) is in the lower-left hand corner of the icon.

Parameters

in	<i>x,y</i>	vertex coordinates
----	------------	--------------------

31.30.3.4 add_vertex() [2/2]

```
short * Fl_File_Icon::add_vertex (
    int x,
    int y ) [inline]
```

Adds a vertex value to the icon array, returning a pointer to it.

The integer version accepts coordinates from 0 to 10000. The origin (0.0) is in the lower-left hand corner of the icon.

Parameters

in	<i>x,y</i>	vertex coordinates
----	------------	--------------------

31.30.3.5 draw()

```
void Fl_File_Icon::draw (
    int x,
    int y,
    int w,
    int h,
    Fl_Color ic,
    int active = 1 )
```

Draws an icon in the indicated area.

Parameters

in	<i>x,y,w,h</i>	position and size
in	<i>ic</i>	icon color
in	<i>active</i>	status, default is active [non-zero]

31.30.3.6 find()

```
Fl_File_Icon * Fl_File_Icon::find (
    const char * filename,
    int filetype = ANY ) [static]
```

Finds an icon that matches the given filename and file type.

Parameters

in	<i>filename</i>	name of file
in	<i>filetype</i>	enumerated file type

Returns

matching file icon or NULL

31.30.3.7 label()

```
void Fl_File_Icon::label (
    Fl_Widget * w )
```

Applies the icon to the widget, registering the [Fl_File_Icon](#) label type as needed.

Parameters

in	<i>w</i>	widget for which this icon will become the label
----	----------	--

31.30.3.8 labeltype()

```
void Fl_File_Icon::labeltype (
    const Fl_Label * o,
    int x,
    int y,
    int w,
    int h,
    Fl_Align a ) [static]
```

Draw the icon label.

Parameters

in	<i>o</i>	label data
in	<i>x,y,w,h</i>	position and size of label
in	<i>a</i>	label alignment [not used]

31.30.3.9 load()

```
void Fl_File_Icon::load (
    const char * f )
```

Loads the specified icon image.
The format is deduced from the filename.

Parameters

in	<i>f</i>	filename
----	----------	----------

31.30.3.10 load_fti()

```
int Fl_File_Icon::load_fti (
    const char * fti )
```

Loads an SGI icon file.

Parameters

in	<i>fti</i>	icon filename
----	------------	---------------

Returns

0 on success, non-zero on error

31.30.3.11 load_image()

```
int Fl_File_Icon::load_image (
    const char * ifile )
```

Load an image icon file from an image filename.

Parameters

in	<i>ifile</i>	image filename
----	--------------	----------------

Returns

0 on success, non-zero on error

31.30.3.12 load_system_icons()

```
void Fl_File_Icon::load_system_icons (
    void ) [static]
```

Loads all system-defined icons.

This call is useful when using the FileChooser widget and should be used when the application starts:

```
Fl\_File\_Icon::load\_system\_icons\(\);
```

31.30.3.13 next()

```
Fl\_File\_Icon * Fl\_File\_Icon::next ( ) [inline]
```

Returns next file icon object.

See [Fl_File_Icon::first\(\)](#)

31.30.3.14 type()

```
int Fl_File_Icon::type ( ) [inline]
```

Returns the filetype associated with the icon, which can be one of the following:

- `Fl_File_Icon::ANY`, any kind of file.
- `Fl_File_Icon::PLAIN`, plain files.
- `Fl_File_Icon::FIFO`, named pipes.
- `Fl_File_Icon::DEVICE`, character and block devices.
- `Fl_File_Icon::LINK`, symbolic links.
- `Fl_File_Icon::DIRECTORY`, directories.

The documentation for this class was generated from the following files:

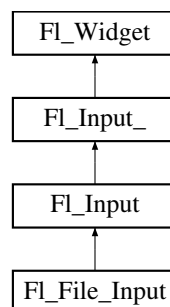
- `Fl_File_Icon.H`
- `Fl_File_Icon.cxx`
- `Fl_File_Icon2.cxx`

31.31 Fl_File_Input Class Reference

This widget displays a pathname in a text input field.

```
#include <Fl_File_Input.H>
```

Inheritance diagram for `Fl_File_Input`:



Public Member Functions

- `Fl_Boxtype down_box () const`
Gets the box type used for the navigation bar.
- `void down_box (Fl_Boxtype b)`
Sets the box type to use for the navigation bar.
- `Fl_Color errorcolor () const`
Gets the current error color.
- `void errorcolor (Fl_Color c)`
Sets the current error color to c.
- `Fl_File_Input (int X, int Y, int W, int H, const char *L=0)`
Creates a new Fl_File_Input widget using the given position, size, and label string.
- `virtual int handle (int event)`
Handle events in the widget.
- `const char * value ()`
Returns the current value, which is a pointer to an internal buffer and is valid only until the next event is handled.
- `int value (const char *str)`
Sets the value of the widget given a new string value.
- `int value (const char *str, int len)`
Sets the value of the widget given a new string value and its length.

Protected Member Functions

- virtual void `draw ()`

Draws the file input widget.

Additional Inherited Members

31.31.1 Detailed Description

This widget displays a pathname in a text input field.

A navigation bar located above the input field allows the user to navigate upward in the directory tree. You may want to handle `FL_WHEN_CHANGED` events for tracking text changes and also `FL_WHEN_RELEASE` for button release when changing to parent dir. `FL_WHEN_RELEASE` callback won't be called if the directory clicked is the same as the current one.

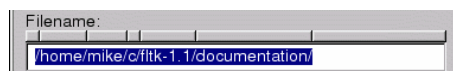


Figure 31.13 `Fl_File_Input`

Note

As all `Fl_Input` derived objects, `Fl_File_Input` may call its callback when losing focus (see `FL_UNFOCUS`) to update its state like its cursor shape. One resulting side effect is that you should call `clear_changed()` early in your callback to avoid reentrant calls if you plan to show another window or dialog box in the callback.

31.31.2 Constructor & Destructor Documentation

31.31.2.1 `Fl_File_Input()`

```
Fl_File_Input::Fl_File_Input (
    int X,
    int Y,
    int W,
    int H,
    const char * L = 0 )
```

Creates a new `Fl_File_Input` widget using the given position, size, and label string. The default boxtype is `FL_DOWN_BOX`.

Parameters

in	<code>X,Y,W,H</code>	position and size of the widget
in	<code>L</code>	widget label, default is no label

31.31.3 Member Function Documentation

31.31.3.1 `down_box()`

```
void Fl_File_Input::down_box (
    Fl_Boxtype b ) [inline]
```

Sets the box type to use for the navigation bar.

31.31.3.2 draw()

```
void Fl_File_Input::draw (
    void ) [protected], [virtual]
```

Draws the file input widget.
Implements [Fl_Widget](#).

31.31.3.3 errorcolor()

```
Fl_Color Fl_File_Input::errorcolor ( ) const [inline]
```

Gets the current error color.

Todo Better docs for [Fl_File_Input::errorcolor\(\)](#) - is it even used?

31.31.3.4 handle()

```
int Fl_File_Input::handle (
    int event ) [virtual]
```

Handle events in the widget.
Return non zero if event is handled.

Parameters

in	<i>event</i>	
----	--------------	--

Reimplemented from [Fl_Widget](#).

31.31.3.5 value() [1/2]

```
int Fl_File_Input::value (
    const char * str )
```

Sets the value of the widget given a new string value.
Returns non 0 on success.

Parameters

in	<i>str</i>	new string value
----	------------	------------------

31.31.3.6 value() [2/2]

```
int Fl_File_Input::value (
    const char * str,
    int len )
```

Sets the value of the widget given a new string value and its length.
Returns non 0 on success.

Parameters

in	<i>str</i>	new string value
in	<i>len</i>	length of value

The documentation for this class was generated from the following files:

- [Fl_File_Input.H](#)

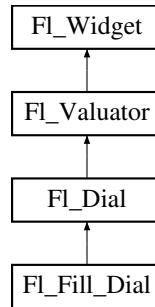
- `Fl_File_Input.cxx`

31.32 `Fl_Fill_Dial` Class Reference

Draws a dial with a filled arc.

```
#include <Fl_Fill_Dial.H>
```

Inheritance diagram for `Fl_Fill_Dial`:



Public Member Functions

- **`Fl_Fill_Dial`** (int X, int Y, int W, int H, const char *L)
Creates a filled dial, also setting its type to `FL_FILL_DIAL`.

Additional Inherited Members

31.32.1 Detailed Description

Draws a dial with a filled arc.

The documentation for this class was generated from the following files:

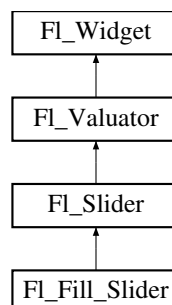
- `Fl_Fill_Dial.H`
- `Fl_Dial.cxx`

31.33 `Fl_Fill_Slider` Class Reference

Widget that draws a filled horizontal slider, useful as a progress or value meter.

```
#include <Fl_Fill_Slider.H>
```

Inheritance diagram for `Fl_Fill_Slider`:



Public Member Functions

- **`Fl_Fill_Slider`** (int X, int Y, int W, int H, const char *L=0)
Creates the slider from its position, size and optional title.

Additional Inherited Members

31.33.1 Detailed Description

Widget that draws a filled horizontal slider, useful as a progress or value meter. The documentation for this class was generated from the following files:

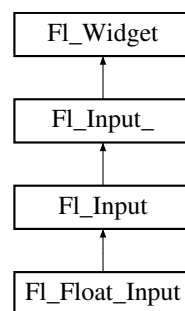
- FL_Fill_Slider.H
- FL_Slider.cxx

31.34 FL_Float_Input Class Reference

The [FL_Float_Input](#) class is a subclass of [FL_Input](#) that only allows the user to type floating point numbers (sign, digits, decimal point, more digits, 'E' or 'e', sign, digits).

```
#include <FL_Float_Input.H>
```

Inheritance diagram for FL_Float_Input:



Public Member Functions

- [FL_Float_Input](#) (int X, int Y, int W, int H, const char *l=0)
Creates a new [FL_Float_Input](#) widget using the given position, size, and label string.

Additional Inherited Members

31.34.1 Detailed Description

The [FL_Float_Input](#) class is a subclass of [FL_Input](#) that only allows the user to type floating point numbers (sign, digits, decimal point, more digits, 'E' or 'e', sign, digits).

31.34.2 Constructor & Destructor Documentation

31.34.2.1 FL_Float_Input()

```

FL_Float_Input::FL_Float_Input (
    int X,
    int Y,
    int W,
    int H,
    const char * l = 0 )
  
```

Creates a new [FL_Float_Input](#) widget using the given position, size, and label string. The default boxtype is FL_DOWN_BOX.

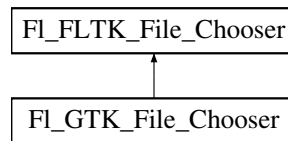
Inherited destructor destroys the widget and any value associated with it.

The documentation for this class was generated from the following files:

- FL_Float_Input.H
- FL_Input.cxx

31.35 FI_FLTK_File_Chooser Class Reference

Inheritance diagram for FI_FLTK_File_Chooser:



Protected Member Functions

- virtual int **count** () const
- const char * **directory** () const
- void **directory** (const char *val)
- const char * **errmsg** () const
- void **errmsg** (const char *msg)
- int **exist_dialog** ()
- virtual const char * **filename** () const
- virtual const char * **filename** (int i) const
- const char * **filter** () const
- void **filter** (const char *)
- int **filter_value** () const
- void **filter_value** (int i)
- int **filters** () const
- **FI_FLTK_File_Chooser** (int val)
- int **options** () const
- void **options** (int)
- void **parse_filter** ()
- const char * **preset_file** () const
- void **preset_file** (const char *)
- virtual int **show** ()
- virtual const char * **title** () const
- virtual void **title** (const char *)
- int **type** () const
- virtual void **type** (int)
- int **type_fl_file** (int)

Protected Attributes

- int **_btype**
- char * **_directory**
- char * **_errmsg**
- [FI_File_Chooser](#) * **_file_chooser**
- char * **_filter**
- int **_filtvalue**
- int **_nfilters**
- int **_options**
- char * **_parsedfilt**
- char * **_preset_file**
- char * **_prevvalue**

Friends

- class [FI_Native_File_Chooser](#)

The documentation for this class was generated from the following files:

- [FI_Native_File_Chooser.H](#)
- [FI_Native_File_Chooser_FLTK.cxx](#)

31.36 FI_Font_Descriptor Class Reference

This a structure for an actual system font, with junk to help choose it and info on character sizes.

```
#include <FI_Font.H>
```

Public Attributes

- [FI_Font_Descriptor](#) * **next**
linked list for this [FI_Fontdesc](#)
- [FI_Fontsize](#) **size**
font size

31.36.1 Detailed Description

This a structure for an actual system font, with junk to help choose it and info on character sizes.

Each [FI_Fontdesc](#) has a linked list of these. These are created the first time each system font/size combination is used.

The documentation for this class was generated from the following file:

- [FI_Font.H](#)

31.37 FI_Fontdesc Struct Reference

Public Attributes

- [FI_Font_Descriptor](#) * **first**
- char **fontname** [128]
- int **n**
- const char * **name**
- char ** **xlist**

The documentation for this struct was generated from the following file:

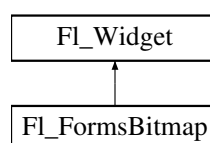
- [FI_Font.H](#)

31.38 FI_FormsBitmap Class Reference

Forms compatibility Bitmap Image Widget.

```
#include <FI_FormsBitmap.H>
```

Inheritance diagram for [FI_FormsBitmap](#):



Public Member Functions

- `Fl_Bitmap * bitmap () const`
Gets a the current associated [Fl_Bitmap](#) objects.
- `void bitmap (Fl_Bitmap *B)`
Sets a new bitmap.
- `Fl_FormsBitmap (Fl_Boxtype, int, int, int, int, const char *=0)`
Creates a bitmap widget from a box type, position, size and optional label specification.
- `void set (int W, int H, const uchar *bits)`
Sets a new bitmap bits with size W,H.

Protected Member Functions

- `void draw ()`
Draws the bitmap and its associated box.

Additional Inherited Members

31.38.1 Detailed Description

Forms compatibility Bitmap Image Widget.

31.38.2 Member Function Documentation

31.38.2.1 draw()

```
void Fl_FormsBitmap::draw (
    void ) [protected], [virtual]
```

Draws the bitmap and its associated box.

Implements [Fl_Widget](#).

31.38.2.2 set()

```
void Fl_FormsBitmap::set (
    int W,
    int H,
    const uchar * bits )
```

Sets a new bitmap bits with size W,H.

Deletes the previous one.

The documentation for this class was generated from the following files:

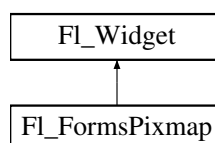
- `Fl_FormsBitmap.H`
- `forms_bitmap.cxx`

31.39 Fl_FormsPixmap Class Reference

Forms pixmap drawing routines.

```
#include <Fl_FormsPixmap.H>
```

Inheritance diagram for `Fl_FormsPixmap`:



Public Member Functions

- [Fl_FormsPixmap](#) ([Fl_Boxtype](#) t, int X, int Y, int W, int H, const char *L=0)
Creates a new [Fl_FormsPixmap](#) widget using the given box type, position, size and label string.
- [Fl_Pixmap](#) * [Pixmap](#) () const
Get the internal pixmap pointer.
- void [Pixmap](#) ([Fl_Pixmap](#) *B)
Set the internal pixmap pointer to an existing pixmap.
- void [set](#) (char *const *bits)
Set/create the internal pixmap using raw data.

Protected Member Functions

- void [draw](#) ()
Draws the widget.

Additional Inherited Members

31.39.1 Detailed Description

Forms pixmap drawing routines.

31.39.2 Constructor & Destructor Documentation

31.39.2.1 Fl_FormsPixmap()

```
Fl_FormsPixmap::Fl_FormsPixmap (
    Fl_Boxtype t,
    int X,
    int Y,
    int W,
    int H,
    const char * L = 0 )
```

Creates a new [Fl_FormsPixmap](#) widget using the given box type, position, size and label string.

Parameters

in	<i>t</i>	box type
in	<i>X,Y,W,H</i>	position and size
in	<i>L</i>	widget label, default is no label

31.39.3 Member Function Documentation

31.39.3.1 draw()

```
void Fl_FormsPixmap::draw ( ) [protected], [virtual]
```

Draws the widget.

Never call this function directly. FLTK will schedule redrawing whenever needed. If your widget must be redrawn as soon as possible, call [redraw\(\)](#) instead.

Override this function to draw your own widgets.

If you ever need to call another widget's draw method *from within your own [draw\(\)](#) method*, e.g. for an embedded scrollbar, you can do it (because [draw\(\)](#) is virtual) like this:

```
Fl_Widget *s = &scroll;           // scroll is an embedded Fl_Scrollbar
s->draw();                       // calls Fl_Scrollbar::draw()
Implements Fl_Widget.
```

31.39.3.2 Pixmap()

```
void Fl_FormsPixmap::Pixmap (
    Fl_Pixmap * B ) [inline]
```

Set the internal pixmap pointer to an existing pixmap.

Parameters

in	<i>B</i>	existing pixmap
----	----------	-----------------

31.39.3.3 set()

```
void Fl_FormsPixmap::set (
    char *const * bits )
```

Set/create the internal pixmap using raw data.

Parameters

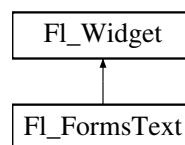
in	<i>bits</i>	raw data
----	-------------	----------

The documentation for this class was generated from the following files:

- Fl_FormsPixmap.H
- forms_pixmap.cxx

31.40 Fl_FormsText Class Reference

Inheritance diagram for Fl_FormsText:



Public Member Functions

- **Fl_FormsText** ([Fl_Boxtype](#) b, int X, int Y, int W, int H, const char *l=0)

Protected Member Functions

- void [draw](#) ()
Draws the widget.

Additional Inherited Members

31.40.1 Member Function Documentation

31.40.1.1 draw()

```
void Fl_FormsText::draw ( ) [protected], [virtual]
```

Draws the widget.

Never call this function directly. FLTK will schedule redrawing whenever needed. If your widget must be redrawn as soon as possible, call [redraw\(\)](#) instead.

Override this function to draw your own widgets.

If you ever need to call another widget's draw method *from within your own [draw\(\)](#) method*, e.g. for an embedded scrollbar, you can do it (because [draw\(\)](#) is virtual) like this:

```
Fl_Widget *s = &scroll;           // scroll is an embedded Fl_Scrollbar
s->draw();                         // calls Fl_Scrollbar::draw()
```

Implements [Fl_Widget](#).

The documentation for this class was generated from the following file:

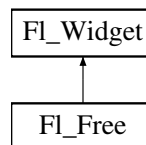
- forms.H

31.41 Fl_Free Class Reference

Emulation of the Forms "free" widget.

```
#include <Fl_Free.H>
```

Inheritance diagram for Fl_Free:



Public Member Functions

- [Fl_Free](#) ([uchar](#) t, int X, int Y, int W, int H, const char *L, [FL_HANDLEPTR](#) hdl)
Create a new [Fl_Free](#) widget with type, position, size, label and handler.
- int [handle](#) (int e)
Handles the specified event.
- [~Fl_Free](#) ()
The destructor will call the handle function with the event [FL_FREE_MEM](#).

Protected Member Functions

- void [draw](#) ()
Draws the widget.

Additional Inherited Members

31.41.1 Detailed Description

Emulation of the Forms "free" widget.

This emulation allows the free demo to run, and appears to be useful for porting programs written in Forms which use the free widget or make subclasses of the Forms widgets.

There are five types of free, which determine when the handle function is called:

- [FL_NORMAL_FREE](#) normal event handling.
- [FL_SLEEPING_FREE](#) deactivates event handling (widget is inactive).
- [FL_INPUT_FREE](#) accepts [FL_FOCUS](#) events.
- [FL_CONTINUOUS_FREE](#) sets a timeout callback 100 times a second and provides an [FL_STEP](#) event. This has obvious detrimental effects on machine performance.
- [FL_ALL_FREE](#) same as [FL_INPUT_FREE](#) and [FL_CONTINUOUS_FREE](#).

31.41.2 Constructor & Destructor Documentation

31.41.2.1 FL_Free()

```
Fl_Free::Fl_Free (
    uchar t,
    int X,
    int Y,
    int W,
    int H,
    const char * L,
    FL_HANDLEPTR hdl )
```

Create a new [FL_Free](#) widget with type, position, size, label and handler.

Parameters

in	<i>t</i>	type
in	<i>X,Y,W,H</i>	position and size
in	<i>L</i>	widget label
in	<i>hdl</i>	handler function

The constructor takes both the type and the handle function. The handle function should be declared as follows:

```
int handle_function(Fl_Widget *w,
    int event,
    float event_x,
    float event_y,
    char key)
```

This function is called from the [handle\(\)](#) method in response to most events, and is called by the [draw\(\)](#) method.

The event argument contains the event type:

```
// old event names for compatibility:
#define FL_MOUSE          FL_DRAG
#define FL_DRAW           0
#define FL_STEP           9
#define FL_FREEMEM        12
#define FL_FREEZE         FL_UNMAP
#define FL_THAW           FL_MAP
```

31.41.3 Member Function Documentation

31.41.3.1 draw()

```
void Fl_Free::draw ( ) [protected], [virtual]
```

Draws the widget.

Never call this function directly. FLTK will schedule redrawing whenever needed. If your widget must be redrawn as soon as possible, call [redraw\(\)](#) instead.

Override this function to draw your own widgets.

If you ever need to call another widget's draw method *from within your own [draw\(\)](#) method*, e.g. for an embedded scrollbar, you can do it (because [draw\(\)](#) is virtual) like this:

```
Fl_Widget *s = &scroll; // scroll is an embedded Fl_Scrollbar
s->draw(); // calls Fl_Scrollbar::draw()
```

Implements [Fl_Widget](#).

31.41.3.2 handle()

```
int Fl_Free::handle (
    int event ) [virtual]
```

Handles the specified event.

You normally don't call this method directly, but instead let FLTK do it when the user interacts with the widget.

When implemented in a widget, this function must return 0 if the widget does not use the event or 1 otherwise. Most of the time, you want to call the inherited [handle\(\)](#) method in your overridden method so that you don't short-circuit events that you don't handle. In this last case you should return the callee retval.

Parameters

in	<i>event</i>	the kind of event received
----	--------------	----------------------------

Return values

0	if the event was not used or understood
1	if the event was used and can be deleted

See also

[Fl_Event](#)

Reimplemented from [Fl_Widget](#).

The documentation for this class was generated from the following files:

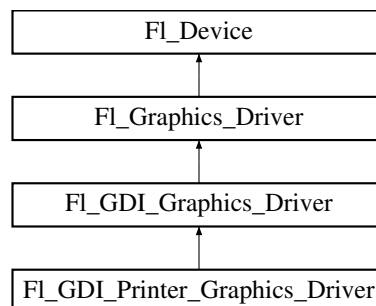
- Fl_Free.H
- forms_free.cxx

31.42 Fl_GDI_Graphics_Driver Class Reference

The MSWindows-specific graphics class.

```
#include <Fl_Device.H>
```

Inheritance diagram for Fl_GDI_Graphics_Driver:



Public Member Functions

- `const char * class_name ()`
Returns the name of the class of this object.
- `void color (Fl_Color c)`
see [fl_color\(Fl_Color c\)](#).
- `void color (uchar r, uchar g, uchar b)`
see [fl_color\(uchar r, uchar g, uchar b\)](#).
- `void copy_offscreen (int x, int y, int w, int h, Fl_Offscreen pixmap, int srcx, int srcy)`
see [fl_copy_offscreen\(\)](#)
- `int descent ()`
see [fl_descent\(\)](#).
- `void draw (const char *str, int n, int x, int y)`
*see [fl_draw\(const char *str, int n, int x, int y\)](#).*

- void `draw (FI_Bitmap *pxm, int XP, int YP, int WP, int HP, int cx, int cy)`
Draws an `FI_Bitmap` object to the device.
- void `draw (FI_Pixmap *pxm, int XP, int YP, int WP, int HP, int cx, int cy)`
Draws an `FI_Pixmap` object to the device.
- void `draw (FI_RGB_Image *img, int XP, int YP, int WP, int HP, int cx, int cy)`
Draws an `FI_RGB_Image` object to the device.
- void `draw (int angle, const char *str, int n, int x, int y)`
*see `fl_draw(int angle, const char *str, int n, int x, int y)`.*
- void `draw_image (const uchar *buf, int X, int Y, int W, int H, int D=3, int L=0)`
see `fl_draw_image(const uchar buf, int X,int Y,int W,int H, int D, int L)`.*
- void `draw_image (FI_Draw_Image_Cb cb, void *data, int X, int Y, int W, int H, int D=3)`
see `fl_draw_image(FI_Draw_Image_Cb cb, void data, int X,int Y,int W,int H, int D)`.*
- void `draw_image_mono (const uchar *buf, int X, int Y, int W, int H, int D=1, int L=0)`
see `fl_draw_image_mono(const uchar buf, int X,int Y,int W,int H, int D, int L)`.*
- void `draw_image_mono (FI_Draw_Image_Cb cb, void *data, int X, int Y, int W, int H, int D=1)`
see `fl_draw_image_mono(FI_Draw_Image_Cb cb, void data, int X,int Y,int W,int H, int D)`.*
- void `font (FI_Font face, FI_Fontsize size)`
see `fl_font(FI_Font face, FI_Fontsize size)`.
- int `height ()`
see `fl_height()`.
- void `rtl_draw (const char *str, int n, int x, int y)`
*see `fl_rtl_draw(const char *str, int n, int x, int y)`.*
- void `text_extents (const char *, int n, int &dx, int &dy, int &w, int &h)`
see `fl_text_extents(const char, int n, int& dx, int& dy, int& w, int& h)`.*
- double `width (const char *str, int n)`
*see `fl_width(const char *str, int n)`.*
- double `width (unsigned int c)`
see `fl_width(unsigned int n)`.

Static Public Attributes

- static const char * `class_id` = "FI_GDI_Graphics_Driver"

Additional Inherited Members

31.42.1 Detailed Description

The MSWindows-specific graphics class.

This class is implemented only on the MSWindows platform.

31.42.2 Member Function Documentation

31.42.2.1 `class_name()`

```
const char * Fl_GDI_Graphics_Driver::class_name ( ) [inline], [virtual]
```

Returns the name of the class of this object.

Use of the `class_name()` function is discouraged because it will be removed from future FLTK versions.

The class of an instance of an `FI_Device` subclass can be checked with code such as:

```
if ( instance->class_name() == Fl_Printer::class_id ) { ... }
```

Reimplemented from `FI_Graphics_Driver`.

Reimplemented in `FI_GDI_Printer_Graphics_Driver`.

31.42.2.2 color() [1/2]

```
void Fl_GDI_Graphics_Driver::color (
    Fl_Color c ) [virtual]
```

see [fl_color\(Fl_Color c\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.42.2.3 color() [2/2]

```
void Fl_GDI_Graphics_Driver::color (
    uchar r,
    uchar g,
    uchar b ) [virtual]
```

see [fl_color\(uchar r, uchar g, uchar b\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.42.2.4 copy_offscreen()

```
void Fl_GDI_Graphics_Driver::copy_offscreen (
    int x,
    int y,
    int w,
    int h,
    Fl_Offscreen pixmap,
    int srcx,
    int srcy ) [virtual]
```

see [fl_copy_offscreen\(\)](#)

Reimplemented from [Fl_Graphics_Driver](#).

31.42.2.5 descent()

```
int Fl_GDI_Graphics_Driver::descent ( ) [virtual]
```

see [fl_descent\(\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.42.2.6 draw() [1/5]

```
void Fl_GDI_Graphics_Driver::draw (
    const char * str,
    int n,
    int x,
    int y ) [virtual]
```

see [fl_draw\(const char *str, int n, int x, int y\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.42.2.7 draw() [2/5]

```
void Fl_GDI_Graphics_Driver::draw (
    Fl_Bitmap * bm,
    int XP,
    int YP,
    int WP,
    int HP,
    int cx,
    int cy ) [virtual]
```

Draws an [Fl_Bitmap](#) object to the device.

Specifies a bounding box for the image, with the origin (upper left-hand corner) of the image offset by the *cx* and *cy* arguments.

Reimplemented from [Fl_Graphics_Driver](#).

Reimplemented in [Fl_GDI_Printer_Graphics_Driver](#).

31.42.2.8 draw() [3/5]

```
void Fl_GDI_Graphics_Driver::draw (
    Fl_Pixmap * pxm,
    int XP,
    int YP,
    int WP,
    int HP,
    int cx,
    int cy ) [virtual]
```

Draws an [Fl_Pixmap](#) object to the device.

Specifies a bounding box for the image, with the origin (upper left-hand corner) of the image offset by the *cx* and *cy* arguments.

Reimplemented from [Fl_Graphics_Driver](#).

Reimplemented in [Fl_GDI_Printer_Graphics_Driver](#).

31.42.2.9 draw() [4/5]

```
void Fl_GDI_Graphics_Driver::draw (
    Fl_RGB_Image * rgb,
    int XP,
    int YP,
    int WP,
    int HP,
    int cx,
    int cy ) [virtual]
```

Draws an [Fl_RGB_Image](#) object to the device.

Specifies a bounding box for the image, with the origin (upper left-hand corner) of the image offset by the *cx* and *cy* arguments.

Reimplemented from [Fl_Graphics_Driver](#).

31.42.2.10 draw() [5/5]

```
void Fl_GDI_Graphics_Driver::draw (
    int angle,
    const char * str,
    int n,
    int x,
    int y ) [virtual]
```

see [fl_draw\(int angle, const char *str, int n, int x, int y\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.42.2.11 draw_image() [1/2]

```
void Fl_GDI_Graphics_Driver::draw_image (
    const uchar * buf,
    int X,
    int Y,
    int W,
```



```
    int H,  
    int D = 3,  
    int L = 0 ) [virtual]
```

see [fl_draw_image\(const uchar* buf, int X,int Y,int W,int H, int D, int L\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.42.2.12 draw_image() [2/2]

```
void Fl_GDI_Graphics_Driver::draw_image (  
    Fl_Draw_Image_Cb cb,  
    void * data,  
    int X,  
    int Y,  
    int W,  
    int H,  
    int D = 3 ) [virtual]
```

see [fl_draw_image\(Fl_Draw_Image_Cb cb, void* data, int X,int Y,int W,int H, int D\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.42.2.13 draw_image_mono() [1/2]

```
void Fl_GDI_Graphics_Driver::draw_image_mono (  
    const uchar * buf,  
    int X,  
    int Y,  
    int W,  
    int H,  
    int D = 1,  
    int L = 0 ) [virtual]
```

see [fl_draw_image_mono\(const uchar* buf, int X,int Y,int W,int H, int D, int L\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.42.2.14 draw_image_mono() [2/2]

```
void Fl_GDI_Graphics_Driver::draw_image_mono (  
    Fl_Draw_Image_Cb cb,  
    void * data,  
    int X,  
    int Y,  
    int W,  
    int H,  
    int D = 1 ) [virtual]
```

see [fl_draw_image_mono\(Fl_Draw_Image_Cb cb, void* data, int X,int Y,int W,int H, int D\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.42.2.15 font()

```
void Fl_GDI_Graphics_Driver::font (  
    Fl_Font face,  
    Fl_Fontsize fsize ) [virtual]
```

see [fl_font\(Fl_Font face, Fl_Fontsize size\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.42.2.16 height()

```
int Fl_GDI_Graphics_Driver::height ( ) [virtual]
```

see [fl_height\(\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.42.2.17 rtl_draw()

```
void Fl_GDI_Graphics_Driver::rtl_draw (
    const char * str,
    int n,
    int x,
    int y ) [virtual]
```

see [fl_rtl_draw\(const char *str, int n, int x, int y\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.42.2.18 text_extents()

```
void Fl_GDI_Graphics_Driver::text_extents (
    const char * t,
    int n,
    int & dx,
    int & dy,
    int & w,
    int & h ) [virtual]
```

see [fl_text_extents\(const char*, int n, int& dx, int& dy, int& w, int& h\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.42.2.19 width() [1/2]

```
double Fl_GDI_Graphics_Driver::width (
    const char * str,
    int n ) [virtual]
```

see [fl_width\(const char *str, int n\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.42.2.20 width() [2/2]

```
double Fl_GDI_Graphics_Driver::width (
    unsigned int c ) [virtual]
```

see [fl_width\(unsigned int n\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

The documentation for this class was generated from the following files:

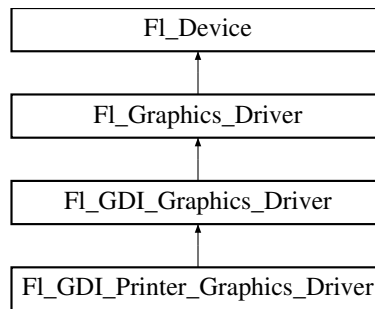
- [Fl_Device.H](#)
- [fl_color_win32.cxx](#)
- [Fl_Device.cxx](#)
- [fl_draw_image_win32.cxx](#)

31.43 Fl_GDI_Printer_Graphics_Driver Class Reference

The graphics driver used when printing on MSWindows.

```
#include <Fl_Device.H>
```

Inheritance diagram for [Fl_GDI_Printer_Graphics_Driver](#):



Public Member Functions

- const char * [class_name](#) ()
Returns the name of the class of this object.
- void [draw](#) ([Fl_Bitmap](#) *bm, int XP, int YP, int WP, int HP, int cx, int cy)
Draws an [Fl_Bitmap](#) object to the device.
- void [draw](#) ([Fl_Pixmap](#) *pixmap, int XP, int YP, int WP, int HP, int cx, int cy)
Draws an [Fl_Pixmap](#) object to the device.
- int [draw_scaled](#) ([Fl_Image](#) *img, int XP, int YP, int WP, int HP)
Draws an [Fl_Image](#) scaled to width W & height H with top-left corner at X,Y .

Static Public Attributes

- static const char * [class_id](#) = "Fl_GDI_Printer_Graphics_Driver"

Additional Inherited Members

31.43.1 Detailed Description

The graphics driver used when printing on MSWindows.

This class is implemented only on the MSWindows platform. It's extremely similar to [Fl_GDI_Graphics_Driver](#).

31.43.2 Member Function Documentation

31.43.2.1 [class_name\(\)](#)

```
const char * Fl_GDI_Printer_Graphics_Driver::class_name ( ) [inline], [virtual]
```

Returns the name of the class of this object.

Use of the [class_name\(\)](#) function is discouraged because it will be removed from future FLTK versions.

The class of an instance of an [Fl_Device](#) subclass can be checked with code such as:

```
if ( instance->class_name() == Fl_Printer::class_id ) { ... }
```

Reimplemented from [Fl_GDI_Graphics_Driver](#).

31.43.2.2 [draw\(\)](#) [1/2]

```
void Fl_GDI_Printer_Graphics_Driver::draw (
    Fl_Bitmap * bm,
    int XP,
    int YP,
    int WP,
    int HP,
    int cx,
    int cy ) [virtual]
```

Draws an [Fl_Bitmap](#) object to the device.

Specifies a bounding box for the image, with the origin (upper left-hand corner) of the image offset by the *cx* and *cy* arguments.

Reimplemented from [Fl_GDI_Graphics_Driver](#).

31.43.2.3 draw() [2/2]

```
void Fl_GDI_Printer_Graphics_Driver::draw (
    Fl_Pixmap * pxm,
    int XP,
    int YP,
    int WP,
    int HP,
    int cx,
    int cy ) [virtual]
```

Draws an [Fl_Pixmap](#) object to the device.

Specifies a bounding box for the image, with the origin (upper left-hand corner) of the image offset by the *cx* and *cy* arguments.

Reimplemented from [Fl_GDI_Graphics_Driver](#).

31.43.2.4 draw_scaled()

```
int Fl_GDI_Printer_Graphics_Driver::draw_scaled (
    Fl_Image * img,
    int X,
    int Y,
    int W,
    int H ) [virtual]
```

Draws an [Fl_Image](#) scaled to width *W* & height *H* with top-left corner at *X,Y*.

Returns

zero when the graphics driver doesn't implement scaled drawing, non-zero if it does implement it.

Reimplemented from [Fl_Graphics_Driver](#).

The documentation for this class was generated from the following files:

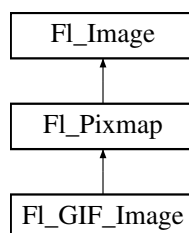
- [Fl_Device.H](#)
- [Fl_Device.cxx](#)

31.44 Fl_GIF_Image Class Reference

The [Fl_GIF_Image](#) class supports loading, caching, and drawing of CompuServe GIFSM images.

```
#include <Fl_GIF_Image.H>
```

Inheritance diagram for [Fl_GIF_Image](#):



Public Member Functions

- [Fl_GIF_Image](#) (const char *filename)

The constructor loads the named GIF image.

Additional Inherited Members

31.44.1 Detailed Description

The [FI_GIF_Image](#) class supports loading, caching, and drawing of Compuserve GIFSM images. The class loads the first image and supports transparency.

31.44.2 Constructor & Destructor Documentation

31.44.2.1 FI_GIF_Image()

```
FI_GIF_Image::FI_GIF_Image (
    const char * infname )
```

The constructor loads the named GIF image.

The destructor frees all memory and server resources that are used by the image.

Use [FI_Image::fail\(\)](#) to check if [FI_GIF_Image](#) failed to load. [fail\(\)](#) returns ERR_FILE_ACCESS if the file could not be opened or read, ERR_FORMAT if the GIF format could not be decoded, and ERR_NO_IMAGE if the image could not be loaded for another reason.

The documentation for this class was generated from the following files:

- FI_GIF_Image.H
- FI_GIF_Image.cxx

31.45 FI_GI_Choice Class Reference

Static Public Member Functions

- static [FI_GI_Choice](#) * [find](#) (int mode, const int *)

Public Attributes

- GLXFBConfig [best_fb](#)
- Colormap [colormap](#)
- XVisualInfo * [vis](#)

The documentation for this class was generated from the following files:

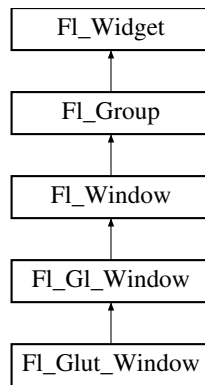
- FI_GI_Choice.H
- FI_GI_Choice.cxx

31.46 FI_GI_Window Class Reference

The [FI_GI_Window](#) widget sets things up so OpenGL works.

```
#include <FI_GI_Window.H>
```

Inheritance diagram for [FI_GI_Window](#):



Public Member Functions

- virtual `Fl_Gl_Window * as_gl_window ()`
Returns an `Fl_Gl_Window` pointer if this widget is an `Fl_Gl_Window`.
- int `can_do ()`
Returns non-zero if the hardware supports the current OpenGL mode.
- int `can_do_overlay ()`
Returns true if the hardware overlay is possible.
- void * `context () const`
Returns a pointer to the `GLContext` that this window is using.
- void `context (void *, int destroy_flag=0)`
Sets a pointer to the `GLContext` that this window is using.
- char `context_valid () const`
Will only be set if the OpenGL context is created or recreated.
- void `context_valid (char v)`
See char `Fl_Gl_Window::context_valid() const`.
- `Fl_Gl_Window (int W, int H, const char *l=0)`
Creates a new `Fl_Gl_Window` widget using the given size, and label string.
- `Fl_Gl_Window (int X, int Y, int W, int H, const char *l=0)`
Creates a new `Fl_Gl_Window` widget using the given position, size, and label string.
- void `flush ()`
Forces the window to be drawn, this window is also made current and calls `draw()`.
- int `handle (int)`
Handle some FLTK events as needed.
- void `hide ()`
Hides the window and destroys the OpenGL context.
- void `hide_overlay ()`
Hides the window if it is not this window, does nothing in WIN32.
- void `invalidate ()`
The `invalidate()` method turns off `valid()` and is equivalent to calling `value(0)`.
- void `make_current ()`
The `make_current()` method selects the OpenGL context for the widget.
- void `make_overlay_current ()`
The `make_overlay_current()` method selects the OpenGL context for the widget's overlay.
- `Fl_Mode mode () const`
Returns the current OpenGL capabilities of the window.
- int `mode (const int *a)`
Set the OpenGL capabilities of the window using platform-specific data.
- int `mode (int a)`

- Set or change the OpenGL capabilities of the window.*

 - void [ortho](#) ()
 - Sets the projection so 0,0 is in the lower left of the window and each pixel is 1 unit wide/tall.*
 - int [pixel_h](#) ()
 - Gives the window height in OpenGL pixels.*
 - int [pixel_w](#) ()
 - Gives the window width in OpenGL pixels.*
 - float [pixels_per_unit](#) ()
 - The number of pixels per FLTK unit of length for the window.*
 - void [redraw_overlay](#) ()
 - This method causes [draw_overlay\(\)](#) to be called at a later time.*
 - void [resize](#) (int, int, int, int)
 - Changes the size and position of the window.*
 - void [show](#) ()
 - Puts the window on the screen.*
 - void [show](#) (int a, char **b)
 - void [swap_buffers](#) ()
 - The [swap_buffers\(\)](#) method swaps the back and front buffers.*
 - char [valid](#) () const
 - Is turned off when FLTK creates a new context for this window or when the window resizes, and is turned on after [draw\(\)](#) is called.*
 - void [valid](#) (char v)
 - See char [FI_GI_Window::valid\(\)](#) const.*
 - [~FI_GI_Window](#) ()
 - The destructor removes the widget and destroys the OpenGL context associated with it.*

Static Public Member Functions

- static int [can_do](#) (const int *m)
 - Returns non-zero if the hardware supports the given OpenGL mode.*
- static int [can_do](#) (int m)
 - Returns non-zero if the hardware supports the given OpenGL mode.*

Protected Member Functions

- virtual void [draw](#) ()
 - Draws the [FI_GI_Window](#).*

Friends

- class [_FI_GI_Overlay](#)

Additional Inherited Members

31.46.1 Detailed Description

The [FI_GI_Window](#) widget sets things up so OpenGL works.

It also keeps an OpenGL "context" for that window, so that changes to the lighting and projection may be reused between redraws. [FI_GI_Window](#) also flushes the OpenGL streams and swaps buffers after [draw\(\)](#) returns.

OpenGL hardware typically provides some overlay bit planes, which are very useful for drawing UI controls atop your 3D graphics. If the overlay hardware is not provided, FLTK tries to simulate the overlay. This works pretty well if your graphics are double buffered, but not very well for single-buffered.

Please note that the FLTK drawing and clipping functions will not work inside an [FI_GI_Window](#). All drawing should be done using OpenGL calls exclusively. Even though [FI_GI_Window](#) is derived from [FI_Group](#), it is not useful to add other FLTK Widgets as children, unless those widgets are modified to draw using OpenGL calls.

31.46.2 Constructor & Destructor Documentation

31.46.2.1 Fl_Gl_Window() [1/2]

```
Fl_Gl_Window::Fl_Gl_Window (
    int W,
    int H,
    const char * l = 0 ) [inline]
```

Creates a new [Fl_Gl_Window](#) widget using the given size, and label string.
The default boxtype is FL_NO_BOX. The default mode is FL_RGB|FL_DOUBLE|FL_DEPTH.

31.46.2.2 Fl_Gl_Window() [2/2]

```
Fl_Gl_Window::Fl_Gl_Window (
    int X,
    int Y,
    int W,
    int H,
    const char * l = 0 ) [inline]
```

Creates a new [Fl_Gl_Window](#) widget using the given position, size, and label string.
The default boxtype is FL_NO_BOX. The default mode is FL_RGB|FL_DOUBLE|FL_DEPTH.

31.46.3 Member Function Documentation

31.46.3.1 as_gl_window()

```
virtual Fl\_Gl\_Window * Fl_Gl_Window::as_gl_window ( ) [inline], [virtual]
```

Returns an [Fl_Gl_Window](#) pointer if this widget is an [Fl_Gl_Window](#).
Use this method if you have a widget (pointer) and need to know whether this widget is derived from [Fl_Gl_Window](#).
If it returns non-NULL, then the widget in question is derived from [Fl_Gl_Window](#).

Return values

NULL	if this widget is not derived from Fl_Gl_Window .
------	---

Note

This method is provided to avoid `dynamic_cast`.

See also

[Fl_Widget::as_group\(\)](#), [Fl_Widget::as_window\(\)](#)

Reimplemented from [Fl_Widget](#).

31.46.3.2 can_do()

```
static int Fl_Gl_Window::can_do (
    const int * m ) [inline], [static]
```

Returns non-zero if the hardware supports the given OpenGL mode.

See also

[Fl_Gl_Window::mode\(const int *a\)](#)

31.46.3.3 can_do_overlay()

```
int Fl_Gl_Window::can_do_overlay ( )
```

Returns true if the hardware overlay is possible.

If this is false, FLTK will try to simulate the overlay, with significant loss of update speed. Calling this will cause FLTK to open the display.

31.46.3.4 context() [1/2]

```
void * Fl_Gl_Window::context ( ) const [inline]
```

Returns a pointer to the GLContext that this window is using.

See also

```
void context(void* v, int destroy_flag)
```

31.46.3.5 context() [2/2]

```
void Fl_Gl_Window::context (
    void * v,
    int destroy_flag = 0 )
```

Sets a pointer to the GLContext that this window is using.

This is a system-dependent structure, but it is portable to copy the context from one window to another. You can also set it to NULL, which will force FLTK to recreate the context the next time [make_current\(\)](#) is called, this is useful for getting around bugs in OpenGL implementations.

If *destroy_flag* is true the context will be destroyed by fltk when the window is destroyed, or when the [mode\(\)](#) is changed, or the next time [context\(x\)](#) is called.

31.46.3.6 context_valid()

```
char Fl_Gl_Window::context_valid ( ) const [inline]
```

Will only be set if the OpenGL context is created or recreated.

It differs from [Fl_Gl_Window::valid\(\)](#) which is also set whenever the context changes size.

31.46.3.7 draw()

```
void Fl_Gl_Window::draw (
    void ) [protected], [virtual]
```

Draws the [Fl_Gl_Window](#).

You **must subclass** [Fl_Gl_Window](#) and provide an implementation for [draw\(\)](#).

You **must override** the [draw\(\)](#) method.

You may also provide an implementation of [draw_overlay\(\)](#) if you want to draw into the overlay planes. You can avoid reinitializing the viewport and lights and other things by checking [valid\(\)](#) at the start of [draw\(\)](#) and only doing the initialization if it is false.

The [draw\(\)](#) method can *only* use OpenGL calls. Do not attempt to call X, any of the functions in `<FL/fl_draw.H>`, or GLX directly. Do not call [gl_start\(\)](#) or [gl_finish\(\)](#).

If double-buffering is enabled in the window, the back and front buffers are swapped after this function is completed.

Reimplemented from [Fl_Window](#).

Reimplemented in [Fl_Glut_Window](#).

31.46.3.8 flush()

```
void Fl_Gl_Window::flush ( ) [virtual]
```

Forces the window to be drawn, this window is also made current and calls [draw\(\)](#).

Reimplemented from [Fl_Window](#).

31.46.3.9 handle()

```
int Fl_Gl_Window::handle (
    int event ) [virtual]
```

Handle some FLTK events as needed.

Reimplemented from [Fl_Window](#).

Reimplemented in [Fl_Glut_Window](#).

31.46.3.10 hide()

```
void Fl_Gl_Window::hide ( ) [virtual]
```

Hides the window and destroys the OpenGL context.

Reimplemented from [Fl_Window](#).

31.46.3.11 make_current()

```
void Fl_Gl_Window::make_current ( )
```

The [make_current\(\)](#) method selects the OpenGL context for the widget.

It is called automatically prior to the [draw\(\)](#) method being called and can also be used to implement feedback and/or selection within the [handle\(\)](#) method.

31.46.3.12 make_overlay_current()

```
void Fl_Gl_Window::make_overlay_current ( )
```

The [make_overlay_current\(\)](#) method selects the OpenGL context for the widget's overlay.

It is called automatically prior to the [draw_overlay\(\)](#) method being called and can also be used to implement feedback and/or selection within the [handle\(\)](#) method.

31.46.3.13 mode() [1/3]

```
Fl_Mode Fl_Gl_Window::mode ( ) const [inline]
```

Returns the current OpenGL capabilities of the window.

Don't use this if capabilities were set through [Fl_Gl_Window::mode\(const int *a\)](#).

31.46.3.14 mode() [2/3]

```
int Fl_Gl_Window::mode (
    const int * a ) [inline]
```

Set the OpenGL capabilities of the window using platform-specific data.

Parameters

<i>a</i>	zero-ending array of platform-specific attributes and attribute values
----------	--

Unix/Linux platform: attributes are GLX attributes adequate for the 3rd argument of the `glXChooseVisual()` function (e.g., `GLX_DOUBLEBUFFER`, defined by including `<GL/glx.h>`).

Note

What attributes are adequate here is subject to change. The preferred, stable public API is [Fl_Gl_Window::mode\(int a\)](#).

MSWindows platform: this member function is of no use.

Mac OS X platform: attributes belong to the `CGLPixelFormatAttribute` enumeration (defined by including `<OpenGL/OpenGL.h>`, e.g., `kCGLPFADoubleBuffer`) and may be followed by adequate attribute values.

31.46.3.15 mode() [3/3]

```
int Fl_Gl_Window::mode (
```

```
int a ) [inline]
```

Set or change the OpenGL capabilities of the window.
The value can be any of the following OR'd together:

- FL_RGB - RGB color (not indexed)
- FL_RGB8 - RGB color with at least 8 bits of each color
- FL_INDEX - Indexed mode
- FL_SINGLE - not double buffered
- FL_DOUBLE - double buffered
- FL_ACCUM - accumulation buffer
- FL_ALPHA - alpha channel in color
- FL_DEPTH - depth buffer
- FL_STENCIL - stencil buffer
- FL_MULTISAMPLE - multisample antialiasing
- FL_OPENGL3 - use OpenGL version 3.0 or more.

FL_RGB and FL_SINGLE have a value of zero, so they are "on" unless you give FL_INDEX or FL_DOUBLE. If the desired combination cannot be done, FLTK will try turning off FL_MULTISAMPLE. If this also fails the `show()` will call `Fl::error()` and not show the window.

You can change the mode while the window is displayed. This is most useful for turning double-buffering on and off. Under X this will cause the old X window to be destroyed and a new one to be created. If this is a top-level window this will unfortunately also cause the window to blink, raise to the top, and be de-iconized, and the `xid()` will change, possibly breaking other code. It is best to make the GL window a child of another window if you wish to do this! `mode()` must not be called within `draw()` since it changes the current context.

The FL_OPENGL3 flag is required to access OpenGL version 3 or more under the X11 and MacOS platforms; it's optional under Windows. See more details in [Using OpenGL 3.0 \(or higher versions\)](#).

Version

the FL_OPENGL3 flag appeared in version 1.3.4

31.46.3.16 ortho()

```
void Fl_Gl_Window::ortho ( )
```

Sets the projection so 0,0 is in the lower left of the window and each pixel is 1 unit wide/tall. If you are drawing 2D images, your `draw()` method may want to call this if `valid()` is false.

31.46.3.17 pixel_h()

```
int Fl_Gl_Window::pixel_h ( ) [inline]
```

Gives the window height in OpenGL pixels.

Generally identical with the result of the `h()` function, but for a window mapped to an Apple 'retina' display, and if `Fl::use_high_res_GL(bool)` is set to true, `pixel_h()` returns $2 * h()$. This method detects when the window has been moved between low and high resolution displays and automatically adjusts the returned value.

Version

1.3.4

31.46.3.18 pixel_w()

```
int Fl_Gl_Window::pixel_w ( ) [inline]
```

Gives the window width in OpenGL pixels.

Generally identical with the result of the [w\(\)](#) function, but for a window mapped to an Apple 'retina' display, and if `Fl::use_high_res_GL(bool)` is set to true, [pixel_w\(\)](#) returns $2 * w()$. This method detects when the window has been moved between low and high resolution displays and automatically adjusts the returned value.

Version

1.3.4

31.46.3.19 pixels_per_unit()

```
float Fl_Gl_Window::pixels_per_unit ( ) [inline]
```

The number of pixels per FLTK unit of length for the window.

Returns 1, except for a window mapped to an Apple 'retina' display, and if `Fl::use_high_res_GL(bool)` is set to true, when it returns 2. This method dynamically adjusts its value when the window is moved to/from a retina display. This method is useful, e.g., to convert, in a window's [handle\(\)](#) method, the FLTK units returned by [Fl::event_x\(\)](#) and [Fl::event_y\(\)](#) to the pixel units used by the OpenGL source code.

Version

1.3.4

31.46.3.20 redraw_overlay()

```
void Fl_Gl_Window::redraw_overlay ( )
```

This method causes [draw_overlay\(\)](#) to be called at a later time.

Initially the overlay is clear. If you want the window to display something in the overlay when it first appears, you must call this immediately after you [show\(\)](#) your window.

31.46.3.21 resize()

```
void Fl_Gl_Window::resize (
    int X,
    int Y,
    int W,
    int H ) [virtual]
```

Changes the size and position of the window.

If [shown\(\)](#) is true, these changes are communicated to the window server (which may refuse that size and cause a further resize). If [shown\(\)](#) is false, the size and position are used when [show\(\)](#) is called. See [Fl_Group](#) for the effect of resizing on the child widgets.

You can also call the [Fl_Widget](#) methods [size\(x,y\)](#) and [position\(w,h\)](#), which are inline wrappers for this virtual function.

A top-level window can not force, but merely suggest a position and size to the operating system. The window manager may not be willing or able to display a window at the desired position or with the given dimensions. It is up to the application developer to verify window parameters after the resize request.

Reimplemented from [Fl_Window](#).

31.46.3.22 show()

```
void Fl_Gl_Window::show ( ) [virtual]
```

Puts the window on the screen.

Usually (on X) this has the side effect of opening the display.

If the window is already shown then it is restored and raised to the top. This is really convenient because your program can call [show\(\)](#) at any time, even if the window is already up. It also means that [show\(\)](#) serves the purpose of [raise\(\)](#) in other toolkits.

[Fl_Window::show\(int argc, char **argv\)](#) is used for top-level windows and allows standard arguments to be parsed from the command-line.

Note

For some obscure reasons [Fl_Window::show\(\)](#) resets the current group by calling [Fl_Group::current\(0\)](#). The comments in the code say "get rid of very common user bug: forgot end()". Although this is true it may have unwanted side effects if you [show\(\)](#) an unrelated window (maybe for an error message or warning) while building a window or any other group widget.

Todo Check if we can remove resetting the current group in a later FLTK version (after 1.3.x). This may break "already broken" programs though if they rely on this "feature".

See also

[Fl_Window::show\(int argc, char **argv\)](#)

Reimplemented from [Fl_Window](#).

31.46.3.23 swap_buffers()

```
void Fl_Gl_Window::swap_buffers ( )
```

The [swap_buffers\(\)](#) method swaps the back and front buffers.

It is called automatically after the [draw\(\)](#) method is called.

31.46.3.24 valid()

```
char Fl_Gl_Window::valid ( ) const [inline]
```

Is turned off when FLTK creates a new context for this window or when the window resizes, and is turned on *after* [draw\(\)](#) is called.

You can use this inside your [draw\(\)](#) method to avoid unnecessarily initializing the OpenGL context. Just do this:

```
void mywindow::draw() {
    if (!valid()) {
        glViewport(0,0,pixel_w(),pixel_h());
        glFrustum(...);
        ...other initialization...
    }
    if (!context_valid()) {
        ...load textures, etc. ...
    }
    ... draw your geometry here ...
}
```

You can turn [valid\(\)](#) on by calling [valid\(1\)](#). You should only do this after fixing the transformation inside a [draw\(\)](#) or after [make_current\(\)](#). This is done automatically after [draw\(\)](#) returns.

The documentation for this class was generated from the following files:

- [Fl_Gl_Window.H](#)
- [Fl_Gl_Overlay.cxx](#)
- [Fl_Gl_Window.cxx](#)

31.47 Fl_Glut_Bitmap_Font Struct Reference

fltk glut font/size attributes used in the glutXXX functions

```
#include <glut.H>
```

Public Attributes

- [Fl_Font font](#)
- [Fl_Fontsize size](#)

31.47.1 Detailed Description

ftlk glut font/size attributes used in the glutXXX functions

The documentation for this struct was generated from the following file:

- glut.H

31.48 FI_Glut_StrokeChar Struct Reference

Public Attributes

- int **Number**
- GLfloat **Right**
- const [FI_Glut_StrokeStrip](#) * **Strips**

The documentation for this struct was generated from the following file:

- glut.H

31.49 FI_Glut_StrokeFont Struct Reference

Public Attributes

- const [FI_Glut_StrokeChar](#) ** **Characters**
- GLfloat **Height**
- char * **Name**
- int **Quantity**

The documentation for this struct was generated from the following file:

- glut.H

31.50 FI_Glut_StrokeStrip Struct Reference

Public Attributes

- int **Number**
- const [FI_Glut_StrokeVertex](#) * **Vertices**

The documentation for this struct was generated from the following file:

- glut.H

31.51 FI_Glut_StrokeVertex Struct Reference

Public Attributes

- GLfloat **X**
- GLfloat **Y**

The documentation for this struct was generated from the following file:

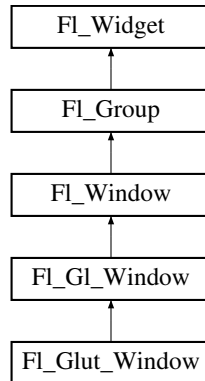
- glut.H

31.52 Fl_Glut_Window Class Reference

GLUT is emulated using this window class and these static variables (plus several more static variables hidden in `glut_compatibility.cxx`):

```
#include <glut.H>
```

Inheritance diagram for Fl_Glut_Window:



Public Member Functions

- **Fl_Glut_Window** (int *w*, int *h*, const char *)
Creates a glut window, registers to the glut windows list.
- **Fl_Glut_Window** (int *x*, int *y*, int *w*, int *h*, const char *)
Creates a glut window, registers to the glut windows list.
- void **make_current** ()
- ~**Fl_Glut_Window** ()
Destroys the glut window, first unregister it from the glut windows list.

Public Attributes

- void(* **display**)()
- void(* **entry**)(int)
- void(* **keyboard**)(uchar, int *x*, int *y*)
- int **menu** [3]
- void(* **motion**)(int *x*, int *y*)
- void(* **mouse**)(int *b*, int *state*, int *x*, int *y*)
- int **number**
- void(* **overlaydisplay**)()
- void(* **passivemotion**)(int *x*, int *y*)
- void(* **reshape**)(int *w*, int *h*)
- void(* **special**)(int, int *x*, int *y*)
- void(* **visibility**)(int)

Protected Member Functions

- void **draw** ()
Draws the Fl_Gl_Window.
- void **draw_overlay** ()
You must implement this virtual function if you want to draw into the overlay.
- int **handle** (int)
Handle some FLTK events as needed.

Additional Inherited Members

31.52.1 Detailed Description

GLUT is emulated using this window class and these static variables (plus several more static variables hidden in `glut_compatibility.cxx`):

31.52.2 Member Function Documentation

31.52.2.1 `draw()`

```
void Fl_Glut_Window::draw (
    void ) [protected], [virtual]
```

Draws the [Fl_Gl_Window](#).

You **must subclass** [Fl_Gl_Window](#) and provide an implementation for `draw()`.

You **must override** the `draw()` method.

You may also provide an implementation of `draw_overlay()` if you want to draw into the overlay planes. You can avoid reinitializing the viewport and lights and other things by checking `valid()` at the start of `draw()` and only doing the initialization if it is false.

The `draw()` method can *only* use OpenGL calls. Do not attempt to call X, any of the functions in `<FL/fl_draw.H>`, or glX directly. Do not call `gl_start()` or `gl_finish()`.

If double-buffering is enabled in the window, the back and front buffers are swapped after this function is completed. Reimplemented from [Fl_Gl_Window](#).

31.52.2.2 `draw_overlay()`

```
void Fl_Glut_Window::draw_overlay ( ) [protected], [virtual]
```

You must implement this virtual function if you want to draw into the overlay.

The overlay is cleared before this is called. You should draw anything that is not clear using OpenGL. You must use `gl_color(i)` to choose colors (it allocates them from the colormap using system-specific calls), and remember that you are in an indexed OpenGL mode and drawing anything other than flat-shaded will probably not work.

Both this function and [Fl_Gl_Window::draw\(\)](#) should check [Fl_Gl_Window::valid\(\)](#) and set the same transformation.

If you don't your code may not work on other systems. Depending on the OS, and on whether overlays are real or simulated, the OpenGL context may be the same or different between the overlay and main window.

Reimplemented from [Fl_Gl_Window](#).

31.52.2.3 `handle()`

```
int Fl_Glut_Window::handle (
    int event ) [protected], [virtual]
```

Handle some FLTK events as needed.

Reimplemented from [Fl_Gl_Window](#).

The documentation for this class was generated from the following files:

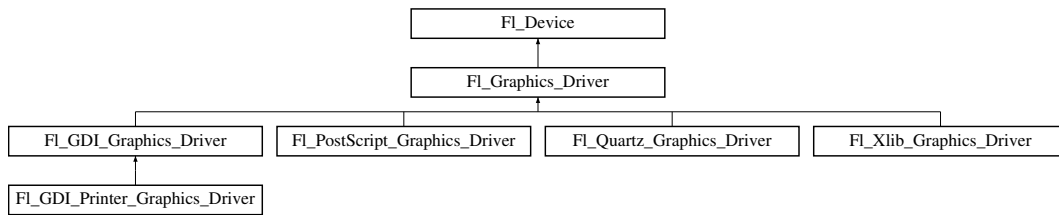
- `glut.H`
- `glut_compatibility.cxx`

31.53 [Fl_Graphics_Driver](#) Class Reference

A virtual class subclassed for each graphics driver FLTK uses.

```
#include <Fl_Device.H>
```

Inheritance diagram for [Fl_Graphics_Driver](#):



Classes

- struct [matrix](#)
A 2D coordinate transformation matrix.

Public Member Functions

- virtual const char * [class_name](#) ()
Returns the name of the class of this object.
- [FI_Color](#) [color](#) ()
see [fl_color\(void\)](#).
- virtual int [descent](#) ()
see [fl_descent\(\)](#).
- virtual int [draw_scaled](#) ([FI_Image](#) *img, int X, int Y, int W, int H)
Draws an [FI_Image](#) scaled to width *W* & height *H* with top-left corner at *X*, *Y*.
- [FI_Font](#) [font](#) ()
see [fl_font\(void\)](#).
- virtual void [font](#) ([FI_Font](#) face, [FI_Fontsize](#) fsize)
see [fl_font\(FI_Font face, FI_Fontsize size\)](#).
- [FI_Font_Descriptor](#) * [font_descriptor](#) ()
Returns a pointer to the current [FI_Font_Descriptor](#) for the graphics driver.
- void [font_descriptor](#) ([FI_Font_Descriptor](#) *d)
Sets the current [FI_Font_Descriptor](#) for the graphics driver.
- virtual int [height](#) ()
see [fl_height\(\)](#).
- [FI_Fontsize](#) [size](#) ()
see [fl_size\(\)](#).
- virtual void [text_extents](#) (const char *, int n, int &dx, int &dy, int &w, int &h)
see [fl_text_extents\(const char*, int n, int& dx, int& dy, int& w, int& h\)](#).
- virtual double [width](#) (const char *str, int n)
see [fl_width\(const char *str, int n\)](#).
- virtual double [width](#) (unsigned int c)
see [fl_width\(unsigned int n\)](#).
- virtual ~[FI_Graphics_Driver](#) ()
The destructor.

Static Public Attributes

- static const char * [class_id](#) = "FI_Graphics_Driver"

Protected Member Functions

- virtual void `arc` (double x, double y, double r, double start, double end)
 see `fl_arc(double x, double y, double r, double start, double end)`.
- virtual void `arc` (int x, int y, int w, int h, double a1, double a2)
 see `fl_arc(int x, int y, int w, int h, double a1, double a2)`.
- virtual void `begin_complex_polygon` ()
 see `fl_begin_complex_polygon()`.
- virtual void `begin_line` ()
 see `fl_begin_line()`.
- virtual void `begin_loop` ()
 see `fl_begin_loop()`.
- virtual void `begin_points` ()
 see `fl_begin_points()`.
- virtual void `begin_polygon` ()
 see `fl_begin_polygon()`.
- virtual void `circle` (double x, double y, double r)
 see `fl_circle(double x, double y, double r)`.
- virtual int `clip_box` (int x, int y, int w, int h, int &X, int &Y, int &W, int &H)
 see `fl_clip_box(int x, int y, int w, int h, int &X, int &Y, int &W, int &H)`.
- FI_Region `clip_region` ()
 see `fl_clip_region()`.
- void `clip_region` (FI_Region r)
 see `fl_clip_region(FI_Region r)`.
- virtual void `color` (FI_Color c)
 see `fl_color(FI_Color c)`.
- virtual void `color` (uchar r, uchar g, uchar b)
 see `fl_color(uchar r, uchar g, uchar b)`.
- virtual void `copy_offscreen` (int x, int y, int w, int h, FI_Offscreen pixmap, int srcx, int srcy)
 see `fl_copy_offscreen()`.
- virtual void `curve` (double X0, double Y0, double X1, double Y1, double X2, double Y2, double X3, double Y3)
 see `fl_curve(double X0, double Y0, double X1, double Y1, double X2, double Y2, double X3, double Y3)`.
- virtual void `draw` (const char *str, int n, int x, int y)
 *see `fl_draw(const char *str, int n, int x, int y)`.*
- virtual void `draw` (FI_Bitmap *bm, int XP, int YP, int WP, int HP, int cx, int cy)
 Draws an FI_Bitmap object to the device.
- virtual void `draw` (FI_Pixmap *pxm, int XP, int YP, int WP, int HP, int cx, int cy)
 Draws an FI_Pixmap object to the device.
- virtual void `draw` (FI_RGB_Image *rgb, int XP, int YP, int WP, int HP, int cx, int cy)
 Draws an FI_RGB_Image object to the device.
- virtual void `draw` (int angle, const char *str, int n, int x, int y)
 *see `fl_draw(int angle, const char *str, int n, int x, int y)`.*
- virtual void `draw_image` (const uchar *buf, int X, int Y, int W, int H, int D=3, int L=0)
 see `fl_draw_image(const uchar buf, int X,int Y,int W,int H, int D, int L)`.*
- virtual void `draw_image` (FI_Draw_Image_Cb cb, void *data, int X, int Y, int W, int H, int D=3)
 see `fl_draw_image(FI_Draw_Image_Cb cb, void data, int X,int Y,int W,int H, int D)`.*
- virtual void `draw_image_mono` (const uchar *buf, int X, int Y, int W, int H, int D=1, int L=0)
 see `fl_draw_image_mono(const uchar buf, int X,int Y,int W,int H, int D, int L)`.*
- virtual void `draw_image_mono` (FI_Draw_Image_Cb cb, void *data, int X, int Y, int W, int H, int D=1)
 see `fl_draw_image_mono(FI_Draw_Image_Cb cb, void data, int X,int Y,int W,int H, int D)`.*
- virtual void `end_complex_polygon` ()

- see [fl_end_complex_polygon\(\)](#).
- virtual void [end_line](#) ()
 - see [fl_end_line\(\)](#).
- virtual void [end_loop](#) ()
 - see [fl_end_loop\(\)](#).
- virtual void [end_points](#) ()
 - see [fl_end_points\(\)](#).
- virtual void [end_polygon](#) ()
 - see [fl_end_polygon\(\)](#).
- **Fl_Graphics_Driver** ()
 - The constructor.
- virtual void [gap](#) ()
 - see [fl_gap\(\)](#).
- virtual void [line](#) (int x, int y, int x1, int y1)
 - see [fl_line\(int x, int y, int x1, int y1\)](#).
- virtual void [line](#) (int x, int y, int x1, int y1, int x2, int y2)
 - see [fl_line\(int x, int y, int x1, int y1, int x2, int y2\)](#).
- virtual void [line_style](#) (int style, int width=0, char *dashes=0)
 - see [fl_line_style\(int style, int width, char* dashes\)](#).
- virtual void [loop](#) (int x0, int y0, int x1, int y1, int x2, int y2)
 - see [fl_loop\(int x0, int y0, int x1, int y1, int x2, int y2\)](#).
- virtual void [loop](#) (int x0, int y0, int x1, int y1, int x2, int y2, int x3, int y3)
 - see [fl_loop\(int x0, int y0, int x1, int y1, int x2, int y2, int x3, int y3\)](#).
- void [mult_matrix](#) (double a, double b, double c, double d, double x, double y)
 - see [fl_mult_matrix\(double a, double b, double c, double d, double x, double y\)](#).
- virtual int [not_clipped](#) (int x, int y, int w, int h)
 - see [fl_not_clipped\(int x, int y, int w, int h\)](#).
- virtual void [pie](#) (int x, int y, int w, int h, double a1, double a2)
 - see [fl_pie\(int x, int y, int w, int h, double a1, double a2\)](#).
- virtual void [point](#) (int x, int y)
 - see [fl_point\(int x, int y\)](#).
- virtual void [polygon](#) (int x0, int y0, int x1, int y1, int x2, int y2)
 - see [fl_polygon\(int x0, int y0, int x1, int y1, int x2, int y2\)](#).
- virtual void [polygon](#) (int x0, int y0, int x1, int y1, int x2, int y2, int x3, int y3)
 - see [fl_polygon\(int x0, int y0, int x1, int y1, int x2, int y2, int x3, int y3\)](#).
- virtual void [pop_clip](#) ()
 - see [fl_pop_clip\(\)](#).
- void [pop_matrix](#) ()
 - see [fl_pop_matrix\(\)](#).
- virtual void [push_clip](#) (int x, int y, int w, int h)
 - see [fl_push_clip\(int x, int y, int w, int h\)](#).
- void [push_matrix](#) ()
 - see [fl_push_matrix\(\)](#).
- virtual void [push_no_clip](#) ()
 - see [fl_push_no_clip\(\)](#).
- virtual void [rect](#) (int x, int y, int w, int h)
 - see [fl_rect\(int x, int y, int w, int h\)](#).
- virtual void [rectf](#) (int x, int y, int w, int h)
 - see [fl_rectf\(int x, int y, int w, int h\)](#).
- void [restore_clip](#) ()
 - see [fl_restore_clip\(\)](#).

- void **rotate** (double d)
see [fl_rotate\(double d\)](#).
- virtual void **rtl_draw** (const char *str, int n, int x, int y)
see [fl_rtl_draw\(const char *str, int n, int x, int y\)](#).
- void **scale** (double x)
see [fl_scale\(double x\)](#).
- void **scale** (double x, double y)
see [fl_scale\(double x, double y\)](#).
- double **transform_dx** (double x, double y)
see [fl_transform_dx\(double x, double y\)](#).
- double **transform_dy** (double x, double y)
see [fl_transform_dy\(double x, double y\)](#).
- double **transform_x** (double x, double y)
see [fl_transform_x\(double x, double y\)](#).
- double **transform_y** (double x, double y)
see [fl_transform_y\(double x, double y\)](#).
- virtual void **transformed_vertex** (double xf, double yf)
see [fl_transformed_vertex\(double xf, double yf\)](#).
- void **translate** (double x, double y)
see [fl_translate\(double x, double y\)](#).
- virtual void **vertex** (double x, double y)
see [fl_vertex\(double x, double y\)](#).
- virtual void **xyline** (int x, int y, int x1)
see [fl_xyline\(int x, int y, int x1\)](#).
- virtual void **xyline** (int x, int y, int x1, int y2)
see [fl_xyline\(int x, int y, int x1, int y2\)](#).
- virtual void **xyline** (int x, int y, int x1, int y2, int x3)
see [fl_xyline\(int x, int y, int x1, int y2, int x3\)](#).
- virtual void **yxline** (int x, int y, int y1)
see [fl_yxline\(int x, int y, int y1\)](#).
- virtual void **yxline** (int x, int y, int y1, int x2)
see [fl_yxline\(int x, int y, int y1, int x2\)](#).
- virtual void **yxline** (int x, int y, int y1, int x2, int y3)
see [fl_yxline\(int x, int y, int y1, int x2, int y3\)](#).

Protected Attributes

- **matrix * fl_matrix**
Points to the current coordinate transformation matrix.

Friends

- void **fl_arc** (double x, double y, double r, double start, double end)
Adds a series of points to the current path on the arc of a circle.
- void **fl_arc** (int x, int y, int w, int h, double a1, double a2)
Draw ellipse sections using integer coordinates.
- void **fl_begin_complex_polygon** ()
Starts drawing a complex filled polygon.
- void **fl_begin_line** ()
Starts drawing a list of lines.
- void **fl_begin_loop** ()

- Starts drawing a closed sequence of lines.*
- void **fl_begin_points** ()
 - Starts drawing a list of points.*
- void **fl_begin_polygon** ()
 - Starts drawing a convex filled polygon.*
- class **FI_Bitmap**
- void **fl_circle** (double x, double y, double r)
 - fl_circle() is equivalent to fl_arc(x,y,r,0,360), but may be faster.*
- int **fl_clip_box** (int x, int y, int w, int h, int &X, int &Y, int &W, int &H)
 - Intersects the rectangle with the current clip region and returns the bounding box of the result.*
- FI_Region **fl_clip_region** ()
 - Returns the current clipping region.*
- void **fl_clip_region** (FI_Region r)
 - Replaces the top of the clipping stack with a clipping region of any shape.*
- void **fl_color** (FI_Color c)
 - Sets the color for all subsequent drawing operations.*
- void **fl_color** (uchar r, uchar g, uchar b)
 - Sets the color for all subsequent drawing operations.*
- FL_EXPORT void **fl_copy_offscreen** (int x, int y, int w, int h, FI_Offscreen pixmap, int srcx, int srcy)
 - Copy a rectangular area of the given offscreen buffer into the current drawing destination.*
- void **fl_curve** (double X0, double Y0, double X1, double Y1, double X2, double Y2, double X3, double Y3)
 - Adds a series of points on a Bezier curve to the path.*
- void **fl_draw** (const char *str, int n, int x, int y)
 - Draws starting at the given x, y location a UTF-8 string of length n bytes.*
- void **fl_draw** (int angle, const char *str, int n, int x, int y)
 - Draws at the given x, y location a UTF-8 string of length n bytes rotating angle degrees counter-clockwise.*
- void **fl_draw_image** (const uchar *buf, int X, int Y, int W, int H, int D, int L)
 - Draws an 8-bit per color RGB or luminance image.*
- void **fl_draw_image** (FI_Draw_Image_Cb cb, void *data, int X, int Y, int W, int H, int D)
 - Draws an image using a callback function to generate image data.*
- void **fl_draw_image_mono** (const uchar *buf, int X, int Y, int W, int H, int D, int L)
 - Draws a gray-scale (1 channel) image.*
- FL_EXPORT void **fl_draw_image_mono** (FI_Draw_Image_Cb cb, void *data, int X, int Y, int W, int H, int D)
 - Draws a gray-scale image using a callback function to generate image data.*
- void **fl_end_complex_polygon** ()
 - Ends complex filled polygon, and draws.*
- void **fl_end_line** ()
 - Ends list of lines, and draws.*
- void **fl_end_loop** ()
 - Ends closed sequence of lines, and draws.*
- void **fl_end_points** ()
 - Ends list of points, and draws.*
- void **fl_end_polygon** ()
 - Ends convex filled polygon, and draws.*
- void **fl_font** (FI_Font face, FI_Fontsize size)
 - Sets the current font, which is then used in various drawing routines.*
- void **fl_gap** ()
 - Call fl_gap() to separate loops of the path.*
- void **fl_line** (int x, int y, int x1, int y1)
 - Draws a line from (x,y) to (x1,y1)*
- void **fl_line** (int x, int y, int x1, int y1, int x2, int y2)

- Draws a line from (x,y) to (x1,y1) and another from (x1,y1) to (x2,y2)*

 - void **fl_line_style** (int style, int width, char *dashes)

Sets how to draw lines (the "pen").
- void **fl_loop** (int x0, int y0, int x1, int y1, int x2, int y2)

Outlines a 3-sided polygon with lines.
- void **fl_loop** (int x0, int y0, int x1, int y1, int x2, int y2, int x3, int y3)

Outlines a 4-sided polygon with lines.
- void **fl_mult_matrix** (double a, double b, double c, double d, double x, double y)

Concatenates another transformation onto the current one.
- int **fl_not_clipped** (int x, int y, int w, int h)

Does the rectangle intersect the current clip region?
- void **fl_pie** (int x, int y, int w, int h, double a1, double a2)

Draw filled ellipse sections using integer coordinates.
- class **FI_Pixmap**
- void **fl_point** (int x, int y)

Draws a single pixel at the given coordinates.
- void **fl_polygon** (int x0, int y0, int x1, int y1, int x2, int y2)

Fills a 3-sided polygon.
- void **fl_polygon** (int x0, int y0, int x1, int y1, int x2, int y2, int x3, int y3)

Fills a 4-sided polygon.
- void **fl_pop_clip** ()

Restores the previous clip region.
- void **fl_pop_matrix** ()

Restores the current transformation matrix from the stack.
- void **fl_push_clip** (int x, int y, int w, int h)

Intersects the current clip region with a rectangle and pushes this new region onto the stack.
- void **fl_push_matrix** ()

Saves the current transformation matrix on the stack.
- void **fl_push_no_clip** ()

Pushes an empty clip region onto the stack so nothing will be clipped.
- void **fl_rect** (int x, int y, int w, int h)

Draws a 1-pixel border inside the given bounding box.
- void **fl_rectf** (int x, int y, int w, int h)

Colors with current color a rectangle that exactly fills the given bounding box.
- void **fl_restore_clip** ()

Undoes any clobbering of clip done by your program.
- class **FI_RGB_Image**
- void **fl_rotate** (double d)

Concatenates rotation transformation onto the current one.
- void **fl_rtl_draw** (const char *str, int n, int x, int y)

Draws a UTF-8 string of length n bytes right to left starting at the given x, y location.
- void **fl_scale** (double x)

Concatenates scaling transformation onto the current one.
- void **fl_scale** (double x, double y)

Concatenates scaling transformation onto the current one.
- double **fl_transform_dx** (double x, double y)

Transforms distance using current transformation matrix.
- double **fl_transform_dy** (double x, double y)

Transforms distance using current transformation matrix.
- double **fl_transform_x** (double x, double y)

Transforms coordinate using the current transformation matrix.

- double [fl_transform_y](#) (double x, double y)
Transforms coordinate using the current transformation matrix.
- void [fl_transformed_vertex](#) (double xf, double yf)
Adds coordinate pair to the vertex list without further transformations.
- void [fl_translate](#) (double x, double y)
Concatenates translation transformation onto the current one.
- void [fl_vertex](#) (double x, double y)
Adds a single vertex to the current path.
- void [fl_xyline](#) (int x, int y, int x1)
Draws a horizontal line from (x,y) to (x1,y)
- void [fl_xyline](#) (int x, int y, int x1, int y2)
Draws a horizontal line from (x,y) to (x1,y), then vertical from (x1,y) to (x1,y2)
- void [fl_xyline](#) (int x, int y, int x1, int y2, int x3)
Draws a horizontal line from (x,y) to (x1,y), then a vertical from (x1,y) to (x1,y2) and then another horizontal from (x1,y2) to (x3,y2)
- void [fl_yxline](#) (int x, int y, int y1)
Draws a vertical line from (x,y) to (x,y1)
- void [fl_yxline](#) (int x, int y, int y1, int x2)
Draws a vertical line from (x,y) to (x,y1), then a horizontal from (x,y1) to (x2,y1)
- void [fl_yxline](#) (int x, int y, int y1, int x2, int y3)
Draws a vertical line from (x,y) to (x,y1) then a horizontal from (x,y1) to (x2,y1), then another vertical from (x2,y1) to (x2,y3)
- FL_EXPORT void [gl_start](#) ()
Creates an OpenGL context.

31.53.1 Detailed Description

A virtual class subclassed for each graphics driver FLTK uses.

Typically, FLTK applications do not use directly objects from this class. Rather, they perform drawing operations (e.g., [fl_rectf\(\)](#)) that operate on the current drawing surface (see [Fl_Surface_Device](#)). Drawing operations are functionally presented in [Drawing Things in FLTK](#) and as function lists in the [Drawing functions](#) and [Color & Font functions](#) modules. The [fl_graphics_driver](#) global variable gives at any time the graphics driver used by all drawing operations. Its value changes when drawing operations are directed to another drawing surface by [Fl_Surface_Device::set_current\(\)](#).

The [Fl_Graphics_Driver](#) class is of interest if one wants to perform new kinds of drawing operations. An example would be to draw to a PDF file. This would involve creating a new [Fl_Graphics_Driver](#) derived class. This new class should implement all virtual methods of the [Fl_Graphics_Driver](#) class to support all FLTK drawing functions.

31.53.2 Member Function Documentation

31.53.2.1 [arc\(\)](#) [1/2]

```
void Fl_Graphics_Driver::arc (
    double x,
    double y,
    double r,
    double start,
    double end ) [protected], [virtual]
```

see [fl_arc\(double x, double y, double r, double start, double end\)](#).

Reimplemented in [Fl_PostScript_Graphics_Driver](#).

31.53.2.2 arc() [2/2]

```
void Fl_Graphics_Driver::arc (
    int x,
    int y,
    int w,
    int h,
    double a1,
    double a2 ) [protected], [virtual]
```

see [fl_arc\(int x, int y, int w, int h, double a1, double a2\)](#).

Reimplemented in [Fl_PostScript_Graphics_Driver](#).

31.53.2.3 begin_complex_polygon()

```
void Fl_Graphics_Driver::begin_complex_polygon ( ) [protected], [virtual]
```

see [fl_begin_complex_polygon\(\)](#).

Reimplemented in [Fl_PostScript_Graphics_Driver](#).

31.53.2.4 begin_line()

```
void Fl_Graphics_Driver::begin_line ( ) [protected], [virtual]
```

see [fl_begin_line\(\)](#).

Reimplemented in [Fl_PostScript_Graphics_Driver](#).

31.53.2.5 begin_loop()

```
void Fl_Graphics_Driver::begin_loop ( ) [protected], [virtual]
```

see [fl_begin_loop\(\)](#).

Reimplemented in [Fl_PostScript_Graphics_Driver](#).

31.53.2.6 begin_points()

```
void Fl_Graphics_Driver::begin_points ( ) [protected], [virtual]
```

see [fl_begin_points\(\)](#).

Reimplemented in [Fl_PostScript_Graphics_Driver](#).

31.53.2.7 begin_polygon()

```
void Fl_Graphics_Driver::begin_polygon ( ) [protected], [virtual]
```

see [fl_begin_polygon\(\)](#).

Reimplemented in [Fl_PostScript_Graphics_Driver](#).

31.53.2.8 circle()

```
void Fl_Graphics_Driver::circle (
    double x,
    double y,
    double r ) [protected], [virtual]
```

see [fl_circle\(double x, double y, double r\)](#).

Reimplemented in [Fl_PostScript_Graphics_Driver](#).

31.53.2.9 class_name()

```
virtual const char * Fl_Graphics_Driver::class_name ( ) [inline], [virtual]
```

Returns the name of the class of this object.

Use of the `class_name()` function is discouraged because it will be removed from future FLTK versions.

The class of an instance of an `Fl_Device` subclass can be checked with code such as:

```
if ( instance->class_name() == Fl_Printer::class_id ) { ... }
```

Reimplemented from `Fl_Device`.

Reimplemented in `Fl_Quartz_Graphics_Driver`, `Fl_GDI_Graphics_Driver`, `Fl_GDI_Printer_Graphics_Driver`, `Fl_Xlib_Graphics_Driver`, and `Fl_PostScript_Graphics_Driver`.

31.53.2.10 clip_box()

```
int Fl_Graphics_Driver::clip_box (
    int x,
    int y,
    int w,
    int h,
    int & X,
    int & Y,
    int & W,
    int & H ) [protected], [virtual]
```

see `fl_clip_box(int x, int y, int w, int h, int &X, int &Y, int &W, int &H)`.

Reimplemented in `Fl_PostScript_Graphics_Driver`.

31.53.2.11 color() [1/2]

```
virtual void Fl_Graphics_Driver::color (
    Fl_Color c ) [inline], [protected], [virtual]
```

see `fl_color(Fl_Color c)`.

Reimplemented in `Fl_Quartz_Graphics_Driver`, `Fl_GDI_Graphics_Driver`, `Fl_Xlib_Graphics_Driver`, and `Fl_PostScript_Graphics_Driver`.

31.53.2.12 color() [2/2]

```
virtual void Fl_Graphics_Driver::color (
    uchar r,
    uchar g,
    uchar b ) [inline], [protected], [virtual]
```

see `fl_color(uchar r, uchar g, uchar b)`.

Reimplemented in `Fl_Quartz_Graphics_Driver`, `Fl_GDI_Graphics_Driver`, `Fl_Xlib_Graphics_Driver`, and `Fl_PostScript_Graphics_Driver`.

31.53.2.13 copy_offscreen()

```
void Fl_Graphics_Driver::copy_offscreen (
    int x,
    int y,
    int w,
    int h,
    Fl_Offscreen pixmap,
    int srcx,
    int srcy ) [protected], [virtual]
```

see `fl_copy_offscreen()`

Reimplemented in `Fl_Quartz_Graphics_Driver`, `Fl_GDI_Graphics_Driver`, and `Fl_Xlib_Graphics_Driver`.

31.53.2.14 curve()

```
void Fl_Graphics_Driver::curve (
    double X0,
    double Y0,
    double X1,
    double Y1,
    double X2,
    double Y2,
    double X3,
    double Y3 ) [protected], [virtual]
```

see [fl_curve\(double X0, double Y0, double X1, double Y1, double X2, double Y2, double X3, double Y3\)](#).

Reimplemented in [Fl_PostScript_Graphics_Driver](#).

31.53.2.15 descent()

```
virtual int Fl_Graphics_Driver::descent ( ) [inline], [virtual]
```

see [fl_descent\(\)](#).

Reimplemented in [Fl_Quartz_Graphics_Driver](#), [Fl_GDI_Graphics_Driver](#), [Fl_Xlib_Graphics_Driver](#), and [Fl_PostScript_Graphics_Driver](#).

31.53.2.16 draw() [1/5]

```
virtual void Fl_Graphics_Driver::draw (
    const char * str,
    int n,
    int x,
    int y ) [inline], [protected], [virtual]
```

see [fl_draw\(const char *str, int n, int x, int y\)](#).

Reimplemented in [Fl_PostScript_Graphics_Driver](#), [Fl_Quartz_Graphics_Driver](#), [Fl_GDI_Graphics_Driver](#), and [Fl_Xlib_Graphics_Driver](#).

31.53.2.17 draw() [2/5]

```
virtual void Fl_Graphics_Driver::draw (
    Fl_Bitmap * bm,
    int XP,
    int YP,
    int WP,
    int HP,
    int cx,
    int cy ) [inline], [protected], [virtual]
```

Draws an [Fl_Bitmap](#) object to the device.

Specifies a bounding box for the image, with the origin (upper left-hand corner) of the image offset by the cx and cy arguments.

Reimplemented in [Fl_PostScript_Graphics_Driver](#), [Fl_GDI_Printer_Graphics_Driver](#), [Fl_Quartz_Graphics_Driver](#), [Fl_GDI_Graphics_Driver](#), and [Fl_Xlib_Graphics_Driver](#).

31.53.2.18 draw() [3/5]

```
virtual void Fl_Graphics_Driver::draw (
    Fl_Pixmap * pxm,
    int XP,
    int YP,
    int WP,
```

```

    int HP,
    int cx,
    int cy ) [inline], [protected], [virtual]

```

Draws an [Fl_Pixmap](#) object to the device.

Specifies a bounding box for the image, with the origin (upper left-hand corner) of the image offset by the *cx* and *cy* arguments.

Reimplemented in [Fl_Quartz_Graphics_Driver](#), [Fl_GDI_Graphics_Driver](#), [Fl_GDI_Printer_Graphics_Driver](#), [Fl_Xlib_Graphics_Driver](#), and [Fl_PostScript_Graphics_Driver](#).

31.53.2.19 draw() [4/5]

```

virtual void Fl_Graphics_Driver::draw (
    Fl_RGB_Image * rgb,
    int XP,
    int YP,
    int WP,
    int HP,
    int cx,
    int cy ) [inline], [protected], [virtual]

```

Draws an [Fl_RGB_Image](#) object to the device.

Specifies a bounding box for the image, with the origin (upper left-hand corner) of the image offset by the *cx* and *cy* arguments.

Reimplemented in [Fl_Quartz_Graphics_Driver](#), [Fl_GDI_Graphics_Driver](#), [Fl_Xlib_Graphics_Driver](#), and [Fl_PostScript_Graphics_Driver](#).

31.53.2.20 draw() [5/5]

```

virtual void Fl_Graphics_Driver::draw (
    int angle,
    const char * str,
    int n,
    int x,
    int y ) [inline], [protected], [virtual]

```

see [fl_draw\(int angle, const char *str, int n, int x, int y\)](#).

Reimplemented in [Fl_Quartz_Graphics_Driver](#), [Fl_GDI_Graphics_Driver](#), [Fl_Xlib_Graphics_Driver](#), and [Fl_PostScript_Graphics_Driver](#).

31.53.2.21 draw_image() [1/2]

```

virtual void Fl_Graphics_Driver::draw_image (
    const uchar * buf,
    int X,
    int Y,
    int W,
    int H,
    int D = 3,
    int L = 0 ) [inline], [protected], [virtual]

```

see [fl_draw_image\(const uchar* buf, int X,int Y,int W,int H, int D, int L\)](#).

Reimplemented in [Fl_Quartz_Graphics_Driver](#), [Fl_GDI_Graphics_Driver](#), [Fl_Xlib_Graphics_Driver](#), and [Fl_PostScript_Graphics_Driver](#).

31.53.2.22 draw_image() [2/2]

```

virtual void Fl_Graphics_Driver::draw_image (
    Fl_Draw_Image_Cb cb,

```

```

void * data,
int X,
int Y,
int W,
int H,
int D = 3 ) [inline], [protected], [virtual]

```

see [fl_draw_image\(Fl_Draw_Image_Cb cb, void* data, int X,int Y,int W,int H, int D\)](#).

Reimplemented in [FI_PostScript_Graphics_Driver](#), [FI_Quartz_Graphics_Driver](#), [FI_GDI_Graphics_Driver](#), and [FI_Xlib_Graphics_Driver](#).

31.53.2.23 draw_image_mono() [1/2]

```

virtual void Fl_Graphics_Driver::draw_image_mono (
    const uchar * buf,
    int X,
    int Y,
    int W,
    int H,
    int D = 1,
    int L = 0 ) [inline], [protected], [virtual]

```

see [fl_draw_image_mono\(const uchar* buf, int X,int Y,int W,int H, int D, int L\)](#).

Reimplemented in [FI_Quartz_Graphics_Driver](#), [FI_GDI_Graphics_Driver](#), [FI_Xlib_Graphics_Driver](#), and [FI_PostScript_Graphics_Driver](#).

31.53.2.24 draw_image_mono() [2/2]

```

virtual void Fl_Graphics_Driver::draw_image_mono (
    Fl_Draw_Image_Cb cb,
    void * data,
    int X,
    int Y,
    int W,
    int H,
    int D = 1 ) [inline], [protected], [virtual]

```

see [fl_draw_image_mono\(Fl_Draw_Image_Cb cb, void* data, int X,int Y,int W,int H, int D\)](#).

Reimplemented in [FI_PostScript_Graphics_Driver](#), [FI_Quartz_Graphics_Driver](#), [FI_GDI_Graphics_Driver](#), and [FI_Xlib_Graphics_Driver](#).

31.53.2.25 draw_scaled()

```

int Fl_Graphics_Driver::draw_scaled (
    Fl_Image * img,
    int X,
    int Y,
    int W,
    int H ) [virtual]

```

Draws an [Fl_Image](#) scaled to width W & height H with top-left corner at X,Y.

Returns

zero when the graphics driver doesn't implement scaled drawing, non-zero if it does implement it.

Reimplemented in [FI_Quartz_Graphics_Driver](#), [FI_GDI_Printer_Graphics_Driver](#), and [FI_PostScript_Graphics_Driver](#).

31.53.2.26 end_complex_polygon()

```

void Fl_Graphics_Driver::end_complex_polygon ( ) [protected], [virtual]

```

see [fl_end_complex_polygon\(\)](#).
Reimplemented in [Fl_PostScript_Graphics_Driver](#).

31.53.2.27 end_line()

void [Fl_Graphics_Driver::end_line](#) () [protected], [virtual]
see [fl_end_line\(\)](#).
Reimplemented in [Fl_PostScript_Graphics_Driver](#).

31.53.2.28 end_loop()

void [Fl_Graphics_Driver::end_loop](#) () [protected], [virtual]
see [fl_end_loop\(\)](#).
Reimplemented in [Fl_PostScript_Graphics_Driver](#).

31.53.2.29 end_points()

void [Fl_Graphics_Driver::end_points](#) () [protected], [virtual]
see [fl_end_points\(\)](#).
Reimplemented in [Fl_PostScript_Graphics_Driver](#).

31.53.2.30 end_polygon()

void [Fl_Graphics_Driver::end_polygon](#) () [protected], [virtual]
see [fl_end_polygon\(\)](#).
Reimplemented in [Fl_PostScript_Graphics_Driver](#).

31.53.2.31 font()

virtual void [Fl_Graphics_Driver::font](#) (
 [Fl_Font](#) *face*,
 [Fl_Fontsize](#) *fsize*) [inline], [virtual]
see [fl_font\(Fl_Font face, Fl_Fontsize size\)](#).
Reimplemented in [Fl_Quartz_Graphics_Driver](#), [Fl_GDI_Graphics_Driver](#), [Fl_Xlib_Graphics_Driver](#), and [Fl_PostScript_Graphics_Driver](#).

31.53.2.32 gap()

void [Fl_Graphics_Driver::gap](#) () [protected], [virtual]
see [fl_gap\(\)](#).
Reimplemented in [Fl_PostScript_Graphics_Driver](#).

31.53.2.33 height()

virtual int [Fl_Graphics_Driver::height](#) () [inline], [virtual]
see [fl_height\(\)](#).
Reimplemented in [Fl_Quartz_Graphics_Driver](#), [Fl_GDI_Graphics_Driver](#), [Fl_Xlib_Graphics_Driver](#), and [Fl_PostScript_Graphics_Driver](#).

31.53.2.34 line() [1/2]

```
void Fl_Graphics_Driver::line (  
    int x,  
    int y,  
    int x1,  
    int y1 ) [protected], [virtual]
```

see [fl_line\(int x, int y, int x1, int y1\)](#).

Reimplemented in [Fl_PostScript_Graphics_Driver](#).

31.53.2.35 line() [2/2]

```
void Fl_Graphics_Driver::line (  
    int x,  
    int y,  
    int x1,  
    int y1,  
    int x2,  
    int y2 ) [protected], [virtual]
```

see [fl_line\(int x, int y, int x1, int y1, int x2, int y2\)](#).

Reimplemented in [Fl_PostScript_Graphics_Driver](#).

31.53.2.36 line_style()

```
void Fl_Graphics_Driver::line_style (  
    int style,  
    int width = 0,  
    char * dashes = 0 ) [protected], [virtual]
```

see [fl_line_style\(int style, int width, char* dashes\)](#).

Reimplemented in [Fl_PostScript_Graphics_Driver](#).

31.53.2.37 loop() [1/2]

```
void Fl_Graphics_Driver::loop (  
    int x0,  
    int y0,  
    int x1,  
    int y1,  
    int x2,  
    int y2 ) [protected], [virtual]
```

see [fl_loop\(int x0, int y0, int x1, int y1, int x2, int y2\)](#).

Reimplemented in [Fl_PostScript_Graphics_Driver](#).

31.53.2.38 loop() [2/2]

```
void Fl_Graphics_Driver::loop (  
    int x0,  
    int y0,  
    int x1,  
    int y1,  
    int x2,  
    int y2,  
    int x3,  
    int y3 ) [protected], [virtual]
```

see [fl_loop\(int x0, int y0, int x1, int y1, int x2, int y2, int x3, int y3\)](#).

Reimplemented in [Fl_PostScript_Graphics_Driver](#).

31.53.2.39 not_clipped()

```
int Fl_Graphics_Driver::not_clipped (
    int x,
    int y,
    int w,
    int h ) [protected], [virtual]
```

see [fl_not_clipped\(int x, int y, int w, int h\)](#).

Reimplemented in [Fl_PostScript_Graphics_Driver](#).

31.53.2.40 pie()

```
void Fl_Graphics_Driver::pie (
    int x,
    int y,
    int w,
    int h,
    double a1,
    double a2 ) [protected], [virtual]
```

see [fl_pie\(int x, int y, int w, int h, double a1, double a2\)](#).

Reimplemented in [Fl_PostScript_Graphics_Driver](#).

31.53.2.41 point()

```
void Fl_Graphics_Driver::point (
    int x,
    int y ) [protected], [virtual]
```

see [fl_point\(int x, int y\)](#).

Reimplemented in [Fl_PostScript_Graphics_Driver](#).

31.53.2.42 polygon() [1/2]

```
void Fl_Graphics_Driver::polygon (
    int x0,
    int y0,
    int x1,
    int y1,
    int x2,
    int y2 ) [protected], [virtual]
```

see [fl_polygon\(int x0, int y0, int x1, int y1, int x2, int y2\)](#).

Reimplemented in [Fl_PostScript_Graphics_Driver](#).

31.53.2.43 polygon() [2/2]

```
void Fl_Graphics_Driver::polygon (
    int x0,
    int y0,
    int x1,
    int y1,
    int x2,
    int y2,
```

```
        int x3,  
        int y3 ) [protected], [virtual]
```

see [fl_polygon\(int x0, int y0, int x1, int y1, int x2, int y2, int x3, int y3\)](#).

Reimplemented in [FI_PostScript_Graphics_Driver](#).

31.53.2.44 pop_clip()

```
void Fl_Graphics_Driver::pop_clip ( ) [protected], [virtual]
```

see [fl_pop_clip\(\)](#).

Reimplemented in [FI_PostScript_Graphics_Driver](#).

31.53.2.45 push_clip()

```
void Fl_Graphics_Driver::push_clip (  
    int x,  
    int y,  
    int w,  
    int h ) [protected], [virtual]
```

see [fl_push_clip\(int x, int y, int w, int h\)](#).

Reimplemented in [FI_PostScript_Graphics_Driver](#).

31.53.2.46 push_no_clip()

```
void Fl_Graphics_Driver::push_no_clip ( ) [protected], [virtual]
```

see [fl_push_no_clip\(\)](#).

Reimplemented in [FI_PostScript_Graphics_Driver](#).

31.53.2.47 rect()

```
void Fl_Graphics_Driver::rect (  
    int x,  
    int y,  
    int w,  
    int h ) [protected], [virtual]
```

see [fl_rect\(int x, int y, int w, int h\)](#).

Reimplemented in [FI_PostScript_Graphics_Driver](#).

31.53.2.48 rectf()

```
void Fl_Graphics_Driver::rectf (  
    int x,  
    int y,  
    int w,  
    int h ) [protected], [virtual]
```

see [fl_rectf\(int x, int y, int w, int h\)](#).

Reimplemented in [FI_PostScript_Graphics_Driver](#).

31.53.2.49 rtl_draw()

```
virtual void Fl_Graphics_Driver::rtl_draw (  
    const char * str,  
    int n,
```



```
int x,  
int y ) [inline], [protected], [virtual]
```

see [fl_rtl_draw\(const char *str, int n, int x, int y\)](#).

Reimplemented in [Fl_PostScript_Graphics_Driver](#), [Fl_Quartz_Graphics_Driver](#), [Fl_GDI_Graphics_Driver](#), and [Fl_Xlib_Graphics_Driver](#).

31.53.2.50 text_extents()

```
void Fl_Graphics_Driver::text_extents (  
    const char * t,  
    int n,  
    int & dx,  
    int & dy,  
    int & w,  
    int & h ) [virtual]
```

see [fl_text_extents\(const char*, int n, int& dx, int& dy, int& w, int& h\)](#).

Reimplemented in [Fl_Quartz_Graphics_Driver](#), [Fl_GDI_Graphics_Driver](#), [Fl_Xlib_Graphics_Driver](#), and [Fl_PostScript_Graphics_Driver](#).

31.53.2.51 transformed_vertex()

```
void Fl_Graphics_Driver::transformed_vertex (  
    double xf,  
    double yf ) [protected], [virtual]
```

see [fl_transformed_vertex\(double xf, double yf\)](#).

Reimplemented in [Fl_PostScript_Graphics_Driver](#).

31.53.2.52 vertex()

```
void Fl_Graphics_Driver::vertex (  
    double x,  
    double y ) [protected], [virtual]
```

see [fl_vertex\(double x, double y\)](#).

Reimplemented in [Fl_PostScript_Graphics_Driver](#).

31.53.2.53 width() [1/2]

```
virtual double Fl_Graphics_Driver::width (  
    const char * str,  
    int n ) [inline], [virtual]
```

see [fl_width\(const char *str, int n\)](#).

Reimplemented in [Fl_PostScript_Graphics_Driver](#), [Fl_Quartz_Graphics_Driver](#), [Fl_GDI_Graphics_Driver](#), and [Fl_Xlib_Graphics_Driver](#).

31.53.2.54 width() [2/2]

```
virtual double Fl_Graphics_Driver::width (  
    unsigned int c ) [inline], [virtual]
```

see [fl_width\(unsigned int n\)](#).

Reimplemented in [Fl_Quartz_Graphics_Driver](#), [Fl_GDI_Graphics_Driver](#), [Fl_Xlib_Graphics_Driver](#), and [Fl_PostScript_Graphics_Driver](#).

31.53.2.55 xyline() [1/3]

```
void Fl_Graphics_Driver::xyline (
    int x,
    int y,
    int x1 ) [protected], [virtual]
```

see [fl_xyline\(int x, int y, int x1\)](#).

Reimplemented in [Fl_PostScript_Graphics_Driver](#).

31.53.2.56 xyline() [2/3]

```
void Fl_Graphics_Driver::xyline (
    int x,
    int y,
    int x1,
    int y2 ) [protected], [virtual]
```

see [fl_xyline\(int x, int y, int x1, int y2\)](#).

Reimplemented in [Fl_PostScript_Graphics_Driver](#).

31.53.2.57 xyline() [3/3]

```
void Fl_Graphics_Driver::xyline (
    int x,
    int y,
    int x1,
    int y2,
    int x3 ) [protected], [virtual]
```

see [fl_xyline\(int x, int y, int x1, int y2, int x3\)](#).

Reimplemented in [Fl_PostScript_Graphics_Driver](#).

31.53.2.58 yxline() [1/3]

```
void Fl_Graphics_Driver::yxline (
    int x,
    int y,
    int y1 ) [protected], [virtual]
```

see [fl_yxline\(int x, int y, int y1\)](#).

Reimplemented in [Fl_PostScript_Graphics_Driver](#).

31.53.2.59 yxline() [2/3]

```
void Fl_Graphics_Driver::yxline (
    int x,
    int y,
    int y1,
    int x2 ) [protected], [virtual]
```

see [fl_yxline\(int x, int y, int y1, int x2\)](#).

Reimplemented in [Fl_PostScript_Graphics_Driver](#).

31.53.2.60 yxline() [3/3]

```
void Fl_Graphics_Driver::yxline (
    int x,
    int y,
```

```

        int y1,
        int x2,
        int y3 ) [protected], [virtual]

```

see [fl_yxline\(int x, int y, int y1, int x2, int y3\)](#).

Reimplemented in [Fl_PostScript_Graphics_Driver](#).

31.53.3 Friends And Related Function Documentation

31.53.3.1 fl_arc [1/2]

```

void fl_arc (
    double x,
    double y,
    double r,
    double start,
    double end ) [friend]

```

Adds a series of points to the current path on the arc of a circle.

You can get elliptical paths by using [scale](#) and [rotate](#) before calling [fl_arc\(\)](#).

Parameters

in	<i>x,y,r</i>	center and radius of circular arc
in	<i>start,end</i>	angles of start and end of arc measured in degrees counter-clockwise from 3 o'clock. If <i>end</i> is less than <i>start</i> then it draws the arc in a clockwise direction.

Examples:

```

// Draw an arc of points
fl_begin_points();
fl_arc(100.0, 100.0, 50.0, 0.0, 180.0);
fl_end_points();
// Draw arc with a line
fl_begin_line();
fl_arc(200.0, 100.0, 50.0, 0.0, 180.0);
fl_end_line();
// Draw filled arc
fl_begin_polygon();
fl_arc(300.0, 100.0, 50.0, 0.0, 180.0);
fl_end_polygon();

```

31.53.3.2 fl_arc [2/2]

```

void fl_arc (
    int x,
    int y,
    int w,
    int h,
    double a1,
    double a2 ) [friend]

```

Draw ellipse sections using integer coordinates.

These functions match the rather limited circle drawing code provided by X and WIN32. The advantage over using [fl_arc](#) with floating point coordinates is that they are faster because they often use the hardware, and they draw much nicer small circles, since the small sizes are often hard-coded bitmaps.

If a complete circle is drawn it will fit inside the passed bounding box. The two angles are measured in degrees counter-clockwise from 3 o'clock and are the starting and ending angle of the arc, *a2* must be greater or equal to *a1*.

[fl_arc\(\)](#) draws a series of lines to approximate the arc. Notice that the integer version of [fl_arc\(\)](#) has a different number of arguments than the double version [fl_arc\(double x, double y, double r, double start, double end\)](#)

Parameters

in	x,y,w,h	bounding box of complete circle
in	$a1,a2$	start and end angles of arc measured in degrees counter-clockwise from 3 o'clock. $a2$ must be greater than or equal to $a1$.

31.53.3.3 fl_begin_complex_polygon

```
void fl_begin_complex_polygon ( ) [friend]
```

Starts drawing a complex filled polygon.

The polygon may be concave, may have holes in it, or may be several disconnected pieces. Call [fl_gap\(\)](#) to separate loops of the path.

To outline the polygon, use [fl_begin_loop\(\)](#) and replace each [fl_gap\(\)](#) with [fl_end_loop\(\)](#); [fl_begin_loop\(\)](#) pairs.

Note

For portability, you should only draw polygons that appear the same whether "even/odd" or "non-zero" winding rules are used to fill them. Holes should be drawn in the opposite direction to the outside loop.

31.53.3.4 fl_begin_points

```
void fl_begin_points ( ) [friend]
```

Starts drawing a list of points.

Points are added to the list with [fl_vertex\(\)](#)

31.53.3.5 fl_circle

```
void fl_circle (
    double x,
    double y,
    double r ) [friend]
```

[fl_circle\(\)](#) is equivalent to [fl_arc\(x,y,r,0,360\)](#), but may be faster.

It must be the *only* thing in the path: if you want a circle as part of a complex polygon you must use [fl_arc\(\)](#)

Parameters

in	x,y,r	center and radius of circle
----	---------	-----------------------------

31.53.3.6 fl_clip_box

```
int fl_clip_box (
    int x,
    int y,
    int w,
    int h,
    int & X,
    int & Y,
    int & W,
    int & H ) [friend]
```

Intersects the rectangle with the current clip region and returns the bounding box of the result.

Returns non-zero if the resulting rectangle is different to the original. This can be used to limit the necessary drawing to a rectangle. W and H are set to zero if the rectangle is completely outside the region.

Parameters

in	<i>x,y,w,h</i>	position and size of rectangle
out	<i>X,Y,W,H</i>	position and size of resulting bounding box.

Returns

Non-zero if the resulting rectangle is different to the original.

31.53.3.7 fl_clip_region

```
void fl_clip_region (
    Fl_Region r ) [friend]
```

Replaces the top of the clipping stack with a clipping region of any shape. Fl_Region is an operating system specific type.

Parameters

in	<i>r</i>	clipping region
----	----------	-----------------

31.53.3.8 fl_color [1/2]

```
void fl_color (
    Fl_Color c ) [friend]
```

Sets the color for all subsequent drawing operations.

For colormapped displays, a color cell will be allocated out of `fl_colormap` the first time you use a color. If the colormap fills up then a least-squares algorithm is used to find the closest color. If no valid graphical context (`fl_gc`) is available, the foreground is not set for the current window.

Parameters

in	<i>c</i>	color
----	----------	-------

31.53.3.9 fl_color [2/2]

```
void fl_color (
    uchar r,
    uchar g,
    uchar b ) [friend]
```

Sets the color for all subsequent drawing operations.

The closest possible match to the RGB color is used. The RGB color is used directly on TrueColor displays. For colormap visuals the nearest index in the gray ramp or color cube is used. If no valid graphical context (`fl_gc`) is available, the foreground is not set for the current window.

Parameters

in	<i>r,g,b</i>	color components
----	--------------	------------------

31.53.3.10 fl_copy_offscreen

```
FL_EXPORT void fl_copy_offscreen (
    int x,
    int y,
    int w,
    int h,
    Fl_Offscreen pixmap,
    int srcx,
    int srcy ) [friend]
```

Copy a rectangular area of the given offscreen buffer into the current drawing destination.

Parameters

<i>x,y</i>	position where to draw the copied rectangle
<i>w,h</i>	size of the copied rectangle
<i>pixmap</i>	offscreen buffer containing the rectangle to copy
<i>srcx,srcy</i>	origin in offscreen buffer of rectangle to copy

31.53.3.11 fl_curve

```
void fl_curve (
    double X0,
    double Y0,
    double X1,
    double Y1,
    double X2,
    double Y2,
    double X3,
    double Y3 ) [friend]
```

Adds a series of points on a Bezier curve to the path.
The curve ends (and two of the points) are at X0,Y0 and X3,Y3.

Parameters

in	<i>X0,Y0</i>	curve start point
in	<i>X1,Y1</i>	curve control point
in	<i>X2,Y2</i>	curve control point
in	<i>X3,Y3</i>	curve end point

31.53.3.12 fl_draw

```
void fl_draw (
    int angle,
    const char * str,
    int n,
    int x,
    int y ) [friend]
```

Draws at the given *x, y* location a UTF-8 string of length *n* bytes rotating *angle* degrees counter-clockwise.

Note

When using X11 (Unix, Linux, Cygwin et al.) this needs Xft to work. Under plain X11 (w/o Xft) rotated text is not supported by FLTK. A warning will be issued to stderr at runtime (only once) if you use this method with an angle other than 0.

31.53.3.13 fl_draw_image [1/2]

```
void fl_draw_image (
    const uchar * buf,
    int X,
    int Y,
    int W,
    int H,
    int D = 3,
    int L = 0 ) [friend]
```

Draws an 8-bit per color RGB or luminance image.

Parameters

in	<i>buf</i>	points at the "r" data of the top-left pixel. Color data must be in <i>r, g, b</i> order. Luminance data is only one <code>gray</code> byte.
in	<i>X, Y</i>	position where to put top-left corner of image
in	<i>W, H</i>	size of the image
in	<i>D</i>	delta to add to the pointer between pixels. It may be any value greater than or equal to 1, or it can be negative to flip the image horizontally
in	<i>L</i>	delta to add to the pointer between lines (if 0 is passed it uses $W * D$), and may be larger than $W * D$ to crop data, or negative to flip the image vertically

It is highly recommended that you put the following code before the first `show()` of *any* window in your program to get rid of the dithering if possible:

```
Fl::visual(FL_RGB);
```

Gray scale (1-channel) images may be drawn. This is done if `abs(D)` is less than 3, or by calling `fl_draw_image_mono()`. Only one 8-bit sample is used for each pixel, and on screens with different numbers of bits for red, green, and blue only gray colors are used. Setting `D` greater than 1 will let you display one channel of a color image.

Note:

The X version does not support all possible visuals. If FLTK cannot draw the image in the current visual it will abort. FLTK supports any visual of 8 bits or less, and all common TrueColor visuals up to 32 bits.

31.53.3.14 fl_draw_image [2/2]

```
void fl_draw_image (
    Fl_Draw_Image_Cb cb,
    void * data,
    int X,
    int Y,
    int W,
    int H,
    int D = 3 ) [friend]
```

Draws an image using a callback function to generate image data.

You can generate the image as it is being drawn, or do arbitrary decompression of stored data, provided it can be decompressed to individual scan lines easily.

Parameters

in	<i>cb</i>	callback function to generate scan line data
in	<i>data</i>	user data passed to callback function
in	<i>X,Y</i>	screen position of top left pixel
in	<i>W,H</i>	image width and height
in	<i>D</i>	data size in bytes (must be greater than 0)

See also

[fl_draw_image\(const uchar* buf, int X,int Y,int W,int H, int D, int L\)](#)

The callback function *cb* is called with the `void* data` user data pointer to allow access to a structure of information about the image, and the *x*, *y*, and *w* of the scan line desired from the image. 0,0 is the upper-left corner of the image, not *x*, *y*. A pointer to a buffer to put the data into is passed. You must copy *w* pixels from scanline *y*, starting at pixel *x*, to this buffer.

Due to cropping, less than the whole image may be requested. So *x* may be greater than zero, the first *y* may be greater than zero, and *w* may be less than *W*. The buffer is long enough to store the entire $W * D$ pixels, this is for convenience with some decompression schemes where you must decompress the entire line at once: decompress it into the buffer, and then if *x* is not zero, copy the data over so the *x*'th pixel is at the start of the buffer.

You can assume the *y*'s will be consecutive, except the first one may be greater than zero.

If *D* is 4 or more, you must fill in the unused bytes with zero.

31.53.3.15 fl_draw_image_mono [1/2]

```
void fl_draw_image_mono (
    const uchar * buf,
    int X,
    int Y,
    int W,
    int H,
    int D = 1,
    int L = 0 ) [friend]
```

Draws a gray-scale (1 channel) image.

See also

[fl_draw_image\(const uchar* buf, int X,int Y,int W,int H, int D, int L\)](#)

31.53.3.16 fl_draw_image_mono [2/2]

```
FL_EXPORT void fl_draw_image_mono (
    Fl_Draw_Image_Cb cb,
    void * data,
    int X,
    int Y,
    int W,
    int H,
    int D = 1 ) [friend]
```

Draws a gray-scale image using a callback function to generate image data.

See also

[fl_draw_image\(Fl_Draw_Image_Cb cb, void* data, int X,int Y,int W,int H, int D\)](#)

31.53.3.17 fl_font

```
void fl_font (
    Fl_Font face,
    Fl_Fontsize size ) [friend]
```

Sets the current font, which is then used in various drawing routines.

You may call this outside a draw context if necessary to call `fl_width()`, but on X this will open the display.

The font is identified by a `face` and a `size`. The size of the font is measured in pixels and not "points". Lines should be spaced `size` pixels apart or more.

31.53.3.18 fl_gap

```
void fl_gap ( ) [friend]
```

Call `fl_gap()` to separate loops of the path.

It is unnecessary but harmless to call `fl_gap()` before the first vertex, after the last vertex, or several times in a row.

31.53.3.19 fl_line_style

```
void fl_line_style (
    int style,
    int width = 0,
    char * dashes = 0 ) [friend]
```

Sets how to draw lines (the "pen").

If you change this it is your responsibility to set it back to the default using `fl_line_style(0)`.

Parameters

in	<i>style</i>	A bitmask which is a bitwise-OR of a line style, a cap style, and a join style. If you don't specify a dash type you will get a solid line. If you don't specify a cap or join type you will get a system-defined default of whatever value is fastest.
in	<i>width</i>	The thickness of the lines in pixels. Zero results in the system defined default, which on both X and Windows is somewhat different and nicer than 1.
in	<i>dashes</i>	A pointer to an array of dash lengths, measured in pixels. The first location is how long to draw a solid portion, the next is how long to draw the gap, then the solid, etc. It is terminated with a zero-length entry. A NULL pointer or a zero-length array results in a solid line. Odd array sizes are not supported and result in undefined behavior.

Note

Because of how line styles are implemented on Win32 systems, you *must* set the line style *after* setting the drawing color. If you set the color after the line style you will lose the line style settings.

The `dashes` array does not work under Windows 95, 98 or Me, since those operating systems do not support complex line styles.

31.53.3.20 fl_mult_matrix

```
void fl_mult_matrix (
    double a,
    double b,
    double c,
    double d,
    double x,
    double y ) [friend]
```

Concatenates another transformation onto the current one.

Parameters

in	a,b,c,d,x,y	transformation matrix elements such that $X' = aX + cY + x$ and $Y' = bX + dY + y$
----	---------------	--

31.53.3.21 fl_not_clipped

```
int fl_not_clipped (
    int x,
    int y,
    int w,
    int h ) [friend]
```

Does the rectangle intersect the current clip region?

Parameters

in	x,y,w,h	position and size of rectangle
----	-----------	--------------------------------

Returns

non-zero if any of the rectangle intersects the current clip region. If this returns 0 you don't have to draw the object.

Note

Under X this returns 2 if the rectangle is partially clipped, and 1 if it is entirely inside the clip region.

31.53.3.22 fl_pie

```
void fl_pie (
    int x,
    int y,
    int w,
    int h,
    double a1,
    double a2 ) [friend]
```

Draw filled ellipse sections using integer coordinates.

Like [fl_arc\(\)](#), but [fl_pie\(\)](#) draws a filled-in pie slice. This slice may extend outside the line drawn by [fl_arc\(\)](#); to avoid this use $w - 1$ and $h - 1$.

Parameters

in	x,y,w,h	bounding box of complete circle
in	$a1,a2$	start and end angles of arc measured in degrees counter-clockwise from 3 o'clock. $a2$ must be greater than or equal to $a1$.

31.53.3.23 fl_polygon [1/2]

```
void fl_polygon (
    int x0,
    int y0,
    int x1,
```

```
    int y1,  
    int x2,  
    int y2 ) [friend]
```

Fills a 3-sided polygon.

The polygon must be convex.

31.53.3.24 fl_polygon [2/2]

```
void fl_polygon (  
    int x0,  
    int y0,  
    int x1,  
    int y1,  
    int x2,  
    int y2,  
    int x3,  
    int y3 ) [friend]
```

Fills a 4-sided polygon.

The polygon must be convex.

31.53.3.25 fl_pop_clip

```
void fl_pop_clip ( ) [friend]
```

Restores the previous clip region.

You must call [fl_pop_clip\(\)](#) once for every time you call [fl_push_clip\(\)](#). Unpredictable results may occur if the clip stack is not empty when you return to FLTK.

31.53.3.26 fl_push_clip

```
void fl_push_clip (  
    int x,  
    int y,  
    int w,  
    int h ) [friend]
```

Intersects the current clip region with a rectangle and pushes this new region onto the stack.

Parameters

in	x,y,w,h	position and size
----	-----------	-------------------

31.53.3.27 fl_push_matrix

```
void fl_push_matrix ( ) [friend]
```

Saves the current transformation matrix on the stack.

The maximum depth of the stack is 32.

31.53.3.28 fl_rect

```
void fl_rect (  
    int x,  
    int y,  
    int w,  
    int h ) [friend]
```

Draws a 1-pixel border *inside* the given bounding box.

This function is meant for quick drawing of simple boxes. The behavior is undefined for line widths that are not 1.

31.53.3.29 fl_rotate

```
void fl_rotate (
    double d ) [friend]
```

Concatenates rotation transformation onto the current one.

Parameters

in	<i>d</i>	- rotation angle, counter-clockwise in degrees (not radians)
----	----------	--

31.53.3.30 fl_scale [1/2]

```
void fl_scale (
    double x ) [friend]
```

Concatenates scaling transformation onto the current one.

Parameters

in	<i>x</i>	scale factor in both x-direction and y-direction
----	----------	--

31.53.3.31 fl_scale [2/2]

```
void fl_scale (
    double x,
    double y ) [friend]
```

Concatenates scaling transformation onto the current one.

Parameters

in	<i>x,y</i>	scale factors in x-direction and y-direction
----	------------	--

31.53.3.32 fl_transform_dx

```
double fl_transform_dx (
    double x,
    double y ) [friend]
```

Transforms distance using current transformation matrix.

Parameters

in	<i>x,y</i>	coordinate
----	------------	------------

31.53.3.33 fl_transform_dy

```
double fl_transform_dy (
    double x,
    double y ) [friend]
```

Transforms distance using current transformation matrix.

Parameters

in	<i>x,y</i>	coordinate
----	------------	------------

31.53.3.34 fl_transform_x

```
double fl_transform_x (  
    double x,  
    double y ) [friend]
```

Transforms coordinate using the current transformation matrix.

Parameters

in	<i>x,y</i>	coordinate
----	------------	------------

31.53.3.35 fl_transform_y

```
double fl_transform_y (  
    double x,  
    double y ) [friend]
```

Transforms coordinate using the current transformation matrix.

Parameters

in	<i>x,y</i>	coordinate
----	------------	------------

31.53.3.36 fl_transformed_vertex

```
void fl_transformed_vertex (  
    double xf,  
    double yf ) [friend]
```

Adds coordinate pair to the vertex list without further transformations.

Parameters

in	<i>xf,yf</i>	transformed coordinate
----	--------------	------------------------

31.53.3.37 fl_translate

```
void fl_translate (  
    double x,  
    double y ) [friend]
```

Concatenates translation transformation onto the current one.

Parameters

in	<i>x,y</i>	translation factor in x-direction and y-direction
----	------------	---

31.53.3.38 fl_vertex

```
void fl_vertex (
    double x,
    double y ) [friend]
```

Adds a single vertex to the current path.

Parameters

in	x,y	coordinate
----	-----	------------

The documentation for this class was generated from the following files:

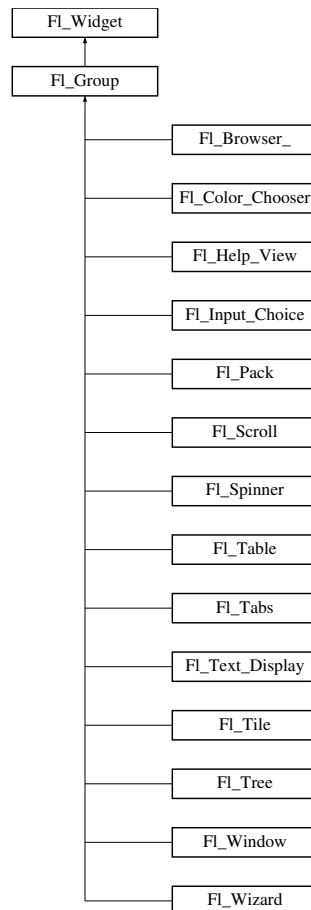
- [Fl_Device.H](#)
- [fl_arc.cxx](#)
- [fl_arci.cxx](#)
- [fl_curve.cxx](#)
- [Fl_Device.cxx](#)
- [Fl_Double_Window.cxx](#)
- [Fl_Image.cxx](#)
- [fl_line_style.cxx](#)
- [fl_rect.cxx](#)
- [fl_vertex.cxx](#)

31.54 Fl_Group Class Reference

The [Fl_Group](#) class is the FLTK container widget.

```
#include <Fl_Group.H>
```

Inheritance diagram for [Fl_Group](#):



Public Member Functions

- [FL_Widget](#) ***& _ddfdesign_kludge** ()
This is for forms compatibility only.
- void **add** ([FL_Widget](#) &)
The widget is removed from its current group (if any) and then added to the end of this group.
- void **add** ([FL_Widget](#) *o)
See void [FL_Group::add\(FL_Widget &w\)](#)
- void **add_resizable** ([FL_Widget](#) &o)
Adds a widget to the group and makes it the resizable widget.
- [FL_Widget](#) *const * **array** () const
Returns a pointer to the array of children.
- virtual [FL_Group](#) * **as_group** ()
Returns an [FL_Group](#) pointer if this widget is an [FL_Group](#).
- void **begin** ()
Sets the current group so you can build the widget tree by just constructing the widgets.
- [FL_Widget](#) * **child** (int n) const
Returns [array\(\)\[n\]](#).
- int **children** () const
Returns how many child widgets the group has.
- void **clear** ()
Deletes all child widgets from memory recursively.
- unsigned int **clip_children** ()
Returns the current clipping mode.

- void `clip_children` (int c)

Controls whether the group widget clips the drawing of child widgets to its bounding box.
- void `end` ()

Exactly the same as `current(this->parent())`.
- int `find` (const `FL_Widget` &o) const

*See int `FL_Group::find(const FL_Widget *w) const`.*
- int `find` (const `FL_Widget` *) const

Searches the child array for the widget and returns the index.
- `FL_Group` (int, int, int, int, const char *s=0)

Creates a new `FL_Group` widget using the given position, size, and label string.
- void `focus` (`FL_Widget` *W)
- void `forms_end` ()

This is for forms compatibility only.
- int `handle` (int)

Handles the specified event.
- void `init_sizes` ()

Resets the internal array of widget sizes and positions.
- void `insert` (`FL_Widget` &, int i)

The widget is removed from its current group (if any) and then inserted into this group.
- void `insert` (`FL_Widget` &o, `FL_Widget` *before)

This does `insert(w, find(before))`.
- void `remove` (`FL_Widget` &)

Removes a widget from the group but does not delete it.
- void `remove` (`FL_Widget` *o)

Removes the widget o from the group.
- void `remove` (int index)

Removes the widget at `index` from the group but does not delete it.
- `FL_Widget` * `resizable` () const

*See void `FL_Group::resizable(FL_Widget *box)`*
- void `resizable` (`FL_Widget` &o)

*See void `FL_Group::resizable(FL_Widget *box)`*
- void `resizable` (`FL_Widget` *o)

The resizable widget defines the resizing box for the group.
- void `resize` (int, int, int, int)

Resizes the `FL_Group` widget and all of its children.
- virtual `~FL_Group` ()

The destructor also deletes all the children.

Static Public Member Functions

- static `FL_Group` * `current` ()

Returns the currently active group.
- static void `current` (`FL_Group` *g)

Sets the current group.

Protected Member Functions

- void `draw` ()
Draws the widget.
- void `draw_child` (`Fl_Widget` &widget) const
Forces a child to redraw.
- void `draw_children` ()
Draws all children of the group.
- void `draw_outside_label` (const `Fl_Widget` &widget) const
Parents normally call this to draw outside labels of child widgets.
- int * `sizes` ()
Returns the internal array of widget sizes and positions.
- void `update_child` (`Fl_Widget` &widget) const
Draws a child only if it needs it.

Additional Inherited Members

31.54.1 Detailed Description

The `Fl_Group` class is the FLTK container widget.

It maintains an array of child widgets. These children can themselves be any widget including `Fl_Group`. The most important subclass of `Fl_Group` is `Fl_Window`, however groups can also be used to control radio buttons or to enforce resize behavior.

The tab and arrow keys are used to move the focus between widgets of this group, and to other groups. The only modifier grabbed is shift (for shift-tab), so that ctrl-tab, alt-up, and such are free for the app to use as shortcuts.

31.54.2 Constructor & Destructor Documentation

31.54.2.1 `Fl_Group()`

```
Fl_Group::Fl_Group (
    int X,
    int Y,
    int W,
    int H,
    const char * l = 0 )
```

Creates a new `Fl_Group` widget using the given position, size, and label string. The default boxtype is `FL_NO_BOX`.

31.54.2.2 `~Fl_Group()`

```
Fl_Group::~Fl_Group ( ) [virtual]
```

The destructor *also deletes all the children*.

This allows a whole tree to be deleted at once, without having to keep a pointer to all the children in the user code. It is allowed that the `Fl_Group` and all of its children are automatic (local) variables, but you must declare the `Fl_Group` *first*, so that it is destroyed last.

If you add static or automatic (local) variables to an `Fl_Group`, then it is your responsibility to remove (or delete) all such static or automatic child widgets *before destroying* the group - otherwise the child widgets' destructors would be called twice!

31.54.3 Member Function Documentation

31.54.3.1 array()

```
Fl_Widget *const * Fl_Group::array ( ) const
```

Returns a pointer to the array of children.

This pointer is only valid until the next time a child is added or removed.

31.54.3.2 as_group()

```
virtual Fl_Group * Fl_Group::as_group ( ) [inline], [virtual]
```

Returns an [Fl_Group](#) pointer if this widget is an [Fl_Group](#).

Use this method if you have a widget (pointer) and need to know whether this widget is derived from [Fl_Group](#). If it returns non-NULL, then the widget in question is derived from [Fl_Group](#), and you can use the returned pointer to access its children or other [Fl_Group](#)-specific methods.

Example:

```
void my_callback (Fl_Widget *w, void *) {
    Fl_Group *g = w->as_group();
    if (g)
        printf ("This group has %d children\n",g->children());
    else
        printf ("This widget is not a group!\n");
}
```

Return values

NULL	if this widget is not derived from Fl_Group .
------	---

Note

This method is provided to avoid `dynamic_cast`.

See also

[Fl_Widget::as_window\(\)](#), [Fl_Widget::as_gl_window\(\)](#)

Reimplemented from [Fl_Widget](#).

31.54.3.3 begin()

```
void Fl_Group::begin ( )
```

Sets the current group so you can build the widget tree by just constructing the widgets.

[begin\(\)](#) is automatically called by the constructor for [Fl_Group](#) (and thus for [Fl_Window](#) as well). [begin\(\)](#) is *exactly the same as* `current(this)`. *Don't forget to [end\(\)](#) the group or window!*

31.54.3.4 child()

```
Fl_Widget * Fl_Group::child (
    int n ) const [inline]
```

Returns [array\(\)](#)[n].

No range checking is done!

31.54.3.5 clear()

```
void Fl_Group::clear ( )
```

Deletes all child widgets from memory recursively.

This method differs from the [remove\(\)](#) method in that it affects all child widgets and deletes them from memory.

31.54.3.6 clip_children() [1/2]

```
unsigned int Fl_Group::clip_children ( ) [inline]
```

Returns the current clipping mode.

Returns

true, if clipping is enabled, false otherwise.

See also

void [Fl_Group::clip_children\(int c\)](#)

31.54.3.7 clip_children() [2/2]

```
void Fl_Group::clip_children (
    int c ) [inline]
```

Controls whether the group widget clips the drawing of child widgets to its bounding box.

Set *c* to 1 if you want to clip the child widgets to the bounding box.

The default is to not clip (0) the drawing of child widgets.

31.54.3.8 current() [1/2]

```
Fl_Group * Fl_Group::current ( ) [static]
```

Returns the currently active group.

The [Fl_Widget](#) constructor automatically does [current\(\)](#)->[add\(widget\)](#) if this is not null. To prevent new widgets from being added to a group, call [Fl_Group::current\(0\)](#).

31.54.3.9 current() [2/2]

```
void Fl_Group::current (
    Fl_Group * g ) [static]
```

Sets the current group.

See also

[Fl_Group::current\(\)](#)

31.54.3.10 draw()

```
void Fl_Group::draw ( ) [protected], [virtual]
```

Draws the widget.

Never call this function directly. FLTK will schedule redrawing whenever needed. If your widget must be redrawn as soon as possible, call [redraw\(\)](#) instead.

Override this function to draw your own widgets.

If you ever need to call another widget's draw method *from within your own [draw\(\)](#) method*, e.g. for an embedded scrollbar, you can do it (because [draw\(\)](#) is virtual) like this:

```
Fl_Widget *s = &scroll;           // scroll is an embedded Fl_Scrollbar
s->draw();                         // calls Fl_Scrollbar::draw()
```

Implements [Fl_Widget](#).

Reimplemented in [Fl_Help_View](#), [Fl_Pack](#), [Fl_Scroll](#), [Fl_Tabs](#), [Fl_Text_Display](#), [Fl_Tree](#), [Fl_Window](#), [Fl_Glut_Window](#), and [Fl_Table](#).

31.54.3.11 draw_child()

```
void Fl_Group::draw_child (
    Fl_Widget & widget ) const [protected]
```

Forces a child to redraw.

This draws a child widget, if it is not clipped. The damage bits are cleared after drawing.

31.54.3.12 draw_children()

```
void Fl_Group::draw_children ( ) [protected]
```

Draws all children of the group.

This is useful, if you derived a widget from [Fl_Group](#) and want to draw a special border or background. You can call [draw_children\(\)](#) from the derived [draw\(\)](#) method after drawing the box, border, or background.

31.54.3.13 end()

```
void Fl_Group::end ( )
```

Exactly the same as [current\(this->parent\(\)\)](#).

Any new widgets added to the widget tree will be added to the parent of the group.

31.54.3.14 find()

```
int Fl_Group::find (
    const Fl_Widget * o ) const
```

Searches the child array for the widget and returns the index.

Returns [children\(\)](#) if the widget is NULL or not found.

31.54.3.15 focus()

```
void Fl_Group::focus (
    Fl_Widget * W ) [inline]
```

Deprecated This is for backwards compatibility only. You should use *W->take_focus()* instead.

See also

[Fl_Widget::take_focus\(\)](#);

31.54.3.16 handle()

```
int Fl_Group::handle (
    int event ) [virtual]
```

Handles the specified event.

You normally don't call this method directly, but instead let FLTK do it when the user interacts with the widget.

When implemented in a widget, this function must return 0 if the widget does not use the event or 1 otherwise.

Most of the time, you want to call the inherited [handle\(\)](#) method in your overridden method so that you don't short-circuit events that you don't handle. In this last case you should return the callee retval.

Parameters

in	<i>event</i>	the kind of event received
----	--------------	----------------------------

Return values

0	if the event was not used or understood
1	if the event was used and can be deleted

See also

[Fl_Event](#)

Reimplemented from [Fl_Widget](#).

Reimplemented in [Fl_Table](#), [Fl_Text_Display](#), [Fl_Text_Editor](#), [Fl_Tree](#), [Fl_Spinner](#), [Fl_Table_Row](#), [Fl_Tile](#), [Fl_Help_View](#), [Fl_Scroll](#), [Fl_Tabs](#), [Fl_Window](#), and [Fl_Glut_Window](#).

31.54.3.17 `init_sizes()`

```
void Fl_Group::init_sizes ( )
```

Resets the internal array of widget sizes and positions.

The `Fl_Group` widget keeps track of the original widget sizes and positions when resizing occurs so that if you resize a window back to its original size the widgets will be in the correct places. If you rearrange the widgets in your group, call this method to register the new arrangement with the `Fl_Group` that contains them.

If you add or remove widgets, this will be done automatically.

Note

The internal array of widget sizes and positions will be allocated and filled when the next `resize()` occurs.

See also

[sizes\(\)](#)

31.54.3.18 `insert()` [1/2]

```
void Fl_Group::insert (
    Fl_Widget & o,
    int index )
```

The widget is removed from its current group (if any) and then inserted into this group.

It is put at index `n` - or at the end, if `n >= children()`. This can also be used to rearrange the widgets inside a group.

31.54.3.19 `insert()` [2/2]

```
void Fl_Group::insert (
    Fl_Widget & o,
    Fl_Widget * before ) [inline]
```

This does `insert(w, find(before))`.

This will append the widget if `before` is not in the group.

31.54.3.20 `remove()` [1/3]

```
void Fl_Group::remove (
    Fl_Widget & o )
```

Removes a widget from the group but does not delete it.

This method does nothing if the widget is not a child of the group.

This method differs from the `clear()` method in that it only affects a single widget and does not delete it from memory.

Note

If you have the child's index anyway, use `remove(int index)` instead, because this doesn't need a child lookup in the group's table of children. This can be much faster, if there are lots of children.

31.54.3.21 `remove()` [2/3]

```
void Fl_Group::remove (
    Fl_Widget * o ) [inline]
```

Removes the widget `o` from the group.

See also

`void remove(Fl_Widget&)`

31.54.3.22 remove() [3/3]

```
void Fl_Group::remove (
    int index )
```

Removes the widget at `index` from the group but does not delete it.

This method does nothing if `index` is out of bounds.

This method differs from the [clear\(\)](#) method in that it only affects a single widget and does not delete it from memory.

Since

FLTK 1.3.0

31.54.3.23 resizable()

```
void Fl_Group::resizable (
    Fl_Widget * o ) [inline]
```

The resizable widget defines the resizing box for the group.

When the group is resized it calculates a new size and position for all of its children. Widgets that are horizontally or vertically inside the dimensions of the box are scaled to the new size. Widgets outside the box are moved.

In these examples the gray area is the resizable:

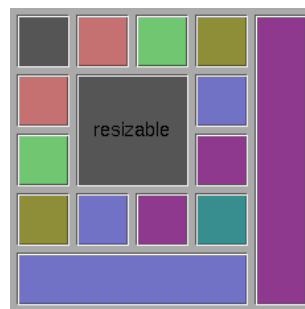


Figure 31.14 before resize



Figure 31.15 after resize

The resizable may be set to the group itself, in which case all the contents are resized. This is the default value for [Fl_Group](#), although NULL is the default for [Fl_Window](#) and [Fl_Pack](#).

If the resizable is NULL then all widgets remain a fixed size and distance from the top-left corner.

It is possible to achieve any type of resize behavior by using an invisible [Fl_Box](#) as the resizable and/or by using a hierarchy of child [Fl_Group](#)'s.

31.54.3.24 `resize()`

```
void Fl_Group::resize (
    int X,
    int Y,
    int W,
    int H ) [virtual]
```

Resizes the `Fl_Group` widget and all of its children.

The `Fl_Group` widget first resizes itself, and then it moves and resizes all its children according to the rules documented for `Fl_Group::resizable(Fl_Widget*)`

See also

[Fl_Group::resizable\(Fl_Widget*\)](#)

[Fl_Group::resizable\(\)](#)

[Fl_Widget::resize\(int,int,int,int\)](#)

Reimplemented from `Fl_Widget`.

Reimplemented in `Fl_Input_Choice`, `Fl_Scroll`, `Fl_Spinner`, `Fl_Table`, `Fl_Text_Display`, `Fl_Tile`, `Fl_Window`, `Fl_Help_View`, `Fl_Overlay_Window`, and `Fl_Tree`.

31.54.3.25 `sizes()`

```
int * Fl_Group::sizes ( ) [protected]
```

Returns the internal array of widget sizes and positions.

If the `sizes()` array does not exist, it will be allocated and filled with the current widget sizes and positions.

Note

You should never need to use this method directly, unless you have special needs to rearrange the children of a `Fl_Group`. `Fl_Tile` uses this to rearrange its widget positions.

See also

[init_sizes\(\)](#)

Todo Should the internal representation of the `sizes()` array be documented?

31.54.3.26 `update_child()`

```
void Fl_Group::update_child (
    Fl_Widget & widget ) const [protected]
```

Draws a child only if it needs it.

This draws a child widget, if it is not clipped *and* if any `damage()` bits are set. The damage bits are cleared after drawing.

See also

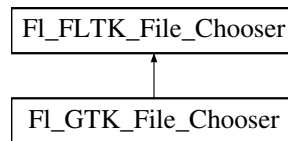
[Fl_Group::draw_child\(Fl_Widget& widget\) const](#)

The documentation for this class was generated from the following files:

- `Fl_Group.H`
- `Fl_Group.cxx`
- `forms_compatibility.cxx`

31.55 FI_GTK_File_Chooser Class Reference

Inheritance diagram for FI_GTK_File_Chooser:



Friends

- class `FI_Native_File_Chooser`

Additional Inherited Members

The documentation for this class was generated from the following files:

- [FI_Native_File_Chooser.H](#)
- [FI_Native_File_Chooser_GTK.cxx](#)

31.56 FI_Help_Block Struct Reference

Public Attributes

- `FI_Color` **bgcolor**
- `uchar` **border**
- `const char *` **end**
- `int` **h**
- `int` **line** [32]
- `const char *` **start**
- `int` **w**
- `int` **x**
- `int` **y**

The documentation for this struct was generated from the following file:

- [FI_Help_View.H](#)

31.57 FI_Help_Dialog Class Reference

The [FI_Help_Dialog](#) widget displays a standard help dialog window using the [FI_Help_View](#) widget.

Public Member Functions

- `FI_Help_Dialog ()`
The constructor creates the dialog pictured above.
- `int h ()`
Returns the position and size of the help dialog.
- `void hide ()`
Hides the [FI_Help_Dialog](#) window.
- `void load (const char *f)`
Loads the specified HTML file into the [FI_Help_View](#) widget.
- `void position (int xx, int yy)`
Set the screen position of the dialog.

- void **resize** (int xx, int yy, int ww, int hh)
Change the position and size of the dialog.
- void **show** ()
Shows the [FI_Help_Dialog](#) window.
- void **show** (int argc, char **argv)
Shows the main Help Dialog Window Delegates call to encapsulated window_ void [FI_Window::show\(int argc, char **argv\)](#) instance method.
- [FI_Fontsize](#) **textsize** ()
Sets or gets the default text size for the help view.
- void **textsize** ([FI_Fontsize](#) s)
Sets or gets the default text size for the help view.
- void **topline** (const char *n)
Sets the top line in the [FI_Help_View](#) widget to the named or numbered line.
- void **topline** (int n)
Sets the top line in the [FI_Help_View](#) widget to the named or numbered line.
- const char * **value** () const
The first form sets the current buffer to the string provided and reformats the text.
- void **value** (const char *f)
The first form sets the current buffer to the string provided and reformats the text.
- int **visible** ()
Returns 1 if the [FI_Help_Dialog](#) window is visible.
- int **w** ()
Returns the position and size of the help dialog.
- int **x** ()
Returns the position and size of the help dialog.
- int **y** ()
Returns the position and size of the help dialog.
- ~[FI_Help_Dialog](#) ()
The destructor destroys the widget and frees all memory that has been allocated for the current file.

31.57.1 Detailed Description

The [FI_Help_Dialog](#) widget displays a standard help dialog window using the [FI_Help_View](#) widget.

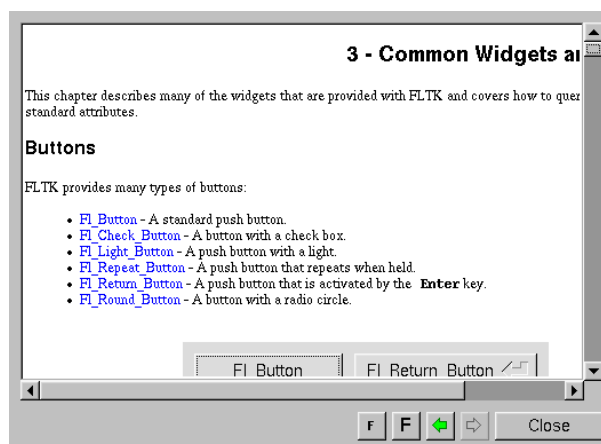


Figure 31.16 [FI_Help_Dialog](#)

31.57.2 Member Function Documentation

31.57.2.1 load()

```
void Fl_Help_Dialog::load (
    const char * f )
```

Loads the specified HTML file into the [Fl_Help_View](#) widget.
The filename can also contain a target name ("filename.html#target").

31.57.2.2 show()

```
void Fl_Help_Dialog::show ( )
```

Shows the [Fl_Help_Dialog](#) window.
Shows the main Help Dialog Window Delegates call to encapsulated window_ void [Fl_Window::show\(\)](#) method.

31.57.2.3 textsize()

```
void Fl_Help_Dialog::textsize (
    Fl_Fontsize s )
```

Sets or gets the default text size for the help view.
Sets the internal [Fl_Help_View](#) instance text size.
Delegates call to encapsulated view_ void [Fl_Help_View::textsize\(Fl_Fontsize s\)](#) instance method

31.57.2.4 value() [1/2]

```
const char * Fl_Help_Dialog::value ( ) const
```

The first form sets the current buffer to the string provided and reformats the text.
It also clears the history of the "back" and "forward" buttons. The second form returns the current buffer contents.

31.57.2.5 value() [2/2]

```
void Fl_Help_Dialog::value (
    const char * v )
```

The first form sets the current buffer to the string provided and reformats the text.
It also clears the history of the "back" and "forward" buttons. The second form returns the current buffer contents.
The documentation for this class was generated from the following files:

- [Fl_Help_Dialog.H](#)
- [Fl_Help_Dialog.cxx](#)
- [Fl_Help_Dialog_Dox.cxx](#)

31.58 Fl_Help_Font_Stack Struct Reference

Public Member Functions

- `size_t count () const`
Gets the current count of font style elements in the stack.
- `Fl_Help_Font_Stack ()`
font stack construction, initialize attributes.
- `void init (Fl_Font f, Fl_Fontsize s, Fl_Color c)`
- `void pop (Fl_Font &f, Fl_Fontsize &s, Fl_Color &c)`
Pops from the stack the font style triplet and calls [fl_font\(\)](#) & [fl_color\(\)](#) adequately.
- `void push (Fl_Font f, Fl_Fontsize s, Fl_Color c)`
Pushes the font style triplet on the stack, also calls [fl_font\(\)](#) & [fl_color\(\)](#) adequately.
- `void top (Fl_Font &f, Fl_Fontsize &s, Fl_Color &c)`
Gets the top (current) element on the stack.

Protected Attributes

- [FI_Help_Font_Style](#) **elts_** [100]
font elements
- **size_t** **nfonts_**
current number of fonts in stack

The documentation for this struct was generated from the following file:

- FI_Help_View.H

31.59 FI_Help_Font_Style Struct Reference

[FI_Help_View](#) font stack element definition.

```
#include <FI_Help_View.H>
```

Public Member Functions

- **FI_Help_Font_Style** ([FI_Font](#) afont, [FI_Fontsize](#) asize, [FI_Color](#) acolor)
- void **get** ([FI_Font](#) &afont, [FI_Fontsize](#) &asize, [FI_Color](#) &acolor)
Gets current font attributes.
- void **set** ([FI_Font](#) afont, [FI_Fontsize](#) asize, [FI_Color](#) acolor)
Sets current font attributes.

Public Attributes

- [FI_Color](#) **c**
Font Color.
- [FI_Font](#) **f**
Font.
- [FI_Fontsize](#) **s**
Font Size.

31.59.1 Detailed Description

[FI_Help_View](#) font stack element definition.

The documentation for this struct was generated from the following file:

- FI_Help_View.H

31.60 FI_Help_Link Struct Reference

Definition of a link for the html viewer.

```
#include <FI_Help_View.H>
```

Public Attributes

- char **filename** [192]
Reference filename.
- int **h**
Height of link text.
- char **name** [32]
Link target (blank if none)
- int **w**
Width of link text.

- int **x**
X offset of link text.
- int **y**
Y offset of link text.

31.60.1 Detailed Description

Definition of a link for the html viewer.

The documentation for this struct was generated from the following file:

- `Fl_Help_View.H`

31.61 `Fl_Help_Target` Struct Reference

`Fl_Help_Target` structure.

```
#include <Fl_Help_View.H>
```

Public Attributes

- char **name** [32]
Target name.
- int **y**
Y offset of target.

31.61.1 Detailed Description

`Fl_Help_Target` structure.

The documentation for this struct was generated from the following file:

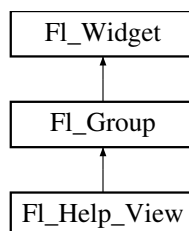
- `Fl_Help_View.H`

31.62 `Fl_Help_View` Class Reference

The `Fl_Help_View` widget displays HTML text.

```
#include <Fl_Help_View.H>
```

Inheritance diagram for `Fl_Help_View`:



Public Member Functions

- void **clear_selection** ()
Removes the current text selection.
- const char * **directory** () const
Returns the current directory for the text in the buffer.
- const char * **filename** () const
Returns the current filename for the text in the buffer.
- int **find** (const char *s, int p=0)

- Finds the specified string *s* at starting position *p*.*

 - **FI_Help_View** (int xx, int yy, int ww, int hh, const char *l=0)
 - The constructor creates the [FI_Help_View](#) widget at the specified position and size.*
 - int **handle** (int)
 - Handles events in the widget.*
 - int **leftline** () const
 - Gets the left position in pixels.*
 - void **leftline** (int)
 - Scrolls the text to the indicated position, given a pixel column.*
 - void **link** (FI_Help_Func *fn)
 - This method assigns a callback function to use when a link is followed or a file is loaded (via [FI_Help_View::load\(\)](#)) that requires a different file or path.*
 - int **load** (const char *f)
 - Loads the specified file.*
 - void **resize** (int, int, int, int)
 - Resizes the help widget.*
 - int **scrollbar_size** () const
 - Gets the current size of the scrollbars' troughs, in pixels.*
 - void **scrollbar_size** (int newSize)
 - Sets the pixel size of the scrollbars' troughs to *newSize*, in pixels.*
 - void **select_all** ()
 - Selects all the text in the view.*
 - int **size** () const
 - Gets the size of the help view.*
 - void **size** (int W, int H)
 - Sets the size of the help view.*
 - **FI_Color textcolor** () const
 - Returns the current default text color.*
 - void **textcolor** (FI_Color c)
 - Sets the default text color.*
 - **FI_Font textfont** () const
 - Returns the current default text font.*
 - void **textfont** (FI_Font f)
 - Sets the default text font.*
 - **FI_Fontsize textsize** () const
 - Gets the default text size.*
 - void **textsize** (FI_Fontsize s)
 - Sets the default text size.*
 - const char * **title** ()
 - Returns the current document title, or NULL if there is no title.*
 - int **topline** () const
 - Returns the current top line in pixels.*
 - void **topline** (const char *n)
 - Scrolls the text to the indicated position, given a named destination.*
 - void **topline** (int)
 - Scrolls the text to the indicated position, given a pixel line.*
 - const char * **value** () const
 - Returns the current buffer contents.*
 - void **value** (const char *val)
 - Sets the current help text buffer to the string provided and reformats the text.*
 - **~FI_Help_View** ()
 - Destroys the [FI_Help_View](#) widget.*

Protected Member Functions

- void `draw` ()
Draws the `Fl_Help_View` widget.

Additional Inherited Members

31.62.1 Detailed Description

The `Fl_Help_View` widget displays HTML text.

Most HTML 2.0 elements are supported, as well as a primitive implementation of tables. GIF, JPEG, and PNG images are displayed inline.

Supported HTML tags:

- A: HREF/NAME
- B
- BODY: BGCOLOR/TEXT/LINK
- BR
- CENTER
- CODE
- DD
- DL
- DT
- EM
- FONT: COLOR/SIZE/FACE=(helvetica/arial/sans/times/serif/symbol/courier)
- H1/H2/H3/H4/H5/H6
- HEAD
- HR
- I
- IMG: SRC/WIDTH/HEIGHT/ALT
- KBD
- LI
- OL
- P
- PRE
- STRONG
- TABLE: TH/TD/TR/BORDER/BGCOLOR/COLSPAN/ALIGN=CENTER|RIGHT|LEFT
- TITLE
- TT
- U
- UL
- VAR

Supported color names:

- black,red,green,yellow,blue,magenta,fuchsia,cyan,aqua,white,gray,grey,lime,maroon,navy,olive,purple,silver,teal.

Supported urls:

- Internal: file:
- External: http: ftp: https: ipp: mailto: news:

Quoted char names:

- Aacute aacute Acirc acirc acute AElig aelig Agrave agrave amp Aring aring Atilde atilde Auml auml
- brvbar bull
- Ccedil ccedil cedil cent copy curren
- deg divide
- Eacute eacute Ecirc ecirc Egrave egrave ETH eth Euml euml euro
- frac12 frac14 frac34
- gt
- Iacute iacute Icirc icirc iexcl Igrave igrave iquest Iuml iuml
- laquo lt
- macr micro middot
- nbsp not Ntilde ntilde
- Oacute oacute Ocirc ocirc Ograve ograve ordf ordm Oslash oslash Otilde otilde Ouml ouml
- para permil plusmn pound
- quot
- raquo reg
- sect shy sup1 sup2 sup3 szlig
- THORN thorn times trade
- Uacute uacute Ucirc ucirc Ugrave ugrave uml Uuml uuml
- Yacute yacute
- yen Yuml yuml

31.62.2 Constructor & Destructor Documentation

31.62.2.1 ~Fl_Help_View()

`Fl_Help_View::~Fl_Help_View ()`

Destroys the [Fl_Help_View](#) widget.

The destructor destroys the widget and frees all memory that has been allocated for the current document.

31.62.3 Member Function Documentation

31.62.3.1 draw()

```
void Fl_Help_View::draw (
    void ) [protected], [virtual]
```

Draws the [Fl_Help_View](#) widget.
Reimplemented from [Fl_Group](#).

31.62.3.2 find()

```
int Fl_Help_View::find (
    const char * s,
    int p = 0 )
```

Finds the specified string *s* at starting position *p*.

Returns

the matching position or -1 if not found

31.62.3.3 handle()

```
int Fl_Help_View::handle (
    int event ) [virtual]
```

Handles events in the widget.
Reimplemented from [Fl_Group](#).

31.62.3.4 leftline()

```
void Fl_Help_View::leftline (
    int left )
```

Scrolls the text to the indicated position, given a pixel column.
If the given pixel value *left* is out of range, then the text is scrolled to the left or right side of the document, resp.

Parameters

<i>in</i>	<i>left</i>	left column number in pixels (0 = left side)
-----------	-------------	--

31.62.3.5 link()

```
void Fl_Help_View::link (
    Fl_Help_Func * fn ) [inline]
```

This method assigns a callback function to use when a link is followed or a file is loaded (via [Fl_Help_View::load\(\)](#)) that requires a different file or path.

The callback function receives a pointer to the [Fl_Help_View](#) widget and the URI or full pathname for the file in question. It must return a pathname that can be opened as a local file or NULL:

```
const char *fn(Fl_Widget *w, const char *uri);
```

The link function can be used to retrieve remote or virtual documents, returning a temporary file that contains the actual data. If the link function returns NULL, the value of the [Fl_Help_View](#) widget will remain unchanged.

If the link callback cannot handle the URI scheme, it should return the uri value unchanged or set the [value\(\)](#) of the widget before returning NULL.

31.62.3.6 load()

```
int Fl_Help_View::load (
    const char * f )
```

Loads the specified file.

This method loads the specified file or URL.

31.62.3.7 `resize()`

```
void Fl_Help_View::resize (
    int xx,
    int yy,
    int ww,
    int hh ) [virtual]
```

Resizes the help widget.

Reimplemented from [Fl_Group](#).

31.62.3.8 `scrollbar_size()` [1/2]

```
int Fl_Help_View::scrollbar_size ( ) const [inline]
```

Gets the current size of the scrollbars' troughs, in pixels.

If this value is zero (default), this widget will use the [Fl::scrollbar_size\(\)](#) value as the scrollbar's width.

Returns

Scrollbar size in pixels, or 0 if the global [Fl::scrollbar_size\(\)](#) is being used.

See also

[Fl::scrollbar_size\(int\)](#)

31.62.3.9 `scrollbar_size()` [2/2]

```
void Fl_Help_View::scrollbar_size (
    int newSize ) [inline]
```

Sets the pixel size of the scrollbars' troughs to `newSize`, in pixels.

Normally you should not need this method, and should use [Fl::scrollbar_size\(int\)](#) instead to manage the size of ALL your widgets' scrollbars. This ensures your application has a consistent UI, is the default behavior, and is normally what you want.

Only use THIS method if you really need to override the global scrollbar size. The need for this should be rare.

Setting `newSize` to the special value of 0 causes the widget to track the global [Fl::scrollbar_size\(\)](#), which is the default.

Parameters

in	<i>newSize</i>	Sets the scrollbar size in pixels. If 0 (default), scrollbar size tracks the global Fl::scrollbar_size()
----	----------------	---

See also

[Fl::scrollbar_size\(\)](#)

31.62.3.10 `topline()` [1/2]

```
void Fl_Help_View::topline (
    const char * n )
```

Scrolls the text to the indicated position, given a named destination.

Parameters

in	<i>n</i>	target name
----	----------	-------------

31.62.3.11 topline() [2/2]

```
void Fl_Help_View::topline (
    int top )
```

Scrolls the text to the indicated position, given a pixel line.

If the given pixel value `top` is out of range, then the text is scrolled to the top or bottom of the document, resp.

Parameters

<code>in</code>	<code>top</code>	top line number in pixels (0 = start of document)
-----------------	------------------	---

31.62.3.12 value()

```
void Fl_Help_View::value (
    const char * val )
```

Sets the current help text buffer to the string provided and reformats the text.

The provided character string `val` is copied internally and will be freed when `value()` is called again, or when the widget is destroyed.

If `val` is NULL, then the widget is cleared.

The documentation for this class was generated from the following files:

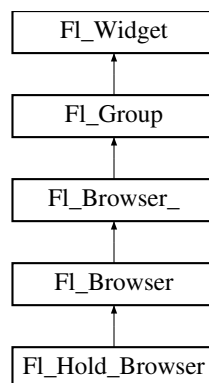
- `Fl_Help_View.H`
- `Fl_Help_View.cxx`

31.63 FI_Hold_Browser Class Reference

The `FI_Hold_Browser` is a subclass of `FI_Browser` which lets the user select a single item, or no items by clicking on the empty space.

```
#include <Fl_Hold_Browser.H>
```

Inheritance diagram for `FI_Hold_Browser`:

**Public Member Functions**

- `FI_Hold_Browser` (int X, int Y, int W, int H, const char *L=0)
Creates a new `FI_Hold_Browser` widget using the given position, size, and label string.

Additional Inherited Members

31.63.1 Detailed Description

The [FI_Hold_Browser](#) is a subclass of [FI_Browser](#) which lets the user select a single item, or no items by clicking on the empty space.

As long as the mouse button is held down the item pointed to by it is highlighted, and this highlighting remains on when the mouse button is released. Normally the callback is done when the user releases the mouse, but you can change this with [when\(\)](#).

See [FI_Browser](#) for methods to add and remove lines from the browser.

31.63.2 Constructor & Destructor Documentation

31.63.2.1 FI_Hold_Browser()

```
FI_Hold_Browser::FI_Hold_Browser (
    int X,
    int Y,
    int W,
    int H,
    const char * L = 0 )
```

Creates a new [FI_Hold_Browser](#) widget using the given position, size, and label string.

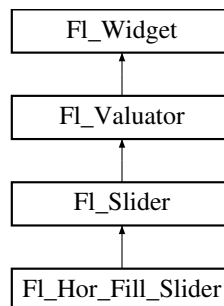
The default boxtype is FL_DOWN_BOX. The constructor specializes [FI_Browser\(\)](#) by setting the type to FL_↔ HOLD_BROWSER. The destructor destroys the widget and frees all memory that has been allocated.

The documentation for this class was generated from the following files:

- FI_Hold_Browser.H
- FI_Browser.cxx

31.64 FI_Hor_Fill_Slider Class Reference

Inheritance diagram for FI_Hor_Fill_Slider:



Public Member Functions

- [FI_Hor_Fill_Slider](#) (int X, int Y, int W, int H, const char *L=0)

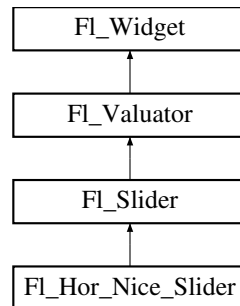
Additional Inherited Members

The documentation for this class was generated from the following files:

- FI_Hor_Fill_Slider.H
- FI_Slider.cxx

31.65 FI_Hor_Nice_Slider Class Reference

Inheritance diagram for FI_Hor_Nice_Slider:



Public Member Functions

- **FI_Hor_Nice_Slider** (int X, int Y, int W, int H, const char *L=0)

Additional Inherited Members

The documentation for this class was generated from the following files:

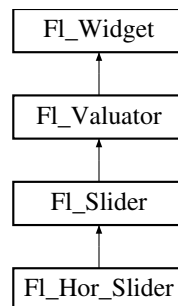
- FI_Hor_Nice_Slider.H
- FI_Slider.cxx

31.66 FI_Hor_Slider Class Reference

Horizontal Slider class.

```
#include <FI_Hor_Slider.H>
```

Inheritance diagram for FI_Hor_Slider:



Public Member Functions

- **FI_Hor_Slider** (int X, int Y, int W, int H, const char *l=0)
Creates a new [FI_Hor_Slider](#) widget using the given position, size, and label string.

Additional Inherited Members

31.66.1 Detailed Description

Horizontal Slider class.

See also

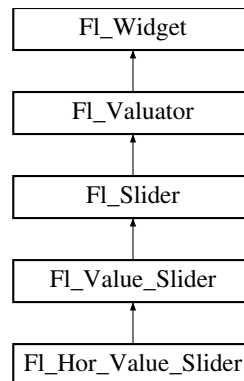
class [Fl_Slider](#).

The documentation for this class was generated from the following files:

- Fl_Hor_Slider.H
- Fl_Slider.cxx

31.67 Fl_Hor_Value_Slider Class Reference

Inheritance diagram for Fl_Hor_Value_Slider:



Public Member Functions

- **Fl_Hor_Value_Slider** (int X, int Y, int W, int H, const char *l=0)

Additional Inherited Members

The documentation for this class was generated from the following files:

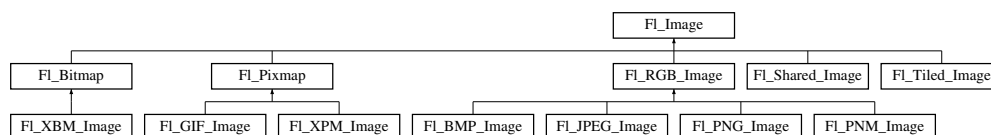
- Fl_Hor_Value_Slider.H
- Fl_Value_Slider.cxx

31.68 Fl_Image Class Reference

Base class for image caching and drawing.

```
#include <Fl_Image.H>
```

Inheritance diagram for Fl_Image:



Public Member Functions

- virtual void [color_average](#) ([Fl_Color](#) c, float i)
The [color_average\(\)](#) method averages the colors in the image with the FLTK color value c.
- [Fl_Image](#) * [copy](#) ()
The [copy\(\)](#) method creates a copy of the specified image.
- virtual [Fl_Image](#) * [copy](#) (int W, int H)

- The *copy()* method creates a copy of the specified image.
- int **count** () const

The *count()* method returns the number of data values associated with the image.
- int **d** () const

Returns the current image depth.
- const char *const * **data** () const

Returns a pointer to the current image data array.
- virtual void **desaturate** ()

The *desaturate()* method converts an image to grayscale.
- void **draw** (int X, int Y)

Draws the image.
- virtual void **draw** (int X, int Y, int W, int H, int cx=0, int cy=0)

Draws the image with a bounding box.
- int **fail** ()

Returns a value that is not 0 if there is currently no image available.
- **FI_Image** (int W, int H, int D)

The constructor creates an empty image with the specified width, height, and depth.
- int **h** () const

Returns the current image height in pixels.
- void **inactive** ()

The *inactive()* method calls *color_average(FI_BACKGROUND_COLOR, 0.33f)* to produce an image that appears grayed out.
- virtual void **label** (**FI_Menu_Item** *m)

The *label()* methods are an obsolete way to set the image attribute of a widget or menu item.
- virtual void **label** (**FI_Widget** *w)

The *label()* methods are an obsolete way to set the image attribute of a widget or menu item.
- int **ld** () const

Returns the current line data size in bytes.
- virtual void **uncache** ()

If the image has been cached for display, delete the cache data.
- int **w** () const

Returns the current image width in pixels.
- virtual ~**FI_Image** ()

The destructor is a virtual method that frees all memory used by the image.

Static Public Member Functions

- static **FI_RGB_Scaling** **RGB_scaling** ()

Returns the currently used RGB image scaling method.
- static void **RGB_scaling** (**FI_RGB_Scaling**)

Sets the RGB image scaling method used for *copy(int, int)*.

Static Public Attributes

- static const int **ERR_FILE_ACCESS** = -2
- static const int **ERR_FORMAT** = -3
- static const int **ERR_NO_IMAGE** = -1

Protected Member Functions

- void **d** (int D)
Sets the current image depth.
- void **data** (const char *const *p, int c)
Sets the current array pointer and count of pointers in the array.
- void **draw_empty** (int X, int Y)
The protected method [draw_empty\(\)](#) draws a box with an X in it.
- void **h** (int H)
Sets the current image height in pixels.
- void **ld** (int LD)
Sets the current line data size in bytes.
- void **w** (int W)
Sets the current image width in pixels.

Static Protected Member Functions

- static void **labeltype** (const [Fl_Label](#) *lo, int lx, int ly, int lw, int lh, [Fl_Align](#) la)
- static void **measure** (const [Fl_Label](#) *lo, int &lw, int &lh)

31.68.1 Detailed Description

Base class for image caching and drawing.

[Fl_Image](#) is the base class used for caching and drawing all kinds of images in FLTK. This class keeps track of common image data such as the pixels, colormap, width, height, and depth. Virtual methods are used to provide type-specific image handling.

Since the [Fl_Image](#) class does not support image drawing by itself, calling the [draw\(\)](#) method results in a box with an X in it being drawn instead.

31.68.2 Constructor & Destructor Documentation

31.68.2.1 Fl_Image()

```
Fl_Image::Fl_Image (
    int W,
    int H,
    int D )
```

The constructor creates an empty image with the specified width, height, and depth.

The width and height are in pixels. The depth is 0 for bitmaps, 1 for pixmap (colormap) images, and 1 to 4 for color images.

31.68.3 Member Function Documentation

31.68.3.1 color_average()

```
void Fl_Image::color_average (
    Fl\_Color c,
    float i ) [virtual]
```

The [color_average\(\)](#) method averages the colors in the image with the FLTK color value c.

The i argument specifies the amount of the original image to combine with the color, so a value of 1.0 results in no color blend, and a value of 0.0 results in a constant image of the specified color.

An internal copy is made of the original image before changes are applied, to avoid modifying the original image.

Reimplemented in [Fl_RGB_Image](#), [Fl_Pixmap](#), [Fl_Shared_Image](#), and [Fl_Tiled_Image](#).

31.68.3.2 copy() [1/2]

```
Fl_Image * Fl_Image::copy ( ) [inline]
```

The `copy()` method creates a copy of the specified image.

If the width and height are provided, the image is resized to the specified size. The image should be deleted (or in the case of `Fl_Shared_Image`, released) when you are done with it.

31.68.3.3 copy() [2/2]

```
Fl_Image * Fl_Image::copy (
    int W,
    int H ) [virtual]
```

The `copy()` method creates a copy of the specified image.

If the width and height are provided, the image is resized to the specified size. The image should be deleted (or in the case of `Fl_Shared_Image`, released) when you are done with it.

Reimplemented in `Fl_Bitmap`, `Fl_RGB_Image`, `Fl_Pixmap`, `Fl_Shared_Image`, and `Fl_Tiled_Image`.

31.68.3.4 count()

```
int Fl_Image::count ( ) const [inline]
```

The `count()` method returns the number of data values associated with the image.

The value will be 0 for images with no associated data, 1 for bitmap and color images, and greater than 2 for pixmap images.

31.68.3.5 d()

```
int Fl_Image::d ( ) const [inline]
```

Returns the current image depth.

The return value will be 0 for bitmaps, 1 for pixmaps, and 1 to 4 for color images.

31.68.3.6 data()

```
const char *const * Fl_Image::data ( ) const [inline]
```

Returns a pointer to the current image data array.

Use the `count()` method to find the size of the data array.

31.68.3.7 desaturate()

```
void Fl_Image::desaturate ( ) [virtual]
```

The `desaturate()` method converts an image to grayscale.

If the image contains an alpha channel (depth = 4), the alpha channel is preserved.

An internal copy is made of the original image before changes are applied, to avoid modifying the original image.

Reimplemented in `Fl_RGB_Image`, `Fl_Pixmap`, `Fl_Shared_Image`, and `Fl_Tiled_Image`.

31.68.3.8 draw() [1/2]

```
void Fl_Image::draw (
    int X,
    int Y ) [inline]
```

Draws the image.

This form specifies the upper-lefthand corner of the image.

31.68.3.9 draw() [2/2]

```
void Fl_Image::draw (
    int X,
    int Y,
    int W,
```



```

    int H,
    int cx = 0,
    int cy = 0 ) [virtual]

```

Draws the image with a bounding box.

Arguments *X*, *Y*, *W*, *H* specify a bounding box for the image, with the origin (upper-left corner) of the image offset by the *cx* and *cy* arguments.

In other words: `fl_push_clip(X, Y, W, H)` is applied, the image is drawn with its upper-left corner at *X*-*cx*, *Y*-*cy* and its own width and height, `fl_pop_clip()` is applied.

Reimplemented in [Fl_Shared_Image](#), [Fl_Tiled_Image](#), [Fl_Bitmap](#), [Fl_RGB_Image](#), and [Fl_Pixmap](#).

31.68.3.10 draw_empty()

```

void Fl_Image::draw_empty (
    int X,
    int Y ) [protected]

```

The protected method `draw_empty()` draws a box with an X in it.

It can be used to draw any image that lacks image data.

31.68.3.11 fail()

```

int Fl_Image::fail ( )

```

Returns a value that is not 0 if there is currently no image available.

Example use:

```

[.]
Fl_Box box(X, Y, W, H);
Fl_JPEG_Image jpg("/tmp/foo.jpg");
switch ( jpg.fail() ) {
    case Fl_Image::ERR_NO_IMAGE:
    case Fl_Image::ERR_FILE_ACCESS:
        fl_alert("/tmp/foo.jpg: %s", strerror(errno)); // shows actual os error to user
        exit(1);
    case Fl_Image::ERR_FORMAT:
        fl_alert("/tmp/foo.jpg: couldn't decode image");
        exit(1);
}
box.image(jpg);
[.]

```

Returns

ERR_NO_IMAGE if no image was found

ERR_FILE_ACCESS if there was a file access related error (errno should be set)

ERR_FORMAT if image decoding failed.

31.68.3.12 inactive()

```

void Fl_Image::inactive ( ) [inline]

```

The `inactive()` method calls `color_average(FL_BACKGROUND_COLOR, 0.33f)` to produce an image that appears grayed out.

An internal copy is made of the original image before changes are applied, to avoid modifying the original image.

31.68.3.13 label() [1/2]

```

void Fl_Image::label (
    Fl_Menu_Item * m ) [virtual]

```

The `label()` methods are an obsolete way to set the image attribute of a widget or menu item.

Use the `image()` or `deimage()` methods of the [Fl_Widget](#) and [Fl_Menu_Item](#) classes instead.

Reimplemented in [Fl_Bitmap](#), [Fl_RGB_Image](#), and [Fl_Pixmap](#).

31.68.3.14 label() [2/2]

```
void Fl_Image::label (
    Fl_Widget * widget ) [virtual]
```

The `label()` methods are an obsolete way to set the image attribute of a widget or menu item. Use the `image()` or `deimage()` methods of the `Fl_Widget` and `Fl_Menu_Item` classes instead. Reimplemented in `Fl_Bitmap`, `Fl_RGB_Image`, and `Fl_Pixmap`.

31.68.3.15 ld() [1/2]

```
int Fl_Image::ld ( ) const [inline]
Returns the current line data size in bytes.
```

See also

[ld\(int\)](#)

31.68.3.16 ld() [2/2]

```
void Fl_Image::ld (
    int LD ) [inline], [protected]
```

Sets the current line data size in bytes.

Color images may contain extra data that is included after every line of color image data and is normally not present.

If `LD` is zero, then line data size is assumed to be `w() * d()` bytes.

If `LD` is non-zero, then it must be positive and larger than `w() * d()` to account for the extra data per line.

31.68.3.17 RGB_scaling()

```
void Fl_Image::RGB_scaling (
    Fl_RGB_Scaling method ) [static]
```

Sets the RGB image scaling method used for `copy(int, int)`.

Applies to all RGB images, defaults to `FL_RGB_SCALING_NEAREST`.

31.68.3.18 uncache()

```
void Fl_Image::uncache ( ) [virtual]
```

If the image has been cached for display, delete the cache data.

This allows you to change the data used for the image and then redraw it without recreating an image object.

Reimplemented in `Fl_Bitmap`, `Fl_RGB_Image`, `Fl_Pixmap`, and `Fl_Shared_Image`.

The documentation for this class was generated from the following files:

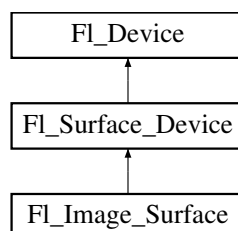
- [Fl_Image.H](#)
- [Fl_Image.cxx](#)

31.69 Fl_Image_Surface Class Reference

Directs all graphics requests to an `Fl_Image`.

```
#include <Fl_Image_Surface.H>
```

Inheritance diagram for `Fl_Image_Surface`:



Public Member Functions

- `const char * class_name ()`
Returns the name of the class of this object.
- `void draw (Fl_Widget *, int delta_x=0, int delta_y=0)`
Draws a widget in the image surface.
- `void draw_decorated_window (Fl_Window *win, int delta_x=0, int delta_y=0)`
Draws a window and its borders and title bar to the image drawing surface.
- `Fl_Image_Surface (int w, int h, int highres=0)`
Constructor with optional high resolution.
- `Fl_Shared_Image * highres_image ()`
Returns a possibly high resolution image made of all drawings sent to the `Fl_Image_Surface` object.
- `Fl_RGB_Image * image ()`
Returns an image made of all drawings sent to the `Fl_Image_Surface` object.
- `void set_current ()`
Make this surface the current drawing surface.
- `~Fl_Image_Surface ()`
The destructor.

Static Public Attributes

- `static const char * class_id = "Fl_Image_Surface"`

Additional Inherited Members

31.69.1 Detailed Description

Directs all graphics requests to an `Fl_Image`.

After creation of an `Fl_Image_Surface` object, call `set_current()` on it, and all subsequent graphics requests will be recorded in the image. It's possible to draw widgets (using `Fl_Image_Surface::draw()`) or to use any of the [Drawing functions](#) or the [Color & Font functions](#). Finally, call `image()` on the object to obtain a newly allocated `Fl_RGB_Image` object.

`Fl_GL_Window` objects can be drawn in the image as well.

Usage example:

```
Fl_Widget *g = ...; // a widget you want to draw in an image
Fl_Image_Surface *img_surf = new Fl_Image_Surface(g->w(), g->h()); // create an Fl_Image_Surface object
img_surf->set_current(); // direct graphics requests to the image
fl_color(FL_WHITE); fl_rectf(0, 0, g->w(), g->h()); // draw a white background
img_surf->draw(g); // draw the g widget in the image
Fl_RGB_Image* image = img_surf->image(); // get the resulting image
delete img_surf; // delete the img_surf object
Fl_Display_Device::display_device()->set_current(); // direct graphics requests back to the display
```

31.69.2 Constructor & Destructor Documentation

31.69.2.1 `Fl_Image_Surface()`

```
Fl_Image_Surface::Fl_Image_Surface (
    int w,
    int h,
    int highres = 0 )
```

Constructor with optional high resolution.

Parameters

<code>w</code>	and
----------------	-----

Parameters

<i>h</i>	give the size in pixels of the resulting image.
<i>highres</i>	if non-zero, the surface pixel size is twice as high and wide as <i>w</i> and <i>h</i> , which is useful to draw it later on a high resolution display (e.g., retina display). This is implemented for the Mac OS platform only. If <i>highres</i> is non-zero, use Fl_Image_Surface::highres_image() to get the image data.

Version

1.3.4 and requires compilation with `-DFL_ABI_VERSION=10304` (1.3.3 without the `highres` parameter)

31.69.3 Member Function Documentation

31.69.3.1 `class_name()`

```
const char * Fl_Image_Surface::class_name ( ) [inline], [virtual]
```

Returns the name of the class of this object.

Use of the `class_name()` function is discouraged because it will be removed from future FLTK versions.

The class of an instance of an [Fl_Device](#) subclass can be checked with code such as:

```
if ( instance->class_name() == Fl_Printer::class_id ) { ... }
```

Reimplemented from [Fl_Surface_Device](#).

31.69.3.2 `draw()`

```
void Fl_Image_Surface::draw (
    Fl_Widget * widget,
    int delta_x = 0,
    int delta_y = 0 )
```

Draws a widget in the image surface.

Parameters

<i>widget</i>	any FLTK widget (e.g., standard, custom, window, GL view) to draw in the image
<i>delta</i> _↔ <i>_x</i>	and
<i>delta</i> _↔ <i>_y</i>	give the position in the image of the top-left corner of the widget

31.69.3.3 `draw_decorated_window()`

```
void Fl_Image_Surface::draw_decorated_window (
    Fl_Window * win,
    int delta_x = 0,
    int delta_y = 0 )
```

Draws a window and its borders and title bar to the image drawing surface.

Parameters

<i>win</i>	an FLTK window to draw in the image
<i>delta</i> _↔ <i>_x</i>	and
<i>delta</i> _↔ <i>_y</i>	give the position in the image of the top-left corner of the window's title bar

31.69.3.4 `highres_image()`

`Fl_Shared_Image * Fl_Image_Surface::highres_image ()`

Returns a possibly high resolution image made of all drawings sent to the `Fl_Image_Surface` object.

The `Fl_Image_Surface` object should have been constructed with `Fl_Image_Surface(W, H, 1)`. The returned image is scaled to a size of WxH drawing units and may have a pixel size twice as wide and high. The returned object should be deallocated with `Fl_Shared_Image::release()` after use.

Version

1.3.4 and requires compilation with `-DFL_ABI_VERSION=10304`

31.69.3.5 `image()`

`Fl_RGB_Image * Fl_Image_Surface::image ()`

Returns an image made of all drawings sent to the `Fl_Image_Surface` object.

The returned object contains its own copy of the RGB data. Prefer `Fl_Image_Surface::highres_image()` if the surface was constructed with the `highres` option on.

31.69.3.6 `set_current()`

```
void Fl_Image_Surface::set_current (
    void ) [virtual]
```

Make this surface the current drawing surface.

This surface will receive all future graphics requests.

Reimplemented from `Fl_Surface_Device`.

The documentation for this class was generated from the following files:

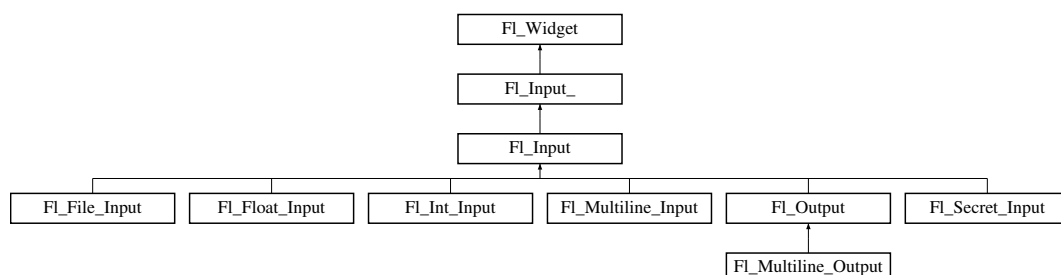
- `Fl_Image_Surface.H`
- `Fl_Image_Surface.cxx`

31.70 Fl_Input Class Reference

This is the FLTK text input widget.

```
#include <Fl_Input.H>
```

Inheritance diagram for `Fl_Input`:



Public Member Functions

- `Fl_Input` (int, int, int, int, const char *=0)
Creates a new `Fl_Input` widget using the given position, size, and label string.
- int `handle` (int)
Handles the specified event.

Protected Member Functions

- void `draw ()`
Draws the widget.

Additional Inherited Members

31.70.1 Detailed Description

This is the FLTK text input widget.

It displays a single line of text and lets the user edit it. Normally it is drawn with an inset box and a white background. The text may contain any characters, and will correctly display any UTF text, using `^X` notation for unprintable control characters. It assumes the font can draw any characters of the used scripts, which is true for standard fonts under MSWindows and Mac OS X. Characters can be input using the keyboard or the character palette/map. Character composition is done using dead keys and/or a compose key as defined by the operating system.

Table 31.190 Keyboard and mouse bindings.

Mouse button 1	Moves the cursor to this point. Drag selects characters. Double click selects words. Triple click selects all line. Shift+click extends the selection. When you select text it is automatically copied to the selection buffer.
Mouse button 2	Insert the selection buffer at the point clicked. You can also select a region and replace it with the selection buffer by selecting the region with mouse button 2.
Mouse button 3	Currently acts like button 1.
Backspace	Deletes one character to the left, or deletes the selected region.
Delete	Deletes one character to the right, or deletes the selected region. Combine with Shift for equivalent of <code>^X</code> (copy+cut).
Enter	May cause the callback, see <code>when()</code> .

Table 31.191 Platform specific keyboard bindings.

Windows/Linux	Mac	Function
^A	Command-A	Selects all text in the widget.
^C	Command-C	Copy the current selection to the clipboard.
^I	^I	Insert a tab.
^J	^J	Insert a Line Feed. (Similar to literal 'Enter' character)
^L	^L	Insert a Form Feed.
^M	^M	Insert a Carriage Return.
^V, Shift-Insert	Command-V	Paste the clipboard. (Macs keyboards don't have "Insert" keys, but if they did, Shift-Insert would work)
^X, Shift-Delete	Command-X, Shift-Delete	Cut. Copy the selection to the clipboard and delete it. (If there's no selection, Shift-Delete acts like Delete)
^Z	Command-Z	Undo. This is a single-level undo mechanism, but all adjacent deletions and insertions are concatenated into a single "undo". Often this will undo a lot more than you expected.

Shift-^Z	Shift-Command-Z	Redo. Currently same behavior as ^Z. Reserved for future multilevel undo/redo.
Arrow Keys	Arrow Keys	Standard cursor movement. Can be combined with Shift to extend selection.
Home	Command-Up, Command-Left	Move to start of line. Can be combined with Shift to extend selection.
End	Command-Down, Command-Right	Move to end of line. Can be combined with Shift to extend selection.
Ctrl-Home	Command-Up, Command-PgUp, Ctrl-Left	Move to top of document/field. In single line input, moves to start of line. In multiline input, moves to start of top line. Can be combined with Shift to extend selection.
Ctrl-End	Command-End, Command-PgDn, Ctrl-Right	Move to bottom of document/field. In single line input, moves to end of line. In multiline input, moves to end of last line. Can be combined with Shift to extend selection.
Ctrl-Left	Alt-Left	Word left. Can be combined with Shift to extend selection.
Ctrl-Right	Alt-Right	Word right. Can be combined with Shift to extend selection.
Ctrl-Backspace	Alt-Backspace	Delete word left.
Ctrl-Delete	Alt-Delete	Delete word right.

31.70.2 Constructor & Destructor Documentation

31.70.2.1 Fl_Input()

```
Fl_Input::Fl_Input (
    int X,
    int Y,
    int W,
    int H,
    const char * l = 0 )
```

Creates a new [Fl_Input](#) widget using the given position, size, and label string. The default boxtype is `FL_DOWN_BOX`.

31.70.3 Member Function Documentation

31.70.3.1 draw()

```
void Fl_Input::draw ( ) [protected], [virtual]
```

Draws the widget.

Never call this function directly. FLTK will schedule redrawing whenever needed. If your widget must be redrawn as soon as possible, call [redraw\(\)](#) instead.

Override this function to draw your own widgets.

If you ever need to call another widget's draw method *from within your own draw() method*, e.g. for an embedded scrollbar, you can do it (because [draw\(\)](#) is virtual) like this:

```
Fl_Widget *s = &scroll; // scroll is an embedded Fl_Scrollbar
```

```
s->draw(); // calls Fl_Scrollbar::draw()
Implements Fl\_Widget.
```

31.70.3.2 handle()

```
int Fl_Input::handle (
    int event ) [virtual]
```

Handles the specified event.

You normally don't call this method directly, but instead let FLTK do it when the user interacts with the widget.

When implemented in a widget, this function must return 0 if the widget does not use the event or 1 otherwise.

Most of the time, you want to call the inherited [handle\(\)](#) method in your overridden method so that you don't short-circuit events that you don't handle. In this last case you should return the callee retval.

Parameters

in	<i>event</i>	the kind of event received
----	--------------	----------------------------

Return values

0	if the event was not used or understood
1	if the event was used and can be deleted

See also

[Fl_Event](#)

Reimplemented from [Fl_Widget](#).

Reimplemented in [Fl_Secret_Input](#).

The documentation for this class was generated from the following files:

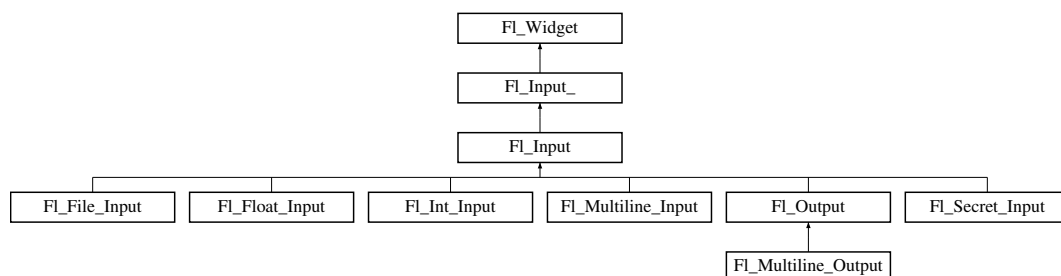
- [Fl_Input.H](#)
- [Fl_Input.cxx](#)

31.71 [Fl_Input_](#) Class Reference

This class provides a low-overhead text input field.

```
#include <Fl_Input_.H>
```

Inheritance diagram for [Fl_Input_](#):



Public Member Functions

- int [copy](#) (int clipboard)
Put the current selection into the clipboard.
- int [copy_cuts](#) ()
Copies the yank buffer to the clipboard.

- [FI_Color cursor_color](#) () const
Gets the color of the cursor.
- void [cursor_color](#) ([FI_Color](#) n)
Sets the color of the cursor.
- int [cut](#) ()
Deletes the current selection.
- int [cut](#) (int a, int b)
Deletes all characters between index a and b.
- int [cut](#) (int n)
Deletes the next n bytes rounded to characters before or after the cursor.
- [FI_Input_](#) (int, int, int, int, const char *=0)
Creates a new [FI_Input_](#) widget.
- [FI_Char index](#) (int i) const
Returns the character at index i.
- int [input_type](#) () const
Gets the input field type.
- void [input_type](#) (int t)
Sets the input field type.
- int [insert](#) (const char *t, int l=0)
Inserts text at the cursor position.
- int [mark](#) () const
Gets the current selection mark.
- int [mark](#) (int m)
Sets the current selection mark.
- int [maximum_size](#) () const
Gets the maximum length of the input field in characters.
- void [maximum_size](#) (int m)
Sets the maximum length of the input field in characters.
- int [position](#) () const
Gets the position of the text cursor.
- int [position](#) (int p)
Sets the cursor position and mark.
- int [position](#) (int p, int m)
Sets the index for the cursor and mark.
- int [readonly](#) () const
Gets the read-only state of the input field.
- void [readonly](#) (int b)
Sets the read-only state of the input field.
- int [replace](#) (int b, int e, const char *text, int ilen=0)
Deletes text from b to e and inserts the new string text.
- void [resize](#) (int, int, int, int)
Changes the size of the widget.
- int [shortcut](#) () const
Return the shortcut key associated with this widget.
- void [shortcut](#) (int s)
Sets the shortcut key associated with this widget.
- int [size](#) () const
Returns the number of bytes in [value\(\)](#).
- void [size](#) (int W, int H)
Sets the width and height of this widget.
- int [static_value](#) (const char *)

- Changes the widget text.*

 - int `static_value` (const char *, int)

Changes the widget text.
- int `tab_nav` () const

Gets whether the Tab key causes focus navigation in multiline input fields or not.
- void `tab_nav` (int val)

Sets whether the Tab key does focus navigation, or inserts tab characters into `FI_Multiline_Input`.
- `FI_Color` `textcolor` () const

Gets the color of the text in the input field.
- void `textcolor` (`FI_Color` n)

Sets the color of the text in the input field.
- `FI_Font` `textfont` () const

Gets the font of the text in the input field.
- void `textfont` (`FI_Font` s)

Sets the font of the text in the input field.
- `FI_Fontsize` `textsize` () const

Gets the size of the text in the input field.
- void `textsize` (`FI_Fontsize` s)

Sets the size of the text in the input field.
- int `undo` ()

Undoes previous changes to the text buffer.
- const char * `value` () const

Returns the text displayed in the widget.
- int `value` (const char *)

Changes the widget text.
- int `value` (const char *, int)

Changes the widget text.
- int `wrap` () const

Gets the word wrapping state of the input field.
- void `wrap` (int b)

Sets the word wrapping state of the input field.
- `~FI_Input_` ()

Destroys the widget.

Protected Member Functions

- void `drawtext` (int, int, int, int)

Draws the text in the passed bounding box.
- void `handle_mouse` (int, int, int, int, int keepmark=0)

Handles mouse clicks and mouse moves.
- int `handletext` (int e, int, int, int, int)

Handles all kinds of text field related events.
- int `line_end` (int i) const

Finds the end of a line.
- int `line_start` (int i) const

Finds the start of a line.
- int `linesPerPage` ()
- void `maybe_do_callback` ()
- int `up_down_position` (int, int keepmark=0)

Moves the cursor to the column given by `up_down_pos`.
- int `word_end` (int i) const

Finds the end of a word.

- int `word_start` (int i) const

Finds the start of a word.

- int `xscroll` () const
- int `yscroll` () const
- void `yscroll` (int yOffset)

Additional Inherited Members

31.71.1 Detailed Description

This class provides a low-overhead text input field.

This is a virtual base class below `FI_Input`. It has all the same interfaces, but lacks the `handle()` and `draw()` method. You may want to subclass it if you are one of those people who likes to change how the editing keys work. It may also be useful for adding scrollbars to the input field.

This can act like any of the subclasses of `FI_Input`, by setting `type()` to one of the following values:

```
#define FL_NORMAL_INPUT      0
#define FL_FLOAT_INPUT      1
#define FL_INT_INPUT        2
#define FL_MULTILINE_INPUT  4
#define FL_SECRET_INPUT     5
#define FL_INPUT_TYPE       7
#define FL_INPUT_READONLY   8
#define FL_NORMAL_OUTPUT    (FL_NORMAL_INPUT | FL_INPUT_READONLY)
#define FL_MULTILINE_OUTPUT (FL_MULTILINE_INPUT | FL_INPUT_READONLY)
#define FL_INPUT_WRAP       16
#define FL_MULTILINE_INPUT_WRAP (FL_MULTILINE_INPUT | FL_INPUT_WRAP)
#define FL_MULTILINE_OUTPUT_WRAP (FL_MULTILINE_INPUT | FL_INPUT_READONLY | FL_INPUT_WRAP)
```

All variables that represent an index into a text buffer are byte-oriented, not character oriented, counting from 0 (at or before the first character) to `size()` (at the end of the buffer, after the last byte). Since UTF-8 characters can be up to six bytes long, simply incrementing such an index will not reliably advance to the next character in the text buffer. Indices and pointers into the text buffer should always point at a 7 bit ASCII character or the beginning of a UTF-8 character sequence. Behavior for false UTF-8 sequences and pointers into the middle of a sequence are undefined.

See also

[FI_Text_Display](#), [FI_Text_Editor](#) for more powerful text handling widgets

31.71.2 Constructor & Destructor Documentation

31.71.2.1 FI_Input_()

```
FI_Input_::FI_Input_ (
    int X,
    int Y,
    int W,
    int H,
    const char * l = 0 )
```

Creates a new `FI_Input_` widget.

This function creates a new `FI_Input_` widget and adds it to the current `FI_Group`. The `value()` is set to `NULL`. The default boxtype is `FL_DOWN_BOX`.

Parameters

<code>X,Y,W,H</code>	the dimensions of the new widget
<code>l</code>	an optional label text

31.71.2.2 ~Fl_Input_()

```
Fl_Input_::~~Fl_Input_ ( )
```

Destroys the widget.

The destructor clears all allocated buffers and removes the widget from the parent [Fl_Group](#).

31.71.3 Member Function Documentation

31.71.3.1 copy()

```
int Fl_Input_::copy (
    int clipboard )
```

Put the current selection into the clipboard.

This function copies the current selection between [mark\(\)](#) and [position\(\)](#) into the specified `clipboard`. This does not replace the old clipboard contents if [position\(\)](#) and [mark\(\)](#) are equal. Clipboard 0 maps to the current text selection and clipboard 1 maps to the cut/paste clipboard.

Parameters

<i>clipboard</i>	the clipboard destination 0 or 1
------------------	----------------------------------

Returns

0 if no text is selected, 1 if the selection was copied

See also

[Fl::copy\(const char *, int, int\)](#)

31.71.3.2 copy_cuts()

```
int Fl_Input_::copy_cuts ( )
```

Copies the *yank* buffer to the clipboard.

This method copies all the previous contiguous cuts from the undo information to the clipboard. This function implements the ^K shortcut key.

Returns

0 if the operation did not change the clipboard

See also

[copy\(int\)](#), [cut\(\)](#)

31.71.3.3 cursor_color() [1/2]

```
Fl_Color Fl_Input_::cursor_color ( ) const [inline]
```

Gets the color of the cursor.

Returns

the current cursor color

31.71.3.4 cursor_color() [2/2]

```
void Fl_Input_::cursor_color (
    Fl_Color n ) [inline]
```

Sets the color of the cursor.

The default color for the cursor is `FL_BLACK`.

Parameters

in	<i>n</i>	the new cursor color
----	----------	----------------------

31.71.3.5 cut() [1/3]

```
int Fl_Input_::cut ( ) [inline]
```

Deletes the current selection.

This function deletes the currently selected text *without* storing it in the clipboard. To use the clipboard, you may call [copy\(\)](#) first or [copy_cuts\(\)](#) after this call.

Returns

0 if no data was copied

31.71.3.6 cut() [2/3]

```
int Fl_Input_::cut (
    int a,
    int b ) [inline]
```

Deletes all characters between index *a* and *b*.

This function deletes the currently selected text *without* storing it in the clipboard. To use the clipboard, you may call [copy\(\)](#) first or [copy_cuts\(\)](#) after this call.

Parameters

<i>a,b</i>	range of bytes rounded to full characters and clamped to the buffer
------------	---

Returns

0 if no data was copied

31.71.3.7 cut() [3/3]

```
int Fl_Input_::cut (
    int n ) [inline]
```

Deletes the next *n* bytes rounded to characters before or after the cursor.

This function deletes the currently selected text *without* storing it in the clipboard. To use the clipboard, you may call [copy\(\)](#) first or [copy_cuts\(\)](#) after this call.

Parameters

<i>n</i>	number of bytes rounded to full characters and clamped to the buffer. A negative number will cut characters to the left of the cursor.
----------	--

Returns

0 if no data was copied

31.71.3.8 drawtext()

```
void Fl_Input_::drawtext (
    int X,
    int Y,
    int W,
    int H ) [protected]
```

Draws the text in the passed bounding box.

If `damage ()` & `FL_DAMAGE_ALL` is true, this assumes the area has already been erased to `color()`. Otherwise it does minimal update and erases the area itself.

Parameters

<code>X,Y,W,H</code>	area that must be redrawn
----------------------	---------------------------

31.71.3.9 handle_mouse()

```
void Fl_Input_::handle_mouse (
    int X,
    int Y,
    int ,
    int ,
    int drag = 0 ) [protected]
```

Handles mouse clicks and mouse moves.

Todo Add comment and parameters

31.71.3.10 handletext()

```
int Fl_Input_::handletext (
    int event,
    int X,
    int Y,
    int W,
    int H ) [protected]
```

Handles all kinds of text field related events.

This is called by derived classes.

Todo Add comment and parameters

31.71.3.11 index()

```
unsigned int Fl_Input_::index (
    int i ) const
```

Returns the character at index `i`.

This function returns the UTF-8 character at `i` as a ucs4 character code.

Parameters

in	<i>i</i>	index into the value field
----	----------	----------------------------

Returns

the character at index *i*

31.71.3.12 input_type() [1/2]

```
int Fl_Input_::input_type ( ) const [inline]
```

Gets the input field type.

Returns

the current input type

31.71.3.13 input_type() [2/2]

```
void Fl_Input_::input_type (
    int t ) [inline]
```

Sets the input field type.

A [redraw\(\)](#) is required to reformat the input field.

Parameters

in	<i>t</i>	new input type
----	----------	----------------

31.71.3.14 insert()

```
int Fl_Input_::insert (
    const char * t,
    int l = 0 ) [inline]
```

Inserts text at the cursor position.

This function inserts the string in *t* at the cursor [position\(\)](#) and moves the new position and mark to the end of the inserted text.

Parameters

in	<i>t</i>	text that will be inserted
in	<i>l</i>	length of text, or 0 if the string is terminated by <code>nul</code> .

Returns

0 if no text was inserted

31.71.3.15 line_end()

```
int Fl_Input_::line_end (
    int i ) const [protected]
```

Finds the end of a line.

This call calculates the end of a line based on the given index *i*.

Parameters

<code>in</code>	<code>i</code>	starting index for the search
-----------------	----------------	-------------------------------

Returns

end of the line

31.71.3.16 line_start()

```
int Fl_Input_::line_start (
    int i ) const [protected]
```

Finds the start of a line.

This call calculates the start of a line based on the given index `i`.

Parameters

<code>in</code>	<code>i</code>	starting index for the search
-----------------	----------------	-------------------------------

Returns

start of the line

31.71.3.17 mark() [1/2]

```
int Fl_Input_::mark ( ) const [inline]
```

Gets the current selection mark.

Returns

index into the text

31.71.3.18 mark() [2/2]

```
int Fl_Input_::mark (
    int m ) [inline]
```

Sets the current selection mark.

`mark(n)` is the same as `position(position(), n)`.

Parameters

<code>m</code>	new index of the mark
----------------	-----------------------

Returns

0 if the mark did not change

See also

[position\(\)](#), [position\(int, int\)](#)

31.71.3.19 maximum_size() [1/2]

```
int Fl_Input_::maximum_size ( ) const [inline]
```

Gets the maximum length of the input field in characters.

See also

[maximum_size\(int\)](#).

31.71.3.20 maximum_size() [2/2]

```
void Fl_Input_::maximum_size (
    int m ) [inline]
```

Sets the maximum length of the input field in characters.

This limits the number of **characters** that can be inserted in the widget.

Since FLTK 1.3 this is different than the buffer size, since one character can be more than one byte in UTF-8 encoding. In FLTK 1.1 this was the same (one byte = one character).

31.71.3.21 position() [1/3]

```
int Fl_Input_::position ( ) const [inline]
```

Gets the position of the text cursor.

Returns

the cursor position as an index in the range 0..[size\(\)](#)

See also

[position\(int, int\)](#)

31.71.3.22 position() [2/3]

```
int Fl_Input_::position (
    int p ) [inline]
```

Sets the cursor position and mark.

`position(n)` is the same as `position(n, n)`.

Parameters

<i>p</i>	new index for cursor and mark
----------	-------------------------------

Returns

0 if no positions changed

See also

[position\(int, int\)](#), [position\(\)](#), [mark\(int\)](#)

31.71.3.23 position() [3/3]

```
int Fl_Input_::position (
    int p,
    int m )
```

Sets the index for the cursor and mark.

The input widget maintains two pointers into the string. The *position* (*p*) is where the cursor is. The *mark* (*m*) is the other end of the selected text. If they are equal then there is no selection. Changing this does not affect the clipboard (use [copy\(\)](#) to do that).

Changing these values causes a [redraw\(\)](#). The new values are bounds checked.

Parameters

<i>p</i>	index for the cursor position
<i>m</i>	index for the mark

Returns

0 if no positions changed

See also

[position\(int\)](#), [position\(\)](#), [mark\(int\)](#)

31.71.3.24 [readonly\(\)](#) [1/2]

```
int Fl_Input_::readonly ( ) const [inline]
```

Gets the read-only state of the input field.

Returns

non-zero if this widget is read-only

31.71.3.25 [readonly\(\)](#) [2/2]

```
void Fl_Input_::readonly (
    int b ) [inline]
```

Sets the read-only state of the input field.

Parameters

in	<i>b</i>	if <i>b</i> is 0, the text in this widget can be edited by the user
----	----------	---

31.71.3.26 [replace\(\)](#)

```
int Fl_Input_::replace (
    int b,
    int e,
    const char * text,
    int ilen = 0 )
```

Deletes text from *b* to *e* and inserts the new string *text*.

All changes to the text buffer go through this function. It deletes the region between *b* and *e* (either one may be less or equal to the other), and then inserts the string *text* at that point and moves the [mark\(\)](#) and [position\(\)](#) to the end of the insertion. Does the callback if [when\(\)](#) & `FL_WHEN_CHANGED` and there is a change.

Set *b* and *e* equal to not delete anything. Set *text* to NULL to not insert anything.

`ilen` can be zero or `strlen(text)`, which saves a tiny bit of time if you happen to already know the length of the insertion, or can be used to insert a portion of a string. If `ilen` is zero, `strlen(text)` is used instead.

`b` and `e` are clamped to the `0..size()` range, so it is safe to pass any values. `b`, `e`, and `ilen` are used as numbers of bytes (not characters), where `b` and `e` count from 0 to `size()` (end of buffer).

If `b` and/or `e` don't point to a valid UTF-8 character boundary, they are adjusted to the previous (`b`) or the next (`e`) valid UTF-8 character boundary, resp..

If the current number of characters in the buffer minus deleted characters plus inserted characters in `text` would overflow the number of allowed characters (`maximum_size()`), then only the first characters of the string are inserted, so that `maximum_size()` is not exceeded.

`cut()` and `insert()` are just inline functions that call `replace()`.

Parameters

in	<i>b</i>	beginning index of text to be deleted
in	<i>e</i>	ending index of text to be deleted and insertion position
in	<i>text</i>	string that will be inserted
in	<i>ilen</i>	length of <code>text</code> or 0 for nul terminated strings

Returns

0 if nothing changed

Note

If `text` does not point to a valid UTF-8 character or includes invalid UTF-8 sequences, the text is inserted nevertheless (counting invalid UTF-8 bytes as one character each).

31.71.3.27 `resize()`

```
void Fl_Input_::resize (
    int X,
    int Y,
    int W,
    int H ) [virtual]
```

Changes the size of the widget.

This call updates the text layout so that the cursor is visible.

Parameters

in	<i>X,Y,W,H</i>	new size of the widget
----	----------------	------------------------

See also

[Fl_Widget::resize\(int, int, int, int\)](#)

Reimplemented from [Fl_Widget](#).

31.71.3.28 `shortcut()` [1/2]

```
int Fl_Input_::shortcut ( ) const [inline]
```

Return the shortcut key associated with this widget.

Returns

shortcut keystroke

See also

[Fl_Button::shortcut\(\)](#)

31.71.3.29 shortcut() [2/2]

```
void Fl_Input_::shortcut (
    int s ) [inline]
```

Sets the shortcut key associated with this widget.
Pressing the shortcut key gives text editing focus to this widget.

Parameters

in	s	new shortcut keystroke
----	---	------------------------

See also

[Fl_Button::shortcut\(\)](#)

31.71.3.30 size() [1/2]

```
int Fl_Input_::size ( ) const [inline]
```

Returns the number of bytes in [value\(\)](#).
This may be greater than `strlen(value())` if there are nul characters in the text.

Returns

number of bytes in the text

31.71.3.31 size() [2/2]

```
void Fl_Input_::size (
    int W,
    int H ) [inline]
```

Sets the width and height of this widget.

Parameters

in	W,H	new width and height
----	-----	----------------------

See also

[Fl_Widget::size\(int, int\)](#)

31.71.3.32 static_value() [1/2]

```
int Fl_Input_::static_value (
    const char * str )
```

Changes the widget text.

This function changes the text and sets the mark and the point to the end of it. The string is *not* copied. If the user edits the string it is copied to the internal buffer then. This can save a great deal of time and memory if your program is rapidly changing the values of text fields, but this will only work if the passed string remains unchanged until either the [Fl_Input](#) is destroyed or [value\(\)](#) is called again.

Parameters

in	<i>str</i>	the new text
----	------------	--------------

Returns

non-zero if the new value is different than the current one

31.71.3.33 static_value() [2/2]

```
int Fl_Input_::static_value (
    const char * str,
    int len )
```

Changes the widget text.

This function changes the text and sets the mark and the point to the end of it. The string is *not* copied. If the user edits the string it is copied to the internal buffer then. This can save a great deal of time and memory if your program is rapidly changing the values of text fields, but this will only work if the passed string remains unchanged until either the [Fl_Input](#) is destroyed or [value\(\)](#) is called again.

You can use the `len` parameter to directly set the length if you know it already or want to put `nul` characters in the text.

Parameters

in	<i>str</i>	the new text
in	<i>len</i>	the length of the new text

Returns

non-zero if the new value is different than the current one

31.71.3.34 tab_nav() [1/2]

```
int Fl_Input_::tab_nav ( ) const [inline]
```

Gets whether the Tab key causes focus navigation in multiline input fields or not.

If enabled (default), hitting Tab causes focus navigation to the next widget.

If disabled, hitting Tab inserts a tab character into the text field.

Returns

1 if Tab advances focus (default), 0 if Tab inserts tab characters.

See also

[tab_nav\(int\)](#), [Fl::OPTION_ARROW_FOCUS](#).

31.71.3.35 tab_nav() [2/2]

```
void Fl_Input_::tab_nav (
    int val ) [inline]
```

Sets whether the Tab key does focus navigation, or inserts tab characters into [Fl_Multiline_Input](#).

By default this flag is enabled to provide the 'normal' behavior most users expect; Tab navigates focus to the next widget. To inserting an actual Tab character, users can use Ctrl-I or copy/paste.

Disabling this flag gives the old FLTK behavior where Tab inserts a tab character into the text field, in which case only the mouse can be used to navigate to the next field.

History: This flag was provided for backwards support of FLTK's old 1.1.x behavior where Tab inserts a tab character instead of navigating focus to the next widget. This behavior was unique to [Fl_Multiline_Input](#). With the advent of [Fl_Text_Editor](#), this old behavior has been deprecated.

Parameters

<code>in</code>	<code>val</code>	If <code>val</code> is 1, Tab advances focus (default). If <code>val</code> is 0, Tab inserts a tab character (old FLTK behavior).
-----------------	------------------	---

See also

[tab_nav\(\)](#), [Fl::OPTION_ARROW_FOCUS](#).

31.71.3.36 textcolor() [1/2]

```
Fl_Color Fl_Input_::textcolor ( ) const [inline]
```

Gets the color of the text in the input field.

Returns

the text color

See also

[textcolor\(Fl_Color\)](#)

31.71.3.37 textcolor() [2/2]

```
void Fl_Input_::textcolor (
    Fl_Color n ) [inline]
```

Sets the color of the text in the input field.

The text color defaults to `FL_FOREGROUND_COLOR`.

Parameters

<code>in</code>	<code>n</code>	new text color
-----------------	----------------	----------------

See also

[textcolor\(\)](#)

31.71.3.38 textfont() [1/2]

```
Fl_Font Fl_Input_::textfont ( ) const [inline]
```

Gets the font of the text in the input field.

Returns

the current `Fl_Font` index

31.71.3.39 textfont() [2/2]

```
void Fl_Input_::textfont (
    Fl_Font s ) [inline]
```

Sets the font of the text in the input field.

The text font defaults to `FL_HELVETICA`.

Parameters

in	s	the new text font
----	---	-------------------

31.71.3.40 `textsize()` [1/2]

```
Fl_Fontsize Fl_Input_::textsize ( ) const [inline]
```

Gets the size of the text in the input field.

Returns

the text height in pixels

31.71.3.41 `textsize()` [2/2]

```
void Fl_Input_::textsize (
    Fl_Fontsize s ) [inline]
```

Sets the size of the text in the input field.

The text height defaults to `FL_NORMAL_SIZE`.

Parameters

in	s	the new font height in pixel units
----	---	------------------------------------

31.71.3.42 `undo()`

```
int Fl_Input_::undo ( )
```

Undoes previous changes to the text buffer.

This call undoes a number of previous calls to [replace\(\)](#).

Returns

non-zero if any change was made.

31.71.3.43 `up_down_position()`

```
int Fl_Input_::up_down_position (
    int i,
    int keepmark = 0 ) [protected]
```

Moves the cursor to the column given by `up_down_pos`.

This function is helpful when implementing up and down cursor movement. It moves the cursor from the beginning of a line to the column indicated by the global variable `up_down_pos` in pixel units.

Parameters

in	<i>i</i>	index into the beginning of a line of text
in	<i>keepmark</i>	if set, move only the cursor, but not the mark

Returns

index to new cursor position

31.71.3.44 value() [1/3]

```
const char * Fl_Input_::value ( ) const [inline]
```

Returns the text displayed in the widget.

This function returns the current value, which is a pointer to the internal buffer and is valid only until the next event is handled.

Returns

pointer to an internal buffer - do not free() this

See also

[Fl_Input_::value\(const char*\)](#)

31.71.3.45 value() [2/3]

```
int Fl_Input_::value (
    const char * str )
```

Changes the widget text.

This function changes the text and sets the mark and the point to the end of it. The string is copied to the internal buffer. Passing NULL is the same as "".

Parameters

in	<i>str</i>	the new text
----	------------	--------------

Returns

non-zero if the new value is different than the current one

See also

[Fl_Input_::value\(const char* str, int len\)](#), [Fl_Input_::value\(\)](#)

31.71.3.46 value() [3/3]

```
int Fl_Input_::value (
    const char * str,
    int len )
```

Changes the widget text.

This function changes the text and sets the mark and the point to the end of it. The string is copied to the internal buffer. Passing NULL is the same as "".

You can use the `length` parameter to directly set the length if you know it already or want to put `nul` characters in the text.

Parameters

in	<i>str</i>	the new text
in	<i>len</i>	the length of the new text

Returns

non-zero if the new value is different than the current one

See also

[Fl_Input_::value\(const char* str\)](#), [Fl_Input_::value\(\)](#)

31.71.3.47 word_end()

```
int Fl_Input_::word_end (
    int i ) const [protected]
```

Finds the end of a word.

Returns the index after the last byte of a word. If the index is already at the end of a word, it will find the end of the following word, so if you call it repeatedly you will move forwards to the end of the text.

Note that this is inconsistent with [line_end\(\)](#).

Parameters

in	<i>i</i>	starting index for the search
----	----------	-------------------------------

Returns

end of the word

31.71.3.48 word_start()

```
int Fl_Input_::word_start (
    int i ) const [protected]
```

Finds the start of a word.

Returns the index of the first byte of a word. If the index is already at the beginning of a word, it will find the beginning of the previous word, so if you call it repeatedly you will move backwards to the beginning of the text.

Note that this is inconsistent with [line_start\(\)](#).

Parameters

in	<i>i</i>	starting index for the search
----	----------	-------------------------------

Returns

start of the word, or previous word

31.71.3.49 wrap() [1/2]

```
int Fl_Input_::wrap ( ) const [inline]
```

Gets the word wrapping state of the input field.

Word wrap is only functional with multi-line input fields.

31.71.3.50 wrap() [2/2]

```
void Fl_Input_::wrap (
    int b ) [inline]
```

Sets the word wrapping state of the input field.

Word wrap is only functional with multi-line input fields.

The documentation for this class was generated from the following files:

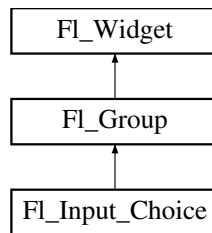
- [Fl_Input_.H](#)
- [Fl_Input_.cxx](#)

31.72 Fl_Input_Choice Class Reference

A combination of the input widget and a menu button.

```
#include <Fl_Input_Choice.H>
```

Inheritance diagram for Fl_Input_Choice:



Public Member Functions

- void **add** (const char *s)
Adds an item to the menu.
- int **changed** () const
Returns the combined [changed\(\)](#) state of the input and menu button widget.
- void **clear** ()
Removes all items from the menu.
- void **clear_changed** ()
Clears the [changed\(\)](#) state of both input and menu button widgets.
- **Fl_Boxtype** **down_box** () const
Gets the box type of the menu button.
- void **down_box** (**Fl_Boxtype** b)
Sets the box type of the menu button.
- **Fl_Input_Choice** (int X, int Y, int W, int H, const char *L=0)
Creates a new [Fl_Input_Choice](#) widget using the given position, size, and label string.
- **Fl_Input** * **input** ()
Returns a pointer to the internal [Fl_Input](#) widget.
- const **Fl_Menu_Item** * **menu** ()
Gets the [Fl_Menu_Item](#) array used for the menu.
- void **menu** (const **Fl_Menu_Item** *m)
Sets the [Fl_Menu_Item](#) array used for the menu.
- **Fl_Menu_Button** * **menubutton** ()
Returns a pointer to the internal [Fl_Menu_Button](#) widget.
- void **resize** (int X, int Y, int W, int H)
Resizes the [Fl_Group](#) widget and all of its children.
- void **set_changed** ()
Sets the [changed\(\)](#) state of both input and menu button widgets to the specified value.
- **Fl_Color** **textcolor** () const
Gets the [Fl_Input](#) text field's text color.
- void **textcolor** (**Fl_Color** c)
Sets the [Fl_Input](#) text field's text color to c.
- **Fl_Font** **textfont** () const
Gets the [Fl_Input](#) text field's font style.
- void **textfont** (**Fl_Font** f)
Sets the [Fl_Input](#) text field's font style to f.
- **Fl_Fontsize** **textsize** () const
Gets the [Fl_Input](#) text field's font size.

- void `textsize` ([Fl_Fontsize](#) s)
Sets the [Fl_Input](#) text field's font size to *s*.
- const char * `value` () const
Returns the [Fl_Input](#) text field's current contents.
- void `value` (const char *val)
Sets the [Fl_Input](#) text field's contents to *val*.
- void `value` (int val)
Chooses item# *val* in the menu, and sets the [Fl_Input](#) text field to that value.

Additional Inherited Members

31.72.1 Detailed Description

A combination of the input widget and a menu button.

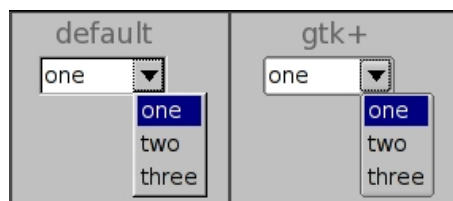


Figure 31.17 `Fl_Input_Choice` widget

The user can either type into the input area, or use the menu button chooser on the right to choose an item which loads the input area with the selected text.

The application can directly access both the internal [Fl_Input](#) and [Fl_Menu_Button](#) widgets respectively using the [input\(\)](#) and [menubutton\(\)](#) accessor methods.

The default behavior is to invoke the [Fl_Input_Choice::callback\(\)](#) if the user changes the input field's contents, either by typing, pasting, or clicking a different item in the choice menu.

The callback can determine if an item was picked vs. typing into the input field by checking the value of [menubutton\(\)->changed\(\)](#), which will be:

- 1: the user picked a different item in the choice menu
- 0: the user typed or pasted directly into the input field

Example use:

```
#include <stdio.h>
#include <FL/Fl.H>
#include <FL/Fl_Double_Window.H>
#include <FL/Fl_Input_Choice.H>
void choice_cb(Fl_Widget *w, void *userdata) {
    // Show info about the picked item
    Fl_Input_Choice *choice = (Fl_Input_Choice*)w;
    const Fl_Menu_Item *item = choice->menubutton()->mvalue();
    printf("*** Choice Callback:\n");
    printf("  item label()='%s'\n", item ? item->label() : "(No item)");
    printf("  item value()=%d\n", choice->menubutton()->value());
    printf("  input value()='%s'\n", choice->input()->value());
    printf("  The user %s\n", choice->menubutton()->changed()
           ? "picked a menu item"
           : "typed text");
}
int main() {
    Fl_Double_Window win(200,100,"Input Choice");
    win.begin();
    Fl_Input_Choice choice(10,10,100,30);
    choice.callback(choice_cb, 0);
    choice.add("Red");
    choice.add("Orange");
    choice.add("Yellow");
    //choice.value("Red"); // uncomment to make "Red" default
    win.end();
    win.show();
    return Fl::run();
}
```

31.72.2 Constructor & Destructor Documentation

31.72.2.1 Fl_Input_Choice()

```
Fl_Input_Choice::Fl_Input_Choice (
    int X,
    int Y,
    int W,
    int H,
    const char * L = 0 )
```

Creates a new [Fl_Input_Choice](#) widget using the given position, size, and label string. Inherited destructor destroys the widget and any values associated with it.

31.72.3 Member Function Documentation

31.72.3.1 add()

```
void Fl_Input_Choice::add (
    const char * s ) [inline]
```

Adds an item to the menu.

You can access the more complex [Fl_Menu_Button::add\(\)](#) methods (setting callbacks, userdata, etc), via [menubutton\(\)](#). Example:

```
Fl_Input_Choice *choice = new Fl_Input_Choice(100,10,120,25,"Fonts");
Fl_Menu_Button *mb = choice->menubutton(); // use Fl_Input_Choice's Fl_Menu_Button
mb->add("Helvetica", 0, MyFont_CB, (void*)mydata); // use Fl_Menu_Button's add() methods
mb->add("Courier", 0, MyFont_CB, (void*)mydata);
mb->add("More..", 0, FontDialog_CB, (void*)mydata);
```

31.72.3.2 input()

```
Fl_Input * Fl_Input_Choice::input ( ) [inline]
```

Returns a pointer to the internal [Fl_Input](#) widget.

This can be used to directly access all of the [Fl_Input](#) widget's methods.

31.72.3.3 menubutton()

```
Fl_Menu_Button * Fl_Input_Choice::menubutton ( ) [inline]
```

Returns a pointer to the internal [Fl_Menu_Button](#) widget.

This can be used to access any of the methods of the menu button, e.g.

```
Fl_Input_Choice *choice = new Fl_Input_Choice(100,10,120,25,"Choice:");
[...
// Print all the items in the choice menu
for ( int t=0; t<choice->menubutton()->size(); t++ ) {
    const Fl_Menu_Item &item = choice->menubutton()->menu()[t];
    printf("item %d -- label=%s\n", t, item.label() ? item.label() : "(Null)");
}
```

31.72.3.4 resize()

```
void Fl_Input_Choice::resize (
    int X,
    int Y,
    int W,
    int H ) [inline], [virtual]
```

Resizes the [Fl_Group](#) widget and all of its children.

The [Fl_Group](#) widget first resizes itself, and then it moves and resizes all its children according to the rules documented for [Fl_Group::resizable\(Fl_Widget*\)](#)

See also

[Fl_Group::resizable\(Fl_Widget*\)](#)
[Fl_Group::resizable\(\)](#)
[Fl_Widget::resize\(int,int,int,int\)](#)

Reimplemented from [Fl_Group](#).

31.72.3.5 value() [1/2]

```
void Fl_Input_Choice::value (
    const char * val ) [inline]
```

Sets the [Fl_Input](#) text field's contents to `val`.
 Does not affect the menu selection.

31.72.3.6 value() [2/2]

```
void Fl_Input_Choice::value (
    int val ) [inline]
```

Chooses item# `val` in the menu, and sets the [Fl_Input](#) text field to that value.
 Any previous text is cleared.

The documentation for this class was generated from the following files:

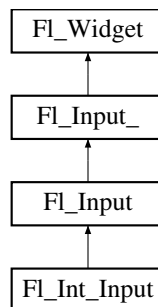
- [Fl_Input_Choice.H](#)
- [Fl_Group.cxx](#)

31.73 Fl_Int_Input Class Reference

The [Fl_Int_Input](#) class is a subclass of [Fl_Input](#) that only allows the user to type decimal digits (or hex numbers of the form 0xae).

```
#include <Fl_Int_Input.H>
```

Inheritance diagram for [Fl_Int_Input](#):



Public Member Functions

- [Fl_Int_Input](#) (int X, int Y, int W, int H, const char *l=0)
Creates a new [Fl_Int_Input](#) widget using the given position, size, and label string.

Additional Inherited Members

31.73.1 Detailed Description

The [Fl_Int_Input](#) class is a subclass of [Fl_Input](#) that only allows the user to type decimal digits (or hex numbers of the form 0xae).

31.73.2 Constructor & Destructor Documentation

31.73.2.1 FI_Int_Input()

```
Fl_Int_Input::Fl_Int_Input (
    int X,
    int Y,
    int W,
    int H,
    const char * l = 0 )
```

Creates a new [FI_Int_Input](#) widget using the given position, size, and label string. The default boxtype is FL_DOWN_BOX.

Inherited destructor destroys the widget and any value associated with it.

The documentation for this class was generated from the following files:

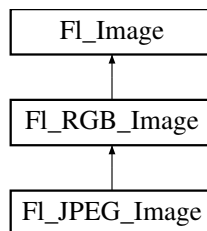
- FI_Int_Input.H
- FI_Input.cxx

31.74 FI_JPEG_Image Class Reference

The [FI_JPEG_Image](#) class supports loading, caching, and drawing of Joint Photographic Experts Group (JPEG) File Interchange Format (JFIF) images.

```
#include <FI_JPEG_Image.H>
```

Inheritance diagram for FI_JPEG_Image:



Public Member Functions

- [FI_JPEG_Image](#) (const char *filename)
The constructor loads the JPEG image from the given jpeg filename.
- [FI_JPEG_Image](#) (const char *name, const unsigned char *data)
The constructor loads the JPEG image from memory.

Additional Inherited Members

31.74.1 Detailed Description

The [FI_JPEG_Image](#) class supports loading, caching, and drawing of Joint Photographic Experts Group (JPEG) File Interchange Format (JFIF) images.

The class supports grayscale and color (RGB) JPEG image files.

31.74.2 Constructor & Destructor Documentation

31.74.2.1 FI_JPEG_Image() [1/2]

```
Fl_JPEG_Image::Fl_JPEG_Image (
    const char * filename )
```

The constructor loads the JPEG image from the given jpeg filename.

The inherited destructor frees all memory and server resources that are used by the image.

Use [Fl_Image::fail\(\)](#) to check if [FI_JPEG_Image](#) failed to load. [fail\(\)](#) returns `ERR_FILE_ACCESS` if the file could not be opened or read, `ERR_FORMAT` if the JPEG format could not be decoded, and `ERR_NO_IMAGE` if the image could not be loaded for another reason. If the image has loaded correctly, [w\(\)](#), [h\(\)](#), and [d\(\)](#) should return values greater than zero.

Parameters

in	<i>filename</i>	a full path and name pointing to a valid jpeg file.
----	-----------------	---

31.74.2.2 FI_JPEG_Image() [2/2]

```
Fl_JPEG_Image::Fl_JPEG_Image (
    const char * name,
    const unsigned char * data )
```

The constructor loads the JPEG image from memory.

Construct an image from a block of memory inside the application. Fluid offers "binary Data" chunks as a great way to add image data into the C++ source code. `name_png` can be NULL. If a name is given, the image is added to the list of shared images (see: [Fl_Shared_Image](#)) and will be available by that name.

The inherited destructor frees all memory and server resources that are used by the image.

Use [Fl_Image::fail\(\)](#) to check if [FI_JPEG_Image](#) failed to load. [fail\(\)](#) returns `ERR_FILE_ACCESS` if the file could not be opened or read, `ERR_FORMAT` if the JPEG format could not be decoded, and `ERR_NO_IMAGE` if the image could not be loaded for another reason. If the image has loaded correctly, [w\(\)](#), [h\(\)](#), and [d\(\)](#) should return values greater than zero.

Parameters

<i>name</i>	A unique name or NULL
<i>data</i>	A pointer to the memory location of the JPEG image

The documentation for this class was generated from the following files:

- [FI_JPEG_Image.H](#)
- [FI_JPEG_Image.cxx](#)

31.75 FI_Label Struct Reference

This struct stores all information for a text or mixed graphics label.

```
#include <Fl_Widget.H>
```

Public Member Functions

- void [draw](#) (int, int, int, int, [Fl_Align](#)) const
Draws the label aligned to the given box.
- void [measure](#) (int &w, int &h) const
Measures the size of the label.

Public Attributes

- [Fl_Align](#) `align_`

- alignment of label*
- [Fl_Color](#) **color**
text color
- [Fl_Image](#) * **deimage**
optional image for a deactivated label
- [Fl_Font](#) **font**
label font used in text
- [Fl_Image](#) * **image**
optional image for an active label
- [Fl_Fontsize](#) **size**
size of label font
- [uchar](#) **type**
type of label.
- `const char *` **value**
label text

31.75.1 Detailed Description

This struct stores all information for a text or mixed graphics label.

Todo There is an aspiration that the [Fl_Label](#) type will become a widget by itself. That way we will be avoiding a lot of code duplication by handling labels in a similar fashion to widgets containing text. We also provide an easy interface for very complex labels, containing html or vector graphics. However, this re-factoring is not in place in this release.

31.75.2 Member Function Documentation

31.75.2.1 draw()

```
void Fl_Label::draw (
    int X,
    int Y,
    int W,
    int H,
    Fl_Align align ) const
```

Draws the label aligned to the given box.

Draws a label with arbitrary alignment in an arbitrary box.

31.75.2.2 measure()

```
void Fl_Label::measure (
    int & W,
    int & H ) const
```

Measures the size of the label.

Parameters

<code>in, out</code>	<code>W,H</code>	: this is the requested size for the label text plus image; on return, this will contain the size needed to fit the label
----------------------	------------------	---

31.75.3 Member Data Documentation

31.75.3.1 type

`uchar Fl_Label::type`
type of label.

See also

[Fl_Labeltype](#)

The documentation for this struct was generated from the following files:

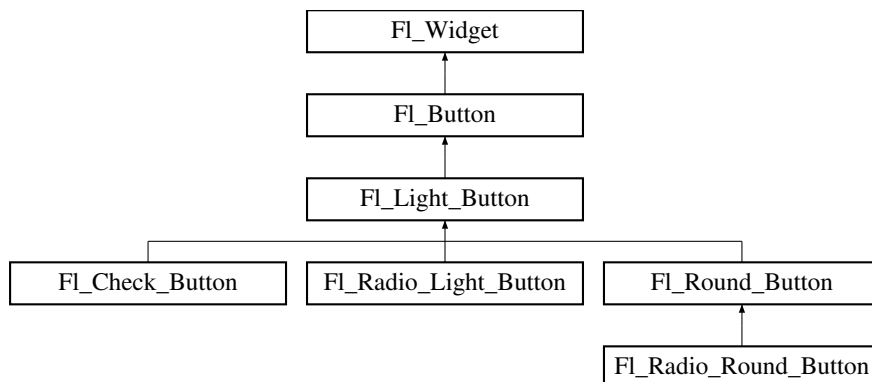
- [Fl_Widget.H](#)
- [fl_labeltype.cxx](#)

31.76 Fl_Light_Button Class Reference

This subclass displays the "on" state by turning on a light, rather than drawing pushed in.

```
#include <Fl_Light_Button.H>
```

Inheritance diagram for Fl_Light_Button:



Public Member Functions

- [Fl_Light_Button](#) (int *x*, int *y*, int *w*, int *h*, const char **l*=0)
Creates a new Fl_Light_Button widget using the given position, size, and label string.
- virtual int [handle](#) (int)
Handles the specified event.

Protected Member Functions

- virtual void [draw](#) ()
Draws the widget.

Additional Inherited Members

31.76.1 Detailed Description

This subclass displays the "on" state by turning on a light, rather than drawing pushed in.

The shape of the "light" is initially set to `FL_DOWN_BOX`. The color of the light when on is controlled with [selection_color\(\)](#), which defaults to `FL_YELLOW`.

Buttons generate callbacks when they are clicked by the user. You control exactly when and how by changing the values for [type\(\)](#) and [when\(\)](#).

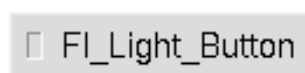


Figure 31.18 Fl_Light_Button

31.76.2 Constructor & Destructor Documentation

31.76.2.1 Fl_Light_Button()

```
Fl_Light_Button::Fl_Light_Button (
    int X,
    int Y,
    int W,
    int H,
    const char * l = 0 )
```

Creates a new [Fl_Light_Button](#) widget using the given position, size, and label string. The destructor deletes the check button.

31.76.3 Member Function Documentation

31.76.3.1 draw()

```
void Fl_Light_Button::draw ( ) [protected], [virtual]
```

Draws the widget.

Never call this function directly. FLTK will schedule redrawing whenever needed. If your widget must be redrawn as soon as possible, call [redraw\(\)](#) instead.

Override this function to draw your own widgets.

If you ever need to call another widget's draw method *from within your own [draw\(\)](#) method*, e.g. for an embedded scrollbar, you can do it (because [draw\(\)](#) is virtual) like this:

```
Fl_Widget *s = &scroll;           // scroll is an embedded Fl_Scrollbar
s->draw();                          // calls Fl_Scrollbar::draw()
```

Reimplemented from [Fl_Button](#).

31.76.3.2 handle()

```
int Fl_Light_Button::handle (
    int event ) [virtual]
```

Handles the specified event.

You normally don't call this method directly, but instead let FLTK do it when the user interacts with the widget.

When implemented in a widget, this function must return 0 if the widget does not use the event or 1 otherwise.

Most of the time, you want to call the inherited [handle\(\)](#) method in your overridden method so that you don't short-circuit events that you don't handle. In this last case you should return the callee retval.

Parameters

in	<i>event</i>	the kind of event received
----	--------------	----------------------------

Return values

0	if the event was not used or understood
1	if the event was used and can be deleted

See also

[Fl_Event](#)

Reimplemented from [Fl_Button](#).

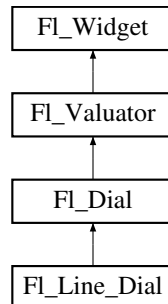
The documentation for this class was generated from the following files:

- [Fl_Light_Button.H](#)

- `Fl_Light_Button.cxx`

31.77 `Fl_Line_Dial` Class Reference

Inheritance diagram for `Fl_Line_Dial`:



Public Member Functions

- `Fl_Line_Dial` (int X, int Y, int W, int H, const char *L=0)

Additional Inherited Members

The documentation for this class was generated from the following files:

- `Fl_Line_Dial.H`
- `Fl_Dial.cxx`

31.78 `Fl_Mac_App_Menu` Class Reference

Mac OS-specific class allowing to customize and localize the application menu.

Static Public Member Functions

- static void `custom_application_menu_items` (const `Fl_Menu_Item` *m)
Adds custom menu item(s) to the application menu of the system menu bar.

Static Public Attributes

- static const char * **about** = "About %@"
Localizable text for the "About xxx" application menu item.
- static const char * **hide** = "Hide %@"
Localizable text for the "Hide xxx" application menu item.
- static const char * **hide_others** = "Hide Others"
Localizable text for the "Hide Others" application menu item.
- static const char * **print** = "Print Front Window"
Localizable text for the "Print Front Window" application menu item.
- static const char * **quit** = "Quit %@"
Localizable text for the "Quit xxx" application menu item.
- static const char * **services** = "Services"
Localizable text for the "Services" application menu item.
- static const char * **show** = "Show All"
Localizable text for the "Show All" application menu item.

31.78.1 Detailed Description

Mac OS-specific class allowing to customize and localize the application menu.

The public class attributes are used to build the application menu. They can be localized at run time to any UTF-8 text by placing instructions such as this before `fl_open_display()` gets called:

```
Fl_Mac_App_Menu::print = "Imprimer la fenêtre";
```

See also

[The Apple OS X Interface](#) for another way to localization.

31.78.2 Member Function Documentation

31.78.2.1 custom_application_menu_items()

```
void Fl_Mac_App_Menu::custom_application_menu_items (  
    const Fl_Menu_Item * m ) [static]
```

Adds custom menu item(s) to the application menu of the system menu bar.

They are positioned after the "Print Front Window" item, or at its place if it was removed with `Fl_Mac_App_Menu::print = ""`.

Parameters

<i>m</i>	zero-ending array of Fl_Menu_Item 's.
----------	---

31.78.3 Member Data Documentation

31.78.3.1 print

```
const char * Fl_Mac_App_Menu::print = "Print Front Window" [static]
```

Localizable text for the "Print Front Window" application menu item.

This menu item won't be displayed if `Fl_Mac_App_Menu::print` is set to an empty string.

The documentation for this class was generated from the following files:

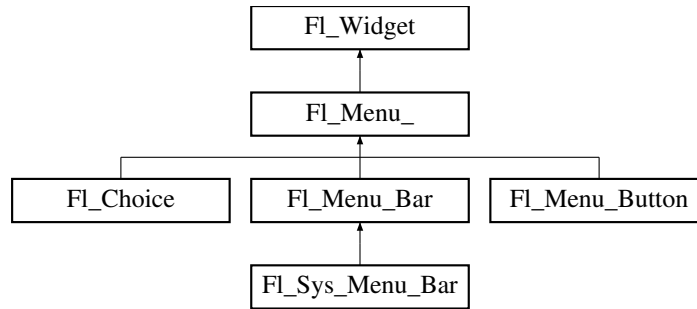
- [mac.H](#)
- [Fl.cxx](#)
- [Fl_Sys_Menu_Bar.mm](#)

31.79 Fl_Menu_ Class Reference

Base class of all widgets that have a menu in FLTK.

```
#include <Fl_Menu_.H>
```

Inheritance diagram for `Fl_Menu_`:



Public Member Functions

- int **add** (const char *)
This is a Forms (and SGI GL library) compatible add function, it adds many menu items, with '|' separating the menu items, and tab separating the menu item names from an optional shortcut string.
- int **add** (const char *, int shortcut, FL_Callback *, void *=0, int=0)
Adds a new menu item.
- int **add** (const char *a, const char *b, FL_Callback *c, void *d=0, int e=0)
See int FL_Menu_::add(const char label, int shortcut, FL_Callback*, void *user_data=0, int flags=0)*
- void **clear** ()
Same as menu(NULL), set the array pointer to null, indicating a zero-length menu.
- int **clear_submenu** (int index)
Clears the specified submenu pointed to by index of all menu items.
- void **copy** (const FL_Menu_Item *m, void *user_data=0)
Sets the menu array pointer with a copy of m that will be automatically deleted.
- FL_Boxtype **down_box** () const
This box type is used to surround the currently-selected items in the menus.
- void **down_box** (FL_Boxtype b)
See FL_Boxtype FL_Menu_::down_box() const
- FL_Color **down_color** () const
For back compatibility, same as selection_color()
- void **down_color** (unsigned c)
For back compatibility, same as selection_color()
- int **find_index** (const char *name) const
Find the menu item index for a given menu pathname, such as "Edit/Copy".
- int **find_index** (const FL_Menu_Item *item) const
Find the index into the menu array for a given item.
- int **find_index** (FL_Callback *cb) const
Find the index into the menu array for a given callback cb.
- const FL_Menu_Item * **find_item** (const char *name)
Find the menu item for a given menu pathname, such as "Edit/Copy".
- const FL_Menu_Item * **find_item** (FL_Callback *)
Find the menu item for the given callback cb.
- FL_Menu_ (int, int, int, int, const char *=0)
Creates a new FL_Menu_ widget using the given position, size, and label string.
- void **global** ()
Make the shortcuts for this menu work no matter what window has the focus when you type it.
- int **insert** (int index, const char *, int shortcut, FL_Callback *, void *=0, int=0)
Inserts a new menu item at the specified index position.
- int **insert** (int index, const char *a, const char *b, FL_Callback *c, void *d=0, int e=0)

- See `int Fl_Menu_::insert(const char* label, int shortcut, Fl_Callback*, void *user_data=0, int flags=0)`
- `int item_pathname` (char *name, int namelen, const `Fl_Menu_Item` *finditem=0) const
Get the menu 'pathname' for the specified menuitem.
 - `const Fl_Menu_Item * menu` () const
Returns a pointer to the array of `Fl_Menu_Items`.
 - `void menu` (const `Fl_Menu_Item` *m)
Sets the menu array pointer directly.
 - `int mode` (int i) const
Gets the flags of item i.
 - `void mode` (int i, int fl)
Sets the flags of item i.
 - `const Fl_Menu_Item * mvalue` () const
Returns a pointer to the last menu item that was picked.
 - `const Fl_Menu_Item * picked` (const `Fl_Menu_Item` *)
When user picks a menu item, call this.
 - `void remove` (int)
Deletes item i from the menu.
 - `void replace` (int, const char *)
Changes the text of item i.
 - `void setonly` (`Fl_Menu_Item` *item)
Turns the radio item "on" for the menu item and turns "off" adjacent radio items of the same group.
 - `void shortcut` (int i, int s)
Changes the shortcut of item i to s.
 - `int size` () const
This returns the number of `Fl_Menu_Item` structures that make up the menu, correctly counting submenus.
 - `void size` (int W, int H)
 - `const Fl_Menu_Item * test_shortcut` ()
Returns the menu item with the entered shortcut (key value).
 - `const char * text` () const
Returns the title of the last item chosen.
 - `const char * text` (int i) const
Returns the title of item i.
 - `Fl_Color textcolor` () const
Get the current color of menu item labels.
 - `void textcolor` (`Fl_Color` c)
Sets the current color of menu item labels.
 - `Fl_Font textfont` () const
Gets the current font of menu item labels.
 - `void textfont` (`Fl_Font` c)
Sets the current font of menu item labels.
 - `Fl_Fontsize textsize` () const
Gets the font size of menu item labels.
 - `void textsize` (`Fl_Fontsize` c)
Sets the font size of menu item labels.
 - `int value` () const
Returns the index into `menu()` of the last item chosen by the user.
 - `int value` (const `Fl_Menu_Item` *)
The value is the index into `menu()` of the last item chosen by the user.
 - `int value` (int i)
The value is the index into `menu()` of the last item chosen by the user.

Protected Member Functions

- int `item_pathname_` (char *name, int namelen, const [Fl_Menu_Item](#) *finditem, const [Fl_Menu_Item](#) *menu=0) const

Protected Attributes

- [uchar](#) `alloc`
- [uchar](#) `down_box_`
- [Fl_Color](#) `textcolor_`
- [Fl_Font](#) `textfont_`
- [Fl_Fontsize](#) `textsize_`

Additional Inherited Members

31.79.1 Detailed Description

Base class of all widgets that have a menu in FLTK.

Currently FLTK provides you with [Fl_Menu_Button](#), [Fl_Menu_Bar](#), and [Fl_Choice](#).

The class contains a pointer to an array of structures of type [Fl_Menu_Item](#). The array may either be supplied directly by the user program, or it may be "private": a dynamically allocated array managed by the [Fl_Menu_](#).

When the user clicks a menu item, `value()` is set to that item and then:

- If the [Fl_Menu_Item](#) has a callback set, that callback is invoked with any userdata configured for it. (The [Fl_Menu_](#) widget's callback is NOT invoked.)
- For any [Fl_Menu_Items](#) that **don't** have a callback set, the [Fl_Menu_](#) widget's callback is invoked with any userdata configured for it. The callback can determine which item was picked using `value()`, `mvalue()`, `item_pathname()`, etc.

31.79.2 Constructor & Destructor Documentation

31.79.2.1 [Fl_Menu_\(\)](#)

```
Fl_Menu_::Fl_Menu_ (
    int X,
    int Y,
    int W,
    int H,
    const char * l = 0 )
```

Creates a new [Fl_Menu_](#) widget using the given position, size, and label string. `menu()` is initialized to null.

31.79.3 Member Function Documentation

31.79.3.1 `add()` [1/2]

```
int Fl_Menu_::add (
    const char * str )
```

This is a Forms (and SGI GL library) compatible add function, it adds many menu items, with '|' separating the menu items, and tab separating the menu item names from an optional shortcut string.

The passed string is split at any '|' characters and then `add(s,0,0,0,0)` is done with each section. This is often useful if you are just using the value, and is compatible with Forms and other GL programs. The section strings use the same special characters as described for the long version of `add()`.

No items must be added to a menu during a callback to the same menu.

Parameters

<i>str</i>	string containing multiple menu labels as described above
------------	---

Returns

the index into the `menu()` array, where the entry was added

31.79.3.2 add() [2/2]

```
int Fl_Menu_::add (
    const char * label,
    int shortcut,
    Fl_Callback * callback,
    void * userdata = 0,
    int flags = 0 )
```

Adds a new menu item.

Parameters

in	<i>label</i>	The text label for the menu item.
in	<i>shortcut</i>	Optional keyboard shortcut that can be an int or string: (FL_CTRL+'a') or "^a". Default 0 if none.
in	<i>callback</i>	Optional callback invoked when user clicks the item. Default 0 if none.
in	<i>userdata</i>	Optional user data passed as an argument to the callback. Default 0 if none.
in	<i>flags</i>	Optional flags that control the type of menu item; see below. Default is 0 for none.

Returns

The index into the `menu()` array, where the entry was added.

Description

If the menu array was directly set with `menu(x)`, then `copy()` is done to make a private array.

Since this method can change the internal menu array, any menu item pointers or indices the application may have cached can become stale, and should be recalculated/refreshed.

A menu item's callback must not `add()` items to its parent menu during the callback.

Detailed Description of Parameters**label**

The menu item's label. This argument is required and must not be NULL.

The characters "&", "/", "\", and "_" are treated as special characters in the label string. The "&" character specifies that the following character is an accelerator and will be underlined. The "/" character is used to escape the next character in the string. Labels starting with the "_" character cause a divider to be placed after that menu item.

A label of the form "File/Quit" will create the submenu "File" with a menu item called "Quit". The "/" character is ignored if it appears as the first character of the label string, e.g. "/File/Quit".

The label string is copied to new memory and can be freed. The other arguments (including the shortcut) are copied into the menu item unchanged.

If an item exists already with that name then it is replaced with this new one. Otherwise this new one is added to the end of the correct menu or submenu. The return value is the offset into the array that the new entry was placed at.

shortcut

The keyboard shortcut for this menu item.

This parameter is optional, and defaults to 0 to indicate no shortcut.

The shortcut can either be a raw integer value (eg. FL_CTRL+'A') or a string (eg. "^c" or "^97").

Raw integer shortcuts can be a combination of keyboard chars (eg. 'A') and optional keyboard modifiers (see [Fl::event_state\(\)](#), e.g. FL_SHIFT, etc). In addition, FL_COMMAND can be used to denote FL_META under Mac OS X and FL_CTRL under other platforms.

String shortcuts can be specified in one of two ways:

```
[#+^]<ascii_value>    e.g. "97", "^97", "+97", "#97"
[#+^]<ascii_char>    e.g. "a", "^a", "+a", "#a"
```

..where <ascii_value> is a decimal value representing an ASCII character (eg. 97 is the ascii code for 'a'), and the optional prefixes enhance the value that follows. Multiple prefixes must appear in the order below.

```
# - Alt
+ - Shift
^ - Control
```

Internally, the text shortcuts are converted to integer values using [fl_old_shortcut\(const char*\)](#).

callback

The callback to invoke when this menu item is selected.

This parameter is optional, and defaults to 0 for no callback.

userdata

The callback's 'user data' that is passed to the callback.

This parameter is optional, and defaults to 0.

flags

These are bit flags to define what kind of menu item this is.

This parameter is optional, and defaults to 0 to define a 'regular' menu item.

These flags can be 'OR'ed together:

```
FL_MENU_INACTIVE    // Deactivate menu item (gray out)
FL_MENU_TOGGLE     // Item is a checkbox toggle (shows checkbox for on/off state)
FL_MENU_VALUE      // The on/off state for checkbox/radio buttons (if set, state is 'on')
FL_MENU_RADIO      // Item is a radio button (one checkbox of many can be on)
FL_MENU_INVISIBLE  // Item will not show up (shortcut will work)
FL_SUBMENU_POINTER // Indicates user_data() is a pointer to another menu array
FL_SUBMENU         // This item is a submenu to other items
FL_MENU_DIVIDER    // Creates divider line below this item. Also ends a group of radio buttons.
```

If FL_SUBMENU is set in an item's flags, then actually two items are added: the first item is the menu item (submenu title), as expected, and the second item is the submenu terminating item with the label and all other members set to 0. If you add submenus with the 'path' technique, then the corresponding submenu terminators (maybe more than one) are added as well.

Todo Raw integer shortcut needs examples. Dependent on responses to <http://fltk.org/newsgroups.php?gfltk.development+v:10086> and results of STR#2344

31.79.3.3 clear()

```
void Fl_Menu_::clear ( )
```

Same as menu(NULL), set the array pointer to null, indicating a zero-length menu. Menus must not be cleared during a callback to the same menu.

31.79.3.4 clear_submenu()

```
int Fl_Menu_::clear_submenu (
    int index )
```

Clears the specified submenu pointed to by `index` of all menu items.

This method is useful for clearing a submenu so that it can be re-populated with new items. Example: a "File/Recent Files/..." submenu that shows the last few files that have been opened.

The specified `index` must point to a submenu.

The submenu is cleared with `remove()`. If the menu array was directly set with `menu(x)`, then `copy()` is done to make a private array.

Warning

Since this method can change the internal menu array, any menu item pointers or indices the application may have cached can become stale, and should be recalculated/refreshed.

Example:

```
int index = menubar->find_index("File/Recent"); // get index of "File/Recent" submenu
if ( index != -1 ) menubar->clear_submenu(index); // clear the submenu
menubar->add("File/Recent/Aaa");
menubar->add("File/Recent/Bbb");
[...]
```

Parameters

<i>index</i>	The index of the submenu to be cleared
--------------	--

Returns

0 on success, -1 if the index is out of range or not a submenu

See also

[remove\(int\)](#)

31.79.3.5 copy()

```
void Fl_Menu_::copy (
    const Fl_Menu_Item * m,
    void * ud = 0 )
```

Sets the menu array pointer with a copy of *m* that will be automatically deleted.

If userdata *ud* is not NULL, then all user data pointers are changed in the menus as well. See [void Fl_Menu_::menu\(const Fl_Menu_Item* m\)](#).

31.79.3.6 down_box()

```
Fl_Boxtype Fl_Menu_::down_box ( ) const [inline]
```

This box type is used to surround the currently-selected items in the menus.

If this is FL_NO_BOX then it acts like FL_THIN_UP_BOX and [selection_color\(\)](#) acts like FL_WHITE, for back compatibility.

31.79.3.7 find_index() [1/3]

```
int Fl_Menu_::find_index (
    const char * pathname ) const
```

Find the menu item index for a given menu *pathname*, such as "Edit/Copy".

This method finds a menu item's index position for the given menu *pathname*, also traversing submenus, but **not** submenu pointers (FL_SUBMENU_POINTER).

To get the menu item pointer for a *pathname*, use [find_item\(\)](#)

Parameters

<i>in</i>	<i>pathname</i>	The path and name of the menu item to find
-----------	-----------------	--

Returns

The index of the matching item, or -1 if not found.

See also

[item_pathname\(\)](#)

31.79.3.8 find_index() [2/3]

```
int Fl_Menu_::find_index (
    const Fl_Menu_Item * item ) const
```

Find the index into the menu array for a given *item*.

A way to convert a menu item pointer into an index.

Does **not** handle items that are in submenu pointers (FL_SUBMENU_POINTER).
 -1 is returned if the item is not in this menu or is part of an FL_SUBMENU_POINTER submenu.
 Current implementation is fast and not expensive.

```
// Convert an index-to-item
int index = 12;
const Fl_Menu_Item *item = mymenu->menu() + index;
// Convert an item-to-index
int index = mymenu->find_index(item);
if ( index == -1 ) { ..error.. }
```

Parameters

in	<i>item</i>	The item to be found
----	-------------	----------------------

Returns

The index of the item, or -1 if not found.

See also

[menu\(\)](#)

31.79.3.9 find_index() [3/3]

```
int Fl_Menu_::find_index (
    Fl_Callback * cb ) const
```

Find the index into the menu array for a given callback *cb*.

This method finds a menu item's index position, also traversing submenus, but **not** submenu pointers (FL_↔SUBMENU_POINTER). This is useful if an application uses internationalisation and a menu item can not be found using its label. This search is also much faster.

Parameters

<i>cb</i>	Find the first item with this callback
-----------	--

Returns

The index of the item with the specific callback, or -1 if not found

See also

[find_index\(const char*\)](#)

31.79.3.10 find_item() [1/2]

```
const Fl_Menu_Item * Fl_Menu_::find_item (
    const char * pathname )
```

Find the menu item for a given menu *pathname*, such as "Edit/Copy".

This method finds a menu item in the menu array, also traversing submenus, but not submenu pointers (FL_↔SUBMENU_POINTER).

To get the menu item's index, use [find_index\(const char*\)](#)

Example:

```
Fl_Menu_Bar *menubar = new Fl_Menu_Bar(..);
menubar->add("File/&Open");
menubar->add("File/&Save");
menubar->add("Edit/&Copy");
// [...]
Fl_Menu_Item *item;
if ( ( item = (Fl_Menu_Item*)menubar->find_item("File/&Open") ) != NULL ) {
    item->labelcolor(FL_RED);
}
```

```

}
if ( ( item = (Fl_Menu_Item*)menubar->find_item("Edit/&Copy") ) != NULL ) {
    item->labelcolor(FL_GREEN);
}

```

Parameters

<i>pathname</i>	The path and name of the menu item
-----------------	------------------------------------

Returns

The item found, or NULL if not found

See also

[find_index\(const char*\)](#), [find_item\(Fl_Callback*\)](#), [item_pathname\(\)](#)

31.79.3.11 find_item() [2/2]

```

const Fl_Menu_Item * Fl_Menu_::find_item (
    Fl_Callback * cb )

```

Find the menu item for the given callback *cb*.

This method finds a menu item in a menu array, also traversing submenus, but not submenu pointers. This is useful if an application uses internationalisation and a menu item can not be found using its label. This search is also much faster.

Parameters

<i>in</i>	<i>cb</i>	find the first item with this callback
-----------	-----------	--

Returns

The item found, or NULL if not found

See also

[find_item\(const char*\)](#)

31.79.3.12 global()

```

void Fl_Menu_::global ( )

```

Make the shortcuts for this menu work no matter what window has the focus when you type it.

This is done by using [Fl::add_handler\(\)](#). This `Fl_Menu_` widget does not have to be visible (ie the window it is in can be hidden, or it does not have to be put in a window at all).

Currently there can be only one `global()` menu. Setting a new one will replace the old one. There is no way to remove the `global()` setting (so don't destroy the widget!)

31.79.3.13 insert()

```

int Fl_Menu_::insert (
    int index,
    const char * label,
    int shortcut,
    Fl_Callback * callback,
    void * userdata = 0,
    int flags = 0 )

```

Inserts a new menu item at the specified `index` position.

If `index` is -1, the menu item is appended; same behavior as [add\(\)](#).

To properly insert a menu item, `label` must be the name of the item (eg. "Quit"), and not a 'menu pathname' (eg. "File/Quit"). If a menu pathname is specified, the value of `index` is *ignored*, the new item's position defined by the pathname.

For more details, see [add\(\)](#). Except for the `index` parameter, [add\(\)](#) has more detailed information on parameters and behavior, and is functionally equivalent.

Parameters

in	<i>index</i>	The menu array's index position where the new item is inserted. If -1, behavior is the same as add() .
in	<i>label</i>	The text label for the menu item. If the label is a menu pathname, <code>index</code> is ignored, and the pathname indicates the position of the new item.
in	<i>shortcut</i>	Optional keyboard shortcut. Can be an int (FL_CTRL+'a') or a string ("^a"). Default is 0.
in	<i>callback</i>	Optional callback invoked when user clicks the item. Default 0 if none.
in	<i>userdata</i>	Optional user data passed as an argument to the callback. Default 0 if none.
in	<i>flags</i>	Optional flags that control the type of menu item; see add() for more info. Default is 0 for none.

Returns

The index into the [menu\(\)](#) array, where the entry was added.

See also

[add\(\)](#)

31.79.3.14 item_pathname()

```
int Fl_Menu_::item_pathname (
    char * name,
    int namelen,
    const Fl_Menu_Item * finditem = 0 ) const
```

Get the menu 'pathname' for the specified menuitem.

If `finditem==NULL`, [mvalue\(\)](#) is used (the most recently picked menuitem).

Example:

```
Fl_Menu_Bar *menubar = 0;
void my_menu_callback(Fl_Widget*,void*) {
    char name[80];
    if ( menubar->item_pathname(name, sizeof(name)-1) == 0 ) { // recently picked item
        if ( strcmp(name, "File/&Open") == 0 ) { .. } // open invoked
        if ( strcmp(name, "File/&Save") == 0 ) { .. } // save invoked
        if ( strcmp(name, "Edit/&Copy") == 0 ) { .. } // copy invoked
    }
}
int main() {
    [...]
    menubar = new Fl_Menu_Bar(..);
    menubar->add("File/&Open", 0, my_menu_callback);
    menubar->add("File/&Save", 0, my_menu_callback);
    menubar->add("Edit/&Copy", 0, my_menu_callback);
    [...]
}
```

Returns

- 0 : OK (name has menuitem's pathname)
- -1 : item not found (name="")
- -2 : 'name' not large enough (name="")

See also

[find_item\(\)](#)

31.79.3.15 menu() [1/2]

```
const Fl_Menu_Item * Fl_Menu_::menu ( ) const [inline]
```

Returns a pointer to the array of `Fl_Menu_Items`.

This will either be the value passed to `menu(value)` or the private copy.

See also

[size\(\)](#) – returns the [size](#) of the `Fl_Menu_Item` array.

Example: How to walk the array:

```
for ( int t=0; t<menubar->size(); t++ ) { // walk array of items
    const Fl_Menu_Item &item = menubar->menu()[t]; // get each item
    fprintf(stderr, "item #%d -- label=%s, value=%s type=%s\n",
        t,
        item.label() ? item.label() : "(Null)", // menu terminators have NULL labels
        (item.flags & FL_MENU_VALUE) ? "set" : "clear", // value of toggle or radio items
        (item.flags & FL_SUBMENU) ? "Submenu" : "Item"); // see if item is a submenu or actual item
}
```

31.79.3.16 menu() [2/2]

```
void Fl_Menu_::menu (
    const Fl_Menu_Item * m )
```

Sets the menu array pointer directly.

If the old menu is private it is deleted. NULL is allowed and acts the same as a zero-length menu. If you try to modify the array (with [add\(\)](#), [replace\(\)](#), or [remove\(\)](#)) a private copy is automatically done.

31.79.3.17 mode() [1/2]

```
int Fl_Menu_::mode (
    int i ) const [inline]
```

Gets the flags of item `i`.

For a list of the flags, see [Fl_Menu_Item](#).

31.79.3.18 mode() [2/2]

```
void Fl_Menu_::mode (
    int i,
    int fl ) [inline]
```

Sets the flags of item `i`.

For a list of the flags, see [Fl_Menu_Item](#).

31.79.3.19 mvalue()

```
const Fl_Menu_Item * Fl_Menu_::mvalue ( ) const [inline]
```

Returns a pointer to the last menu item that was picked.

31.79.3.20 picked()

```
const Fl_Menu_Item * Fl_Menu_::picked (
    const Fl_Menu_Item * v )
```

When user picks a menu item, call this.

It will do the callback. Unfortunately this also casts away const for the checkboxes, but this was necessary so non-checkbox menus can really be declared const...

31.79.3.21 remove()

```
void Fl_Menu_::remove (
    int i )
```

Deletes item *i* from the menu.

If the menu array was directly set with menu(x) then [copy\(\)](#) is done to make a private array.

No items must be removed from a menu during a callback to the same menu.

Parameters

<i>i</i>	index into menu array
----------	-----------------------

31.79.3.22 replace()

```
void Fl_Menu_::replace (
    int i,
    const char * str )
```

Changes the text of item *i*.

This is the only way to get slash into an [add\(\)](#)'ed menu item. If the menu array was directly set with menu(x) then [copy\(\)](#) is done to make a private array.

Parameters

<i>i</i>	index into menu array
<i>str</i>	new label for menu item at index <i>i</i>

31.79.3.23 size()

```
int Fl_Menu_::size ( ) const
```

This returns the number of [Fl_Menu_Item](#) structures that make up the menu, correctly counting submenus.

This includes the "terminator" item at the end. To copy a menu array you need to copy [size\(\)*sizeof\(Fl_Menu_Item\)](#) bytes. If the menu is NULL this returns zero (an empty menu will return 1).

31.79.3.24 test_shortcut()

```
const Fl_Menu_Item * Fl_Menu_::test_shortcut ( ) [inline]
```

Returns the menu item with the entered shortcut (key value).

This searches the complete [menu\(\)](#) for a shortcut that matches the entered key value. It must be called for a FL_KEYBOARD or FL_SHORTCUT event.

If a match is found, the menu's callback will be called.

Returns

matched [Fl_Menu_Item](#) or NULL.

31.79.3.25 text() [1/2]

```
const char * Fl_Menu_::text ( ) const [inline]
```

Returns the title of the last item chosen.

31.79.3.26 text() [2/2]

```
const char * Fl_Menu_::text (
    int i ) const [inline]
```

Returns the title of item i.

31.79.3.27 textcolor()

```
Fl_Color Fl_Menu_::textcolor ( ) const [inline]
```

Get the current color of menu item labels.

31.79.3.28 textfont() [1/2]

```
Fl_Font Fl_Menu_::textfont ( ) const [inline]
```

Gets the current font of menu item labels.

31.79.3.29 textfont() [2/2]

```
void Fl_Menu_::textfont (
    Fl_Font c ) [inline]
```

Sets the current font of menu item labels.

31.79.3.30 textsize() [1/2]

```
Fl_Fontsize Fl_Menu_::textsize ( ) const [inline]
```

Gets the font size of menu item labels.

31.79.3.31 textsize() [2/2]

```
void Fl_Menu_::textsize (
    Fl_Fontsize c ) [inline]
```

Sets the font size of menu item labels.

31.79.3.32 value() [1/3]

```
int Fl_Menu_::value ( ) const [inline]
```

Returns the index into [menu\(\)](#) of the last item chosen by the user. It is zero initially.

31.79.3.33 value() [2/3]

```
int Fl_Menu_::value (
    const Fl_Menu_Item * m )
```

The value is the index into [menu\(\)](#) of the last item chosen by the user.

It is zero initially. You can set it as an integer, or set it with a pointer to a menu item. The set routines return non-zero if the new value is different than the old one.

31.79.3.34 value() [3/3]

```
int Fl_Menu_::value (
    int i ) [inline]
```

The value is the index into [menu\(\)](#) of the last item chosen by the user.

It is zero initially. You can set it as an integer, or set it with a pointer to a menu item. The set routines return non-zero if the new value is different than the old one.

The documentation for this class was generated from the following files:

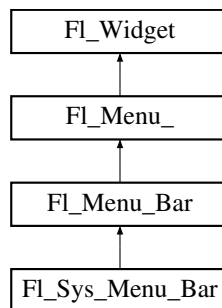
- [Fl_Menu_.H](#)
- [Fl_Menu_.cxx](#)
- [Fl_Menu_add.cxx](#)
- [Fl_Menu_global.cxx](#)

31.80 FI_Menu_Bar Class Reference

This widget provides a standard menubar interface.

```
#include <Fl_Menu_Bar.H>
```

Inheritance diagram for [Fl_Menu_Bar](#):

**Public Member Functions**

- [Fl_Menu_Bar](#) (int X, int Y, int W, int H, const char *l=0)
Creates a new [Fl_Menu_Bar](#) widget using the given position, size, and label string.
- int [handle](#) (int)
Handles the specified event.

Protected Member Functions

- void [draw](#) ()
Draws the widget.

Additional Inherited Members**31.80.1 Detailed Description**

This widget provides a standard menubar interface.

Usually you will put this widget along the top edge of your window. The height of the widget should be 30 for the menu titles to draw correctly with the default font.

The items on the bar and the menus they bring up are defined by a single [Fl_Menu_Item](#) array. Because a [Fl_Menu_Item](#) array defines a hierarchy, the top level menu defines the items in the menubar, while the submenus define the pull-down menus. Sub-sub menus and lower pop up to the right of the submenus.

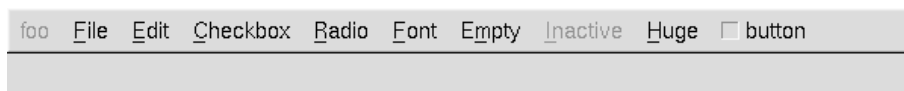


Figure 31.19 menubar

If there is an item in the top menu that is not a title of a submenu, then it acts like a "button" in the menubar. Clicking on it will pick it.

When the user clicks a menu item, `value()` is set to that item and then:

- The item's callback is done if one has been set; the `Fl_Menu_Bar` is passed as the `Fl_Widget*` argument, along with any userdata configured for the callback.
- If the item does not have a callback, the `Fl_Menu_Bar`'s callback is done instead, along with any userdata configured for the callback. The callback can determine which item was picked using `value()`, `mvalue()`, `item_pathname()`, etc.

Submenus will also pop up in response to shortcuts indicated by putting a '&' character in the name field of the menu item. If you put a '&' character in a top-level "button" then the shortcut picks it. The '&' character in submenus is ignored until the menu is popped up.

Typing the `shortcut()` of any of the menu items will cause callbacks exactly the same as when you pick the item with the mouse.

31.80.2 Constructor & Destructor Documentation

31.80.2.1 Fl_Menu_Bar()

```
Fl_Menu_Bar::Fl_Menu_Bar (
    int X,
    int Y,
    int W,
    int H,
    const char * l = 0 )
```

Creates a new `Fl_Menu_Bar` widget using the given position, size, and label string.

The default boxtype is `FL_UP_BOX`.

The constructor sets `menu()` to `NULL`. See `Fl_Menu_` for the methods to set or change the menu.

`labelsize()`, `labelfont()`, and `labelcolor()` are used to control how the menubar items are drawn. They are initialized from the `Fl_Menu` static variables, but you can change them if desired.

`label()` is ignored unless you change `align()` to put it outside the menubar.

The destructor removes the `Fl_Menu_Bar` widget and all of its menu items.

31.80.3 Member Function Documentation

31.80.3.1 draw()

```
void Fl_Menu_Bar::draw ( ) [protected], [virtual]
```

Draws the widget.

Never call this function directly. FLTK will schedule redrawing whenever needed. If your widget must be redrawn as soon as possible, call `redraw()` instead.

Override this function to draw your own widgets.

If you ever need to call another widget's draw method *from within your own `draw()` method*, e.g. for an embedded scrollbar, you can do it (because `draw()` is virtual) like this:

```
Fl_Widget *s = &scroll; // scroll is an embedded Fl_Scrollbar
s->draw(); // calls Fl_Scrollbar::draw()
```

Implements `Fl_Widget`.

Reimplemented in `Fl_Sys_Menu_Bar`.

31.80.3.2 handle()

```
int Fl_Menu_Bar::handle (
    int event ) [virtual]
```

Handles the specified event.

You normally don't call this method directly, but instead let FLTK do it when the user interacts with the widget.

When implemented in a widget, this function must return 0 if the widget does not use the event or 1 otherwise.

Most of the time, you want to call the inherited [handle\(\)](#) method in your overridden method so that you don't short-circuit events that you don't handle. In this last case you should return the callee retval.

Parameters

<code>in</code>	<code>event</code>	the kind of event received
-----------------	--------------------	----------------------------

Return values

<code>0</code>	if the event was not used or understood
<code>1</code>	if the event was used and can be deleted

See also

[FI_Event](#)

Reimplemented from [FI_Widget](#).

The documentation for this class was generated from the following files:

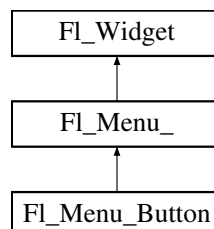
- [FI_Menu_Bar.H](#)
- [FI_Menu_Bar.cxx](#)

31.81 FI_Menu_Button Class Reference

This is a button that when pushed pops up a menu (or hierarchy of menus) defined by an array of [FI_Menu_Item](#) objects.

```
#include <FI_Menu_Button.H>
```

Inheritance diagram for [FI_Menu_Button](#):



Public Types

- enum [popup_buttons](#) {
[POPUP1](#) = 1 , [POPUP2](#) , [POPUP12](#) , [POPUP3](#) ,
[POPUP13](#) , [POPUP23](#) , [POPUP123](#) }

indicate what mouse buttons pop up the menu.

Public Member Functions

- [FI_Menu_Button](#) (int, int, int, int, const char *=0)
Creates a new [FI_Menu_Button](#) widget using the given position, size, and label string.
- int [handle](#) (int)
Handles the specified event.
- const [FI_Menu_Item](#) * [popup](#) ()
Act exactly as though the user clicked the button or typed the shortcut key.

Protected Member Functions

- void `draw ()`
Draws the widget.

Additional Inherited Members

31.81.1 Detailed Description

This is a button that when pushed pops up a menu (or hierarchy of menus) defined by an array of `Fl_Menu_Item` objects.



Figure 31.20 menu_button

Normally any mouse button will pop up a menu and it is lined up below the button as shown in the picture. However an `Fl_Menu_Button` may also control a pop-up menu. This is done by setting the `type()`. If `type()` is zero a normal menu button is produced. If it is nonzero then this is a pop-up menu. The bits in `type()` indicate what mouse buttons pop up the menu (see `Fl_Menu_Button::popup_buttons`).

The menu will also pop up in response to shortcuts indicated by putting a '&' character in the `label()`.

Typing the `shortcut()` of any of the menu items will cause callbacks exactly the same as when you pick the item with the mouse. The '&' character in menu item names are only looked at when the menu is popped up, however.

When the user clicks a menu item, `value()` is set to that item and then:

- The item's callback is done if one has been set; the `Fl_Menu_Button` is passed as the `Fl_Widget*` argument, along with any userdata configured for the callback.
- If the item does not have a callback, the `Fl_Menu_Button`'s callback is done instead, along with any userdata configured for it. The callback can determine which item was picked using `value()`, `mvalue()`, `item_pathname()`, etc.

31.81.2 Member Enumeration Documentation

31.81.2.1 popup_buttons

enum `Fl_Menu_Button::popup_buttons`

indicate what mouse buttons pop up the menu.

Values for `type()` used to indicate what mouse buttons pop up the menu. `Fl_Menu_Button::POPUP3` is usually what you want.

Enumerator

POPUP1	pops up with the mouse 1st button.
POPUP2	pops up with the mouse 2nd button.
POPUP12	pops up with the mouse 1st or 2nd buttons.
POPUP3	pops up with the mouse 3rd button.
POPUP13	pops up with the mouse 1st or 3rd buttons.
POPUP23	pops up with the mouse 2nd or 3rd buttons.
POPUP123	pops up with any mouse button.

31.81.3 Constructor & Destructor Documentation

31.81.3.1 Fl_Menu_Button()

```
Fl_Menu_Button::Fl_Menu_Button (
    int X,
    int Y,
    int W,
    int H,
    const char * l = 0 )
```

Creates a new [Fl_Menu_Button](#) widget using the given position, size, and label string. The default boxtype is FL_UP_BOX.

The constructor sets [menu\(\)](#) to NULL. See [Fl_Menu_](#) for the methods to set or change the menu.

31.81.4 Member Function Documentation

31.81.4.1 draw()

```
void Fl_Menu_Button::draw ( ) [protected], [virtual]
```

Draws the widget.

Never call this function directly. FLTK will schedule redrawing whenever needed. If your widget must be redrawn as soon as possible, call [redraw\(\)](#) instead.

Override this function to draw your own widgets.

If you ever need to call another widget's draw method *from within your own [draw\(\)](#) method*, e.g. for an embedded scrollbar, you can do it (because [draw\(\)](#) is virtual) like this:

```
Fl_Widget *s = &scroll;           // scroll is an embedded Fl_Scrollbar
s->draw();                         // calls Fl_Scrollbar::draw()
```

Implements [Fl_Widget](#).

31.81.4.2 handle()

```
int Fl_Menu_Button::handle (
    int event ) [virtual]
```

Handles the specified event.

You normally don't call this method directly, but instead let FLTK do it when the user interacts with the widget.

When implemented in a widget, this function must return 0 if the widget does not use the event or 1 otherwise.

Most of the time, you want to call the inherited [handle\(\)](#) method in your overridden method so that you don't short-circuit events that you don't handle. In this last case you should return the callee retval.

Parameters

in	<i>event</i>	the kind of event received
----	--------------	----------------------------

Return values

0	if the event was not used or understood
1	if the event was used and can be deleted

See also

[FI_Event](#)

Reimplemented from [FI_Widget](#).

31.81.4.3 popup()

```
const FI_Menu_Item * FI_Menu_Button::popup ( )
```

Act exactly as though the user clicked the button or typed the shortcut key.

The menu appears, it waits for the user to pick an item, and if they pick one it sets [value\(\)](#) and does the callback or sets [changed\(\)](#) as described above. The menu item is returned or NULL if the user dismisses the menu.

The documentation for this class was generated from the following files:

- [FI_Menu_Button.H](#)
- [FI_Menu_Button.cxx](#)

31.82 FI_Menu_Item Struct Reference

The [FI_Menu_Item](#) structure defines a single menu item that is used by the [FI_Menu_](#) class.

```
#include <FI_Menu_Item.H>
```

Public Member Functions

- void **activate** ()
Allows a menu item to be picked.
- int **active** () const
Gets whether or not the item can be picked.
- int **activevisible** () const
Returns non 0 if FL_INACTIVE and FL_INVISIBLE are cleared, 0 otherwise.
- int **add** (const char *, int [shortcut](#), [FI_Callback](#) *, void *=0, int=0)
Adds a menu item.
- int **add** (const char *a, const char *b, [FI_Callback](#) *c, void *d=0, int e=0)
See int add(const char, int shortcut, FI_Callback*, void*, int)*
- long **argument** () const
Gets the [user_data\(\)](#) argument that is sent to the callback function.
- void **argument** (long v)
Sets the [user_data\(\)](#) argument that is sent to the callback function.
- [FI_Callback_p](#) **callback** () const
Returns the callback function that is set for the menu item.
- void **callback** ([FI_Callback](#) *c)
Sets the menu item's callback function.
- void **callback** ([FI_Callback](#) *c, void *p)
Sets the menu item's callback function and userdata() argument.
- void **callback** ([FI_Callback0](#) *c)
Sets the menu item's callback function.
- void **callback** ([FI_Callback1](#) *c, long p=0)
Sets the menu item's callback function and userdata() argument.

- void **check** ()
back compatibility only
- int **checkbox** () const
Returns true if a checkbox will be drawn next to this item.
- int **checked** () const
back compatibility only
- void **clear** ()
Turns the check or radio item "off" for the menu item.
- void **deactivate** ()
Prevents a menu item from being picked.
- void **do_callback** (FL_Widget *o) const
Calls the FL_Menu_Item item's callback, and provides the FL_Widget argument.
- void **do_callback** (FL_Widget *o, long arg) const
Calls the FL_Menu_Item item's callback, and provides the FL_Widget argument.
- void **do_callback** (FL_Widget *o, void *arg) const
Calls the FL_Menu_Item item's callback, and provides the FL_Widget argument.
- void **draw** (int x, int y, int w, int h, const FL_Menu_ *, int t=0) const
Draws the menu item in bounding box x,y,w,h, optionally selects the item.
- const FL_Menu_Item * **find_shortcut** (int *ip=0, const bool require_alt=false) const
Search only the top level menu for a shortcut.
- FL_Menu_Item * **first** ()
Returns the first menu item, same as next(0).
- const FL_Menu_Item * **first** () const
Returns the first menu item, same as next(0).
- void **hide** ()
Hides an item in the menu.
- void **image** (FL_Image &a)
compatibility api for FLUID, same as a.label(this)
- void **image** (FL_Image *a)
compatibility api for FLUID, same as a->label(this)
- int **insert** (int, const char *, int, FL_Callback *, void **=0, int=0)
Inserts an item at position index.
- const char * **label** () const
Returns the title of the item.
- void **label** (const char *a)
See const char FL_Menu_Item::label() const*

- void **label** (FL_Labeltype a, const char *b)
See const char FL_Menu_Item::label() const*

- FL_Color **labelcolor** () const
Gets the menu item's label color.
- void **labelcolor** (FL_Color a)
Sets the menu item's label color.
- FL_Font **labelfont** () const
Gets the menu item's label font.
- void **labelfont** (FL_Font a)
Sets the menu item's label font.
- FL_Fontsize **labelsize** () const
Gets the label font pixel size/height.
- void **labelsize** (FL_Fontsize a)

- Sets the label font pixel size/height.*

 - `FI_Labeltype labeltype () const`

Returns the menu item's labeltype.
- `void labeltype (FI_Labeltype a)`
- Sets the menu item's labeltype.*
- `int measure (int *h, const FI_Menu_ *) const`
- Measures width of label, including effect of & characters.*
- `FI_Menu_Item * next (int i=1)`
- Advances a pointer by n items through a menu array, skipping the contents of submenus and invisible items.*
- `const FI_Menu_Item * next (int=1) const`
- Advance a pointer by n items through a menu array, skipping the contents of submenus and invisible items.*
- `const FI_Menu_Item * popup (int X, int Y, const char *title=0, const FI_Menu_Item *picked=0, const FI_Menu_Item *s=0) const`
- This method is called by widgets that want to display menus.*
- `const FI_Menu_Item * pulldown (int X, int Y, int W, int H, const FI_Menu_Item *picked=0, const FI_Menu_Item *s=0, const FI_Menu_Item *title=0, int menubar=0) const`
- Pulldown() is similar to popup(), but a rectangle is provided to position the menu.*
- `int radio () const`
- Returns true if this item is a radio item.*
- `void set ()`
- Turns the check or radio item "on" for the menu item.*
- `void setonly ()`
- Turns the radio item "on" for the menu item and turns "off" adjacent radio items set.*
- `int shortcut () const`
- Gets what key combination shortcut will trigger the menu item.*
- `void shortcut (int s)`
- Sets exactly what key combination will trigger the menu item.*
- `void show ()`
- Makes an item visible in the menu.*
- `int size () const`
- Size of the menu starting from this menu item.*
- `int submenu () const`
- Returns true if either FL_SUBMENU or FL_SUBMENU_POINTER is on in the flags.*
- `const FI_Menu_Item * test_shortcut () const`
- This is designed to be called by a widgets handle() method in response to a FL_SHORTCUT event.*
- `void uncheck ()`
- back compatibility only*
- `void * user_data () const`
- Gets the user_data() argument that is sent to the callback function.*
- `void user_data (void *v)`
- Sets the user_data() argument that is sent to the callback function.*
- `int value () const`
- Returns the current value of the check or radio item.*
- `int visible () const`
- Gets the visibility of an item.*

Public Attributes

- [Fl_Callback](#) * **callback_**
menu item callback
- int **flags**
menu item flags like FL_MENU_TOGGLE, FL_MENU_RADIO
- [Fl_Color](#) **labelcolor_**
menu item text color
- [Fl_Font](#) **labelfont_**
which font for this menu item text
- [Fl_Fontsize](#) **labelsize_**
size of menu item text
- [uchar](#) **labeltype_**
how the menu item text looks like
- int **shortcut_**
menu item shortcut
- const char * **text**
menu item text, returned by [label\(\)](#)
- void * **user_data_**
menu item user_data for the menu's callback

31.82.1 Detailed Description

The [Fl_Menu_Item](#) structure defines a single menu item that is used by the [Fl_Menu_](#) class.

```
struct Fl_Menu_Item {
  const char*  text;           // label()
  ulong       shortcut_;
  Fl_Callback* callback_;
  void*       user_data_;
  int         flags;
  uchar       labeltype_;
  uchar       labelfont_;
  uchar       labelsize_;
  uchar       labelcolor_;
};
enum { // values for flags:
  FL_MENU_INACTIVE   = 1,    // Deactivate menu item (gray out)
  FL_MENU_TOGGLE    = 2,    // Item is a checkbox toggle (shows checkbox for on/off state)
  FL_MENU_VALUE     = 4,    // The on/off state for checkbox/radio buttons (if set, state is 'on')
  FL_MENU_RADIO     = 8,    // Item is a radio button (one checkbox of many can be on)
  FL_MENU_INVISIBLE = 0x10, // Item will not show up (shortcut will work)
  FL_SUBMENU_POINTER = 0x20, // Indicates user_data() is a pointer to another menu array
  FL_SUBMENU        = 0x40, // This item is a submenu to other items
  FL_MENU_DIVIDER   = 0x80, // Creates divider line below this item. Also ends a group of radio buttons.
  FL_MENU_HORIZONTAL = 0x100 // ??? -- reserved
};
```

Typically menu items are statically defined; for example:

```
Fl_Menu_Item popup[] = {
  {"&alpha",    FL_ALT+'a', the_cb, (void*)1},
  {"&beta",     FL_ALT+'b', the_cb, (void*)2},
  {"&gamma",    FL_ALT+'c', the_cb, (void*)3, FL_MENU_DIVIDER},
  {"&strange",  0,         strange_cb},
  {"&charm",    0,         charm_cb},
  {"&truth",    0,         truth_cb},
  {"&beauty",   0,         beauty_cb},
  {"sub&menu",  0,         0, 0, FL_SUBMENU},
  {"one"},
  {"two"},
  {"three"},
  {},
  {"inactive", FL_ALT+'i', 0, 0, FL_MENU_INACTIVE|FL_MENU_DIVIDER},
  {"invisible",FL_ALT+'i', 0, 0, FL_MENU_INVISIBLE},
  {"check",    FL_ALT+'i', 0, 0, FL_MENU_TOGGLE|FL_MENU_VALUE},
  {"box",      FL_ALT+'i', 0, 0, FL_MENU_TOGGLE},
  {}
};
```

produces:

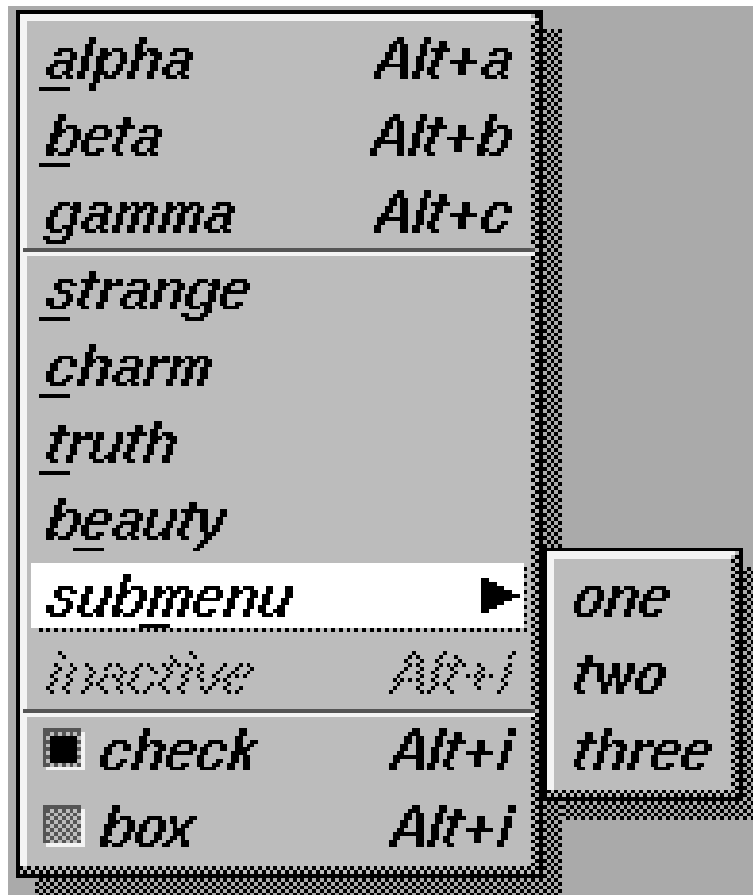


Figure 31.21 menu

A submenu title is identified by the bit `FL_SUBMENU` in the flags field, and ends with a `label()` that is `NULL`. You can nest menus to any depth. A pointer to the first item in the submenu can be treated as an `Fl_Menu` array itself. It is also possible to make separate submenu arrays with `FL_SUBMENU_POINTER` flags. You should use the method functions to access structure members and not access them directly to avoid compatibility problems with future releases of FLTK.

31.82.2 Member Function Documentation

31.82.2.1 `add()`

```
int Fl_Menu_Item::add (
    const char * mytext,
    int sc,
    Fl_Callback * cb,
    void * data = 0,
    int myflags = 0 )
```

Adds a menu item.

The text is split at `'/'` characters to automatically produce submenus (actually a totally unnecessary feature as you can now add submenu titles directly by setting `FL_SUBMENU` in the flags).

Returns

the index into the `menu()` array, where the entry was added

See also

[Fl_Menu_Item::insert\(int, const char*, int, Fl_Callback*, void*, int\)](#)

31.82.2.2 argument() [1/2]

```
long Fl_Menu_Item::argument ( ) const [inline]
```

Gets the [user_data\(\)](#) argument that is sent to the callback function.

For convenience you can also define the callback as taking a long argument. This method casts the stored [user_data\(\)](#) argument to long and returns it as a *long* value.

31.82.2.3 argument() [2/2]

```
void Fl_Menu_Item::argument (
    long v ) [inline]
```

Sets the [user_data\(\)](#) argument that is sent to the callback function.

For convenience you can also define the callback as taking a long argument. This method casts the given argument *v* to void* and stores it in the menu item's [userdata\(\)](#) member. This may not be portable to some machines.

31.82.2.4 callback() [1/5]

```
Fl_Callback_p Fl_Menu_Item::callback ( ) const [inline]
```

Returns the callback function that is set for the menu item.

Each item has space for a callback function and an argument for that function. Due to back compatibility, the [Fl_Menu_Item](#) itself is not passed to the callback, instead you have to get it by calling `((Fl_Menu_*)w)->mvalue()` where *w* is the widget argument.

31.82.2.5 callback() [2/5]

```
void Fl_Menu_Item::callback (
    Fl_Callback * c ) [inline]
```

Sets the menu item's callback function.

This method does not set the [userdata\(\)](#) argument.

See also

[Fl_Callback_p](#) [Fl_MenuItem::callback\(\)](#) const

31.82.2.6 callback() [3/5]

```
void Fl_Menu_Item::callback (
    Fl_Callback * c,
    void * p ) [inline]
```

Sets the menu item's callback function and [userdata\(\)](#) argument.

See also

[Fl_Callback_p](#) [Fl_MenuItem::callback\(\)](#) const

31.82.2.7 callback() [4/5]

```
void Fl_Menu_Item::callback (
    Fl_Callback0 * c ) [inline]
```

Sets the menu item's callback function.

This method does not set the [userdata\(\)](#) argument.

See also

[Fl_Callback_p](#) [Fl_MenuItem::callback\(\)](#) const

31.82.2.8 callback() [5/5]

```
void Fl_Menu_Item::callback (
    Fl_Callback1 * c,
    long p = 0 ) [inline]
```

Sets the menu item's callback function and userdata() argument.

This method does not set the userdata() argument. The argument `is` is cast to `void*` and stored as the `userdata()` for the menu item's callback function.

See also

[Fl_Callback_p](#) `Fl_MenuItem::callback()` `const`

31.82.2.9 check()

```
void Fl_Menu_Item::check ( ) [inline]
back compatibility only
```

Deprecated

31.82.2.10 checkbox()

```
int Fl_Menu_Item::checkbox ( ) const [inline]
```

Returns true if a checkbox will be drawn next to this item.

This is true if `FL_MENU_TOGGLE` or `FL_MENU_RADIO` is set in the flags.

31.82.2.11 checked()

```
int Fl_Menu_Item::checked ( ) const [inline]
back compatibility only
```

Deprecated

31.82.2.12 deactivate()

```
void Fl_Menu_Item::deactivate ( ) [inline]
```

Prevents a menu item from being picked.

Note that this will also cause the menu item to appear grayed-out.

31.82.2.13 do_callback() [1/3]

```
void Fl_Menu_Item::do_callback (
    Fl_Widget * o ) const [inline]
```

Calls the `Fl_Menu_Item` item's callback, and provides the `Fl_Widget` argument.

The callback is called with the stored `user_data()` as its second argument. You must first check that `callback()` is non-zero before calling this.

31.82.2.14 do_callback() [2/3]

```
void Fl_Menu_Item::do_callback (
    Fl_Widget * o,
    long arg ) const [inline]
```

Calls the `Fl_Menu_Item` item's callback, and provides the `Fl_Widget` argument.

This call overrides the callback's second argument with the given value `arg`. `long arg` is cast to `void*` when calling the callback. You must first check that `callback()` is non-zero before calling this.

31.82.2.15 do_callback() [3/3]

```
void Fl_Menu_Item::do_callback (
    Fl_Widget * o,
    void * arg ) const [inline]
```

Calls the [Fl_Menu_Item](#) item's callback, and provides the [Fl_Widget](#) argument.

This call overrides the callback's second argument with the given value `arg`. You must first check that `callback()` is non-zero before calling this.

31.82.2.16 find_shortcut()

```
const Fl_Menu_Item * Fl_Menu_Item::find_shortcut (
    int * ip = 0,
    const bool require_alt = false ) const
```

Search only the top level menu for a shortcut.

Either `&x` in the label or the shortcut fields are used.

This tests the current event, which must be an `FL_KEYBOARD` or `FL_SHORTCUT`, against a shortcut value.

Parameters

<code>ip</code>	returns the index of the item, if <code>ip</code> is not NULL.
<code>require_alt</code>	if true: match only if Alt key is pressed.

Returns

found [Fl_Menu_Item](#) or NULL

31.82.2.17 insert()

```
int Fl_Menu_Item::insert (
    int index,
    const char * mytext,
    int sc,
    Fl_Callback * cb,
    void * data = 0,
    int myflags = 0 )
```

Inserts an item at position `index`.

If `index` is -1, the item is added the same way as [Fl_Menu_Item::add\(\)](#).

If 'mytext' contains any un-escaped front slashes (`/`), it's assumed a menu pathname is being specified, and the value of `index` will be ignored.

In all other aspects, the behavior of [insert\(\)](#) is the same as [add\(\)](#).

Parameters

in	<code>index</code>	insert new items here
in	<code>mytext</code>	new label string, details see above
in	<code>sc</code>	keyboard shortcut for new item
in	<code>cb</code>	callback function for new item
in	<code>data</code>	user data for new item
in	<code>myflags</code>	menu flags as described in Fl_Menu_Item

Returns

the index into the `menu()` array, where the entry was added

31.82.2.18 label()

```
const char * Fl_Menu_Item::label ( ) const [inline]
```

Returns the title of the item.

A NULL here indicates the end of the menu (or of a submenu). A '&' in the item will print an underscore under the next letter, and if the menu is popped up that letter will be a "shortcut" to pick that item. To get a real '&' put two in a row.

31.82.2.19 labelcolor() [1/2]

```
Fl_Color Fl_Menu_Item::labelcolor ( ) const [inline]
```

Gets the menu item's label color.

This color is passed to the `labeltype` routine, and is typically the color of the label text. This defaults to `FL_BLACK`. If this color is not black fltk will **not** use overlay bitplanes to draw the menu - this is so that images put in the menu draw correctly.

31.82.2.20 labelcolor() [2/2]

```
void Fl_Menu_Item::labelcolor (
    Fl_Color a ) [inline]
```

Sets the menu item's label color.

See also

[Fl_Color Fl_Menu_Item::labelcolor\(\) const](#)

31.82.2.21 labelfont() [1/2]

```
Fl_Font Fl_Menu_Item::labelfont ( ) const [inline]
```

Gets the menu item's label font.

Fonts are identified by small 8-bit indexes into a table. See the enumeration list for predefined fonts. The default value is a Helvetica font. The function [Fl::set_font\(\)](#) can define new fonts.

31.82.2.22 labelfont() [2/2]

```
void Fl_Menu_Item::labelfont (
    Fl_Font a ) [inline]
```

Sets the menu item's label font.

Fonts are identified by small 8-bit indexes into a table. See the enumeration list for predefined fonts. The default value is a Helvetica font. The function [Fl::set_font\(\)](#) can define new fonts.

31.82.2.23 labeltype() [1/2]

```
Fl_Labeltype Fl_Menu_Item::labeltype ( ) const [inline]
```

Returns the menu item's labeltype.

A labeltype identifies a routine that draws the label of the widget. This can be used for special effects such as emboss, or to use the [label\(\)](#) pointer as another form of data such as a bitmap. The value `FL_NORMAL_LABEL` prints the label as text.

31.82.2.24 labeltype() [2/2]

```
void Fl_Menu_Item::labeltype (
    Fl_Labeltype a ) [inline]
```

Sets the menu item's labeltype.

A labeltype identifies a routine that draws the label of the widget. This can be used for special effects such as emboss, or to use the [label\(\)](#) pointer as another form of data such as a bitmap. The value `FL_NORMAL_LABEL` prints the label as text.

31.82.2.25 measure()

```
int Fl_Menu_Item::measure (
    int * hp,
    const Fl_Menu_ * m ) const
```

Measures width of label, including effect of & characters.
Optionally, can get height if hp is not NULL.

31.82.2.26 next() [1/2]

```
Fl_Menu_Item * Fl_Menu_Item::next (
    int i = 1 ) [inline]
```

Advances a pointer by n items through a menu array, skipping the contents of submenus and invisible items.
There are two calls so that you can advance through const and non-const data.

31.82.2.27 next() [2/2]

```
const Fl_Menu_Item * Fl_Menu_Item::next (
    int n = 1 ) const
```

Advance a pointer by n items through a menu array, skipping the contents of submenus and invisible items.
There are two calls so that you can advance through const and non-const data.

31.82.2.28 popup()

```
const Fl_Menu_Item * Fl_Menu_Item::popup (
    int X,
    int Y,
    const char * title = 0,
    const Fl_Menu_Item * picked = 0,
    const Fl_Menu_ * button = 0 ) const
```

This method is called by widgets that want to display menus.

The menu stays up until the user picks an item or dismisses it. The selected item (or NULL if none) is returned. *This does not do the callbacks or change the state of check or radio items.*

X,Y is the position of the mouse cursor, relative to the window that got the most recent event (usually you can pass [Fl::event_x\(\)](#) and [Fl::event_y\(\)](#) unchanged here).

title is a character string title for the menu. If non-zero a small box appears above the menu with the title in it.

The menu is positioned so the cursor is centered over the item picked. This will work even if picked is in a submenu. If picked is zero or not in the menu item table the menu is positioned with the cursor in the top-left corner.

button is a pointer to an [Fl_Menu_](#) from which the color and boxtypes for the menu are pulled. If NULL then defaults are used.

31.82.2.29 pulldown()

```
const Fl_Menu_Item * Fl_Menu_Item::pulldown (
    int X,
    int Y,
    int W,
    int H,
    const Fl_Menu_Item * initial_item = 0,
    const Fl_Menu_ * pbutton = 0,
    const Fl_Menu_Item * t = 0,
    int menubar = 0 ) const
```

Pulldown() is similar to [popup\(\)](#), but a rectangle is provided to position the menu.

The menu is made at least W wide, and the picked item is centered over the rectangle (like [Fl_Choice](#) uses). If picked is zero or not found, the menu is aligned just below the rectangle (like a pulldown menu).

The title and menubar arguments are used internally by the [Fl_Menu_Bar](#) widget.

31.82.2.30 radio()

```
int Fl_Menu_Item::radio ( ) const [inline]
```

Returns true if this item is a radio item.

When a radio button is selected all "adjacent" radio buttons are turned off. A set of radio items is delimited by an item that has `radio()` false, or by an item with `FL_MENU_DIVIDER` turned on.

31.82.2.31 set()

```
void Fl_Menu_Item::set ( ) [inline]
```

Turns the check or radio item "on" for the menu item.

Note that this does not turn off any adjacent radio items like `set_only()` does.

31.82.2.32 setonly()

```
void Fl_Menu_Item::setonly ( )
```

Turns the radio item "on" for the menu item and turns "off" adjacent radio items set.

Deprecated This method is dangerous if radio items are first in the menu. Use `Fl_Menu_::setonly(Fl_Menu_Item*)` instead.

31.82.2.33 shortcut()

```
void Fl_Menu_Item::shortcut (
    int s ) [inline]
```

Sets exactly what key combination will trigger the menu item.

The value is a logical 'or' of a key and a set of shift flags, for instance `FL_ALT+'a'` or `FL_ALT+FL_F+10` or just 'a'. A value of zero disables the shortcut.

The key can be any value returned by `Fl::event_key()`, but will usually be an ASCII letter. Use a lower-case letter unless you require the shift key to be held down.

The shift flags can be any set of values accepted by `Fl::event_state()`. If the bit is on that shift key must be pushed. Meta, Alt, Ctrl, and Shift must be off if they are not in the shift flags (zero for the other bits indicates a "don't care" setting).

31.82.2.34 size()

```
int Fl_Menu_Item::size ( ) const
```

Size of the menu starting from this menu item.

This method counts all menu items starting with `this` menu item, including all menu items in the same (sub)menu level, all nested submenus, **and** the terminating empty (0) menu item.

It does **not** count menu items referred to by `FL_SUBMENU_POINTER` menu items (except the single menu item with `FL_SUBMENU_POINTER`).

All menu items counted are consecutive in memory (one array).

Example:

```
schemechoice = new Fl_Choice(X+125,Y,140,25,"FLTK Scheme");
schemechoice->add("none");
schemechoice->add("plastic");
schemechoice->add("gtk+");
schemechoice->add("gleam");
printf("schemechoice->menu()->size() = %d\n", schemechoice->menu()->size());
```

Output:

```
schemechoice->menu()->size() = 5
```

31.82.2.35 submenu()

```
int Fl_Menu_Item::submenu ( ) const [inline]
```

Returns true if either `FL_SUBMENU` or `FL_SUBMENU_POINTER` is on in the flags.

`FL_SUBMENU` indicates an embedded submenu that goes from the next item through the next one with a `NULL label()`. `FL_SUBMENU_POINTER` indicates that `user_data()` is a pointer to another menu array.

31.82.2.36 test_shortcut()

```
const Fl_Menu_Item * Fl_Menu_Item::test_shortcut ( ) const
```

This is designed to be called by a widget's handle() method in response to a FL_SHORTCUT event.

If the current event matches one of the items shortcut, that item is returned. If the keystroke does not match any shortcuts then NULL is returned. This only matches the [shortcut\(\)](#) fields, not the letters in the title preceded by ' '.

31.82.2.37 uncheck()

```
void Fl_Menu_Item::uncheck ( ) [inline]
```

back compatibility only

Deprecated

31.82.2.38 value()

```
int Fl_Menu_Item::value ( ) const [inline]
```

Returns the current value of the check or radio item.

This is zero (0) if the menu item is not checked and non-zero otherwise. You should not rely on a particular value, only zero or non-zero.

Note

The returned value for a checked menu item as of FLTK 1.3.2 is FL_MENU_VALUE (4), but may be 1 in a future version.

The documentation for this struct was generated from the following files:

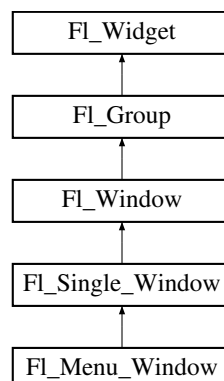
- [Fl_Menu_Item.H](#)
- [Fl_Menu.cxx](#)
- [Fl_Menu_.cxx](#)
- [Fl_Menu_add.cxx](#)

31.83 Fl_Menu_Window Class Reference

The [Fl_Menu_Window](#) widget is a window type used for menus.

```
#include <Fl_Menu_Window.H>
```

Inheritance diagram for [Fl_Menu_Window](#):



Public Member Functions

- void [clear_overlay](#) ()
Tells FLTK to use normal drawing planes instead of overlay planes.
- void [erase](#) ()

- Erases the window, does nothing if HAVE_OVERLAY is not defined config.h.*
- **Fl_Menu_Window** (int W, int H, const char *l=0)

Creates a new [Fl_Menu_Window](#) widget using the given size, and label string.
 - **Fl_Menu_Window** (int X, int Y, int W, int H, const char *l=0)

Creates a new [Fl_Menu_Window](#) widget using the given position, size, and label string.
 - void [flush](#) ()

Forces the window to be drawn, this window is also made current and calls [draw\(\)](#).
 - void [hide](#) ()

Removes the window from the screen.
 - unsigned int [overlay](#) ()

Tells if hardware overlay mode is set.
 - void [set_overlay](#) ()

Tells FLTK to use hardware overlay planes if they are available.
 - void [show](#) ()

Puts the window on the screen.
 - **~Fl_Menu_Window** ()

Destroys the window and all of its children.

Additional Inherited Members

31.83.1 Detailed Description

The [Fl_Menu_Window](#) widget is a window type used for menus.

By default the window is drawn in the hardware overlay planes if they are available so that the menu don't force the rest of the window to redraw.

31.83.2 Member Function Documentation

31.83.2.1 [clear_overlay\(\)](#)

```
void Fl_Menu_Window::clear_overlay ( ) [inline]
```

Tells FLTK to use normal drawing planes instead of overlay planes.

This is usually necessary if your menu contains multi-color pixmap.

31.83.2.2 [flush\(\)](#)

```
void Fl_Menu_Window::flush ( ) [virtual]
```

Forces the window to be drawn, this window is also made current and calls [draw\(\)](#).

Reimplemented from [Fl_Window](#).

31.83.2.3 [hide\(\)](#)

```
void Fl_Menu_Window::hide ( ) [virtual]
```

Removes the window from the screen.

If the window is already hidden or has not been shown then this does nothing and is harmless.

Reimplemented from [Fl_Window](#).

31.83.2.4 [set_overlay\(\)](#)

```
void Fl_Menu_Window::set_overlay ( ) [inline]
```

Tells FLTK to use hardware overlay planes if they are available.

31.83.2.5 show()

```
void Fl_Menu_Window::show ( ) [virtual]
```

Puts the window on the screen.

Usually (on X) this has the side effect of opening the display.

If the window is already shown then it is restored and raised to the top. This is really convenient because your program can call `show()` at any time, even if the window is already up. It also means that `show()` serves the purpose of `raise()` in other toolkits.

`Fl_Window::show(int argc, char **argv)` is used for top-level windows and allows standard arguments to be parsed from the command-line.

Note

For some obscure reasons `Fl_Window::show()` resets the current group by calling `Fl_Group::current(0)`. The comments in the code say "get rid of very common user bug: forgot end()". Although this is true it may have unwanted side effects if you `show()` an unrelated window (maybe for an error message or warning) while building a window or any other group widget.

Todo Check if we can remove resetting the current group in a later FLTK version (after 1.3.x). This may break "already broken" programs though if they rely on this "feature".

See also

[Fl_Window::show\(int argc, char **argv\)](#)

Reimplemented from [Fl_Window](#).

The documentation for this class was generated from the following files:

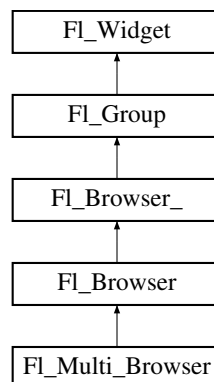
- `Fl_Menu_Window.H`
- `Fl_Menu_Window.cxx`

31.84 FI_Multi_Browser Class Reference

The `Fl_Multi_Browser` class is a subclass of `Fl_Browser` which lets the user select any set of the lines.

```
#include <Fl_Multi_Browser.H>
```

Inheritance diagram for `Fl_Multi_Browser`:



Public Member Functions

- `Fl_Multi_Browser` (int X, int Y, int W, int H, const char *L=0)
Creates a new `Fl_Multi_Browser` widget using the given position, size, and label string.

Additional Inherited Members

31.84.1 Detailed Description

The `Fl_Multi_Browser` class is a subclass of `Fl_Browser` which lets the user select any set of the lines.

The user interface is Macintosh style: clicking an item turns off all the others and selects that one, dragging selects all the items the mouse moves over, and ctrl + click (Cmd+click on the Mac OS platform) toggles the items. Shift + click extends the selection until the clicked item. This is different from how forms did it. Normally the callback is done when the user releases the mouse, but you can change this with [when\(\)](#). See [Fl_Browser](#) for methods to add and remove lines from the browser.

31.84.2 Constructor & Destructor Documentation

31.84.2.1 Fl_Multi_Browser()

```
Fl_Multi_Browser::Fl_Multi_Browser (
    int X,
    int Y,
    int W,
    int H,
    const char * L = 0 )
```

Creates a new [Fl_Multi_Browser](#) widget using the given position, size, and label string.

The default boxtype is FL_DOWN_BOX. The constructor specializes [Fl_Browser\(\)](#) by setting the type to FL_↔MULTI_BROWSER. The destructor destroys the widget and frees all memory that has been allocated.

The documentation for this class was generated from the following files:

- [Fl_Multi_Browser.H](#)
- [Fl_Browser.cxx](#)

31.85 Fl_Multi_Label Struct Reference

This struct allows multiple labels to be added to objects that might normally have only one label.

```
#include <Fl_Multi_Label.H>
```

Public Member Functions

- void [label](#) ([Fl_Menu_Item](#) *)
This method is used to associate a [Fl_Multi_Label](#) with a [Fl_Menu_Item](#).
- void [label](#) ([Fl_Widget](#) *)
This method is used to associate a [Fl_Multi_Label](#) with a [Fl_Widget](#).

Public Attributes

- const char * [labela](#)
Holds the "leftmost" of the two elements in the composite label.
- const char * [labelb](#)
Holds the "rightmost" of the two elements in the composite label.
- [uchar](#) [typea](#)
Holds the "type" of [labela](#).
- [uchar](#) [typeb](#)
Holds the "type" of [labelb](#).

31.85.1 Detailed Description

This struct allows multiple labels to be added to objects that might normally have only one label.

This struct allows a mixed text and/or graphics label to be applied to an object that would normally only have a single (usually text only) label.

Most regular FLTK widgets now support the ability to associate both images and text with a label but some special cases, notably the non-widget [Fl_Menu_Item](#) objects, do not. [Fl_Multi_Label](#) may be used to create menu items

that have an icon and text, which would not normally be possible for an [FI_Menu_Item](#). For example, [FI_Multi_Label](#) is used in the New->Code submenu in fluid, and others.

Each [FI_Multi_Label](#) holds two elements, `labela` and `labelb`; each may hold either a text label (`const char*`) or an image (`FI_Image*`). When displayed, `labela` is drawn first and `labelb` is drawn immediately to its right.

More complex labels might be constructed by setting `labelb` as another [FI_Multi_Label](#) and thus chaining up a series of label elements.

When assigning a label element to one of `labela` or `labelb`, they should be explicitly cast to (`const char*`) if they are not of that type already.

See also

[FI_Label](#) and [FI_Labeltype](#)

31.85.2 Member Data Documentation

31.85.2.1 `labela`

```
const char* Fl_Multi_Label::labela
```

Holds the "leftmost" of the two elements in the composite label.

Typically this would be assigned either a text string (`const char*`), a (`FI_Image*`) or a (`FI_Multi_Label*`).

31.85.2.2 `labelb`

```
const char* Fl_Multi_Label::labelb
```

Holds the "rightmost" of the two elements in the composite label.

Typically this would be assigned either a text string (`const char*`), a (`FI_Image*`) or a (`FI_Multi_Label*`).

31.85.2.3 `typea`

```
uchar Fl_Multi_Label::typea
```

Holds the "type" of `labela`.

Typically this is set to `FL_NORMAL_LABEL` for a text label, `_FL_IMAGE_LABEL` for an image (based on `FI_image`) or `_FL_MULTI_LABEL` if "chaining" multiple [FI_Multi_Label](#) elements together.

31.85.2.4 `typeb`

```
uchar Fl_Multi_Label::typeb
```

Holds the "type" of `labelb`.

Typically this is set to `FL_NORMAL_LABEL` for a text label, `_FL_IMAGE_LABEL` for an image (based on `FI_image`) or `_FL_MULTI_LABEL` if "chaining" multiple [FI_Multi_Label](#) elements together.

The documentation for this struct was generated from the following files:

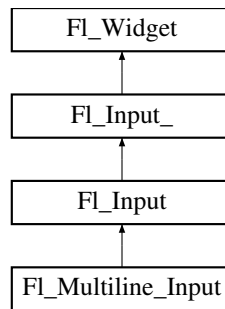
- `FI_Multi_Label.H`
- `FI_Multi_Label.cxx`

31.86 FI_Multiline_Input Class Reference

This input field displays '\n' characters as new lines rather than ^J, and accepts the Return, Tab, and up and down arrow keys.

```
#include <Fl_Multiline_Input.H>
```

Inheritance diagram for `FI_Multiline_Input`:



Public Member Functions

- [Fl_Multiline_Input](#) (int X, int Y, int W, int H, const char *l=0)

Creates a new [Fl_Multiline_Input](#) widget using the given position, size, and label string.

Additional Inherited Members

31.86.1 Detailed Description

This input field displays '\n' characters as new lines rather than ^J, and accepts the Return, Tab, and up and down arrow keys.

This is for editing multiline text.

This is far from the nirvana of text editors, and is probably only good for small bits of text, 10 lines at most. Note that this widget does not support scrollbars or per-character color control.

If you are presenting large amounts of text and need scrollbars or full color control of characters, you probably want [Fl_Text_Editor](#) instead.

In FLTK 1.3.x, the default behavior of the 'Tab' key was changed to support consistent focus navigation. To get the older FLTK 1.1.x behavior, set [Fl_Input_::tab_nav\(\)](#) to 0. Newer programs should consider using [Fl_Text_Editor](#).

31.86.2 Constructor & Destructor Documentation

31.86.2.1 Fl_Multiline_Input()

```

Fl_Multiline_Input::Fl_Multiline_Input (
    int X,
    int Y,
    int W,
    int H,
    const char * l = 0 )
  
```

Creates a new [Fl_Multiline_Input](#) widget using the given position, size, and label string.

The default boxtype is FL_DOWN_BOX.

Inherited destructor destroys the widget and any value associated with it.

The documentation for this class was generated from the following files:

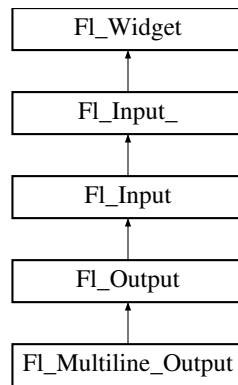
- [Fl_Multiline_Input.H](#)
- [Fl_Input.cxx](#)

31.87 Fl_Multiline_Output Class Reference

This widget is a subclass of [Fl_Output](#) that displays multiple lines of text.

```
#include <Fl_Multiline_Output.H>
```

Inheritance diagram for [Fl_Multiline_Output](#):



Public Member Functions

- [Fl_Multiline_Output](#) (int X, int Y, int W, int H, const char *l=0)
Creates a new [Fl_Multiline_Output](#) widget using the given position, size, and label string.

Additional Inherited Members

31.87.1 Detailed Description

This widget is a subclass of [Fl_Output](#) that displays multiple lines of text.

It also displays tab characters as whitespace to the next column.

Note that this widget does not support scrollbars, or per-character color control.

If you are presenting large amounts of read-only text and need scrollbars, or full color control of characters, then use [Fl_Text_Display](#). If you want to display HTML text, use [Fl_Help_View](#).

31.87.2 Constructor & Destructor Documentation

31.87.2.1 Fl_Multiline_Output()

```

Fl_Multiline_Output::Fl_Multiline_Output (
    int X,
    int Y,
    int W,
    int H,
    const char * l = 0 )
  
```

Creates a new [Fl_Multiline_Output](#) widget using the given position, size, and label string.

The default boxtype is FL_DOWN_BOX.

Inherited destructor destroys the widget and any value associated with it.

The documentation for this class was generated from the following files:

- Fl_Multiline_Output.H
- Fl_Input.cxx

31.88 Fl_Native_File_Chooser Class Reference

This class lets an FLTK application easily and consistently access the operating system's native file chooser.

```
#include <Fl_Native_File_Chooser.H>
```

Public Types

- enum [Option](#) {
[NO_OPTIONS](#) = 0x0000 , [SAVEAS_CONFIRM](#) = 0x0001 , [NEW_FOLDER](#) = 0x0002 , [PREVIEW](#) = 0x0004 ,
[USE_FILTER_EXT](#) = 0x0008 }

- enum `Type` {
`BROWSE_FILE` = 0 , `BROWSE_DIRECTORY` , `BROWSE_MULTI_FILE` , `BROWSE_MULTI_DIRECTORY` ,
`BROWSE_SAVE_FILE` , `BROWSE_SAVE_DIRECTORY` }

Public Member Functions

- int `count` () const
Returns the number of filenames (or directory names) the user selected.
- const char * `directory` () const
Returns the current preset `directory()` value.
- void `directory` (const char *val)
Preset the directory the browser will show when opened.
- const char * `errmsg` () const
Returns a system dependent error message for the last method that failed.
- const char * `filename` () const
Return the filename the user chose.
- const char * `filename` (int i) const
Return one of the filenames the user selected.
- const char * `filter` () const
Returns the filter string last set.
- void `filter` (const char *f)
Sets the filename filters used for browsing.
- int `filter_value` () const
Returns which filter value was last selected by the user.
- void `filter_value` (int i)
Sets which filter will be initially selected.
- int `filters` () const
Gets how many filters were available, not including "All Files".
- `FI_Native_File_Chooser` (int val=`BROWSE_FILE`)
The constructor.
- int `options` () const
Gets the platform specific `FI_Native_File_Chooser::Option` flags.
- void `options` (int o)
Sets the platform specific chooser options to `val`.
- const char * `preset_file` () const
Get the preset filename.
- void `preset_file` (const char *f)
Sets the default filename for the chooser.
- int `show` ()
Post the chooser's dialog.
- const char * `title` () const
Get the title of the file chooser's dialog window.
- void `title` (const char *t)
Set the title of the file chooser's dialog window.
- int `type` () const
Gets the current `FI_Native_File_Chooser::Type` of browser.
- void `type` (int t)
Sets the current `FI_Native_File_Chooser::Type` of browser.
- `~FI_Native_File_Chooser` ()
Destructor.

Static Public Attributes

- static const char * **file_exists_message** = "File exists. Are you sure you want to overwrite?"
Localizable message.

31.88.1 Detailed Description

This class lets an FLTK application easily and consistently access the operating system's native file chooser. Some operating systems have very complex and specific file choosers that many users want access to specifically, instead of FLTK's default file chooser(s).

In cases where there is no native file browser, FLTK's own file browser is used instead.

To use this widget, use the following include in your code:

```
#include <FL/Fl_Native_File_Chooser.H>
```

The following example shows how to pick a single file:

```
// Create and post the local native file chooser
#include <FL/Fl_Native_File_Chooser.H>
[...
Fl_Native_File_Chooser fnfc;
fnfc.title("Pick a file");
fnfc.type(Fl_Native_File_Chooser::BROWSE_FILE);
fnfc.filter("Text\t*.txt\n"
           "C Files\t*.{cxx,h,c}");
fnfc.directory("/var/tmp"); // default directory to use
// Show native chooser
switch ( fnfc.show() ) {
  case -1: printf("ERROR: %s\n", fnfc.errmsg()); break; // ERROR
  case 1:  printf("CANCEL\n"); break; // CANCEL
  default: printf("PICKED: %s\n", fnfc.filename()); break; // FILE CHOSEN
}
]
```

The `Fl_Native_File_Chooser` widget transmits UTF-8 encoded filenames to its user. It is recommended to open files that may have non-ASCII names with the `fl_fopen()` or `fl_open()` utility functions that handle these names in a cross-platform way (whereas the standard `fopen()/open()` functions fail on the MSWindows platform to open files with a non-ASCII name).

Platform Specific Caveats

- Under X windows, and if `FL::OPTION_FNFC_USES_GTK` has not been switched off, the widget attempts to use standard GTK file chooser dialogs if they are available at run-time on the platform, and falls back to use FLTK's `Fl_File_Chooser` if they are not. In the latter case, it's best if you call `Fl_File_Icon::load_system_icons()` at the start of `main()`, to enable the nicer looking file browser widgets. Use the static public attributes of class `Fl_File_Chooser` to localize the browser.
- Some operating systems support certain OS specific options; see `Fl_Native_File_Chooser::options()` for a list.



Figure 31.22 The Fl_Native_File_Chooser on different platforms

31.88.2 Member Enumeration Documentation

31.88.2.1 Option

```
enum Fl_Native_File_Chooser::Option
```

Enumerator

NO_OPTIONS	no options enabled
SAVEAS_CONFIRM	Show native 'Save As' overwrite confirm dialog.
NEW_FOLDER	Show 'New Folder' icon (if supported)
PREVIEW	enable preview mode (if supported)
USE_FILTER_EXT	Chooser filter pilots the output file extension (if supported)

31.88.2.2 Type

```
enum Fl_Native_File_Chooser::Type
```

Enumerator

BROWSE_FILE	browse files (lets user choose one file)
BROWSE_DIRECTORY	browse directories (lets user choose one directory)
BROWSE_MULTI_FILE	browse files (lets user choose multiple files)
BROWSE_MULTI_DIRECTORY	browse directories (lets user choose multiple directories)

Enumerator

BROWSE_SAVE_FILE	browse to save a file
BROWSE_SAVE_DIRECTORY	browse to save a directory

31.88.3 Constructor & Destructor Documentation

31.88.3.1 Fl_Native_File_Chooser()

```
Fl_Native_File_Chooser::Fl_Native_File_Chooser (
    int val = BROWSE_FILE )
```

The constructor.

Internally allocates the native widgets. Optional `val` presets the type of browser this will be, which can also be changed with `type()`.

31.88.3.2 ~Fl_Native_File_Chooser()

```
Fl_Native_File_Chooser::~Fl_Native_File_Chooser ( )
```

Destructor.

Deallocates any resources allocated to this widget.

31.88.4 Member Function Documentation

31.88.4.1 count()

```
int Fl_Native_File_Chooser::count ( ) const
```

Returns the number of filenames (or directory names) the user selected.

Example:

```
if ( fnfc->show() == 0 ) {
    // Print all filenames user selected
    for (int n=0; n<fnfc->count(); n++) {
        printf("%d) '%s'\n", n, fnfc->filename(n));
    }
}
```

31.88.4.2 directory()

```
void Fl_Native_File_Chooser::directory (
    const char * val )
```

Preset the directory the browser will show when opened.

If `val` is NULL, or no directory is specified, the chooser will attempt to use the last non-cancelled folder.

31.88.4.3 errmsg()

```
const char * Fl_Native_File_Chooser::errmsg ( ) const
```

Returns a system dependent error message for the last method that failed.

This message should at least be flagged to the user in a dialog box, or to some kind of error log. Contents will be valid only for methods that document `errmsg()` will have info on failures.

31.88.4.4 filename() [1/2]

```
const char * Fl_Native_File_Chooser::filename ( ) const
```

Return the filename the user chose.

Use this if only expecting a single filename. If more than one filename is expected, use `filename(int)` instead. Return value may be "" if no filename was chosen (eg. user cancelled).

31.88.4.5 filename() [2/2]

```
const char * Fl_Native_File_Chooser::filename (
    int i ) const
```

Return one of the filenames the user selected.

Use `count()` to determine how many filenames the user selected.

Example:

```
if ( fnfc->show() == 0 ) {
    // Print all filenames user selected
    for (int n=0; n<fnfc->count(); n++ ) {
        printf("%d) '%s'\n", n, fnfc->filename(n));
    }
}
```

31.88.4.6 filter() [1/2]

```
const char * Fl_Native_File_Chooser::filter ( ) const
```

Returns the filter string last set.

Can be NULL if no filter was set.

31.88.4.7 filter() [2/2]

```
void Fl_Native_File_Chooser::filter (
    const char * f )
```

Sets the filename filters used for browsing.

The default is NULL, which browses all files.

The filter string can be any of:

- A single wildcard (eg. "*.txt")
- Multiple wildcards (eg. ".*{cxx,h,H}")
- A descriptive name followed by a "\t" and a wildcard (eg. "Text Files\t*.txt")
- A list of separate wildcards with a "\n" between each (eg. ".*{cxx,H}\n*.txt")
- A list of descriptive names and wildcards (eg. "C++ Files\t*.{cxx,H}\nTxt Files\t*.txt")

The format of each filter is a wildcard, or an optional user description followed by '\t' and the wildcard.

On most platforms, each filter is available to the user via a pulldown menu in the file chooser. The 'All Files' option is always available to the user.

31.88.4.8 filter_value() [1/2]

```
int Fl_Native_File_Chooser::filter_value ( ) const
```

Returns which filter value was last selected by the user.

This is only valid if the chooser returns success.

31.88.4.9 filter_value() [2/2]

```
void Fl_Native_File_Chooser::filter_value (
    int i )
```

Sets which filter will be initially selected.

The first filter is indexed as 0. If `filter_value()==filters()`, then "All Files" was chosen. If `filter_value() > filters()`, then a custom filter was set.

31.88.4.10 options()

```
void Fl_Native_File_Chooser::options (
    int o )
```

Sets the platform specific chooser options to `val`.

`val` is expected to be one or more `Fl_Native_File_Chooser::Option` flags ORed together. Some platforms have OS-specific functions that can be enabled/disabled via this method.

Flag	Description	Win	Mac	Other
<code>NEW_FOLDER</code> Used	Shows the 'New Folder' button.			Ignored Used
<code>PREVIEW</code>	Enables the 'Preview' mode by default.		Ignored	Ignored Used
<code>SAVEAS_CONFIRM</code>	Confirm dialog if <code>BROWSE_SAVE_FILE</code> file exists.	Used	Used	Used
<code>USE_FILTER_EXT</code>	Chooser filter pilots the output file extension.	Ignored	Used	Used (GTK)

31.88.4.11 preset_file()

```
void Fl_Native_File_Chooser::preset_file (
    const char * f )
```

Sets the default filename for the chooser.

Use [directory\(\)](#) to set the default directory. Mainly used to preset the filename for save dialogs, and on most platforms can be used for opening files as well.

31.88.4.12 show()

```
int Fl_Native_File_Chooser::show ( )
```

Post the chooser's dialog.

Blocks until dialog has been completed or cancelled.

Returns

- 0 – user picked a file
- 1 – user cancelled
- -1 – failed; [errmsg\(\)](#) has reason

31.88.4.13 title() [1/2]

```
const char * Fl_Native_File_Chooser::title ( ) const
```

Get the title of the file chooser's dialog window.

Return value may be NULL if no title was set.

31.88.4.14 title() [2/2]

```
void Fl_Native_File_Chooser::title (
    const char * t )
```

Set the title of the file chooser's dialog window.

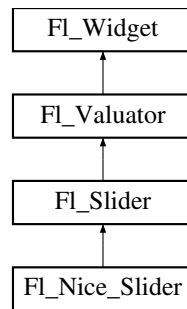
Can be NULL if no title desired. The default title varies according to the platform, so you are advised to set the title explicitly.

The documentation for this class was generated from the following files:

- [Fl_Native_File_Chooser.H](#)
- [Fl_Native_File_Chooser.cxx](#)
- [Fl_Native_File_Chooser_FLTK.cxx](#)

31.89 Fl_Nice_Slider Class Reference

Inheritance diagram for Fl_Nice_Slider:



Public Member Functions

- **Fl_Nice_Slider** (int X, int Y, int W, int H, const char *L=0)

Additional Inherited Members

The documentation for this class was generated from the following files:

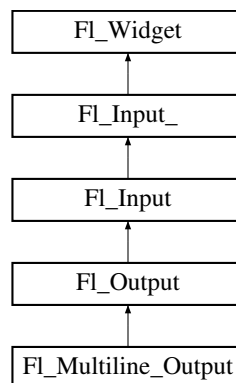
- Fl_Nice_Slider.H
- Fl_Slider.cxx

31.90 Fl_Output Class Reference

This widget displays a piece of text.

```
#include <Fl_Output.H>
```

Inheritance diagram for Fl_Output:



Public Member Functions

- **Fl_Output** (int X, int Y, int W, int H, const char *l=0)
Creates a new *Fl_Output* widget using the given position, size, and label string.

Additional Inherited Members

31.90.1 Detailed Description

This widget displays a piece of text.

When you set the `value()`, *Fl_Output* does a `strcpy()` to its own storage, which is useful for program-generated values. The user may select portions of the text using the mouse and paste the contents into other fields or programs.

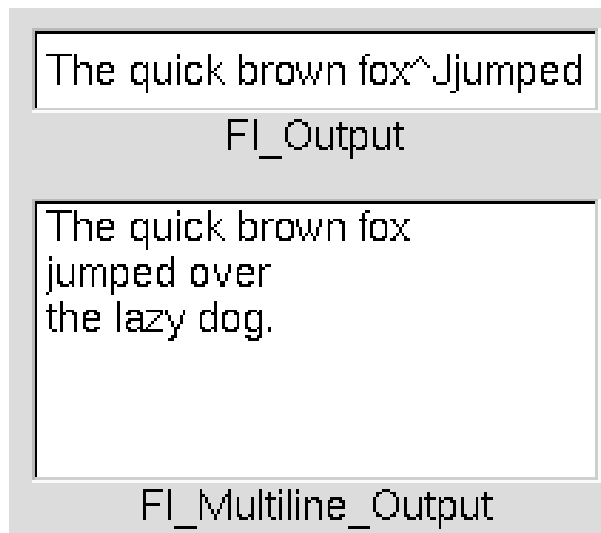


Figure 31.23 Fl_Output

There is a single subclass, [Fl_Multiline_Output](#), which allows you to display multiple lines of text. [Fl_Multiline_Output](#) does not provide scroll bars. If a more complete text editing widget is needed, use [Fl_Text_Display](#) instead. The text may contain any characters except `\0`, and will correctly display anything, using `^X` notation for unprintable control characters and `\nnn` notation for unprintable characters with the high bit set. It assumes the font can draw any characters in the ISO-Latin1 character set.

31.90.2 Constructor & Destructor Documentation

31.90.2.1 Fl_Output()

```
Fl_Output::Fl_Output (
    int X,
    int Y,
    int W,
    int H,
    const char * l = 0 )
```

Creates a new [Fl_Output](#) widget using the given position, size, and label string. The default boxtype is `FL_DOWN_BOX`.

Inherited destructor destroys the widget and any value associated with it.

The documentation for this class was generated from the following files:

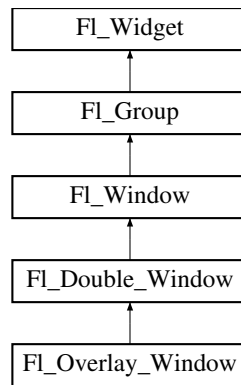
- [Fl_Output.H](#)
- [Fl_Input.cxx](#)

31.91 Fl_Overlay_Window Class Reference

This window provides double buffering and also the ability to draw the "overlay" which is another picture placed on top of the main image.

```
#include <Fl_Overlay_Window.H>
```

Inheritance diagram for [Fl_Overlay_Window](#):



Public Member Functions

- int **can_do_overlay** ()
Returns non-zero if there's hardware overlay support.
- void **flush** ()
Forces the window to be redrawn.
- void **hide** ()
Removes the window from the screen.
- void **redraw_overlay** ()
Call this to indicate that the overlay data has changed and needs to be redrawn.
- void **resize** (int, int, int, int)
Changes the size and position of the window.
- void **show** ()
Puts the window on the screen.
- void **show** (int a, char **b)
- ~**FI_Overlay_Window** ()
Destroys the window and all child widgets.

Protected Member Functions

- virtual void **draw_overlay** ()=0
You must subclass [FI_Overlay_Window](#) and provide this method.
- **FI_Overlay_Window** (int W, int H, const char *l=0)
*See [FI_Overlay_Window::FI_Overlay_Window\(int X, int Y, int W, int H, const char *l=0\)](#)*
- [FI_Overlay_Window](#) (int X, int Y, int W, int H, const char *l=0)
Creates a new [FI_Overlay_Window](#) widget using the given position, size, and label (title) string.

Additional Inherited Members

31.91.1 Detailed Description

This window provides double buffering and also the ability to draw the "overlay" which is another picture placed on top of the main image.

The overlay is designed to be a rapidly-changing but simple graphic such as a mouse selection box. [FI_Overlay_Window](#) uses the overlay planes provided by your graphics hardware if they are available.

If no hardware support is found the overlay is simulated by drawing directly into the on-screen copy of the double-buffered window, and "erased" by copying the backbuffer over it again. This means the overlay will blink if you change the image in the window.

31.91.2 Constructor & Destructor Documentation

31.91.2.1 Fl_Overlay_Window()

```
Fl_Overlay_Window::Fl_Overlay_Window (
    int X,
    int Y,
    int W,
    int H,
    const char * I = 0 ) [protected]
```

Creates a new [Fl_Overlay_Window](#) widget using the given position, size, and label (title) string. If the positions (x,y) are not given, then the window manager will choose them.

31.91.3 Member Function Documentation

31.91.3.1 draw_overlay()

```
virtual void Fl_Overlay_Window::draw_overlay ( ) [protected], [pure virtual]
```

You must subclass [Fl_Overlay_Window](#) and provide this method.

It is just like a [draw\(\)](#) method, except it draws the overlay. The overlay will have already been "cleared" when this is called. You can use any of the routines described in [<FL/fl_draw.H>](#).

31.91.3.2 flush()

```
void Fl_Overlay_Window::flush ( ) [virtual]
```

Forces the window to be redrawn.

Reimplemented from [Fl_Double_Window](#).

31.91.3.3 hide()

```
void Fl_Overlay_Window::hide ( ) [virtual]
```

Removes the window from the screen.

If the window is already hidden or has not been shown then this does nothing and is harmless.

Reimplemented from [Fl_Double_Window](#).

31.91.3.4 redraw_overlay()

```
void Fl_Overlay_Window::redraw_overlay ( )
```

Call this to indicate that the overlay data has changed and needs to be redrawn.

The overlay will be clear until the first time this is called, so if you want an initial display you must call this after calling [show\(\)](#).

31.91.3.5 resize()

```
void Fl_Overlay_Window::resize (
    int X,
    int Y,
    int W,
    int H ) [virtual]
```

Changes the size and position of the window.

If [shown\(\)](#) is true, these changes are communicated to the window server (which may refuse that size and cause a further resize). If [shown\(\)](#) is false, the size and position are used when [show\(\)](#) is called. See [Fl_Group](#) for the effect of resizing on the child widgets.

You can also call the [Fl_Widget](#) methods [size\(x,y\)](#) and [position\(w,h\)](#), which are inline wrappers for this virtual function.

A top-level window can not force, but merely suggest a position and size to the operating system. The window manager may not be willing or able to display a window at the desired position or with the given dimensions. It is up to the application developer to verify window parameters after the resize request.

Reimplemented from [Fl_Double_Window](#).

31.91.3.6 show()

```
void Fl_Overlay_Window::show ( ) [virtual]
```

Puts the window on the screen.

Usually (on X) this has the side effect of opening the display.

If the window is already shown then it is restored and raised to the top. This is really convenient because your program can call [show\(\)](#) at any time, even if the window is already up. It also means that [show\(\)](#) serves the purpose of [raise\(\)](#) in other toolkits.

[Fl_Window::show\(int argc, char **argv\)](#) is used for top-level windows and allows standard arguments to be parsed from the command-line.

Note

For some obscure reasons [Fl_Window::show\(\)](#) resets the current group by calling [Fl_Group::current\(0\)](#). The comments in the code say "get rid of very common user bug: forgot end()". Although this is true it may have unwanted side effects if you [show\(\)](#) an unrelated window (maybe for an error message or warning) while building a window or any other group widget.

Todo Check if we can remove resetting the current group in a later FLTK version (after 1.3.x). This may break "already broken" programs though if they rely on this "feature".

See also

[Fl_Window::show\(int argc, char **argv\)](#)

Reimplemented from [Fl_Double_Window](#).

The documentation for this class was generated from the following files:

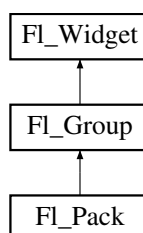
- [Fl_Overlay_Window.H](#)
- [Fl_Double_Window.cxx](#)
- [Fl_Overlay_Window.cxx](#)

31.92 Fl_Pack Class Reference

This widget was designed to add the functionality of compressing and aligning widgets.

```
#include <Fl_Pack.H>
```

Inheritance diagram for [Fl_Pack](#):



Public Types

- enum { **VERTICAL** = 0 , **HORIZONTAL** = 1 }

Public Member Functions

- [Fl_Pack](#) (int *x*, int *y*, int *w*, int *h*, const char **l*=0)
Creates a new [Fl_Pack](#) widget using the given position, size, and label string.
- [uchar horizontal](#) () const

Same as [Fl_Group::type\(\)](#)

- int **spacing** () const
Gets the number of extra pixels of blank space that are added between the children.
- void **spacing** (int i)
Sets the number of extra pixels of blank space that are added between the children.

Protected Member Functions

- void **draw** ()
Draws the widget.

Additional Inherited Members

31.92.1 Detailed Description

This widget was designed to add the functionality of compressing and aligning widgets.

If [type\(\)](#) is `Fl_Pack::HORIZONTAL` all the children are resized to the height of the [Fl_Pack](#), and are moved next to each other horizontally. If [type\(\)](#) is not `Fl_Pack::HORIZONTAL` then the children are resized to the width and are stacked below each other. Then the [Fl_Pack](#) resizes itself to surround the child widgets.

This widget is needed for the [Fl_Tabs](#). In addition you may want to put the [Fl_Pack](#) inside an [Fl_Scroll](#).

The resizable for [Fl_Pack](#) is set to `NULL` by default.

See also: [Fl_Group::resizable\(\)](#)

31.92.2 Constructor & Destructor Documentation

31.92.2.1 Fl_Pack()

```
Fl_Pack::Fl_Pack (
    int X,
    int Y,
    int W,
    int H,
    const char * l = 0 )
```

Creates a new [Fl_Pack](#) widget using the given position, size, and label string.

The default boxtype is `FL_NO_BOX`.

The destructor *also deletes all the children*. This allows a whole tree to be deleted at once, without having to keep a pointer to all the children in the user code. A kludge has been done so the [Fl_Pack](#) and all of it's children can be automatic (local) variables, but you must declare the [Fl_Pack](#)*first*, so that it is destroyed last.

31.92.3 Member Function Documentation

31.92.3.1 draw()

```
void Fl_Pack::draw ( ) [protected], [virtual]
```

Draws the widget.

Never call this function directly. FLTK will schedule redrawing whenever needed. If your widget must be redrawn as soon as possible, call [redraw\(\)](#) instead.

Override this function to draw your own widgets.

If you ever need to call another widget's draw method *from within your own draw() method*, e.g. for an embedded scrollbar, you can do it (because [draw\(\)](#) is virtual) like this:

```
Fl_Widget *s = &scroll;           // scroll is an embedded Fl_Scrollbar
s->draw();                          // calls Fl_Scrollbar::draw()
```

Reimplemented from [Fl_Group](#).

The documentation for this class was generated from the following files:

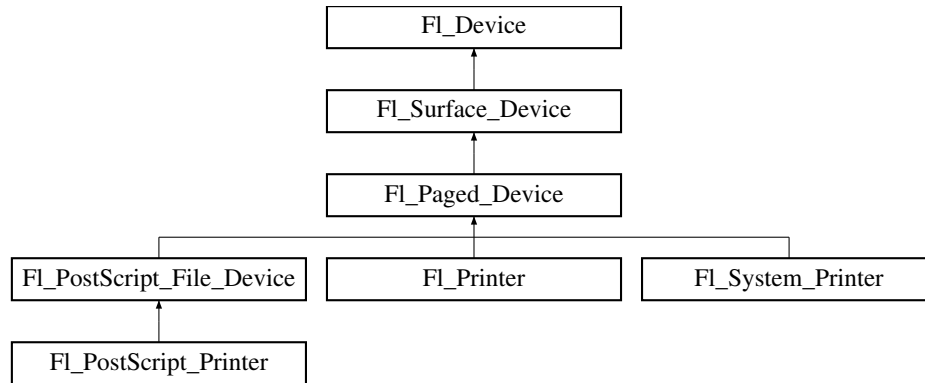
- [Fl_Pack.H](#)
- [Fl_Pack.cxx](#)

31.93 Fl_Paged_Device Class Reference

Represents page-structured drawing surfaces.

```
#include <Fl_Paged_Device.H>
```

Inheritance diagram for `Fl_Paged_Device`:



Classes

- struct [page_format](#)
width, height and name of a page format

Public Types

- enum [Page_Format](#) {
A0 = 0 , **A1** , **A2** , **A3** ,
A4 , **A5** , **A6** , **A7** ,
A8 , **A9** , **B0** , **B1** ,
B2 , **B3** , **B4** , **B5** ,
B6 , **B7** , **B8** , **B9** ,
B10 , **C5E** , **DLE** , **EXECUTIVE** ,
FOLIO , **LEDGER** , **LEGAL** , **LETTER** ,
TABLOID , **ENVELOPE** , **MEDIA** = 0x1000 }
Possible page formats.
- enum [Page_Layout](#) { **PORTRAIT** = 0 , **LANDSCAPE** = 0x100 , **REVERSED** = 0x200 , **ORIENTATION** = 0x300 }
Possible page layouts.

Public Member Functions

- const char * [class_name](#) ()
Returns the name of the class of this object.
- virtual void [end_job](#) (void)
To be called at the end of a print job.
- virtual int [end_page](#) (void)
To be called at the end of each page.
- virtual void [margins](#) (int *left, int *top, int *right, int *bottom)
Computes the dimensions of margins that lie between the printable page area and the full page.
- virtual void [origin](#) (int *x, int *y)

- Computes the page coordinates of the current origin of graphics functions.*

 - virtual void [origin](#) (int x, int y)
- Sets the position in page coordinates of the origin of graphics functions.*

 - virtual void [print_widget](#) ([FI_Widget](#) *widget, int delta_x=0, int delta_y=0)
- Draws the widget on the printed page.*

 - void [print_window](#) ([FI_Window](#) *win, int x_offset=0, int y_offset=0)
- Prints a window with its title bar and frame if any.*

 - virtual void [print_window_part](#) ([FI_Window](#) *win, int x, int y, int w, int h, int delta_x=0, int delta_y=0)
- Prints a rectangular part of an on-screen window.*

 - virtual int [printable_rect](#) (int *w, int *h)
- Computes the width and height of the printable area of the page.*

 - virtual void [rotate](#) (float angle)
- Rotates the graphics operations relatively to paper.*

 - virtual void [scale](#) (float scale_x, float scale_y=0.)
- Changes the scaling of page coordinates.*

 - virtual int [start_job](#) (int pagecount, int *frompage=NULL, int *topage=NULL)
- Starts a print job.*

 - virtual int [start_page](#) (void)
- Starts a new printed page.*

 - virtual void [translate](#) (int x, int y)
- Translates the current graphics origin accounting for the current rotation.*

 - virtual void [untranslate](#) (void)
- Undoes the effect of a previous [translate\(\)](#) call.*

 - virtual ~[FI_Paged_Device](#) ()
- The destructor.*

Static Public Attributes

- static const char * [class_id](#) = "FI_Paged_Device"
- static const [page_format](#) [page_formats](#) [[NO_PAGE_FORMATS](#)]
width, height and name of all elements of the enum [Page_Format](#).

Protected Member Functions

- [FI_Paged_Device](#) ()
The constructor.

Protected Attributes

- int [x_offset](#)
horizontal offset to the origin of graphics coordinates
- int [y_offset](#)
vertical offset to the origin of graphics coordinates

Friends

- class [FI_Copy_Surface](#)
- class [FI_Image_Surface](#)

Additional Inherited Members

31.93.1 Detailed Description

Represents page-structured drawing surfaces.

This class has no public constructor: don't instantiate it; use [FI_Printer](#) or [FI_PostScript_File_Device](#) instead.

31.93.2 Member Enumeration Documentation

31.93.2.1 Page_Format

enum `Fl_Paged_Device::Page_Format`

Possible page formats.

All paper formats with pre-defined width and height.

Enumerator

A0	A0 format.
A4	A4 format.
LETTER	Letter format.

31.93.2.2 Page_Layout

enum `Fl_Paged_Device::Page_Layout`

Possible page layouts.

Enumerator

PORTRAIT	Portrait orientation.
LANDSCAPE	Landscape orientation.
REVERSED	Reversed orientation.
ORIENTATION	orientation

31.93.3 Member Function Documentation

31.93.3.1 class_name()

```
const char * Fl_Paged_Device::class_name ( ) [inline], [virtual]
```

Returns the name of the class of this object.

Use of the `class_name()` function is discouraged because it will be removed from future FLTK versions.

The class of an instance of an `Fl_Device` subclass can be checked with code such as:

```
if ( instance->class_name() == Fl_Printer::class_id ) { ... }
```

Reimplemented from `Fl_Surface_Device`.

Reimplemented in `Fl_PostScript_File_Device`, `Fl_System_Printer`, `Fl_PostScript_Printer`, and `Fl_Printer`.

31.93.3.2 end_job()

```
void Fl_Paged_Device::end_job (
    void ) [virtual]
```

To be called at the end of a print job.

Reimplemented in `Fl_PostScript_File_Device`, `Fl_System_Printer`, and `Fl_Printer`.

31.93.3.3 end_page()

```
int Fl_Paged_Device::end_page (
    void ) [virtual]
```

To be called at the end of each page.

Returns

0 if OK, non-zero if any error.

Reimplemented in [Fl_PostScript_File_Device](#), [Fl_System_Printer](#), and [Fl_Printer](#).

31.93.3.4 margins()

```
void Fl_Paged_Device::margins (
    int * left,
    int * top,
    int * right,
    int * bottom ) [virtual]
```

Computes the dimensions of margins that lie between the printable page area and the full page. Values are in the same unit as that used by FLTK drawing functions. They are changed by [scale\(\)](#) calls.

Parameters

out	<i>left</i>	If non-null, *left is set to the left margin size.
out	<i>top</i>	If non-null, *top is set to the top margin size.
out	<i>right</i>	If non-null, *right is set to the right margin size.
out	<i>bottom</i>	If non-null, *bottom is set to the bottom margin size.

Reimplemented in [Fl_PostScript_File_Device](#), [Fl_System_Printer](#), and [Fl_Printer](#).

31.93.3.5 origin() [1/2]

```
void Fl_Paged_Device::origin (
    int * x,
    int * y ) [virtual]
```

Computes the page coordinates of the current origin of graphics functions.

Parameters

out	<i>x</i>	If non-null, *x is set to the horizontal page offset of graphics origin.
out	<i>y</i>	Same as above, vertically.

Reimplemented in [Fl_PostScript_File_Device](#), [Fl_System_Printer](#), and [Fl_Printer](#).

31.93.3.6 origin() [2/2]

```
void Fl_Paged_Device::origin (
    int x,
    int y ) [virtual]
```

Sets the position in page coordinates of the origin of graphics functions.

Arguments should be expressed relatively to the result of a previous [printable_rect\(\)](#) call. That is, `printable_rect(&w, &h); origin(w/2, 0);` sets the graphics origin at the top center of the page printable area. Origin() calls are not affected by [rotate\(\)](#) calls. Successive [origin\(\)](#) calls don't combine their effects.

Parameters

in	<i>x</i>	Horizontal position in page coordinates of the desired origin of graphics functions.
in	<i>y</i>	Same as above, vertically.

Reimplemented in [Fl_PostScript_File_Device](#), [Fl_System_Printer](#), and [Fl_Printer](#).

31.93.3.7 print_widget()

```
void Fl_Paged_Device::print_widget (
    Fl_Widget * widget,
    int delta_x = 0,
    int delta_y = 0 ) [virtual]
```

Draws the widget on the printed page.

The widget's position on the printed page is determined by the last call to [origin\(\)](#) and by the optional `delta_x` and `delta_y` arguments. Its dimensions are in points unless there was a previous call to [scale\(\)](#).

Parameters

in	<i>widget</i>	Any FLTK widget (e.g., standard, custom, window).
in	<i>delta_x</i>	Optional horizontal offset for positioning the widget relatively to the current origin of graphics functions.
in	<i>delta_y</i>	Same as above, vertically.

Reimplemented in [Fl_Printer](#).

31.93.3.8 print_window()

```
void Fl_Paged_Device::print_window (
    Fl_Window * win,
    int x_offset = 0,
    int y_offset = 0 )
```

Prints a window with its title bar and frame if any.

`x_offset` and `y_offset` are optional coordinates of where to position the window top left. Equivalent to [print_widget\(\)](#) if `win` is a subwindow or has no border. Use [Fl_Window::decorated_w\(\)](#) and [Fl_Window::decorated_h\(\)](#) to get the size of the printed window.

31.93.3.9 print_window_part()

```
void Fl_Paged_Device::print_window_part (
    Fl_Window * win,
    int x,
    int y,
    int w,
    int h,
    int delta_x = 0,
    int delta_y = 0 ) [virtual]
```

Prints a rectangular part of an on-screen window.

Parameters

<i>win</i>	The window from where to capture.
<i>x</i>	The rectangle left
<i>y</i>	The rectangle top
<i>w</i>	The rectangle width
<i>h</i>	The rectangle height
<i>delta_x</i>	Optional horizontal offset from current graphics origin where to print the captured rectangle.
<i>delta_y</i>	As above, vertically.

Reimplemented in [Fl_Printer](#).

31.93.3.10 printable_rect()

```
int Fl_Paged_Device::printable_rect (
    int * w,
    int * h ) [virtual]
```

Computes the width and height of the printable area of the page.

Values are in the same unit as that used by FLTK drawing functions, are unchanged by calls to [origin\(\)](#), but are changed by [scale\(\)](#) calls. Values account for the user-selected paper type and print orientation.

Returns

0 if OK, non-zero if any error

Reimplemented in [Fl_PostScript_File_Device](#), [Fl_System_Printer](#), and [Fl_Printer](#).

31.93.3.11 rotate()

```
void Fl_Paged_Device::rotate (
    float angle ) [virtual]
```

Rotates the graphics operations relatively to paper.

The rotation is centered on the current graphics origin. Successive [rotate\(\)](#) calls don't combine their effects.

Parameters

<i>angle</i>	Rotation angle in counter-clockwise degrees.
--------------	--

Reimplemented in [Fl_PostScript_File_Device](#), [Fl_System_Printer](#), and [Fl_Printer](#).

31.93.3.12 scale()

```
void Fl_Paged_Device::scale (
    float scale_x,
    float scale_y = 0. ) [virtual]
```

Changes the scaling of page coordinates.

This function also resets the origin of graphics functions at top left of printable page area. After a [scale\(\)](#) call, do a [printable_rect\(\)](#) call to get the new dimensions of the printable page area. Successive [scale\(\)](#) calls don't combine their effects.

Parameters

<i>scale</i> <i>_x</i>	Horizontal dimensions of plot are multiplied by this quantity.
<i>scale</i> <i>_y</i>	Same as above, vertically. The value 0. is equivalent to setting <code>scale_y = scale_x</code> . Thus, <code>scale(factor);</code> is equivalent to <code>scale(factor, factor);</code>

Reimplemented in [Fl_PostScript_File_Device](#), [Fl_System_Printer](#), and [Fl_Printer](#).

31.93.3.13 start_job()

```
int Fl_Paged_Device::start_job (
    int pagecount,
    int * frompage = NULL,
    int * topage = NULL ) [virtual]
```

Starts a print job.

Parameters

in	<i>pagecount</i>	the total number of pages of the job (or 0 if you don't know the number of pages)
out	<i>frompage</i>	if non-null, *frompage is set to the first page the user wants printed
out	<i>topage</i>	if non-null, *topage is set to the last page the user wants printed

Returns

0 if OK, non-zero if any error

Reimplemented in [FI_PostScript_File_Device](#), [FI_System_Printer](#), [FI_Printer](#), and [FI_PostScript_Printer](#).

31.93.3.14 start_page()

```
int Fl_Paged_Device::start_page (
    void ) [virtual]
```

Starts a new printed page.

The page coordinates are initially in points, i.e., 1/72 inch, and with origin at the top left of the printable page area.

Returns

0 if OK, non-zero if any error

Reimplemented in [FI_PostScript_File_Device](#), [FI_System_Printer](#), and [FI_Printer](#).

31.93.3.15 translate()

```
void Fl_Paged_Device::translate (
    int x,
    int y ) [virtual]
```

Translates the current graphics origin accounting for the current rotation.

This function is only useful after a [rotate\(\)](#) call. Each [translate\(\)](#) call must be matched by an [untranslate\(\)](#) call.

Successive [translate\(\)](#) calls add up their effects.

Reimplemented in [FI_PostScript_File_Device](#), [FI_System_Printer](#), and [FI_Printer](#).

31.93.3.16 untranslate()

```
void Fl_Paged_Device::untranslate (
    void ) [virtual]
```

Undoes the effect of a previous [translate\(\)](#) call.

Reimplemented in [FI_PostScript_File_Device](#), [FI_System_Printer](#), and [FI_Printer](#).

The documentation for this class was generated from the following files:

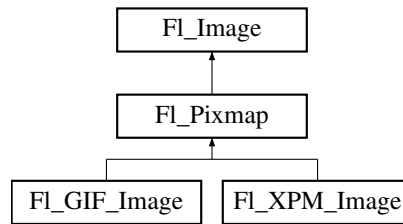
- [FI_Paged_Device.H](#)
- [FI_Paged_Device.cxx](#)

31.94 FI_Pixmap Class Reference

The [FI_Pixmap](#) class supports caching and drawing of colormap (pixmap) images, including transparency.

```
#include <Fl_Pixmap.H>
```

Inheritance diagram for [FI_Pixmap](#):



Public Member Functions

- virtual void `color_average` (`FI_Color` c, float i)
The `color_average()` method averages the colors in the image with the FLTK color value c.
- `FI_Image * copy` ()
- virtual `FI_Image * copy` (int W, int H)
The `copy()` method creates a copy of the specified image.
- virtual void `desaturate` ()
The `desaturate()` method converts an image to grayscale.
- void `draw` (int X, int Y)
- virtual void `draw` (int X, int Y, int W, int H, int cx=0, int cy=0)
Draws the image with a bounding box.
- `FI_Pixmap` (char *const *D)
The constructors create a new pixmap from the specified XPM data.
- `FI_Pixmap` (const char *const *D)
The constructors create a new pixmap from the specified XPM data.
- `FI_Pixmap` (const uchar *const *D)
The constructors create a new pixmap from the specified XPM data.
- `FI_Pixmap` (uchar *const *D)
The constructors create a new pixmap from the specified XPM data.
- virtual void `label` (`FI_Menu_Item` *m)
The `label()` methods are an obsolete way to set the image attribute of a widget or menu item.
- virtual void `label` (`FI_Widget` *w)
The `label()` methods are an obsolete way to set the image attribute of a widget or menu item.
- virtual void `uncache` ()
If the image has been cached for display, delete the cache data.
- virtual `~FI_Pixmap` ()
The destructor frees all memory and server resources that are used by the pixmap.

Public Attributes

- int `alloc_data`

Protected Member Functions

- void `measure` ()

Friends

- class `FI_GDI_Graphics_Driver`
- class `FI_GDI_Printer_Graphics_Driver`
- class `FI_Quartz_Graphics_Driver`
- class `FI_Xlib_Graphics_Driver`

Additional Inherited Members

31.94.1 Detailed Description

The [Fl_Pixmap](#) class supports caching and drawing of colormap (pixmap) images, including transparency.

31.94.2 Constructor & Destructor Documentation

31.94.2.1 Fl_Pixmap() [1/4]

```
Fl_Pixmap::Fl_Pixmap (
    char *const * D ) [inline], [explicit]
```

The constructors create a new pixmap from the specified XPM data.

31.94.2.2 Fl_Pixmap() [2/4]

```
Fl_Pixmap::Fl_Pixmap (
    uchar *const * D ) [inline], [explicit]
```

The constructors create a new pixmap from the specified XPM data.

31.94.2.3 Fl_Pixmap() [3/4]

```
Fl_Pixmap::Fl_Pixmap (
    const char *const * D ) [inline], [explicit]
```

The constructors create a new pixmap from the specified XPM data.

31.94.2.4 Fl_Pixmap() [4/4]

```
Fl_Pixmap::Fl_Pixmap (
    const uchar *const * D ) [inline], [explicit]
```

The constructors create a new pixmap from the specified XPM data.

31.94.3 Member Function Documentation

31.94.3.1 color_average()

```
void Fl_Pixmap::color_average (
    Fl_Color c,
    float i ) [virtual]
```

The [color_average\(\)](#) method averages the colors in the image with the FLTK color value *c*.

The *i* argument specifies the amount of the original image to combine with the color, so a value of 1.0 results in no color blend, and a value of 0.0 results in a constant image of the specified color.

An internal copy is made of the original image before changes are applied, to avoid modifying the original image.

Reimplemented from [Fl_Image](#).

31.94.3.2 copy()

```
Fl_Image * Fl_Pixmap::copy (
    int W,
    int H ) [virtual]
```

The `copy()` method creates a copy of the specified image.

If the width and height are provided, the image is resized to the specified size. The image should be deleted (or in the case of [Fl_Shared_Image](#), released) when you are done with it.

Reimplemented from [Fl_Image](#).

31.94.3.3 desaturate()

```
void Fl_Pixmap::desaturate ( ) [virtual]
```

The `desaturate()` method converts an image to grayscale.

If the image contains an alpha channel (depth = 4), the alpha channel is preserved.

An internal copy is made of the original image before changes are applied, to avoid modifying the original image.

Reimplemented from [Fl_Image](#).

31.94.3.4 draw()

```
void Fl_Pixmap::draw (
    int X,
    int Y,
    int W,
    int H,
    int cx = 0,
    int cy = 0 ) [virtual]
```

Draws the image with a bounding box.

Arguments `X`, `Y`, `W`, `H` specify a bounding box for the image, with the origin (upper-left corner) of the image offset by the `cx` and `cy` arguments.

In other words: `fl_push_clip(X, Y, W, H)` is applied, the image is drawn with its upper-left corner at `X-cx`, `Y-cy` and its own width and height, `fl_pop_clip()` is applied.

Reimplemented from [Fl_Image](#).

31.94.3.5 label() [1/2]

```
void Fl_Pixmap::label (
    Fl_Menu_Item * m ) [virtual]
```

The `label()` methods are an obsolete way to set the image attribute of a widget or menu item.

Use the `image()` or `deimage()` methods of the [Fl_Widget](#) and [Fl_Menu_Item](#) classes instead.

Reimplemented from [Fl_Image](#).

31.94.3.6 label() [2/2]

```
void Fl_Pixmap::label (
    Fl_Widget * widget ) [virtual]
```

The `label()` methods are an obsolete way to set the image attribute of a widget or menu item.

Use the `image()` or `deimage()` methods of the [Fl_Widget](#) and [Fl_Menu_Item](#) classes instead.

Reimplemented from [Fl_Image](#).

31.94.3.7 uncache()

```
void Fl_Pixmap::uncache ( ) [virtual]
```

If the image has been cached for display, delete the cache data.

This allows you to change the data used for the image and then redraw it without recreating an image object. Reimplemented from [Fl_Image](#).

The documentation for this class was generated from the following files:

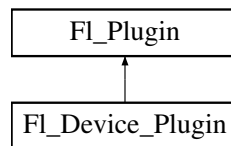
- [Fl_Pixmap.H](#)
- [Fl_Pixmap.cxx](#)

31.95 Fl_Plugin Class Reference

[Fl_Plugin](#) allows link-time and run-time integration of binary modules.

```
#include <Fl_Plugin.H>
```

Inheritance diagram for [Fl_Plugin](#):



Public Member Functions

- [Fl_Plugin](#) (const char *klass, const char *name)
Create a plugin.
- virtual [~Fl_Plugin](#) ()
Clear the plugin and remove it from the database.

31.95.1 Detailed Description

[Fl_Plugin](#) allows link-time and run-time integration of binary modules.

[Fl_Plugin](#) and [Fl_Plugin_Manager](#) provide a small and simple solution for linking C++ classes at run-time, or optionally linking modules at compile time without the need to change the main application.

[Fl_Plugin_Manager](#) uses static initialisation to create the plugin interface early during startup. Plugins are stored in a temporary database, organized in classes.

Plugins should derive a new class from [Fl_Plugin](#) as a base:

```
class My_Plugin : public Fl_Plugin {
public:
    My_Plugin() : Fl_Plugin("effects", "blur") {}
    void do_something(...);
};
My_Plugin blur_plugin();
```

Plugins can be put into modules and either linked before distribution, or loaded from dynamically linkable files. An

[Fl_Plugin_Manager](#) is used to list and access all currently loaded plugins.

```
Fl_Plugin_Manager mgr("effects");
int i, n = mgr.plugins();
for (i=0; i<n; i++) {
    My_Plugin *pin = (My_Plugin*)mgr.plugin(i);
    pin->do_something();
}
```

31.95.2 Constructor & Destructor Documentation

31.95.2.1 Fl_Plugin()

```
Fl_Plugin::Fl_Plugin (
    const char * klass,
    const char * name )
```

Create a plugin.

Parameters

in	<i>klass</i>	plugins are grouped in classes
in	<i>name</i>	every plugin should have a unique name

The documentation for this class was generated from the following files:

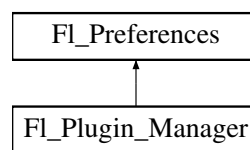
- FI_Plugin.H
- FI_Preferences.cxx

31.96 FI_Plugin_Manager Class Reference

[FI_Plugin_Manager](#) manages link-time and run-time plugin binaries.

```
#include <FI_Plugin.H>
```

Inheritance diagram for FI_Plugin_Manager:



Public Member Functions

- [FI_Preferences::ID addPlugin](#) (const char *name, FI_Plugin *plugin)
This function adds a new plugin to the database.
- [FI_Plugin_Manager](#) (const char *klass)
Manage all plugins belonging to one class.
- [FI_Plugin * plugin](#) (const char *name)
Return the address of a plugin by name.
- [FI_Plugin * plugin](#) (int index)
Return the address of a plugin by index.
- int [plugins](#) ()
Return the number of plugins in the klass.
- [~FI_Plugin_Manager](#) ()
Remove the plugin manager.

Static Public Member Functions

- static int [load](#) (const char *filename)
Load a module from disk.
- static int [loadAll](#) (const char *filepath, const char *pattern=0)
Use this function to load a whole directory full of modules.
- static void [removePlugin](#) (FI_Preferences::ID id)
Remove any plugin.

Additional Inherited Members

31.96.1 Detailed Description

[FI_Plugin_Manager](#) manages link-time and run-time plugin binaries.

See also

[FI_Plugin](#)

31.96.2 Constructor & Destructor Documentation

31.96.2.1 ~Fl_Plugin_Manager()

```
Fl_Plugin_Manager::~~Fl_Plugin_Manager ( )
```

Remove the plugin manager.

Calling this does not remove the database itself or any plugins. It just removes the reference to the database.

31.96.3 Member Function Documentation

31.96.3.1 addPlugin()

```
Fl_Preferences::ID Fl_Plugin_Manager::addPlugin (
    const char * name,
    Fl_Plugin * plugin )
```

This function adds a new plugin to the database.

There is no need to call this function explicitly. Every [Fl_Plugin](#) constructor will call this function at initialization time.

31.96.3.2 load()

```
int Fl_Plugin_Manager::load (
    const char * filename ) [static]
```

Load a module from disk.

A module must be a dynamically linkable file for the given operating system. When loading a module, its `+init` function will be called which in turn calls the constructor of all statically initialized [Fl_Plugin](#) classes and adds them to the database.

31.96.3.3 removePlugin()

```
void Fl_Plugin_Manager::removePlugin (
    Fl_Preferences::ID id ) [static]
```

Remove any plugin.

There is no need to call this function explicitly. Every [Fl_Plugin](#) destructor will call this function at destruction time.

The documentation for this class was generated from the following files:

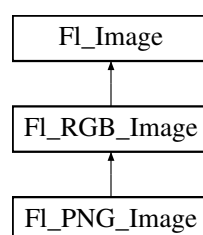
- [Fl_Plugin.H](#)
- [Fl_Preferences.cxx](#)

31.97 FI_PNG_Image Class Reference

The [FI_PNG_Image](#) class supports loading, caching, and drawing of Portable Network Graphics (PNG) image files.

```
#include <FI_PNG_Image.H>
```

Inheritance diagram for [FI_PNG_Image](#):



Public Member Functions

- [FI_PNG_Image](#) (const char *filename)
The constructor loads the named PNG image from the given png filename.
- [FI_PNG_Image](#) (const char *name_png, const unsigned char *buffer, int datasize)
Constructor that reads a PNG image from memory.

Additional Inherited Members

31.97.1 Detailed Description

The [FI_PNG_Image](#) class supports loading, caching, and drawing of Portable Network Graphics (PNG) image files. The class loads colormapped and full-color images and handles color- and alpha-based transparency.

31.97.2 Constructor & Destructor Documentation

31.97.2.1 FI_PNG_Image() [1/2]

```
FI_PNG_Image::FI_PNG_Image (
    const char * filename )
```

The constructor loads the named PNG image from the given png filename.

The destructor frees all memory and server resources that are used by the image.

Use [FI_Image::fail\(\)](#) to check if [FI_PNG_Image](#) failed to load. [fail\(\)](#) returns `ERR_FILE_ACCESS` if the file could not be opened or read, `ERR_FORMAT` if the PNG format could not be decoded, and `ERR_NO_IMAGE` if the image could not be loaded for another reason.

Parameters

in	<i>filename</i>	Name of PNG file to read
----	-----------------	--------------------------

31.97.2.2 FI_PNG_Image() [2/2]

```
FI_PNG_Image::FI_PNG_Image (
    const char * name_png,
    const unsigned char * buffer,
    int maxsize )
```

Constructor that reads a PNG image from memory.

Construct an image from a block of memory inside the application. Fluid offers "binary Data" chunks as a great way to add image data into the C++ source code. `name_png` can be NULL. If a name is given, the image is added to the list of shared images (see: [FI_Shared_Image](#)) and will be available by that name.

Parameters

<i>name_png</i>	A name given to this image or NULL
<i>buffer</i>	Pointer to the start of the PNG image in memory
<i>maxsize</i>	Size in bytes of the memory buffer containing the PNG image

The documentation for this class was generated from the following files:

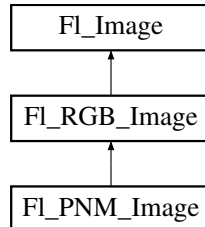
- [FI_PNG_Image.H](#)
- [FI_PNG_Image.cxx](#)

31.98 FI_PNM_Image Class Reference

The [FI_PNM_Image](#) class supports loading, caching, and drawing of Portable Anymap (PNM, PBM, PGM, PPM) image files.

```
#include <Fl_PNM_Image.H>
```

Inheritance diagram for [FI_PNM_Image](#):



Public Member Functions

- [FI_PNM_Image](#) (const char *filename)
The constructor loads the named PNM image.

Additional Inherited Members

31.98.1 Detailed Description

The [FI_PNM_Image](#) class supports loading, caching, and drawing of Portable Anymap (PNM, PBM, PGM, PPM) image files.

The class loads bitmap, grayscale, and full-color images in both ASCII and binary formats.

31.98.2 Constructor & Destructor Documentation

31.98.2.1 FI_PNM_Image()

```
Fl_PNM_Image::Fl_PNM_Image (
    const char * name )
```

The constructor loads the named PNM image.

The destructor frees all memory and server resources that are used by the image.

Use [FI_Image::fail\(\)](#) to check if [FI_PNM_Image](#) failed to load. [fail\(\)](#) returns `ERR_FILE_ACCESS` if the file could not be opened or read, `ERR_FORMAT` if the PNM format could not be decoded, and `ERR_NO_IMAGE` if the image could not be loaded for another reason.

The documentation for this class was generated from the following files:

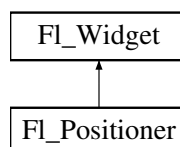
- [FI_PNM_Image.H](#)
- [FI_PNM_Image.cxx](#)

31.99 FI_Positioner Class Reference

This class is provided for Forms compatibility.

```
#include <Fl_Positioner.H>
```

Inheritance diagram for [FI_Positioner](#):



Public Member Functions

- [FI_Positioner](#) (int **x**, int **y**, int **w**, int **h**, const char *l=0)
Creates a new [FI_Positioner](#) widget using the given position, size, and label string.
- int [handle](#) (int)
Handles the specified event.
- int [value](#) (double, double)
Returns the current position in x and y.
- void [xbounds](#) (double, double)
Sets the X axis bounds.
- double [xmaximum](#) () const
Gets the X axis maximum.
- void [xmaximum](#) (double a)
Same as [xbounds](#)([xminimum](#)(), a)
- double [xminimum](#) () const
Gets the X axis minimum.
- void [xminimum](#) (double a)
Same as [xbounds](#)(a, [xmaximum](#)())
- void [xstep](#) (double a)
Sets the stepping value for the X axis.
- double [xvalue](#) () const
Gets the X axis coordinate.
- int [xvalue](#) (double)
Sets the X axis coordinate.
- void [ybounds](#) (double, double)
Sets the Y axis bounds.
- double [ymaximum](#) () const
Gets the Y axis maximum.
- void [ymaximum](#) (double a)
Same as [ybounds](#)([yminimum](#)(), a)
- double [yminimum](#) () const
Gets the Y axis minimum.
- void [yminimum](#) (double a)
Same as [ybounds](#)(a, [ymaximum](#)())
- void [ystep](#) (double a)
Sets the stepping value for the Y axis.
- double [yvalue](#) () const
Gets the Y axis coordinate.
- int [yvalue](#) (double)
Sets the Y axis coordinate.

Protected Member Functions

- void [draw](#) ()
Draws the widget.
- void [draw](#) (int, int, int, int)
- int [handle](#) (int, int, int, int, int)

Additional Inherited Members

31.99.1 Detailed Description

This class is provided for Forms compatibility.

It provides 2D input. It would be useful if this could be put atop another widget so that the crosshairs are on top, but this is not implemented. The color of the crosshairs is [selection_color\(\)](#).



Figure 31.24 Fl_Positioner

31.99.2 Constructor & Destructor Documentation

31.99.2.1 Fl_Positioner()

```
Fl_Positioner::Fl_Positioner (
    int X,
    int Y,
    int W,
    int H,
    const char * l = 0 )
```

Creates a new [Fl_Positioner](#) widget using the given position, size, and label string. The default boxtype is FL_NO_BOX.

31.99.3 Member Function Documentation

31.99.3.1 draw()

```
void Fl_Positioner::draw ( ) [protected], [virtual]
```

Draws the widget.

Never call this function directly. FLTK will schedule redrawing whenever needed. If your widget must be redrawn as soon as possible, call [redraw\(\)](#) instead.

Override this function to draw your own widgets.

If you ever need to call another widget's draw method *from within your own [draw\(\)](#) method*, e.g. for an embedded scrollbar, you can do it (because [draw\(\)](#) is virtual) like this:

```
Fl_Widget *s = &scroll;           // scroll is an embedded Fl_Scrollbar
s->draw();                          // calls Fl_Scrollbar::draw()
```

Implements [Fl_Widget](#).

31.99.3.2 handle()

```
int Fl_Positioner::handle (
    int event ) [virtual]
```

Handles the specified event.

You normally don't call this method directly, but instead let FLTK do it when the user interacts with the widget.

When implemented in a widget, this function must return 0 if the widget does not use the event or 1 otherwise.

Most of the time, you want to call the inherited [handle\(\)](#) method in your overridden method so that you don't short-circuit events that you don't handle. In this last case you should return the callee retval.

Parameters

in	<i>event</i>	the kind of event received
----	--------------	----------------------------

Return values

0	if the event was not used or understood
1	if the event was used and can be deleted

See also

[Fl_Event](#)

Reimplemented from [Fl_Widget](#).

The documentation for this class was generated from the following files:

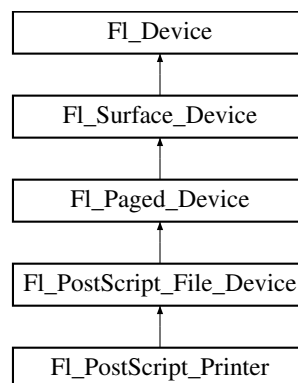
- Fl_Positioner.H
- Fl_Positioner.cxx

31.100 Fl_PostScript_File_Device Class Reference

To send graphical output to a PostScript file.

```
#include <Fl_PostScript.H>
```

Inheritance diagram for Fl_PostScript_File_Device:



Public Member Functions

- const char * [class_name](#) ()
Returns the name of the class of this object.
- void [end_job](#) (void)
To be called at the end of a print job.
- int [end_page](#) (void)
To be called at the end of each page.
- **Fl_PostScript_File_Device** ()
The constructor.
- void [margins](#) (int *left, int *top, int *right, int *bottom)
Computes the dimensions of margins that lie between the printable page area and the full page.
- void [origin](#) (int *x, int *y)
Computes the page coordinates of the current origin of graphics functions.
- void [origin](#) (int x, int y)
Sets the position in page coordinates of the origin of graphics functions.

- int [printable_rect](#) (int *w, int *h)
Computes the width and height of the printable area of the page.
- void [rotate](#) (float angle)
Rotates the graphics operations relatively to paper.
- void [scale](#) (float scale_x, float scale_y=0.)
Changes the scaling of page coordinates.
- int [start_job](#) (FILE *ps_output, int pagecount, enum [FI_Paged_Device::Page_Format](#) format=[FI_Paged_Device::A4](#), enum [FI_Paged_Device::Page_Layout](#) layout=[FI_Paged_Device::PORTRAIT](#))
Begins the session where all graphics requests will go to FILE pointer.
- int [start_job](#) (int pagecount, enum [FI_Paged_Device::Page_Format](#) format=[FI_Paged_Device::A4](#), enum [FI_Paged_Device::Page_Layout](#) layout=[FI_Paged_Device::PORTRAIT](#))
Begins the session where all graphics requests will go to a local PostScript file.
- int [start_job](#) (int pagecount, int *from, int *to)
Don't use with this class.
- int [start_page](#) (void)
Starts a new printed page.
- void [translate](#) (int x, int y)
Translates the current graphics origin accounting for the current rotation.
- void [untranslate](#) (void)
Undoes the effect of a previous [translate\(\)](#) call.
- [~FI_PostScript_File_Device](#) ()
The destructor.

Static Public Attributes

- static const char * [class_id](#) = "FI_PostScript_File_Device"
- static const char * [file_chooser_title](#) = "Select a .ps file"
Label of the PostScript file chooser window.

Protected Member Functions

- [FI_PostScript_Graphics_Driver](#) * [driver](#) ()
Returns the PostScript driver of this drawing surface.

Additional Inherited Members

31.100.1 Detailed Description

To send graphical output to a PostScript file.

This class is used exactly as the [FI_Printer](#) class except for the [start_job\(\)](#) call, two variants of which are usable and allow to specify what page format and layout are desired.

31.100.2 Member Function Documentation

31.100.2.1 [class_name\(\)](#)

```
const char * Fl_PostScript_File_Device::class_name ( ) [inline], [virtual]
```

Returns the name of the class of this object.

Use of the [class_name\(\)](#) function is discouraged because it will be removed from future FLTK versions.

The class of an instance of an [FI_Device](#) subclass can be checked with code such as:

```
if ( instance->class_name() == Fl_Printer::class_id ) { ... }
```

Reimplemented from [FI_Paged_Device](#).

Reimplemented in [FI_PostScript_Printer](#).

31.100.2.2 end_job()

```
void Fl_PostScript_File_Device::end_job (
    void ) [virtual]
```

To be called at the end of a print job.

Reimplemented from [Fl_Paged_Device](#).

31.100.2.3 end_page()

```
int Fl_PostScript_File_Device::end_page (
    void ) [virtual]
```

To be called at the end of each page.

Returns

0 if OK, non-zero if any error.

Reimplemented from [Fl_Paged_Device](#).

31.100.2.4 margins()

```
void Fl_PostScript_File_Device::margins (
    int * left,
    int * top,
    int * right,
    int * bottom ) [virtual]
```

Computes the dimensions of margins that lie between the printable page area and the full page.

Values are in the same unit as that used by FLTK drawing functions. They are changed by [scale\(\)](#) calls.

Parameters

out	<i>left</i>	If non-null, *left is set to the left margin size.
out	<i>top</i>	If non-null, *top is set to the top margin size.
out	<i>right</i>	If non-null, *right is set to the right margin size.
out	<i>bottom</i>	If non-null, *bottom is set to the bottom margin size.

Reimplemented from [Fl_Paged_Device](#).

31.100.2.5 origin() [1/2]

```
void Fl_PostScript_File_Device::origin (
    int * x,
    int * y ) [virtual]
```

Computes the page coordinates of the current origin of graphics functions.

Parameters

out	<i>x</i>	If non-null, *x is set to the horizontal page offset of graphics origin.
out	<i>y</i>	Same as above, vertically.

Reimplemented from [Fl_Paged_Device](#).

31.100.2.6 origin() [2/2]

```
void Fl_PostScript_File_Device::origin (
```

```

    int x,
    int y ) [virtual]

```

Sets the position in page coordinates of the origin of graphics functions.

Arguments should be expressed relatively to the result of a previous `printable_rect()` call. That is, `printable_rect(&w, &h); origin(w/2, 0);` sets the graphics origin at the top center of the page printable area. `Origin()` calls are not affected by `rotate()` calls. Successive `origin()` calls don't combine their effects.

Parameters

in	<i>x</i>	Horizontal position in page coordinates of the desired origin of graphics functions.
in	<i>y</i>	Same as above, vertically.

Reimplemented from [Fl_Paged_Device](#).

31.100.2.7 printable_rect()

```

int Fl_PostScript_File_Device::printable_rect (
    int * w,
    int * h ) [virtual]

```

Computes the width and height of the printable area of the page.

Values are in the same unit as that used by FLTK drawing functions, are unchanged by calls to `origin()`, but are changed by `scale()` calls. Values account for the user-selected paper type and print orientation.

Returns

0 if OK, non-zero if any error

Reimplemented from [Fl_Paged_Device](#).

31.100.2.8 rotate()

```

void Fl_PostScript_File_Device::rotate (
    float angle ) [virtual]

```

Rotates the graphics operations relatively to paper.

The rotation is centered on the current graphics origin. Successive `rotate()` calls don't combine their effects.

Parameters

<i>angle</i>	Rotation angle in counter-clockwise degrees.
--------------	--

Reimplemented from [Fl_Paged_Device](#).

31.100.2.9 scale()

```

void Fl_PostScript_File_Device::scale (
    float scale_x,
    float scale_y = 0. ) [virtual]

```

Changes the scaling of page coordinates.

This function also resets the origin of graphics functions at top left of printable page area. After a `scale()` call, do a `printable_rect()` call to get the new dimensions of the printable page area. Successive `scale()` calls don't combine their effects.

Parameters

<i>scale_x</i>	Horizontal dimensions of plot are multiplied by this quantity.
----------------	--

Parameters

<i>scale_x</i> <i>_y</i>	Same as above, vertically. The value 0. is equivalent to setting <code>scale_y = scale_x</code> . Thus, <code>scale(factor)</code> ; is equivalent to <code>scale(factor, factor)</code> ;
-----------------------------	--

Reimplemented from [Fl_Paged_Device](#).

31.100.2.10 start_job() [1/3]

```
int Fl_PostScript_File_Device::start_job (
    FILE * ps_output,
    int pagecount,
    enum Fl_Paged_Device::Page_Format format = Fl_Paged_Device::A4,
    enum Fl_Paged_Device::Page_Layout layout = Fl_Paged_Device::PORTRAIT )
```

Begins the session where all graphics requests will go to FILE pointer.

Parameters

<i>ps_output</i>	A writable FILE pointer that will receive PostScript output and that should not be closed until after end_job() has been called.
<i>pagecount</i>	The total number of pages to be created. Use 0 if this number is unknown when this function is called.
<i>format</i>	Desired page format.
<i>layout</i>	Desired page layout.

Returns

always 0.

31.100.2.11 start_job() [2/3]

```
int Fl_PostScript_File_Device::start_job (
    int pagecount,
    enum Fl_Paged_Device::Page_Format format = Fl_Paged_Device::A4,
    enum Fl_Paged_Device::Page_Layout layout = Fl_Paged_Device::PORTRAIT )
```

Begins the session where all graphics requests will go to a local PostScript file.

Opens a file dialog entitled with [Fl_PostScript_File_Device::file_chooser_title](#) to select an output PostScript file.

Parameters

<i>pagecount</i>	The total number of pages to be created. Use 0 if this number is unknown when this function is called.
<i>format</i>	Desired page format.
<i>layout</i>	Desired page layout.

Returns

0 if OK, 1 if user cancelled the file dialog, 2 if fopen failed on user-selected output file.

31.100.2.12 start_job() [3/3]

```
int Fl_PostScript_File_Device::start_job (
    int pagecount,
```

```

    int * from,
    int * to ) [virtual]

```

Don't use with this class.

Reimplemented from [Fl_Paged_Device](#).

Reimplemented in [Fl_PostScript_Printer](#).

31.100.2.13 start_page()

```

int Fl_PostScript_File_Device::start_page (
    void ) [virtual]

```

Starts a new printed page.

The page coordinates are initially in points, i.e., 1/72 inch, and with origin at the top left of the printable page area.

Returns

0 if OK, non-zero if any error

Reimplemented from [Fl_Paged_Device](#).

31.100.2.14 translate()

```

void Fl_PostScript_File_Device::translate (
    int x,
    int y ) [virtual]

```

Translates the current graphics origin accounting for the current rotation.

This function is only useful after a [rotate\(\)](#) call. Each [translate\(\)](#) call must be matched by an [untranslate\(\)](#) call.

Successive [translate\(\)](#) calls add up their effects.

Reimplemented from [Fl_Paged_Device](#).

31.100.2.15 untranslate()

```

void Fl_PostScript_File_Device::untranslate (
    void ) [virtual]

```

Undoes the effect of a previous [translate\(\)](#) call.

Reimplemented from [Fl_Paged_Device](#).

The documentation for this class was generated from the following files:

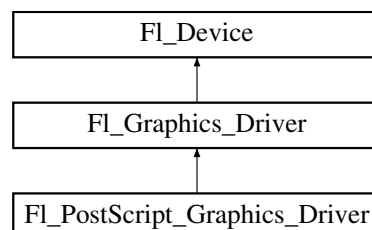
- [Fl_PostScript.H](#)
- [Fl_PostScript.cxx](#)

31.101 Fl_PostScript_Graphics_Driver Class Reference

PostScript graphical backend.

```
#include <Fl_PostScript.H>
```

Inheritance diagram for [Fl_PostScript_Graphics_Driver](#):



Public Member Functions

- void [arc](#) (double x, double y, double r, double start, double a)
 - see [fl_arc\(double x, double y, double r, double start, double end\)](#).*
- void [arc](#) (int x, int y, int w, int h, double a1, double a2)
 - see [fl_arc\(int x, int y, int w, int h, double a1, double a2\)](#).*
- void [begin_complex_polygon](#) ()
 - see [fl_begin_complex_polygon\(\)](#).*
- void [begin_line](#) ()
 - see [fl_begin_line\(\)](#).*
- void [begin_loop](#) ()
 - see [fl_begin_loop\(\)](#).*
- void [begin_points](#) ()
 - see [fl_begin_points\(\)](#).*
- void [begin_polygon](#) ()
 - see [fl_begin_polygon\(\)](#).*
- void [circle](#) (double x, double y, double r)
 - see [fl_circle\(double x, double y, double r\)](#).*
- const char * [class_name](#) ()
 - Returns the name of the class of this object.*
- int [clip_box](#) (int x, int y, int w, int h, int &X, int &Y, int &W, int &H)
 - see [fl_clip_box\(int x, int y, int w, int h, int &X, int &Y, int &W, int &H\)](#).*
- int [clocale_printf](#) (const char *format,...)
 - Shields output PostScript data from modifications of the current locale.*
- void [color](#) (FI_Color c)
 - see [fl_color\(FI_Color c\)](#).*
- void [color](#) (uchar r, uchar g, uchar b)
 - see [fl_color\(uchar r, uchar g, uchar b\)](#).*
- void [curve](#) (double x, double y, double x1, double y1, double x2, double y2, double x3, double y3)
 - see [fl_curve\(double X0, double Y0, double X1, double Y1, double X2, double Y2, double X3, double Y3\)](#).*
- int [descent](#) ()
 - see [fl_descent\(\)](#).*
- void [draw](#) (const char *s, int nBytes, int x, int y)
 - see [fl_draw\(const char *str, int n, int x, int y\)](#).*
- void [draw](#) (FI_Bitmap *bitmap, int XP, int YP, int WP, int HP, int cx, int cy)
 - Draws an [FI_Bitmap](#) object to the device.*
- void [draw](#) (FI_Pixmap *pixmap, int XP, int YP, int WP, int HP, int cx, int cy)
 - Draws an [FI_Pixmap](#) object to the device.*
- void [draw](#) (FI_RGB_Image *rgb, int XP, int YP, int WP, int HP, int cx, int cy)
 - Draws an [FI_RGB_Image](#) object to the device.*
- void [draw](#) (int angle, const char *str, int n, int x, int y)
 - see [fl_draw\(int angle, const char *str, int n, int x, int y\)](#).*
- void [draw_image](#) (const uchar *d, int x, int y, int w, int h, int delta=3, int ldelta=0)
 - see [fl_draw_image\(const uchar* buf, int X,int Y,int W,int H, int D, int L\)](#).*
- void [draw_image](#) (FI_Draw_Image_Cb call, void *data, int x, int y, int w, int h, int delta=3)
 - see [fl_draw_image\(FI_Draw_Image_Cb cb, void* data, int X,int Y,int W,int H, int D\)](#).*
- void [draw_image_mono](#) (const uchar *d, int x, int y, int w, int h, int delta=1, int ld=0)
 - see [fl_draw_image_mono\(const uchar* buf, int X,int Y,int W,int H, int D, int L\)](#).*
- void [draw_image_mono](#) (FI_Draw_Image_Cb call, void *data, int x, int y, int w, int h, int delta=1)
 - see [fl_draw_image_mono\(FI_Draw_Image_Cb cb, void* data, int X,int Y,int W,int H, int D\)](#).*
- int [draw_scaled](#) (FI_Image *img, int XP, int YP, int WP, int HP)

- Draws an [FL_Image](#) scaled to width W & height H with top-left corner at X, Y .*
- void [end_complex_polygon](#) ()
see [fl_end_complex_polygon\(\)](#).
 - void [end_line](#) ()
see [fl_end_line\(\)](#).
 - void [end_loop](#) ()
see [fl_end_loop\(\)](#).
 - void [end_points](#) ()
see [fl_end_points\(\)](#).
 - void [end_polygon](#) ()
see [fl_end_polygon\(\)](#).
 - **[FI_PostScript_Graphics_Driver](#)** ()
The constructor.
 - void [font](#) (int face, int size)
see [fl_font\(FI_Font face, FI_Fontsize size\)](#).
 - void [gap](#) ()
see [fl_gap\(\)](#).
 - int [height](#) ()
see [fl_height\(\)](#).
 - void [line](#) (int x1, int y1, int x2, int y2)
see [fl_line\(int x, int y, int x1, int y1\)](#).
 - void [line](#) (int x1, int y1, int x2, int y2, int x3, int y3)
see [fl_line\(int x, int y, int x1, int y1, int x2, int y2\)](#).
 - void [line_style](#) (int style, int width=0, char *dashes=0)
see [fl_line_style\(int style, int width, char dashes\)](#).*
 - void [loop](#) (int x0, int y0, int x1, int y1, int x2, int y2)
see [fl_loop\(int x0, int y0, int x1, int y1, int x2, int y2\)](#).
 - void [loop](#) (int x0, int y0, int x1, int y1, int x2, int y2, int x3, int y3)
see [fl_loop\(int x0, int y0, int x1, int y1, int x2, int y2, int x3, int y3\)](#).
 - int [not_clipped](#) (int x, int y, int w, int h)
see [fl_not_clipped\(int x, int y, int w, int h\)](#).
 - void [pie](#) (int x, int y, int w, int h, double a1, double a2)
see [fl_pie\(int x, int y, int w, int h, double a1, double a2\)](#).
 - void [point](#) (int x, int y)
see [fl_point\(int x, int y\)](#).
 - void [polygon](#) (int x0, int y0, int x1, int y1, int x2, int y2)
see [fl_polygon\(int x0, int y0, int x1, int y1, int x2, int y2\)](#).
 - void [polygon](#) (int x0, int y0, int x1, int y1, int x2, int y2, int x3, int y3)
see [fl_polygon\(int x0, int y0, int x1, int y1, int x2, int y2, int x3, int y3\)](#).
 - void [pop_clip](#) ()
see [fl_pop_clip\(\)](#).
 - void [push_clip](#) (int x, int y, int w, int h)
see [fl_push_clip\(int x, int y, int w, int h\)](#).
 - void [push_no_clip](#) ()
see [fl_push_no_clip\(\)](#).
 - void [rect](#) (int x, int y, int w, int h)
see [fl_rect\(int x, int y, int w, int h\)](#).
 - void [rectf](#) (int x, int y, int w, int h)
see [fl_rectf\(int x, int y, int w, int h\)](#).
 - void [rtl_draw](#) (const char *s, int n, int x, int y)
*see [fl_rtl_draw\(const char *str, int n, int x, int y\)](#).*

- void `text_extents` (const char *c, int n, int &dx, int &dy, int &w, int &h)
see `fl_text_extents(const char*, int n, int& dx, int& dy, int& w, int& h)`.
- void `transformed_vertex` (double x, double y)
see `fl_transformed_vertex(double xf, double yf)`.
- void `vertex` (double x, double y)
see `fl_vertex(double x, double y)`.
- double `width` (const char *, int)
see `fl_width(const char *str, int n)`.
- double `width` (unsigned int u)
see `fl_width(unsigned int n)`.
- void `xyline` (int x, int y, int x1)
see `fl_xyline(int x, int y, int x1)`.
- void `xyline` (int x, int y, int x1, int y2)
see `fl_xyline(int x, int y, int x1, int y2)`.
- void `xyline` (int x, int y, int x1, int y2, int x3)
see `fl_xyline(int x, int y, int x1, int y2, int x3)`.
- void `yxline` (int x, int y, int y1)
see `fl_yxline(int x, int y, int y1)`.
- void `yxline` (int x, int y, int y1, int x2)
see `fl_yxline(int x, int y, int y1, int x2)`.
- void `yxline` (int x, int y, int y1, int x2, int y3)
see `fl_yxline(int x, int y, int y1, int x2, int y3)`.
- `~Fl_PostScript_Graphics_Driver` ()
The destructor.

Static Public Attributes

- static const char * `class_id` = "Fl_PostScript_Graphics_Driver"

Additional Inherited Members

31.101.1 Detailed Description

PostScript graphical backend.

PostScript text uses vectorial fonts when using the FLTK standard fonts

and the latin alphabet or a few other characters listed in the following table. The latin alphabet means all unicode characters between U+0020 and U+017F, or, in other words, the ASCII, Latin-1 Supplement and Latin Extended-A charts.

Char	Code-point	Name	Char	Code-point	Name	Char	Code-point	Name
<i>f</i>	U+0192	florin	,	U+201A	quotesinglbase [™]		U+2122	trademark
^	U+02C6	circumflex	“	U+201C	quotedblleft		U+2202	partialdiff
ˇ	U+02C7	caron	”	U+201D	quotedblright		U+2206	Delta
˘	U+02D8	breve	„	U+201E	quotedblbase		U+2211	summation
˙	U+02D9	dotaccent	†	U+2020	dagger		U+221A	radical
	U+02DA	ring	‡	U+2021	daggerdbl		U+221E	infinity
◌̣	U+02DB	ogonek	•	U+2022	bullet		U+2260	notequal
~	U+02DC	tilde	…	U+2026	ellipsis		U+2264	lessequal
”	U+02DD	hungarumlaut	‰	U+2030	perthousand		U+2265	greaterequal
–	U+2013	endash	◁	U+2039	guilsinglleft		U+25CA	lozenge
—	U+2014	emdash	▷	U+203A	guilsinglright	fi	U+FB01	fi
‘	U+2018	quoteleft	/	U+2044	fraction	fl	U+FB02	fl

Char	Code-point	Name	Char	Code-point	Name	Char	Code-point	Name
'	U+2019	quoteright	€	U+20AC	Euro		U+F8FF	apple (Mac OS only)

All other unicode characters or all other fonts (FL_FREE_FONT and above) are output as a bitmap. FLTK standard fonts are output using the corresponding PostScript standard fonts.

31.101.2 Member Function Documentation

31.101.2.1 arc() [1/2]

```
void Fl_PostScript_Graphics_Driver::arc (
    double x,
    double y,
    double r,
    double start,
    double end ) [virtual]
```

see [fl_arc\(double x, double y, double r, double start, double end\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.101.2.2 arc() [2/2]

```
void Fl_PostScript_Graphics_Driver::arc (
    int x,
    int y,
    int w,
    int h,
    double a1,
    double a2 ) [virtual]
```

see [fl_arc\(int x, int y, int w, int h, double a1, double a2\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.101.2.3 begin_complex_polygon()

```
void Fl_PostScript_Graphics_Driver::begin_complex_polygon ( ) [inline], [virtual]
```

see [fl_begin_complex_polygon\(\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.101.2.4 begin_line()

```
void Fl_PostScript_Graphics_Driver::begin_line ( ) [virtual]
```

see [fl_begin_line\(\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.101.2.5 begin_loop()

```
void Fl_PostScript_Graphics_Driver::begin_loop ( ) [virtual]
```

see [fl_begin_loop\(\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.101.2.6 begin_points()

```
void Fl_PostScript_Graphics_Driver::begin_points ( ) [virtual]
```

see [fl_begin_points\(\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.101.2.7 begin_polygon()

```
void Fl_PostScript_Graphics_Driver::begin_polygon ( ) [virtual]
```

see [fl_begin_polygon\(\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.101.2.8 circle()

```
void Fl_PostScript_Graphics_Driver::circle (
    double x,
    double y,
    double r ) [virtual]
```

see [fl_circle\(double x, double y, double r\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.101.2.9 class_name()

```
const char * Fl_PostScript_Graphics_Driver::class_name ( ) [inline], [virtual]
```

Returns the name of the class of this object.

Use of the [class_name\(\)](#) function is discouraged because it will be removed from future FLTK versions.

The class of an instance of an [Fl_Device](#) subclass can be checked with code such as:

```
if ( instance->class_name() == Fl_Printer::class_id ) { ... }
```

Reimplemented from [Fl_Graphics_Driver](#).

31.101.2.10 clip_box()

```
int Fl_PostScript_Graphics_Driver::clip_box (
    int x,
    int y,
    int w,
    int h,
    int & X,
    int & Y,
    int & W,
    int & H ) [virtual]
```

see [fl_clip_box\(int x, int y, int w, int h, int &X, int &Y, int &W, int &H\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.101.2.11 clocale_printf()

```
int Fl_PostScript_Graphics_Driver::clocale_printf (
    const char * format,
    ... )
```

Shields output PostScript data from modifications of the current locale.

It typically avoids PostScript errors caused if the current locale uses comma instead of dot as "decimal point".

Parameters

<i>format</i>	directives controlling output PostScript data
---------------	---

Returns

value returned by `vfprintf()` call

31.101.2.12 color() [1/2]

```
void Fl_PostScript_Graphics_Driver::color (
    Fl_Color c ) [virtual]
```

see [fl_color\(Fl_Color c\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.101.2.13 color() [2/2]

```
void Fl_PostScript_Graphics_Driver::color (
    uchar r,
    uchar g,
    uchar b ) [virtual]
```

see [fl_color\(uchar r, uchar g, uchar b\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.101.2.14 curve()

```
void Fl_PostScript_Graphics_Driver::curve (
    double X0,
    double Y0,
    double X1,
    double Y1,
    double X2,
    double Y2,
    double X3,
    double Y3 ) [virtual]
```

see [fl_curve\(double X0, double Y0, double X1, double Y1, double X2, double Y2, double X3, double Y3\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.101.2.15 descent()

```
int Fl_PostScript_Graphics_Driver::descent ( ) [virtual]
```

see [fl_descent\(\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.101.2.16 draw() [1/5]

```
void Fl_PostScript_Graphics_Driver::draw (
    const char * str,
    int n,
    int x,
    int y ) [inline], [virtual]
```

see [fl_draw\(const char *str, int n, int x, int y\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.101.2.17 draw() [2/5]

```
void Fl_PostScript_Graphics_Driver::draw (
    Fl_Bitmap * bm,
```

```
int XP,  
int YP,  
int WP,  
int HP,  
int cx,  
int cy ) [virtual]
```

Draws an [Fl_Bitmap](#) object to the device.

Specifies a bounding box for the image, with the origin (upper left-hand corner) of the image offset by the *cx* and *cy* arguments.

Reimplemented from [Fl_Graphics_Driver](#).

31.101.2.18 draw() [3/5]

```
void Fl_PostScript_Graphics_Driver::draw (  
    Fl_Pixmap * pxm,  
    int XP,  
    int YP,  
    int WP,  
    int HP,  
    int cx,  
    int cy ) [virtual]
```

Draws an [Fl_Pixmap](#) object to the device.

Specifies a bounding box for the image, with the origin (upper left-hand corner) of the image offset by the *cx* and *cy* arguments.

Reimplemented from [Fl_Graphics_Driver](#).

31.101.2.19 draw() [4/5]

```
void Fl_PostScript_Graphics_Driver::draw (  
    Fl_RGB_Image * rgb,  
    int XP,  
    int YP,  
    int WP,  
    int HP,  
    int cx,  
    int cy ) [virtual]
```

Draws an [Fl_RGB_Image](#) object to the device.

Specifies a bounding box for the image, with the origin (upper left-hand corner) of the image offset by the *cx* and *cy* arguments.

Reimplemented from [Fl_Graphics_Driver](#).

31.101.2.20 draw() [5/5]

```
void Fl_PostScript_Graphics_Driver::draw (  
    int angle,  
    const char * str,  
    int n,  
    int x,  
    int y ) [virtual]
```

see [fl_draw\(int angle, const char *str, int n, int x, int y\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.101.2.21 draw_image() [1/2]

```
void Fl_PostScript_Graphics_Driver::draw_image (  

```

```
    const uchar * buf,  
    int X,  
    int Y,  
    int W,  
    int H,  
    int D = 3,  
    int L = 0 ) [virtual]
```

see [fl_draw_image\(const uchar* buf, int X,int Y,int W,int H, int D, int L\)](#).
Reimplemented from [Fl_Graphics_Driver](#).

31.101.2.22 draw_image() [2/2]

```
void Fl_PostScript_Graphics_Driver::draw_image (  
    Fl_Draw_Image_Cb cb,  
    void * data,  
    int X,  
    int Y,  
    int W,  
    int H,  
    int D = 3 ) [virtual]
```

see [fl_draw_image\(Fl_Draw_Image_Cb cb, void* data, int X,int Y,int W,int H, int D\)](#).
Reimplemented from [Fl_Graphics_Driver](#).

31.101.2.23 draw_image_mono() [1/2]

```
void Fl_PostScript_Graphics_Driver::draw_image_mono (  
    const uchar * buf,  
    int X,  
    int Y,  
    int W,  
    int H,  
    int D = 1,  
    int L = 0 ) [virtual]
```

see [fl_draw_image_mono\(const uchar* buf, int X,int Y,int W,int H, int D, int L\)](#).
Reimplemented from [Fl_Graphics_Driver](#).

31.101.2.24 draw_image_mono() [2/2]

```
void Fl_PostScript_Graphics_Driver::draw_image_mono (  
    Fl_Draw_Image_Cb cb,  
    void * data,  
    int X,  
    int Y,  
    int W,  
    int H,  
    int D = 1 ) [virtual]
```

see [fl_draw_image_mono\(Fl_Draw_Image_Cb cb, void* data, int X,int Y,int W,int H, int D\)](#).
Reimplemented from [Fl_Graphics_Driver](#).

31.101.2.25 draw_scaled()

```
int Fl_PostScript_Graphics_Driver::draw_scaled (  
    Fl_Image * img,  
    int X,  
    int Y,
```

```
        int W,  
        int H ) [virtual]
```

Draws an [FI_Image](#) scaled to width W & height H with top-left corner at X,Y.

Returns

zero when the graphics driver doesn't implement scaled drawing, non-zero if it does implement it.

Reimplemented from [FI_Graphics_Driver](#).

31.101.2.26 end_complex_polygon()

```
void Fl_PostScript_Graphics_Driver::end_complex_polygon ( ) [inline], [virtual]
```

see [fl_end_complex_polygon\(\)](#).

Reimplemented from [FI_Graphics_Driver](#).

31.101.2.27 end_line()

```
void Fl_PostScript_Graphics_Driver::end_line ( ) [virtual]
```

see [fl_end_line\(\)](#).

Reimplemented from [FI_Graphics_Driver](#).

31.101.2.28 end_loop()

```
void Fl_PostScript_Graphics_Driver::end_loop ( ) [virtual]
```

see [fl_end_loop\(\)](#).

Reimplemented from [FI_Graphics_Driver](#).

31.101.2.29 end_points()

```
void Fl_PostScript_Graphics_Driver::end_points ( ) [virtual]
```

see [fl_end_points\(\)](#).

Reimplemented from [FI_Graphics_Driver](#).

31.101.2.30 end_polygon()

```
void Fl_PostScript_Graphics_Driver::end_polygon ( ) [virtual]
```

see [fl_end_polygon\(\)](#).

Reimplemented from [FI_Graphics_Driver](#).

31.101.2.31 font()

```
void Fl_PostScript_Graphics_Driver::font (
```

```
        int face,  
        int fsize ) [virtual]
```

see [fl_font\(FI_Font face, FI_Fontsize size\)](#).

Reimplemented from [FI_Graphics_Driver](#).

31.101.2.32 gap()

```
void Fl_PostScript_Graphics_Driver::gap ( ) [inline], [virtual]
```

see [fl_gap\(\)](#).

Reimplemented from [FI_Graphics_Driver](#).

31.101.2.33 height()

```
int Fl_PostScript_Graphics_Driver::height ( ) [virtual]
```

see [fl_height\(\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.101.2.34 line() [1/2]

```
void Fl_PostScript_Graphics_Driver::line (
    int x,
    int y,
    int x1,
    int y1 ) [virtual]
```

see [fl_line\(int x, int y, int x1, int y1\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.101.2.35 line() [2/2]

```
void Fl_PostScript_Graphics_Driver::line (
    int x,
    int y,
    int x1,
    int y1,
    int x2,
    int y2 ) [virtual]
```

see [fl_line\(int x, int y, int x1, int y1, int x2, int y2\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.101.2.36 line_style()

```
void Fl_PostScript_Graphics_Driver::line_style (
    int style,
    int width = 0,
    char * dashes = 0 ) [virtual]
```

see [fl_line_style\(int style, int width, char* dashes\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.101.2.37 loop() [1/2]

```
void Fl_PostScript_Graphics_Driver::loop (
    int x0,
    int y0,
    int x1,
    int y1,
    int x2,
    int y2 ) [virtual]
```

see [fl_loop\(int x0, int y0, int x1, int y1, int x2, int y2\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.101.2.38 loop() [2/2]

```
void Fl_PostScript_Graphics_Driver::loop (
    int x0,
    int y0,
```

```
    int x1,  
    int y1,  
    int x2,  
    int y2,  
    int x3,  
    int y3 ) [virtual]
```

see [fl_loop\(int x0, int y0, int x1, int y1, int x2, int y2, int x3, int y3\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.101.2.39 not_clipped()

```
int Fl_PostScript_Graphics_Driver::not_clipped (  
    int x,  
    int y,  
    int w,  
    int h ) [virtual]
```

see [fl_not_clipped\(int x, int y, int w, int h\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.101.2.40 pie()

```
void Fl_PostScript_Graphics_Driver::pie (  
    int x,  
    int y,  
    int w,  
    int h,  
    double a1,  
    double a2 ) [virtual]
```

see [fl_pie\(int x, int y, int w, int h, double a1, double a2\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.101.2.41 point()

```
void Fl_PostScript_Graphics_Driver::point (  
    int x,  
    int y ) [virtual]
```

see [fl_point\(int x, int y\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.101.2.42 polygon() [1/2]

```
void Fl_PostScript_Graphics_Driver::polygon (  
    int x0,  
    int y0,  
    int x1,  
    int y1,  
    int x2,  
    int y2 ) [virtual]
```

see [fl_polygon\(int x0, int y0, int x1, int y1, int x2, int y2\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.101.2.43 polygon() [2/2]

```
void Fl_PostScript_Graphics_Driver::polygon (  

```

```
int x0,  
int y0,  
int x1,  
int y1,  
int x2,  
int y2,  
int x3,  
int y3 ) [virtual]
```

see [fl_polygon\(int x0, int y0, int x1, int y1, int x2, int y2, int x3, int y3\)](#).
Reimplemented from [Fl_Graphics_Driver](#).

31.101.2.44 pop_clip()

```
void Fl_PostScript_Graphics_Driver::pop_clip ( ) [virtual]
```

see [fl_pop_clip\(\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.101.2.45 push_clip()

```
void Fl_PostScript_Graphics_Driver::push_clip (  
int x,  
int y,  
int w,  
int h ) [virtual]
```

see [fl_push_clip\(int x, int y, int w, int h\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.101.2.46 push_no_clip()

```
void Fl_PostScript_Graphics_Driver::push_no_clip ( ) [virtual]
```

see [fl_push_no_clip\(\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.101.2.47 rect()

```
void Fl_PostScript_Graphics_Driver::rect (  
int x,  
int y,  
int w,  
int h ) [virtual]
```

see [fl_rect\(int x, int y, int w, int h\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.101.2.48 rectf()

```
void Fl_PostScript_Graphics_Driver::rectf (  
int x,  
int y,  
int w,  
int h ) [virtual]
```

see [fl_rectf\(int x, int y, int w, int h\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.101.2.49 rtl_draw()

```
void Fl_PostScript_Graphics_Driver::rtl_draw (
    const char * str,
    int n,
    int x,
    int y ) [virtual]
```

see [fl_rtl_draw\(const char *str, int n, int x, int y\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.101.2.50 text_extents()

```
void Fl_PostScript_Graphics_Driver::text_extents (
    const char * t,
    int n,
    int & dx,
    int & dy,
    int & w,
    int & h ) [virtual]
```

see [fl_text_extents\(const char*, int n, int& dx, int& dy, int& w, int& h\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.101.2.51 transformed_vertex()

```
void Fl_PostScript_Graphics_Driver::transformed_vertex (
    double xf,
    double yf ) [virtual]
```

see [fl_transformed_vertex\(double xf, double yf\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.101.2.52 vertex()

```
void Fl_PostScript_Graphics_Driver::vertex (
    double x,
    double y ) [virtual]
```

see [fl_vertex\(double x, double y\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.101.2.53 width() [1/2]

```
double Fl_PostScript_Graphics_Driver::width (
    const char * str,
    int n ) [virtual]
```

see [fl_width\(const char *str, int n\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.101.2.54 width() [2/2]

```
double Fl_PostScript_Graphics_Driver::width (
    unsigned int c ) [virtual]
```

see [fl_width\(unsigned int n\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.101.2.55 xyline() [1/3]

```
void Fl_PostScript_Graphics_Driver::xyline (
    int x,
    int y,
    int x1 ) [virtual]
```

see [fl_xyline\(int x, int y, int x1\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.101.2.56 xyline() [2/3]

```
void Fl_PostScript_Graphics_Driver::xyline (
    int x,
    int y,
    int x1,
    int y2 ) [virtual]
```

see [fl_xyline\(int x, int y, int x1, int y2\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.101.2.57 xyline() [3/3]

```
void Fl_PostScript_Graphics_Driver::xyline (
    int x,
    int y,
    int x1,
    int y2,
    int x3 ) [virtual]
```

see [fl_xyline\(int x, int y, int x1, int y2, int x3\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.101.2.58 yxline() [1/3]

```
void Fl_PostScript_Graphics_Driver::yxline (
    int x,
    int y,
    int y1 ) [virtual]
```

see [fl_yxline\(int x, int y, int y1\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.101.2.59 yxline() [2/3]

```
void Fl_PostScript_Graphics_Driver::yxline (
    int x,
    int y,
    int y1,
    int x2 ) [virtual]
```

see [fl_yxline\(int x, int y, int y1, int x2\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.101.2.60 yxline() [3/3]

```
void Fl_PostScript_Graphics_Driver::yxline (
    int x,
    int y,
```

```

    int y1,
    int x2,
    int y3 ) [virtual]

```

see [fl_yxline\(int x, int y, int y1, int x2, int y3\)](#).

Reimplemented from [FI_Graphics_Driver](#).

The documentation for this class was generated from the following files:

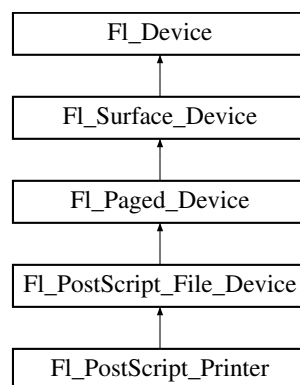
- [FI_PostScript.H](#)
- [FI_PostScript.cxx](#)

31.102 FI_PostScript_Printer Class Reference

Print support under Unix/Linux.

```
#include <FI_Printer.H>
```

Inheritance diagram for FI_PostScript_Printer:



Public Member Functions

- const char * [class_name](#) ()
Returns the name of the class of this object.
- int [start_job](#) (int pages, int *firstpage=NULL, int *lastpage=NULL)
Starts a print job.

Static Public Attributes

- static const char * [class_id](#) = FI_Printer::class_id

Protected Member Functions

- [FI_PostScript_Printer](#) (void)
The constructor.

Friends

- class [FI_Printer](#)

Additional Inherited Members

31.102.1 Detailed Description

Print support under Unix/Linux.

Class [FI_PostScript_Printer](#) is implemented only on the Unix/Linux platform. It has no public constructor. Use [FI_Printer](#) instead that is cross-platform and has the same API.

31.102.2 Member Function Documentation

31.102.2.1 class_name()

```
const char * Fl_PostScript_Printer::class_name ( ) [inline], [virtual]
```

Returns the name of the class of this object.

Use of the `class_name()` function is discouraged because it will be removed from future FLTK versions.

The class of an instance of an `Fl_Device` subclass can be checked with code such as:

```
if ( instance->class_name() == Fl_Printer::class_id ) { ... }
```

Reimplemented from `Fl_PostScript_File_Device`.

31.102.2.2 start_job()

```
int Fl_PostScript_Printer::start_job (
    int pages,
    int * firstpage = NULL,
    int * lastpage = NULL ) [virtual]
```

Starts a print job.

Reimplemented from `Fl_PostScript_File_Device`.

The documentation for this class was generated from the following files:

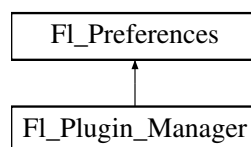
- `Fl_Printer.H`
- `Fl_PostScript.cxx`
- `Fl_Printer.cxx`

31.103 FI_Preferences Class Reference

`Fl_Preferences` provides methods to store user settings between application starts.

```
#include <Fl_Preferences.H>
```

Inheritance diagram for `Fl_Preferences`:



Classes

- struct `Entry`
- class `Name`

'Name' provides a simple method to create numerical or more complex procedural names for entries and groups on the fly.

- class `Node`
- class `RootNode`

Public Types

- typedef void * `ID`

Every `Fl_Preferences-Group` has a unique ID.

- enum `Root` { `SYSTEM` =0 , `USER` }

Define the scope of the preferences.

Public Member Functions

- char **clear** ()
Delete all groups and all entries.
- char **deleteAllEntries** ()
Delete all entries.
- char **deleteAllGroups** ()
Delete all groups.
- char **deleteEntry** (const char *entry)
Deletes a single name/value pair.
- char **deleteGroup** (const char *group)
Deletes a group.
- int **entries** ()
Returns the number of entries (name/value pairs) in a group.
- const char * **entry** (int index)
Returns the name of an entry.
- char **entryExists** (const char *key)
Returns non-zero if an entry with this name exists.
- **FI_Preferences** (const char *path, const char *vendor, const char *application)
Use this constructor to create or read a preferences file at an arbitrary position in the file system.
- **FI_Preferences** (const **FI_Preferences** &)
Create another reference to a Preferences group.
- **FI_Preferences** (**FI_Preferences** &parent, const char *group)
Generate or read a new group of entries within another group.
- **FI_Preferences** (**FI_Preferences** &parent, int groupIndex)
Open a child group using a given index.
- **FI_Preferences** (**FI_Preferences** *parent, const char *group)
Create or access a group of preferences using a name.
- **FI_Preferences** (**FI_Preferences** *parent, int groupIndex)
- **FI_Preferences** (ID id)
Create a new dataset access point using a dataset ID.
- **FI_Preferences** (**Root** root, const char *vendor, const char *application)
The constructor creates a group that manages name/value pairs and child groups.
- void **flush** ()
Writes all preferences to disk.
- char **get** (const char *entry, char *&value, const char *defaultValue)
Reads an entry from the group.
- char **get** (const char *entry, char *value, const char *defaultValue, int maxSize)
Reads an entry from the group.
- char **get** (const char *entry, double &value, double defaultValue)
Reads an entry from the group.
- char **get** (const char *entry, float &value, float defaultValue)
Reads an entry from the group.
- char **get** (const char *entry, int &value, int defaultValue)
Reads an entry from the group.
- char **get** (const char *entry, void *&value, const void *defaultValue, int defaultSize)
Reads an entry from the group.
- char **get** (const char *entry, void *value, const void *defaultValue, int defaultSize, int maxSize)
Reads an entry from the group.
- char **getUserdataPath** (char *path, int pathlen)
Creates a path that is related to the preferences file and that is usable for additional application data.

- `const char * group (int num_group)`
Returns the name of the Nth (num_group) group.
- `char groupExists (const char *key)`
Returns non-zero if a group with this name exists.
- `int groups ()`
Returns the number of groups that are contained within a group.
- `ID id ()`
Return an ID that can later be reused to open more references to this dataset.
- `const char * name ()`
Return the name of this entry.
- `const char * path ()`
Return the full path to this entry.
- `char set (const char *entry, const char *value)`
Sets an entry (name/value pair).
- `char set (const char *entry, const void *value, int size)`
Sets an entry (name/value pair).
- `char set (const char *entry, double value)`
Sets an entry (name/value pair).
- `char set (const char *entry, double value, int precision)`
Sets an entry (name/value pair).
- `char set (const char *entry, float value)`
Sets an entry (name/value pair).
- `char set (const char *entry, float value, int precision)`
Sets an entry (name/value pair).
- `char set (const char *entry, int value)`
Sets an entry (name/value pair).
- `int size (const char *entry)`
Returns the size of the value part of an entry.
- `virtual ~FI_Preferences ()`
The destructor removes allocated resources.

Static Public Member Functions

- `static const char * newUUID ()`
Returns a UUID as generated by the system.
- `static char remove (ID id_)`
Remove the group with this ID from a database.

Protected Attributes

- `Node * node`
- `RootNode * rootNode`

Friends

- class `Node`
- class `RootNode`

31.103.1 Detailed Description

[Fl_Preferences](#) provides methods to store user settings between application starts.

It is similar to the Registry on WIN32 and Preferences on MacOS, and provides a simple configuration mechanism for UNIX.

[Fl_Preferences](#) uses a hierarchy to store data. It bundles similar data into groups and manages entries into those groups as name/value pairs.

Preferences are stored in text files that can be edited manually. The file format is easy to read and relatively forgiving. Preferences files are the same on all platforms. User comments in preference files are preserved. Filenames are unique for each application by using a vendor/application naming scheme. The user must provide default values for all entries to ensure proper operation should preferences be corrupted or not yet exist.

Entries can be of any length. However, the size of each preferences file should be kept small for performance reasons. One application can have multiple preferences files. Extensive binary data however should be stored in separate files: see [getUserdataPath\(\)](#).

Note

Starting with FLTK 1.3, preference databases are expected to be in UTF-8 encoding. Previous databases were stored in the current character set or code page which renders them incompatible for text entries using international characters.

31.103.2 Member Typedef Documentation

31.103.2.1 ID

```
typedef void* Fl_Preferences::ID
```

Every Fl_Preferences-Group has a unique ID.

ID's can be retrieved from an Fl_Preferences-Group and can then be used to create more Fl_Preference references to the same data set, as long as the database remains open.

31.103.3 Member Enumeration Documentation

31.103.3.1 Root

```
enum Fl_Preferences::Root
```

Define the scope of the preferences.

Enumerator

SYSTEM	Preferences are used system-wide.
USER	Preferences apply only to the current user.

31.103.4 Constructor & Destructor Documentation

31.103.4.1 Fl_Preferences() [1/7]

```
Fl_Preferences::Fl_Preferences (
    Root root,
    const char * vendor,
    const char * application )
```

The constructor creates a group that manages name/value pairs and child groups.

Groups are ready for reading and writing at any time. The root argument is either [Fl_Preferences::USER](#) or [Fl_Preferences::SYSTEM](#).

This constructor creates the *base* instance for all following entries and reads existing databases into memory. The *vendor* argument is a unique text string identifying the development team or vendor of an application. A domain name or an EMail address are great unique names, e.g. "researchATmatthiasm.com" or "fltk.org". The application argument can be the working title or final name of your application. Both *vendor* and *application* must be valid relative UNIX pathnames and may contain '/'s to create deeper file structures.

A set of Preferences marked "run-time" exists exactly one per application and only as long as the application runs. It can be used as a database for volatile information. FLTK uses it to register plugins at run-time.

Parameters

in	<i>root</i>	can be USER or SYSTEM for user specific or system wide preferences
in	<i>vendor</i>	unique text describing the company or author of this file
in	<i>application</i>	unique text describing the application

31.103.4.2 Fl_Preferences() [2/7]

```
Fl_Preferences::Fl_Preferences (
    const char * path,
    const char * vendor,
    const char * application )
```

Use this constructor to create or read a preferences file at an arbitrary position in the file system.

The file name is generated in the form *path/application.prefs*. If *application* is NULL, *path* must contain the full file name.

Parameters

in	<i>path</i>	path to the directory that contains the preferences file
in	<i>vendor</i>	unique text describing the company or author of this file
in	<i>application</i>	unique text describing the application

31.103.4.3 Fl_Preferences() [3/7]

```
Fl_Preferences::Fl_Preferences (
    Fl_Preferences & parent,
    const char * group )
```

Generate or read a new group of entries within another group.

Use the *group* argument to name the group that you would like to access. *Group* can also contain a path to a group further down the hierarchy by separating group names with a forward slash '/'.

Parameters

in	<i>parent</i>	reference object for the new group
in	<i>group</i>	name of the group to access (may contain '/'s)

31.103.4.4 Fl_Preferences() [4/7]

```
Fl_Preferences::Fl_Preferences (
    Fl_Preferences * parent,
    const char * group )
```

Create or access a group of preferences using a name.

Parameters

in	<i>parent</i>	the parameter <i>parent</i> is a pointer to the parent group. <i>Parent</i> may be NULL. It then refers to an application internal database which exists only once, and remains in RAM only until the application quits. This database is used to manage plugins and other data indexes by strings.
in	<i>group</i>	a group name that is used as a key into the database

See also

[FI_Preferences\(FI_Preferences&, const char *group \)](#)

31.103.4.5 FI_Preferences() [5/7]

```
Fl_Preferences::Fl_Preferences (
    Fl_Preferences & parent,
    int groupIndex )
```

Open a child group using a given index.

Use the *groupIndex* argument to find the group that you would like to access. If the given index is invalid (negative or too high), a new group is created with a UUID as a name.

The index needs to be fixed. It is currently backward. Index 0 points to the last member in the 'list' of preferences.

Parameters

in	<i>parent</i>	reference object for the new group
in	<i>groupIndex</i>	zero based index into child groups

31.103.4.6 FI_Preferences() [6/7]

```
Fl_Preferences::Fl_Preferences (
    Fl_Preferences * parent,
    int groupIndex )
```

See also

[FI_Preferences\(FI_Preferences&, int groupIndex \)](#)

31.103.4.7 FI_Preferences() [7/7]

```
Fl_Preferences::Fl_Preferences (
    Fl_Preferences::ID id )
```

Create a new dataset access point using a dataset ID.

ID's are a great way to remember shortcuts to database entries that are deeply nested in a preferences database, as long as the database root is not deleted. An ID can be retrieved from any [FI_Preferences](#) dataset, and can then be used to create multiple new references to the same dataset.

ID's can be very helpful when put into the *user_data()* field of widget callbacks.

31.103.4.8 ~FI_Preferences()

```
Fl_Preferences::~Fl_Preferences ( ) [virtual]
```

The destructor removes allocated resources.

When used on the *base* preferences group, the destructor flushes all changes to the preferences file and deletes all internal databases.

The destructor does not remove any data from the database. It merely deletes your reference to the database.

31.103.5 Member Function Documentation

31.103.5.1 deleteEntry()

```
char Fl_Preferences::deleteEntry (
    const char * key )
```

Deletes a single name/value pair.

This function removes the entry `key` from the database.

Parameters

in	<i>key</i>	name of entry to delete
----	------------	-------------------------

Returns

0 if deleting the entry failed

31.103.5.2 deleteGroup()

```
char Fl_Preferences::deleteGroup (
    const char * group )
```

Deletes a group.

Removes a group and all keys and groups within that group from the database.

Parameters

in	<i>group</i>	name of the group to delete
----	--------------	-----------------------------

Returns

0 if call failed

31.103.5.3 entries()

```
int Fl_Preferences::entries ( )
```

Returns the number of entries (name/value pairs) in a group.

Returns

number of entries

31.103.5.4 entry()

```
const char * Fl_Preferences::entry (
    int index )
```

Returns the name of an entry.

There is no guaranteed order of entry names. The index must be within the range given by [entries\(\)](#).

Parameters

in	<i>index</i>	number indexing the requested entry
----	--------------	-------------------------------------

Returns

pointer to value cstring

31.103.5.5 entryExists()

```
char Fl_Preferences::entryExists (
    const char * key )
```

Returns non-zero if an entry with this name exists.

Parameters

in	key	name of entry that is searched for
----	-----	------------------------------------

Returns

0 if entry was not found

31.103.5.6 flush()

```
void Fl_Preferences::flush ( )
```

Writes all preferences to disk.

This function works only with the base preferences group. This function is rarely used as deleting the base preferences flushes automatically.

31.103.5.7 get() [1/7]

```
char Fl_Preferences::get (
    const char * key,
    char *& text,
    const char * defaultValue )
```

Reads an entry from the group.

A default value must be supplied. The return value indicates if the value was available (non-zero) or the default was used (0). [get\(\)](#) allocates memory of sufficient size to hold the value. The buffer must be free'd by the developer using 'free(value)'.

Parameters

in	key	name of entry
out	text	returned from preferences or default value if none was set
in	defaultValue	default value to be used if no preference was set

Returns

0 if the default value was used

31.103.5.8 get() [2/7]

```
char Fl_Preferences::get (
    const char * key,
    char * text,
    const char * defaultValue,
    int maxSize )
```

Reads an entry from the group.

A default value must be supplied. The return value indicates if the value was available (non-zero) or the default was used (0). 'maxSize' is the maximum length of text that will be read. The text buffer must allow for one additional byte for a trailing zero.

Parameters

in	<i>key</i>	name of entry
out	<i>text</i>	returned from preferences or default value if none was set
in	<i>defaultValue</i>	default value to be used if no preference was set
in	<i>maxSize</i>	maximum length of value plus one byte for a trailing zero

Returns

0 if the default value was used

31.103.5.9 `get()` [3/7]

```
char Fl_Preferences::get (
    const char * key,
    double & value,
    double defaultValue )
```

Reads an entry from the group.

A default value must be supplied. The return value indicates if the value was available (non-zero) or the default was used (0).

Parameters

in	<i>key</i>	name of entry
out	<i>value</i>	returned from preferences or default value if none was set
in	<i>defaultValue</i>	default value to be used if no preference was set

Returns

0 if the default value was used

31.103.5.10 `get()` [4/7]

```
char Fl_Preferences::get (
    const char * key,
    float & value,
    float defaultValue )
```

Reads an entry from the group.

A default value must be supplied. The return value indicates if the value was available (non-zero) or the default was used (0).

Parameters

in	<i>key</i>	name of entry
out	<i>value</i>	returned from preferences or default value if none was set
in	<i>defaultValue</i>	default value to be used if no preference was set

Returns

0 if the default value was used

31.103.5.11 get() [5/7]

```
char Fl_Preferences::get (
    const char * key,
    int & value,
    int defaultValue )
```

Reads an entry from the group.

A default value must be supplied. The return value indicates if the value was available (non-zero) or the default was used (0).

Parameters

in	<i>key</i>	name of entry
out	<i>value</i>	returned from preferences or default value if none was set
in	<i>defaultValue</i>	default value to be used if no preference was set

Returns

0 if the default value was used

31.103.5.12 get() [6/7]

```
char Fl_Preferences::get (
    const char * key,
    void *& data,
    const void * defaultValue,
    int defaultSize )
```

Reads an entry from the group.

A default value must be supplied. The return value indicates if the value was available (non-zero) or the default was used (0). [get\(\)](#) allocates memory of sufficient size to hold the value. The buffer must be free'd by the developer using 'free(value)'.

Parameters

in	<i>key</i>	name of entry
out	<i>data</i>	returned from preferences or default value if none was set
in	<i>defaultValue</i>	default value to be used if no preference was set
in	<i>defaultSize</i>	size of default value array

Returns

0 if the default value was used

31.103.5.13 get() [7/7]

```
char Fl_Preferences::get (
    const char * key,
    void * data,
    const void * defaultValue,
```

```

    int defaultSize,
    int maxSize )

```

Reads an entry from the group.

A default value must be supplied. The return value indicates if the value was available (non-zero) or the default was used (0). 'maxSize' is the maximum length of text that will be read.

Parameters

in	<i>key</i>	name of entry
out	<i>data</i>	value returned from preferences or default value if none was set
in	<i>defaultValue</i>	default value to be used if no preference was set
in	<i>defaultSize</i>	size of default value array
in	<i>maxSize</i>	maximum length of value

Returns

0 if the default value was used

Todo maxSize should receive the number of bytes that were read.

31.103.5.14 getUserdataPath()

```

char Fl_Preferences::getUserdataPath (
    char * path,
    int pathlen )

```

Creates a path that is related to the preferences file and that is usable for additional application data.

This function creates a directory that is named after the preferences database without the .prefs extension and located in the same directory. It then fills the given buffer with the complete path name.

Example:

```

Fl_Preferences prefs( USER, "matthiasm.com", "test" );
char path[FL_PATH_MAX];
prefs.getUserdataPath( path );

```

..creates the preferences database in (MS Windows):
c:/Documents and Settings/matt/Application Data/matthiasm.com/test.prefs

..and returns the userdata path:
c:/Documents and Settings/matt/Application Data/matthiasm.com/test/

Parameters

out	<i>path</i>	buffer for user data path
in	<i>pathlen</i>	size of path buffer (should be at least FL_PATH_MAX)

Returns

0 if path was not created or pathname can't fit into buffer

31.103.5.15 group()

```

const char * Fl_Preferences::group (
    int num_group )

```

Returns the name of the Nth (num_group) group.

There is no guaranteed order of group names. The index must be within the range given by [groups\(\)](#).

Parameters

in	<i>num_group</i>	number indexing the requested group
----	------------------	-------------------------------------

Returns

'C' string pointer to the group name

31.103.5.16 groupExists()

```
char Fl_Preferences::groupExists (
    const char * key )
```

Returns non-zero if a group with this name exists.

Group names are relative to the Preferences node and can contain a path. "." describes the current node, "/" describes the topmost node. By preceding a groupname with a "./", its path becomes relative to the topmost node.

Parameters

in	<i>key</i>	name of group that is searched for
----	------------	------------------------------------

Returns

0 if no group by that name was found

31.103.5.17 groups()

```
int Fl_Preferences::groups ( )
```

Returns the number of groups that are contained within a group.

Returns

0 for no groups at all

31.103.5.18 newUUID()

```
const char * Fl_Preferences::newUUID ( ) [static]
```

Returns a UUID as generated by the system.

A UUID is a "universally unique identifier" which is commonly used in configuration files to create identities. A UUID in ASCII looks like this: 937C4900-51AA-4C11-8DD3-7AB59944F03E. It has always 36 bytes plus a trailing zero.

Returns

a pointer to a static buffer containing the new UUID in ASCII format. The buffer is overwritten during every call to this function!

31.103.5.19 set() [1/7]

```
char Fl_Preferences::set (
    const char * key,
    const char * text )
```

Sets an entry (name/value pair).

The return value indicates if there was a problem storing the data in memory. However it does not reflect if the value was actually stored in the preferences file.

Parameters

in	<i>key</i>	name of entry
in	<i>text</i>	set this entry to value

Returns

0 if setting the value failed

31.103.5.20 set() [2/7]

```
char Fl_Preferences::set (
    const char * key,
    const void * data,
    int dsize )
```

Sets an entry (name/value pair).

The return value indicates if there was a problem storing the data in memory. However it does not reflect if the value was actually stored in the preferences file.

Parameters

in	<i>key</i>	name of entry
in	<i>data</i>	set this entry to value
in	<i>dsize</i>	size of data array

Returns

0 if setting the value failed

31.103.5.21 set() [3/7]

```
char Fl_Preferences::set (
    const char * key,
    double value )
```

Sets an entry (name/value pair).

The return value indicates if there was a problem storing the data in memory. However it does not reflect if the value was actually stored in the preferences file.

Parameters

in	<i>key</i>	name of entry
in	<i>value</i>	set this entry to value

Returns

0 if setting the value failed

31.103.5.22 set() [4/7]

```
char Fl_Preferences::set (
    const char * key,
    double value,
    int precision )
```

Sets an entry (name/value pair).

The return value indicates if there was a problem storing the data in memory. However it does not reflect if the value was actually stored in the preferences file.

Parameters

in	<i>key</i>	name of entry
in	<i>value</i>	set this entry to <i>value</i>
in	<i>precision</i>	number of decimal digits to represent value

Returns

0 if setting the value failed

31.103.5.23 set() [5/7]

```
char Fl_Preferences::set (
    const char * key,
    float value )
```

Sets an entry (name/value pair).

The return value indicates if there was a problem storing the data in memory. However it does not reflect if the value was actually stored in the preferences file.

Parameters

in	<i>key</i>	name of entry
in	<i>value</i>	set this entry to <i>value</i>

Returns

0 if setting the value failed

31.103.5.24 set() [6/7]

```
char Fl_Preferences::set (
    const char * key,
    float value,
    int precision )
```

Sets an entry (name/value pair).

The return value indicates if there was a problem storing the data in memory. However it does not reflect if the value was actually stored in the preferences file.

Parameters

in	<i>key</i>	name of entry
in	<i>value</i>	set this entry to <i>value</i>
in	<i>precision</i>	number of decimal digits to represent value

Returns

0 if setting the value failed

31.103.5.25 set() [7/7]

```
char Fl_Preferences::set (
    const char * key,
```

```
int value )
```

Sets an entry (name/value pair).

The return value indicates if there was a problem storing the data in memory. However it does not reflect if the value was actually stored in the preferences file.

Parameters

in	<i>key</i>	name of entry
in	<i>value</i>	set this entry to <i>value</i>

Returns

0 if setting the value failed

31.103.5.26 size()

```
int Fl_Preferences::size (
    const char * key )
```

Returns the size of the value part of an entry.

Parameters

in	<i>key</i>	name of entry
----	------------	---------------

Returns

size of value

The documentation for this class was generated from the following files:

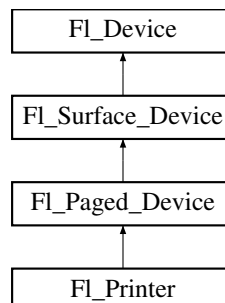
- Fl_Preferences.H
- Fl_Preferences.cxx

31.104 FI_Printer Class Reference

OS-independent print support.

```
#include <Fl_Printer.H>
```

Inheritance diagram for Fl_Printer:



Public Member Functions

- `const char * class_name ()`
Returns the name of the class of this object.
- `Fl_Graphics_Driver * driver (void)`

- void `end_job` (void)
To be called at the end of a print job.
- int `end_page` (void)
To be called at the end of each page.
- **FI_Printer** (void)
The constructor.
- void `margins` (int *left, int *top, int *right, int *bottom)
Computes the dimensions of margins that lie between the printable page area and the full page.
- void `origin` (int *x, int *y)
Computes the page coordinates of the current origin of graphics functions.
- void `origin` (int x, int y)
Sets the position in page coordinates of the origin of graphics functions.
- void `print_widget` (FI_Widget *widget, int delta_x=0, int delta_y=0)
Draws the widget on the printed page.
- void `print_window_part` (FI_Window *win, int x, int y, int w, int h, int delta_x=0, int delta_y=0)
Prints a rectangular part of an on-screen window.
- int `printable_rect` (int *w, int *h)
Computes the width and height of the printable area of the page.
- void `rotate` (float angle)
Rotates the graphics operations relatively to paper.
- void `scale` (float scale_x, float scale_y=0.)
Changes the scaling of page coordinates.
- void `set_current` (void)
Make this surface the current drawing surface.
- int `start_job` (int pagecount, int *frompage=NULL, int *topage=NULL)
Starts a print job.
- int `start_page` (void)
Starts a new printed page.
- void `translate` (int x, int y)
Translates the current graphics origin accounting for the current rotation.
- void `untranslate` (void)
Undoes the effect of a previous `translate()` call.
- `~FI_Printer` (void)
The destructor.

Static Public Attributes

- static const char * `class_id` = "FI_Printer"

These attributes are effective under the Xlib platform only.

- static const char * `dialog_title` = "Print"
[this text may be customized at run-time]
- static const char * `dialog_printer` = "Printer:"
[this text may be customized at run-time]
- static const char * `dialog_range` = "Print Range"
[this text may be customized at run-time]
- static const char * `dialog_copies` = "Copies"
[this text may be customized at run-time]
- static const char * `dialog_all` = "All"
[this text may be customized at run-time]
- static const char * `dialog_pages` = "Pages"
[this text may be customized at run-time]
- static const char * `dialog_from` = "From:"

- *[this text may be customized at run-time]*
static const char * **dialog_to** = "To:"
- *[this text may be customized at run-time]*
static const char * **dialog_properties** = "Properties..."
- *[this text may be customized at run-time]*
static const char * **dialog_copyNo** = "# Copies:"
- *[this text may be customized at run-time]*
static const char * **dialog_print_button** = "Print"
- *[this text may be customized at run-time]*
static const char * **dialog_cancel_button** = "Cancel"
- *[this text may be customized at run-time]*
static const char * **dialog_print_to_file** = "Print To File"
- *[this text may be customized at run-time]*
static const char * **property_title** = "Printer Properties"
- *[this text may be customized at run-time]*
static const char * **property_pagesize** = "Page Size:"
- *[this text may be customized at run-time]*
static const char * **property_mode** = "Output Mode:"
- *[this text may be customized at run-time]*
static const char * **property_use** = "Use"
- *[this text may be customized at run-time]*
static const char * **property_save** = "Save"
- *[this text may be customized at run-time]*
static const char * **property_cancel** = "Cancel"

Additional Inherited Members

31.104.1 Detailed Description

OS-independent print support.

[FL_Printer](#) allows to use all drawing, color, text, image, and clip FLTK functions, and to have them operate on printed page(s). There are two main, non exclusive, ways to use it.

- Print any widget (standard, custom, [FL_Window](#), [FL_Gl_Window](#)) as it appears on screen, with optional translation, scaling and rotation. This is done by calling [print_widget\(\)](#), [print_window\(\)](#) or [print_window_part\(\)](#).
- Use a series of FLTK graphics commands (e.g., font, text, lines, colors, clip, image) to compose a page appropriately shaped for printing.

In both cases, begin by [start_job\(\)](#), [start_page\(\)](#), [printable_rect\(\)](#) and [origin\(\)](#) calls and finish by [end_page\(\)](#) and [end_job\(\)](#) calls.

Example of use: print a widget centered in a page

```
#include <FL/Fl_Printer.H>
#include <FL/fl_draw.H>
int width, height;
Fl_Widget *widget = ... // a widget we want printed
Fl_Printer *printer = new Fl_Printer();
if (printer->start_job(1) == 0) {
    printer->start_page();
    printer->printable_rect(&width, &height);
    fl_color(FL_BLACK);
    fl_line_style(FL_SOLID, 2);
    fl_rect(0, 0, width, height);
    fl_font(FL_COURIER, 12);
    time_t now; time(&now); fl_draw(ctime(&now), 0, fl_height());
    printer->origin(width/2, height/2);
    printer->print_widget(widget, -widget->w()/2, -widget->h()/2);
    printer->end_page();
    printer->end_job();
}
delete printer;
```

Platform specifics

- Unix/Linux platforms: Unless it has been previously changed, the default paper size is A4. To change that, press the "Properties" button of the "Print" dialog window opened by an [FL_Printer::start_job\(\)](#) call. This opens a "Printer Properties" window where it's possible to select the adequate paper size. Finally press the "Save"

button therein to assign the chosen paper size to the chosen printer for this and all further print operations. Class [FI_RGB_Image](#) prints but loses its transparency if it has one. See class [FI_PostScript_Graphics_Driver](#) for a description of how UTF-8 strings appear in print. Use the static public attributes of this class to set the print dialog to other languages than English. For example, the "Printer:" dialog item [FI_Printer::dialog_printer](#) can be set to French with:

```
FI_Printer::dialog_printer = "Imprimante:";
```

before creation of the [FI_Printer](#) object. Use [FI_PostScript_File_Device::file_chooser_title](#) to customize the title of the file chooser dialog that opens when using the "Print To File" option of the print dialog.

- MSWindows platform: Transparent [FI_RGB_Image](#) 's don't print with exact transparency on most printers. [FI_RGB_Image](#) 's don't [rotate\(\)](#) well. A workaround is to use the [print_window_part\(\)](#) call.
- Mac OS X platform: all graphics requests print as on display.

31.104.2 Member Function Documentation

31.104.2.1 class_name()

```
const char * FI_Printer::class_name ( ) [inline], [virtual]
```

Returns the name of the class of this object.

Use of the [class_name\(\)](#) function is discouraged because it will be removed from future FLTK versions.

The class of an instance of an [FI_Device](#) subclass can be checked with code such as:

```
if ( instance->class_name() == FI_Printer::class_id ) { ... }
```

Reimplemented from [FI_Paged_Device](#).

31.104.2.2 end_job()

```
void FI_Printer::end_job (
    void ) [virtual]
```

To be called at the end of a print job.

Reimplemented from [FI_Paged_Device](#).

31.104.2.3 end_page()

```
int FI_Printer::end_page (
    void ) [virtual]
```

To be called at the end of each page.

Returns

0 if OK, non-zero if any error.

Reimplemented from [FI_Paged_Device](#).

31.104.2.4 margins()

```
void FI_Printer::margins (
    int * left,
    int * top,
    int * right,
    int * bottom ) [virtual]
```

Computes the dimensions of margins that lie between the printable page area and the full page.

Values are in the same unit as that used by FLTK drawing functions. They are changed by [scale\(\)](#) calls.

Parameters

out	<i>left</i>	If non-null, *left is set to the left margin size.
-----	-------------	--

Parameters

out	<i>top</i>	If non-null, *top is set to the top margin size.
out	<i>right</i>	If non-null, *right is set to the right margin size.
out	<i>bottom</i>	If non-null, *bottom is set to the bottom margin size.

Reimplemented from [Fl_Paged_Device](#).

31.104.2.5 origin() [1/2]

```
void Fl_Printer::origin (
    int * x,
    int * y ) [virtual]
```

Computes the page coordinates of the current origin of graphics functions.

Parameters

out	<i>x</i>	If non-null, *x is set to the horizontal page offset of graphics origin.
out	<i>y</i>	Same as above, vertically.

Reimplemented from [Fl_Paged_Device](#).

31.104.2.6 origin() [2/2]

```
void Fl_Printer::origin (
    int x,
    int y ) [virtual]
```

Sets the position in page coordinates of the origin of graphics functions.

Arguments should be expressed relatively to the result of a previous [printable_rect\(\)](#) call. That is, `printable_rect(&w, &h); origin(w/2, 0);` sets the graphics origin at the top center of the page printable area. Origin() calls are not affected by [rotate\(\)](#) calls. Successive [origin\(\)](#) calls don't combine their effects.

Parameters

in	<i>x</i>	Horizontal position in page coordinates of the desired origin of graphics functions.
in	<i>y</i>	Same as above, vertically.

Reimplemented from [Fl_Paged_Device](#).

31.104.2.7 print_widget()

```
void Fl_Printer::print_widget (
    Fl_Widget * widget,
    int delta_x = 0,
    int delta_y = 0 ) [virtual]
```

Draws the widget on the printed page.

The widget's position on the printed page is determined by the last call to [origin\(\)](#) and by the optional `delta_x` and `delta_y` arguments. Its dimensions are in points unless there was a previous call to [scale\(\)](#).

Parameters

in	<i>widget</i>	Any FLTK widget (e.g., standard, custom, window).
in	<i>delta_x</i>	Optional horizontal offset for positioning the widget relatively to the current origin of graphics functions.

Parameters

in	<i>delta</i> _↔ <i>_y</i>	Same as above, vertically.
----	--	----------------------------

Reimplemented from [Fl_Paged_Device](#).

31.104.2.8 print_window_part()

```
void Fl_Printer::print_window_part (
    Fl_Window * win,
    int x,
    int y,
    int w,
    int h,
    int delta_x = 0,
    int delta_y = 0 ) [virtual]
```

Prints a rectangular part of an on-screen window.

Parameters

<i>win</i>	The window from where to capture.
<i>x</i>	The rectangle left
<i>y</i>	The rectangle top
<i>w</i>	The rectangle width
<i>h</i>	The rectangle height
<i>delta</i> _↔ <i>_x</i>	Optional horizontal offset from current graphics origin where to print the captured rectangle.
<i>delta</i> _↔ <i>_y</i>	As above, vertically.

Reimplemented from [Fl_Paged_Device](#).

31.104.2.9 printable_rect()

```
int Fl_Printer::printable_rect (
    int * w,
    int * h ) [virtual]
```

Computes the width and height of the printable area of the page.

Values are in the same unit as that used by FLTK drawing functions, are unchanged by calls to [origin\(\)](#), but are changed by [scale\(\)](#) calls. Values account for the user-selected paper type and print orientation.

Returns

0 if OK, non-zero if any error

Reimplemented from [Fl_Paged_Device](#).

31.104.2.10 rotate()

```
void Fl_Printer::rotate (
    float angle ) [virtual]
```

Rotates the graphics operations relatively to paper.

The rotation is centered on the current graphics origin. Successive [rotate\(\)](#) calls don't combine their effects.

Parameters

<i>angle</i>	Rotation angle in counter-clockwise degrees.
--------------	--

Reimplemented from [Fl_Paged_Device](#).

31.104.2.11 scale()

```
void Fl_Printer::scale (
    float scale_x,
    float scale_y = 0. ) [virtual]
```

Changes the scaling of page coordinates.

This function also resets the origin of graphics functions at top left of printable page area. After a [scale\(\)](#) call, do a [printable_rect\(\)](#) call to get the new dimensions of the printable page area. Successive [scale\(\)](#) calls don't combine their effects.

Parameters

<i>scale</i> <i>_x</i>	Horizontal dimensions of plot are multiplied by this quantity.
<i>scale</i> <i>_y</i>	Same as above, vertically. The value 0. is equivalent to setting <code>scale_y = scale_x</code> . Thus, <code>scale(factor)</code> ; is equivalent to <code>scale(factor, factor)</code> ;

Reimplemented from [Fl_Paged_Device](#).

31.104.2.12 set_current()

```
void Fl_Printer::set_current (
    void ) [virtual]
```

Make this surface the current drawing surface.

This surface will receive all future graphics requests.

Reimplemented from [Fl_Surface_Device](#).

31.104.2.13 start_job()

```
int Fl_Printer::start_job (
    int pagecount,
    int * frompage = NULL,
    int * topage = NULL ) [virtual]
```

Starts a print job.

Opens a platform-specific dialog window allowing the user to set several options including the desired printer and the page orientation. Optionally, the user can also select a range of pages to be printed. This range is returned to the caller that is in charge of sending only these pages for printing.

Parameters

in	<i>pagecount</i>	the total number of pages of the job (or 0 if you don't know the number of pages)
out	<i>frompage</i>	if non-null, *frompage is set to the first page the user wants printed
out	<i>topage</i>	if non-null, *topage is set to the last page the user wants printed

Returns

0 if OK, non-zero if any error occurred or if the user cancelled the print request.

Reimplemented from [Fl_Paged_Device](#).

31.104.2.14 start_page()

```
int Fl_Printer::start_page (
    void ) [virtual]
```

Starts a new printed page.

The page coordinates are initially in points, i.e., 1/72 inch, and with origin at the top left of the printable page area.

Returns

0 if OK, non-zero if any error

Reimplemented from [Fl_Paged_Device](#).

31.104.2.15 translate()

```
void Fl_Printer::translate (
    int x,
    int y ) [virtual]
```

Translates the current graphics origin accounting for the current rotation.

This function is only useful after a [rotate\(\)](#) call. Each [translate\(\)](#) call must be matched by an [untranslate\(\)](#) call.

Successive [translate\(\)](#) calls add up their effects.

Reimplemented from [Fl_Paged_Device](#).

31.104.2.16 untranslate()

```
void Fl_Printer::untranslate (
    void ) [virtual]
```

Undoes the effect of a previous [translate\(\)](#) call.

Reimplemented from [Fl_Paged_Device](#).

The documentation for this class was generated from the following files:

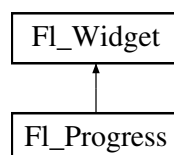
- [Fl_Printer.H](#)
- [Fl_Printer.cxx](#)

31.105 Fl_Progress Class Reference

Displays a progress bar for the user.

```
#include <Fl_Progress.H>
```

Inheritance diagram for Fl_Progress:

**Public Member Functions**

- [Fl_Progress](#) (int x, int y, int w, int h, const char *l=0)
The constructor creates the progress bar using the position, size, and label.
- float [maximum](#) () const
Gets the maximum value in the progress widget.
- void [maximum](#) (float v)
Sets the maximum value in the progress widget.

- float [minimum](#) () const
Gets the minimum value in the progress widget.
- void [minimum](#) (float v)
Sets the minimum value in the progress widget.
- float [value](#) () const
Gets the current value in the progress widget.
- void [value](#) (float v)
Sets the current value in the progress widget.

Protected Member Functions

- virtual void [draw](#) ()
Draws the progress bar.

Additional Inherited Members

31.105.1 Detailed Description

Displays a progress bar for the user.

31.105.2 Constructor & Destructor Documentation

31.105.2.1 Fl_Progress()

```
Fl_Progress::Fl_Progress (
    int X,
    int Y,
    int W,
    int H,
    const char * L = 0 )
```

The constructor creates the progress bar using the position, size, and label.

You can set the background color with [color\(\)](#) and the progress bar color with [selection_color\(\)](#), or you can set both colors together with `color(unsigned bg, unsigned sel)`.

The default colors are FL_BACKGROUND2_COLOR and FL_YELLOW, resp.

31.105.3 Member Function Documentation

31.105.3.1 draw()

```
void Fl_Progress::draw (
    void ) [protected], [virtual]
```

Draws the progress bar.

Implements [Fl_Widget](#).

31.105.3.2 maximum() [1/2]

```
float Fl_Progress::maximum ( ) const [inline]
```

Gets the maximum value in the progress widget.

31.105.3.3 maximum() [2/2]

```
void Fl_Progress::maximum (
    float v ) [inline]
```

Sets the maximum value in the progress widget.

31.105.3.4 minimum() [1/2]

```
float Fl_Progress::minimum ( ) const [inline]
```

Gets the minimum value in the progress widget.

31.105.3.5 minimum() [2/2]

```
void Fl_Progress::minimum (
    float v ) [inline]
```

Sets the minimum value in the progress widget.

31.105.3.6 value() [1/2]

```
float Fl_Progress::value ( ) const [inline]
```

Gets the current value in the progress widget.

31.105.3.7 value() [2/2]

```
void Fl_Progress::value (
    float v ) [inline]
```

Sets the current value in the progress widget.

The documentation for this class was generated from the following files:

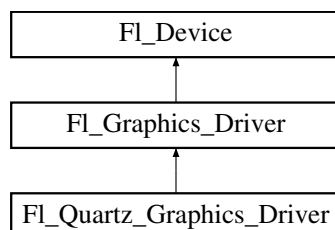
- Fl_Progress.H
- Fl_Progress.cxx

31.106 Fl_Quartz_Graphics_Driver Class Reference

The Mac OS X-specific graphics class.

```
#include <Fl_Device.H>
```

Inheritance diagram for Fl_Quartz_Graphics_Driver:



Public Member Functions

- const char * [class_name](#) ()
Returns the name of the class of this object.
- void [color](#) ([FI_Color](#) c)
see [fl_color\(FI_Color c\)](#).
- void [color](#) (uchar r, uchar g, uchar b)
see [fl_color\(uchar r, uchar g, uchar b\)](#).
- void [copy_offscreen](#) (int x, int y, int w, int h, [FI_Offscreen](#) pixmap, int srcx, int srcy)
see [fl_copy_offscreen\(\)](#)
- int [descent](#) ()
see [fl_descent\(\)](#).
- void [draw](#) (const char *str, int n, int x, int y)
*see [fl_draw\(const char *str, int n, int x, int y\)](#).*
- void [draw](#) ([FI_Bitmap](#) *pixmap, int XP, int YP, int WP, int HP, int cx, int cy)
Draws an [FI_Bitmap](#) object to the device.
- void [draw](#) ([FI_Pixmap](#) *pixmap, int XP, int YP, int WP, int HP, int cx, int cy)
Draws an [FI_Pixmap](#) object to the device.
- void [draw](#) ([FI_RGB_Image](#) *img, int XP, int YP, int WP, int HP, int cx, int cy)
Draws an [FI_RGB_Image](#) object to the device.
- void [draw](#) (int angle, const char *str, int n, int x, int y)
*see [fl_draw\(int angle, const char *str, int n, int x, int y\)](#).*
- void [draw_image](#) (const uchar *buf, int X, int Y, int W, int H, int D=3, int L=0)
see [fl_draw_image\(const uchar buf, int X,int Y,int W,int H, int D, int L\)](#).*
- void [draw_image](#) ([FI_Draw_Image_Cb](#) cb, void *data, int X, int Y, int W, int H, int D=3)
see [fl_draw_image\(FI_Draw_Image_Cb cb, void data, int X,int Y,int W,int H, int D\)](#).*
- void [draw_image_mono](#) (const uchar *buf, int X, int Y, int W, int H, int D=1, int L=0)
see [fl_draw_image_mono\(const uchar buf, int X,int Y,int W,int H, int D, int L\)](#).*
- void [draw_image_mono](#) ([FI_Draw_Image_Cb](#) cb, void *data, int X, int Y, int W, int H, int D=1)
see [fl_draw_image_mono\(FI_Draw_Image_Cb cb, void data, int X,int Y,int W,int H, int D\)](#).*
- int [draw_scaled](#) ([FI_Image](#) *img, int XP, int YP, int WP, int HP)
*Draws an [FI_Image](#) scaled to width *W* & height *H* with top-left corner at *X,Y*.*
- void [font](#) ([FI_Font](#) face, [FI_Fontsize](#) size)
see [fl_font\(FI_Font face, FI_Fontsize size\)](#).
- int [height](#) ()
see [fl_height\(\)](#).
- void [rtl_draw](#) (const char *str, int n, int x, int y)
*see [fl_rtl_draw\(const char *str, int n, int x, int y\)](#).*
- void [text_extents](#) (const char *, int n, int &dx, int &dy, int &w, int &h)
see [fl_text_extents\(const char, int n, int& dx, int& dy, int& w, int& h\)](#).*
- double [width](#) (const char *str, int n)
*see [fl_width\(const char *str, int n\)](#).*
- double [width](#) (unsigned int c)
see [fl_width\(unsigned int n\)](#).

Static Public Attributes

- static const char * [class_id](#) = "FI_Quartz_Graphics_Driver"

Additional Inherited Members

31.106.1 Detailed Description

The Mac OS X-specific graphics class.

This class is implemented only on the Mac OS X platform.

31.106.2 Member Function Documentation

31.106.2.1 class_name()

```
const char * Fl_Quartz_Graphics_Driver::class_name ( ) [inline], [virtual]
```

Returns the name of the class of this object.

Use of the [class_name\(\)](#) function is discouraged because it will be removed from future FLTK versions.

The class of an instance of an [Fl_Device](#) subclass can be checked with code such as:

```
if ( instance->class_name() == Fl_Printer::class_id ) { ... }
```

Reimplemented from [Fl_Graphics_Driver](#).

31.106.2.2 color() [1/2]

```
void Fl_Quartz_Graphics_Driver::color (
    Fl_Color c ) [virtual]
```

see [fl_color\(Fl_Color c\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.106.2.3 color() [2/2]

```
void Fl_Quartz_Graphics_Driver::color (
    uchar r,
    uchar g,
    uchar b ) [virtual]
```

see [fl_color\(uchar r, uchar g, uchar b\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.106.2.4 descent()

```
int Fl_Quartz_Graphics_Driver::descent ( ) [virtual]
```

see [fl_descent\(\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.106.2.5 draw() [1/5]

```
void Fl_Quartz_Graphics_Driver::draw (
    const char * str,
    int n,
    int x,
    int y ) [virtual]
```

see [fl_draw\(const char *str, int n, int x, int y\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.106.2.6 draw() [2/5]

```
void Fl_Quartz_Graphics_Driver::draw (
    Fl_Bitmap * bm,
    int XP,
    int YP,
    int WP,
    int HP,
    int cx,
    int cy ) [virtual]
```

Draws an [Fl_Bitmap](#) object to the device.

Specifies a bounding box for the image, with the origin (upper left-hand corner) of the image offset by the *cx* and *cy* arguments.

Reimplemented from [Fl_Graphics_Driver](#).

31.106.2.7 draw() [3/5]

```
void Fl_Quartz_Graphics_Driver::draw (
    Fl_Pixmap * pxm,
    int XP,
    int YP,
    int WP,
    int HP,
    int cx,
    int cy ) [virtual]
```

Draws an [Fl_Pixmap](#) object to the device.

Specifies a bounding box for the image, with the origin (upper left-hand corner) of the image offset by the *cx* and *cy* arguments.

Reimplemented from [Fl_Graphics_Driver](#).

31.106.2.8 draw() [4/5]

```
void Fl_Quartz_Graphics_Driver::draw (
    Fl_RGB_Image * rgb,
    int XP,
    int YP,
    int WP,
    int HP,
    int cx,
    int cy ) [virtual]
```

Draws an [Fl_RGB_Image](#) object to the device.

Specifies a bounding box for the image, with the origin (upper left-hand corner) of the image offset by the *cx* and *cy* arguments.

Reimplemented from [Fl_Graphics_Driver](#).

31.106.2.9 draw() [5/5]

```
void Fl_Quartz_Graphics_Driver::draw (
    int angle,
    const char * str,
    int n,
    int x,
    int y ) [virtual]
```

see [fl_draw\(int angle, const char *str, int n, int x, int y\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.106.2.10 draw_image() [1/2]

```
void Fl_Quartz_Graphics_Driver::draw_image (
    const uchar * buf,
    int X,
    int Y,
    int W,
    int H,
    int D = 3,
    int L = 0 ) [virtual]
```

see [fl_draw_image\(const uchar* buf, int X,int Y,int W,int H, int D, int L\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.106.2.11 draw_image() [2/2]

```
void Fl_Quartz_Graphics_Driver::draw_image (
    Fl_Draw_Image_Cb cb,
    void * data,
    int X,
    int Y,
    int W,
    int H,
    int D = 3 ) [virtual]
```

see [fl_draw_image\(Fl_Draw_Image_Cb cb, void* data, int X,int Y,int W,int H, int D\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.106.2.12 draw_image_mono() [1/2]

```
void Fl_Quartz_Graphics_Driver::draw_image_mono (
    const uchar * buf,
    int X,
    int Y,
    int W,
    int H,
    int D = 1,
    int L = 0 ) [virtual]
```

see [fl_draw_image_mono\(const uchar* buf, int X,int Y,int W,int H, int D, int L\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.106.2.13 draw_image_mono() [2/2]

```
void Fl_Quartz_Graphics_Driver::draw_image_mono (
    Fl_Draw_Image_Cb cb,
    void * data,
    int X,
    int Y,
    int W,
    int H,
    int D = 1 ) [virtual]
```

see [fl_draw_image_mono\(Fl_Draw_Image_Cb cb, void* data, int X,int Y,int W,int H, int D\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.106.2.14 draw_scaled()

```
int Fl_Quartz_Graphics_Driver::draw_scaled (
    Fl_Image * img,
```

```

    int X,
    int Y,
    int W,
    int H ) [virtual]

```

Draws an [Fl_Image](#) scaled to width W & height H with top-left corner at X,Y.

Returns

zero when the graphics driver doesn't implement scaled drawing, non-zero if it does implement it.

Reimplemented from [Fl_Graphics_Driver](#).

31.106.2.15 font()

```

void Fl_Quartz_Graphics_Driver::font (
    Fl_Font face,
    Fl_Fontsize fsize ) [virtual]

```

see [fl_font\(Fl_Font face, Fl_Fontsize size\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.106.2.16 height()

```

int Fl_Quartz_Graphics_Driver::height ( ) [virtual]
see fl\_height\(\).

```

Reimplemented from [Fl_Graphics_Driver](#).

31.106.2.17 rtl_draw()

```

void Fl_Quartz_Graphics_Driver::rtl_draw (
    const char * str,
    int n,
    int x,
    int y ) [virtual]

```

see [fl_rtl_draw\(const char *str, int n, int x, int y\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.106.2.18 text_extents()

```

void Fl_Quartz_Graphics_Driver::text_extents (
    const char * t,
    int n,
    int & dx,
    int & dy,
    int & w,
    int & h ) [virtual]

```

see [fl_text_extents\(const char*, int n, int& dx, int& dy, int& w, int& h\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.106.2.19 width() [1/2]

```

double Fl_Quartz_Graphics_Driver::width (
    const char * str,
    int n ) [virtual]

```

see [fl_width\(const char *str, int n\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.106.2.20 width() [2/2]

```
double Fl_Quartz_Graphics_Driver::width (
    unsigned int c ) [virtual]
```

see [fl_width\(unsigned int n\)](#).

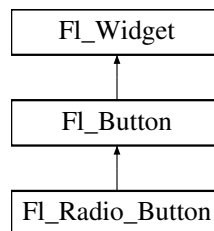
Reimplemented from [Fl_Graphics_Driver](#).

The documentation for this class was generated from the following files:

- [Fl_Device.H](#)
- [fl_color_mac.cxx](#)
- [Fl_Device.cxx](#)
- [Fl_Double_Window.cxx](#)
- [fl_draw_image_mac.cxx](#)

31.107 FI_Radio_Button Class Reference

Inheritance diagram for [Fl_Radio_Button](#):

**Public Member Functions**

- [Fl_Radio_Button](#) (int X, int Y, int W, int H, const char *L=0)

The constructor creates the button using the given position, size, and label.

Additional Inherited Members**31.107.1 Constructor & Destructor Documentation****31.107.1.1 Fl_Radio_Button()**

```
Fl_Radio_Button::Fl_Radio_Button (
    int X,
    int Y,
    int W,
    int H,
    const char * L = 0 )
```

The constructor creates the button using the given position, size, and label. The Button [type\(\)](#) is set to `FL_RADIO_BUTTON`.

Parameters

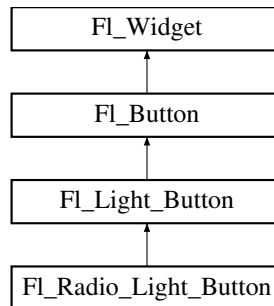
in	<i>X, Y, W, H</i>	position and size of the widget
in	<i>L</i>	widget label, default is no label

The documentation for this class was generated from the following files:

- [Fl_Radio_Button.H](#)
- [Fl_Button.cxx](#)

31.108 FI_Radio_Light_Button Class Reference

Inheritance diagram for FI_Radio_Light_Button:



Public Member Functions

- **FI_Radio_Light_Button** (int X, int Y, int W, int H, const char *l=0)

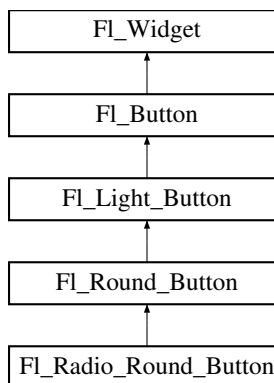
Additional Inherited Members

The documentation for this class was generated from the following files:

- FI_Radio_Light_Button.H
- FI_Light_Button.cxx

31.109 FI_Radio_Round_Button Class Reference

Inheritance diagram for FI_Radio_Round_Button:



Public Member Functions

- **FI_Radio_Round_Button** (int X, int Y, int W, int H, const char *L=0)
Creates a new *FI_Radio_Button* widget using the given position, size, and label string.

Additional Inherited Members

31.109.1 Constructor & Destructor Documentation

31.109.1.1 Fl_Radio_Round_Button()

```
Fl_Radio_Round_Button::Fl_Radio_Round_Button (
    int X,
    int Y,
    int W,
    int H,
    const char * L = 0 )
```

Creates a new [Fl_Radio_Button](#) widget using the given position, size, and label string. The button `type()` is set to `FL_RADIO_BUTTON`.

Parameters

in	<i>X,Y,W,H</i>	position and size of the widget
in	<i>L</i>	widget label, default is no label

The documentation for this class was generated from the following files:

- `Fl_Radio_Round_Button.H`
- `Fl_Round_Button.cxx`

31.110 Fl_Scroll::Fl_Region_LRTB Struct Reference

A local struct to manage a region defined by left/right/top/bottom.

```
#include <Fl_Scroll.H>
```

Public Attributes

- `int b`
(b)ottom "y" position, aka y2
- `int l`
(l)eft "x" position, aka x1
- `int r`
(r)ight "x" position, aka x2
- `int t`
(t)op "y" position, aka y1

31.110.1 Detailed Description

A local struct to manage a region defined by left/right/top/bottom.

The documentation for this struct was generated from the following file:

- `Fl_Scroll.H`

31.111 Fl_Scroll::Fl_Region_XYWH Struct Reference

A local struct to manage a region defined by xywh.

```
#include <Fl_Scroll.H>
```

Public Attributes

- `int h`
- `int w`
- `int x`
- `int y`

31.111.1 Detailed Description

A local struct to manage a region defined by xywh.

The documentation for this struct was generated from the following file:

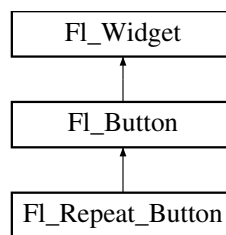
- `Fl_Scroll.H`

31.112 `Fl_Repeat_Button` Class Reference

The `Fl_Repeat_Button` is a subclass of `Fl_Button` that generates a callback when it is pressed and then repeatedly generates callbacks as long as it is held down.

```
#include <Fl_Repeat_Button.H>
```

Inheritance diagram for `Fl_Repeat_Button`:



Public Member Functions

- void `deactivate` ()
- `Fl_Repeat_Button` (int X, int Y, int W, int H, const char *l=0)
 - Creates a new `Fl_Repeat_Button` widget using the given position, size, and label string.*
- int `handle` (int)
 - Handles the specified event.*

Additional Inherited Members

31.112.1 Detailed Description

The `Fl_Repeat_Button` is a subclass of `Fl_Button` that generates a callback when it is pressed and then repeatedly generates callbacks as long as it is held down.

The speed of the repeat is fixed and depends on the implementation.

31.112.2 Constructor & Destructor Documentation

31.112.2.1 `Fl_Repeat_Button()`

```
Fl_Repeat_Button::Fl_Repeat_Button (
    int X,
    int Y,
    int W,
    int H,
    const char * l = 0 )
```

Creates a new `Fl_Repeat_Button` widget using the given position, size, and label string. The default boxtype is `FL_UP_BOX`. Deletes the button.

31.112.3 Member Function Documentation

31.112.3.1 handle()

```
int Fl_Repeat_Button::handle (
    int event ) [virtual]
```

Handles the specified event.

You normally don't call this method directly, but instead let FLTK do it when the user interacts with the widget. When implemented in a widget, this function must return 0 if the widget does not use the event or 1 otherwise. Most of the time, you want to call the inherited [handle\(\)](#) method in your overridden method so that you don't short-circuit events that you don't handle. In this last case you should return the callee retval.

Parameters

in	<i>event</i>	the kind of event received
----	--------------	----------------------------

Return values

0	if the event was not used or understood
1	if the event was used and can be deleted

See also

[Fl_Event](#)

Reimplemented from [Fl_Button](#).

The documentation for this class was generated from the following files:

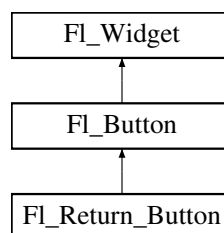
- [Fl_Repeat_Button.H](#)
- [Fl_Repeat_Button.cxx](#)

31.113 Fl_Return_Button Class Reference

The [Fl_Return_Button](#) is a subclass of [Fl_Button](#) that generates a callback when it is pressed or when the user presses the Enter key.

```
#include <Fl_Return_Button.H>
```

Inheritance diagram for [Fl_Return_Button](#):



Public Member Functions

- [Fl_Return_Button](#) (int X, int Y, int W, int H, const char *l=0)
Creates a new [Fl_Return_Button](#) widget using the given position, size, and label string.
- int [handle](#) (int)
Handles the specified event.

Protected Member Functions

- void [draw](#) ()
Draws the widget.

Additional Inherited Members

31.113.1 Detailed Description

The [Fl_Return_Button](#) is a subclass of [Fl_Button](#) that generates a callback when it is pressed or when the user presses the Enter key.

A carriage-return symbol is drawn next to the button label.

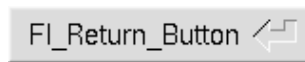


Figure 31.25 [Fl_Return_Button](#)

31.113.2 Constructor & Destructor Documentation

31.113.2.1 [Fl_Return_Button\(\)](#)

```
Fl_Return_Button::Fl_Return_Button (
    int X,
    int Y,
    int W,
    int H,
    const char * l = 0 )
```

Creates a new [Fl_Return_Button](#) widget using the given position, size, and label string.

The default boxtype is `FL_UP_BOX`.

The inherited destructor deletes the button.

31.113.3 Member Function Documentation

31.113.3.1 [draw\(\)](#)

```
void Fl_Return_Button::draw ( ) [protected], [virtual]
```

Draws the widget.

Never call this function directly. FLTK will schedule redrawing whenever needed. If your widget must be redrawn as soon as possible, call [redraw\(\)](#) instead.

Override this function to draw your own widgets.

If you ever need to call another widget's draw method *from within your own [draw\(\)](#) method*, e.g. for an embedded scrollbar, you can do it (because [draw\(\)](#) is virtual) like this:

```
Fl_Widget *s = &scroll;           // scroll is an embedded Fl_Scrollbar
s->draw();                          // calls Fl_Scrollbar::draw()
```

Reimplemented from [Fl_Button](#).

31.113.3.2 [handle\(\)](#)

```
int Fl_Return_Button::handle (
    int event ) [virtual]
```

Handles the specified event.

You normally don't call this method directly, but instead let FLTK do it when the user interacts with the widget.

When implemented in a widget, this function must return 0 if the widget does not use the event or 1 otherwise.

Most of the time, you want to call the inherited [handle\(\)](#) method in your overridden method so that you don't short-circuit events that you don't handle. In this last case you should return the callee retval.

Parameters

in	<i>event</i>	the kind of event received
----	--------------	----------------------------

Return values

0	if the event was not used or understood
1	if the event was used and can be deleted

See also

[FI_Event](#)

Reimplemented from [FI_Button](#).

The documentation for this class was generated from the following files:

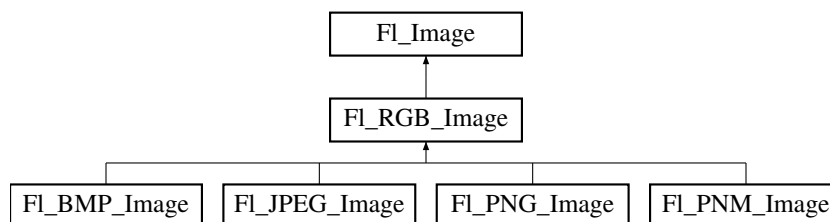
- FI_Return_Button.H
- FI_Return_Button.cxx

31.114 FI_RGB_Image Class Reference

The [FI_RGB_Image](#) class supports caching and drawing of full-color images with 1 to 4 channels of color information.

```
#include <FI_Image.H>
```

Inheritance diagram for FI_RGB_Image:



Public Member Functions

- virtual void [color_average](#) ([FI_Color](#) c, float i)
The [color_average\(\)](#) method averages the colors in the image with the FLTK color value c.
- [FI_Image](#) * [copy](#) ()
- virtual [FI_Image](#) * [copy](#) (int W, int H)
The [copy\(\)](#) method creates a copy of the specified image.
- virtual void [desaturate](#) ()
The [desaturate\(\)](#) method converts an image to grayscale.
- void [draw](#) (int X, int Y)
- virtual void [draw](#) (int X, int Y, int W, int H, int cx=0, int cy=0)
Draws the image with a bounding box.
- [FI_RGB_Image](#) ([const FI_Pixmap](#) *pxm, [FI_Color](#) bg=FL_GRAY)
The constructor creates a new RGBA image from the specified [FI_Pixmap](#).
- [FI_RGB_Image](#) ([const uchar](#) *bits, int W, int H, int D=3, int LD=0)
The constructor creates a new image from the specified data.
- virtual void [label](#) ([FI_Menu_Item](#) *m)
The [label\(\)](#) methods are an obsolete way to set the image attribute of a widget or menu item.
- virtual void [label](#) ([FI_Widget](#) *w)
The [label\(\)](#) methods are an obsolete way to set the image attribute of a widget or menu item.
- virtual void [uncache](#) ()
If the image has been cached for display, delete the cache data.
- virtual ~[FI_RGB_Image](#) ()
The destructor frees all memory and server resources that are used by the image.

Static Public Member Functions

- static `size_t max_size ()`
Returns the maximum allowed image size in bytes when creating an [FI_RGB_Image](#) object.
- static void `max_size (size_t size)`
Sets the maximum allowed image size in bytes when creating an [FI_RGB_Image](#) object.

Public Attributes

- int `alloc_array`
If non-zero, the object's data array is delete[]'d when deleting the object.
- const `uchar * array`
Points to the start of the object's data array.

Friends

- class [FI_GDI_Graphics_Driver](#)
- class [FI_GDI_Printer_Graphics_Driver](#)
- class [FI_Quartz_Graphics_Driver](#)
- class [FI_Xlib_Graphics_Driver](#)

Additional Inherited Members

31.114.1 Detailed Description

The [FI_RGB_Image](#) class supports caching and drawing of full-color images with 1 to 4 channels of color information.

Images with an even number of channels are assumed to contain alpha information, which is used to blend the image with the contents of the screen.

[FI_RGB_Image](#) is defined in [<FL/FI_Image.H>](#), however for compatibility reasons [<FL/FI_RGB_Image.H>](#) should be included.

31.114.2 Constructor & Destructor Documentation

31.114.2.1 FI_RGB_Image() [1/2]

```
Fl_RGB_Image::Fl_RGB_Image (
    const uchar * bits,
    int W,
    int H,
    int D = 3,
    int LD = 0 )
```

The constructor creates a new image from the specified data.

The data array `bits` must contain sufficient data to provide $W * H * D$ image bytes and optional line padding, see `LD`.

`W` and `H` are the width and height of the image in pixels, resp.

`D` is the image depth and can be:

- `D=1`: each `uchar` in `bits[]` is a grayscale pixel value
- `D=2`: each `uchar` pair in `bits[]` is a grayscale + alpha pixel value
- `D=3`: each `uchar` triplet in `bits[]` is an R/G/B pixel value
- `D=4`: each `uchar` quad in `bits[]` is an R/G/B/A pixel value

`LD` specifies the line data size of the array, see [Fl_Image::ld\(int\)](#). If `LD` is zero, then $W * D$ is assumed, otherwise `LD` must be greater than or equal to $W * D$ to account for (unused) extra data per line (padding).

The caller is responsible that the image data array `bits` persists as long as the image is used.

This constructor sets [Fl_RGB_Image::alloc_array](#) to 0. To have the image object control the deallocation of the data array `bits`, set `alloc_array` to non-zero after construction.

Parameters

in	<i>bits</i>	The image data array.
in	<i>W</i>	The width of the image in pixels.
in	<i>H</i>	The height of the image in pixels.
in	<i>D</i>	The image depth, or 'number of channels' (default=3).
in	<i>LD</i>	Line data size (default=0).

See also

[Fl_Image::data\(\)](#), [Fl_Image::w\(\)](#), [Fl_Image::h\(\)](#), [Fl_Image::d\(\)](#), [Fl_Image::ld\(int\)](#)

31.114.2.2 Fl_RGB_Image() [2/2]

```
Fl_RGB_Image::Fl_RGB_Image (
    const Fl_Pixmap * pxm,
    Fl_Color bg = FL_GRAY )
```

The constructor creates a new RGBA image from the specified [Fl_Pixmap](#).

The RGBA image is built fully opaque except for the transparent area of the pixmap that is assigned the `bg` color with full transparency.

This constructor creates a new internal data array and sets [Fl_RGB_Image::alloc_array](#) to 1 so the data array is deleted when the image is destroyed.

31.114.3 Member Function Documentation

31.114.3.1 color_average()

```
void Fl_RGB_Image::color_average (
    Fl_Color c,
    float i ) [virtual]
```

The [color_average\(\)](#) method averages the colors in the image with the FLTK color value `c`.

The `i` argument specifies the amount of the original image to combine with the color, so a value of 1.0 results in no color blend, and a value of 0.0 results in a constant image of the specified color.

An internal copy is made of the original image before changes are applied, to avoid modifying the original image.

Reimplemented from [Fl_Image](#).

31.114.3.2 copy()

```
Fl_Image * Fl_RGB_Image::copy (
    int W,
    int H ) [virtual]
```

The `copy()` method creates a copy of the specified image.

If the width and height are provided, the image is resized to the specified size. The image should be deleted (or in the case of [Fl_Shared_Image](#), released) when you are done with it.

Reimplemented from [Fl_Image](#).

31.114.3.3 desaturate()

```
void Fl_RGB_Image::desaturate ( ) [virtual]
```

The `desaturate()` method converts an image to grayscale.

If the image contains an alpha channel (depth = 4), the alpha channel is preserved.

An internal copy is made of the original image before changes are applied, to avoid modifying the original image.

Reimplemented from [Fl_Image](#).

31.114.3.4 draw()

```
void Fl_RGB_Image::draw (
    int X,
    int Y,
    int W,
    int H,
    int cx = 0,
    int cy = 0 ) [virtual]
```

Draws the image with a bounding box.

Arguments `X`, `Y`, `W`, `H` specify a bounding box for the image, with the origin (upper-left corner) of the image offset by the `cx` and `cy` arguments.

In other words: `fl_push_clip(X,Y,W,H)` is applied, the image is drawn with its upper-left corner at `X-cx`, `Y-cy` and its own width and height, `fl_pop_clip()` is applied.

Reimplemented from [Fl_Image](#).

31.114.3.5 label() [1/2]

```
void Fl_RGB_Image::label (
    Fl_Menu_Item * m ) [virtual]
```

The `label()` methods are an obsolete way to set the image attribute of a widget or menu item.

Use the `image()` or `deimage()` methods of the [Fl_Widget](#) and [Fl_Menu_Item](#) classes instead.

Reimplemented from [Fl_Image](#).

31.114.3.6 label() [2/2]

```
void Fl_RGB_Image::label (
    Fl_Widget * widget ) [virtual]
```

The `label()` methods are an obsolete way to set the image attribute of a widget or menu item.

Use the `image()` or `deimage()` methods of the [Fl_Widget](#) and [Fl_Menu_Item](#) classes instead.

Reimplemented from [Fl_Image](#).

31.114.3.7 max_size() [1/2]

```
static size_t Fl_RGB_Image::max_size ( ) [inline], [static]
```

Returns the maximum allowed image size in bytes when creating an [Fl_RGB_Image](#) object.

See also

```
void Fl_RGB_Image::max_size(size_t)
```

31.114.3.8 max_size() [2/2]

```
static void Fl_RGB_Image::max_size (
    size_t size ) [inline], [static]
```

Sets the maximum allowed image size in bytes when creating an [Fl_RGB_Image](#) object.

The image size in bytes of an `Fl_RGB_Image` object is the value of the product $w() * h() * d()$. If this product exceeds size, the created object of a derived class of `Fl_RGB_Image` won't be loaded with the image data. This does not apply to direct RGB image creation with `Fl_RGB_Image::Fl_RGB_Image(const uchar *bits, int W, int H, int D, int LD)`. The default `max_size()` value is essentially infinite.

31.114.3.9 `uncache()`

```
void Fl_RGB_Image::uncache ( ) [virtual]
```

If the image has been cached for display, delete the cache data.

This allows you to change the data used for the image and then redraw it without recreating an image object.

Reimplemented from `Fl_Image`.

The documentation for this class was generated from the following files:

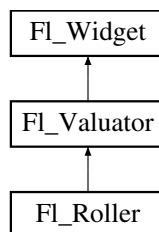
- `Fl_Image.H`
- `Fl_Image.cxx`

31.115 `Fl_Roller` Class Reference

The `Fl_Roller` widget is a "dolly" control commonly used to move 3D objects.

```
#include <Fl_Roller.H>
```

Inheritance diagram for `Fl_Roller`:



Public Member Functions

- `Fl_Roller` (int X, int Y, int W, int H, const char *L=0)
Creates a new `Fl_Roller` widget using the given position, size, and label string.
- int `handle` (int)
Handles the specified event.

Protected Member Functions

- void `draw` ()
Draws the widget.

Additional Inherited Members

31.115.1 Detailed Description

The `Fl_Roller` widget is a "dolly" control commonly used to move 3D objects.

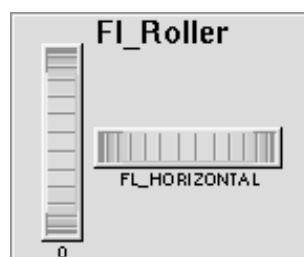


Figure 31.26 `Fl_Roller`

31.115.2 Constructor & Destructor Documentation

31.115.2.1 Fl_Roller()

```
Fl_Roller::Fl_Roller (
    int X,
    int Y,
    int W,
    int H,
    const char * L = 0 )
```

Creates a new [Fl_Roller](#) widget using the given position, size, and label string. The default boxtype is FL_NO_BOX.

Inherited destructor destroys the valuator.

31.115.3 Member Function Documentation

31.115.3.1 draw()

```
void Fl_Roller::draw ( ) [protected], [virtual]
```

Draws the widget.

Never call this function directly. FLTK will schedule redrawing whenever needed. If your widget must be redrawn as soon as possible, call [redraw\(\)](#) instead.

Override this function to draw your own widgets.

If you ever need to call another widget's draw method *from within your own [draw\(\)](#) method*, e.g. for an embedded scrollbar, you can do it (because [draw\(\)](#) is virtual) like this:

```
Fl_Widget *s = &scroll;           // scroll is an embedded Fl_Scrollbar
s->draw();                          // calls Fl_Scrollbar::draw()
```

Implements [Fl_Widget](#).

31.115.3.2 handle()

```
int Fl_Roller::handle (
    int event ) [virtual]
```

Handles the specified event.

You normally don't call this method directly, but instead let FLTK do it when the user interacts with the widget.

When implemented in a widget, this function must return 0 if the widget does not use the event or 1 otherwise.

Most of the time, you want to call the inherited [handle\(\)](#) method in your overridden method so that you don't short-circuit events that you don't handle. In this last case you should return the callee retval.

Parameters

in	<i>event</i>	the kind of event received
----	--------------	----------------------------

Return values

0	if the event was not used or understood
1	if the event was used and can be deleted

See also

[Fl_Event](#)

Reimplemented from [Fl_Widget](#).

The documentation for this class was generated from the following files:

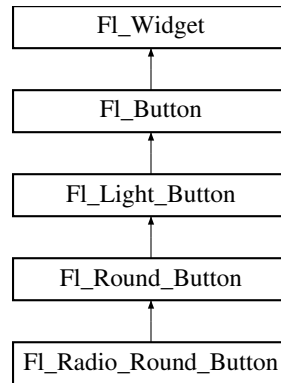
- `Fl_Roller.H`
- `Fl_Roller.cxx`

31.116 `Fl_Round_Button` Class Reference

Buttons generate callbacks when they are clicked by the user.

```
#include <Fl_Round_Button.H>
```

Inheritance diagram for `Fl_Round_Button`:



Public Member Functions

- `Fl_Round_Button` (int *x*, int *y*, int *w*, int *h*, const char **l*=0)
Creates a new `Fl_Round_Button` widget using the given position, size, and label string.

Additional Inherited Members

31.116.1 Detailed Description

Buttons generate callbacks when they are clicked by the user.

You control exactly when and how by changing the values for `type()` and `when()`.

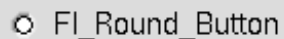


Figure 31.27 `Fl_Round_Button`

The `Fl_Round_Button` subclass display the "on" state by turning on a light, rather than drawing pushed in. The shape of the "light" is initially set to `FL_ROUND_DOWN_BOX`. The color of the light when on is controlled with `selection_color()`, which defaults to `FL_FOREGROUND_COLOR`.

31.116.2 Constructor & Destructor Documentation

31.116.2.1 `Fl_Round_Button()`

```
Fl_Round_Button::Fl_Round_Button (
    int X,
    int Y,
    int W,
    int H,
    const char * L = 0 )
```

Creates a new `Fl_Round_Button` widget using the given position, size, and label string.

○ FI_Round_Button

Figure 31.28 FI_Round_Button

The `FI_Round_Button` subclass displays the "ON" state by turning on a light, rather than drawing pushed in. The default box type is `FL_NO_BOX`, which draws the label w/o a box right of the checkmark. The shape of the "light" is set with `down_box()` and its default value is `FL_ROUND_DOWN_BOX`. The color of the light when on is controlled with `selection_color()`, which defaults to `FL_FOREGROUND_COLOR` (usually black).

Parameters

in	<i>X,Y,W,H</i>	position and size of the widget
in	<i>L</i>	widget label, default is no label

The documentation for this class was generated from the following files:

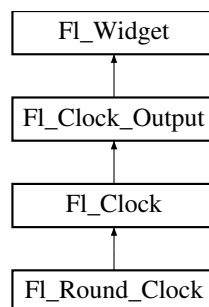
- `FI_Round_Button.H`
- `FI_Round_Button.cxx`

31.117 FI_Round_Clock Class Reference

A clock widget of type `FL_ROUND_CLOCK`.

```
#include <Fl_Round_Clock.H>
```

Inheritance diagram for `FI_Round_Clock`:



Public Member Functions

- **FI_Round_Clock** (int X, int Y, int W, int H, const char *L=0)
Creates the clock widget, setting his type and box.

Additional Inherited Members

31.117.1 Detailed Description

A clock widget of type `FL_ROUND_CLOCK`.

Has no box.

The documentation for this class was generated from the following files:

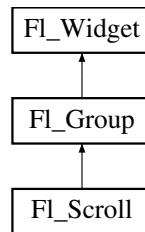
- `FI_Round_Clock.H`
- `FI_Clock.cxx`

31.118 FI_Scroll Class Reference

This container widget lets you maneuver around a set of widgets much larger than your window.

```
#include <Fl_Scroll.H>
```

Inheritance diagram for Fl_Scroll:



Classes

- struct [Fl_Region_LRTB](#)
A local struct to manage a region defined by left/right/top/bottom.
- struct [Fl_Region_XYWH](#)
A local struct to manage a region defined by xywh.
- struct [Fl_Scrollbar_Data](#)
A local struct to manage a scrollbar's xywh region and tab values.
- struct [ScrollInfo](#)
Structure to manage scrollbar and widget interior sizes.

Public Types

- enum {
HORIZONTAL = 1 , **VERTICAL** = 2 , **BOTH** = 3 , **ALWAYS_ON** = 4 ,
HORIZONTAL_ALWAYS = 5 , **VERTICAL_ALWAYS** = 6 , **BOTH_ALWAYS** = 7 }

Public Member Functions

- void **clear** ()
Clear all but the scrollbars...
- [Fl_Scroll](#) (int X, int Y, int W, int H, const char *l=0)
Creates a new [Fl_Scroll](#) widget using the given position, size, and label string.
- int **handle** (int)
Handles the specified event.
- void **resize** (int X, int Y, int W, int H)
Resizes the [Fl_Scroll](#) widget and moves its children if necessary.
- void **scroll_to** (int, int)
Moves the contents of the scroll group to a new position.
- int **scrollbar_size** () const
Gets the current size of the scrollbars' troughs, in pixels.
- void **scrollbar_size** (int newSize)
*Sets the pixel size of the scrollbars' troughs to *newSize*, in pixels.*
- int **xposition** () const
Gets the current horizontal scrolling position.
- int **yposition** () const
Gets the current vertical scrolling position.

Public Attributes

- [Fl_Scrollbar](#) **hscrollbar**
- [Fl_Scrollbar](#) **scrollbar**

Protected Member Functions

- void `bbox` (int &, int &, int &, int &)
Returns the bounding box for the interior of the scrolling area, inside the scrollbars.
- void `draw` ()
Draws the widget.
- void `recalc_scrollbars` (ScrollInfo &si)
Calculate visibility/size/position of scrollbars, find children's bounding box.

Additional Inherited Members

31.118.1 Detailed Description

This container widget lets you maneuver around a set of widgets much larger than your window. If the child widgets are larger than the size of this object then scrollbars will appear so that you can scroll over to them:

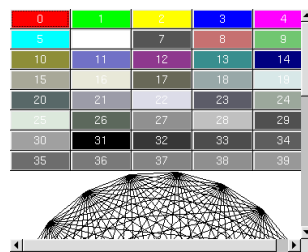


Figure 31.29 FI_Scroll

If all of the child widgets are packed together into a solid rectangle then you want to set `box()` to `FL_NO_BOX` or one of the `_FRAME` types. This will result in the best output. However, if the child widgets are a sparse arrangement you must set `box()` to a real `_BOX` type. This can result in some blinking during redrawing, but that can be solved by using a `FI_Double_Window`.

By default you can scroll in both directions, and the scrollbars disappear if the data will fit in the area of the scroll. Use `FI_Scroll::type()` to change this as follows :

- 0 - No scrollbars
- `FI_Scroll::HORIZONTAL` - Only a horizontal scrollbar.
- `FI_Scroll::VERTICAL` - Only a vertical scrollbar.
- `FI_Scroll::BOTH` - The default is both scrollbars.
- `FI_Scroll::HORIZONTAL_ALWAYS` - Horizontal scrollbar always on, vertical always off.
- `FI_Scroll::VERTICAL_ALWAYS` - Vertical scrollbar always on, horizontal always off.
- `FI_Scroll::BOTH_ALWAYS` - Both always on.

Use `scrollbar.align(int)` (see void `FI_Widget::align(FI_Align)`) : to change what side the scrollbars are drawn on.

If the `FL_ALIGN_LEFT` bit is on, the vertical scrollbar is on the left. If the `FL_ALIGN_TOP` bit is on, the horizontal scrollbar is on the top. Note that only the alignment flags in scrollbar are considered. The flags in `hscrollbar` however are ignored.

This widget can also be used to pan around a single child widget "canvas". This child widget should be of your own class, with a `draw()` method that draws the contents. The scrolling is done by changing the `x()` and `y()` of the widget, so this child must use the `x()` and `y()` to position its drawing. To speed up drawing it should test `fl_not_clipped(int x,int y,int w,int h)` to find out if a particular area of the widget must be drawn.

Another very useful child is a single `FI_Pack`, which is itself a group that packs its children together and changes size to surround them. Filling the `FI_Pack` with `FI_Tabs` groups (and then putting normal widgets inside those) gives you a very powerful scrolling list of individually-openable panels.

Fluid lets you create these, but you can only lay out objects that fit inside the [Fl_Scroll](#) without scrolling. Be sure to leave space for the scrollbars, as Fluid won't show these either.

You cannot use [Fl_Window](#) as a child of this since the clipping is not conveyed to it when drawn, and it will draw over the scrollbars and neighboring objects.

31.118.2 Constructor & Destructor Documentation

31.118.2.1 Fl_Scroll()

```
Fl_Scroll::Fl_Scroll (
    int X,
    int Y,
    int W,
    int H,
    const char * L = 0 )
```

Creates a new [Fl_Scroll](#) widget using the given position, size, and label string.

The default boxtype is FL_NO_BOX.

The destructor *also deletes all the children*. This allows a whole tree to be deleted at once, without having to keep a pointer to all the children in the user code. A kludge has been done so the [Fl_Scroll](#) and all of its children can be automatic (local) variables, but you must declare the [Fl_Scroll](#) *first*, so that it is destroyed last.

31.118.3 Member Function Documentation

31.118.3.1 bbox()

```
void Fl_Scroll::bbox (
    int & X,
    int & Y,
    int & W,
    int & H ) [protected]
```

Returns the bounding box for the interior of the scrolling area, inside the scrollbars.

Currently this is only reliable after [draw\(\)](#), and before any resizing of the [Fl_Scroll](#) or any child widgets occur.

Todo The visibility of the scrollbars ought to be checked/calculated outside of the [draw\(\)](#) method (STR #1895).

31.118.3.2 draw()

```
void Fl_Scroll::draw ( ) [protected], [virtual]
```

Draws the widget.

Never call this function directly. FLTK will schedule redrawing whenever needed. If your widget must be redrawn as soon as possible, call [redraw\(\)](#) instead.

Override this function to draw your own widgets.

If you ever need to call another widget's draw method *from within your own [draw\(\)](#) method*, e.g. for an embedded scrollbar, you can do it (because [draw\(\)](#) is virtual) like this:

```
Fl_Widget *s = &scroll; // scroll is an embedded Fl_Scrollbar
s->draw(); // calls Fl_Scrollbar::draw()
```

Reimplemented from [Fl_Group](#).

31.118.3.3 handle()

```
int Fl_Scroll::handle (
    int event ) [virtual]
```

Handles the specified event.

You normally don't call this method directly, but instead let FLTK do it when the user interacts with the widget. When implemented in a widget, this function must return 0 if the widget does not use the event or 1 otherwise. Most of the time, you want to call the inherited [handle\(\)](#) method in your overridden method so that you don't short-circuit events that you don't handle. In this last case you should return the callee retval.

Parameters

in	<i>event</i>	the kind of event received
----	--------------	----------------------------

Return values

0	if the event was not used or understood
1	if the event was used and can be deleted

See also

[Fl_Event](#)

Reimplemented from [Fl_Group](#).

31.118.3.4 recalc_scrollbars()

```
void Fl_Scroll::recalc_scrollbars (
    ScrollInfo & si ) [protected]
```

Calculate visibility/size/position of scrollbars, find children's bounding box.

The *si* parameter will be filled with data from the calculations. Derived classes can make use of this call to figure out the scrolling area eg. during [resize\(\)](#) handling.

Parameters

in	<i>si</i>	– ScrollInfo structure
----	-----------	--

Returns

Structure containing the calculated info.

31.118.3.5 resize()

```
void Fl_Scroll::resize (
    int X,
    int Y,
    int W,
    int H ) [virtual]
```

Resizes the [Fl_Scroll](#) widget and moves its children if necessary.

The [Fl_Scroll](#) widget first resizes itself, and then it moves all its children if (and only if) the [Fl_Scroll](#) widget has been moved. The children are moved by the same amount as the [Fl_Scroll](#) widget has been moved, hence all children keep their relative positions.

Note

[Fl_Scroll::resize\(\)](#) does **not** call [Fl_Group::resize\(\)](#), and child widgets are **not** resized.

Since children of an [Fl_Scroll](#) are not resized, the [resizable\(\)](#) widget is ignored (if it is set).

The scrollbars are moved to their proper positions, as given by [Fl_Scroll::scrollbar.align\(\)](#), and switched on or off as necessary.

Note

Due to current (FLTK 1.3.x) implementation constraints some of this may effectively be postponed until the [Fl_Scroll](#) is drawn the next time. This may change in a future release.

See also

[Fl_Group::resizable\(\)](#)

[Fl_Widget::resize\(int,int,int,int\)](#)

Reimplemented from [Fl_Group](#).

31.118.3.6 scroll_to()

```
void Fl_Scroll::scroll_to (
    int X,
    int Y )
```

Moves the contents of the scroll group to a new position.

This is like moving the scrollbars of the [Fl_Scroll](#) around. For instance:

```
Fl_Scroll scroll (10,10,200,200);
Fl_Box b1 ( 10, 10,50,50,"b1"); // relative (x,y) = (0,0)
Fl_Box b2 ( 60, 60,50,50,"b2"); // relative (x,y) = (50,50)
Fl_Box b3 ( 60,110,50,50,"b3"); // relative (x,y) = (50,100)
// populate scroll with more children ...
scroll.end();
scroll.scroll_to(50,100);
```

will move the logical origin of the internal scroll area to (-50,-100) relative to the origin of the [Fl_Scroll](#) (10,10), i.e. [Fl_Box](#) b3 will be visible in the top left corner of the scroll area.

31.118.3.7 scrollbar_size() [1/2]

```
int Fl_Scroll::scrollbar_size ( ) const [inline]
```

Gets the current size of the scrollbars' troughs, in pixels.

If this value is zero (default), this widget will use the [Fl::scrollbar_size\(\)](#) value as the scrollbar's width.

Returns

Scrollbar size in pixels, or 0 if the global [Fl::scrollbar_size\(\)](#) is being used.

See also

[Fl::scrollbar_size\(int\)](#)

31.118.3.8 scrollbar_size() [2/2]

```
void Fl_Scroll::scrollbar_size (
    int newSize ) [inline]
```

Sets the pixel size of the scrollbars' troughs to `newSize`, in pixels.

Normally you should not need this method, and should use [Fl::scrollbar_size\(int\)](#) instead to manage the size of ALL your widgets' scrollbars. This ensures your application has a consistent UI, is the default behavior, and is normally what you want.

Only use THIS method if you really need to override the global scrollbar size. The need for this should be rare.

Setting `newSize` to the special value of 0 causes the widget to track the global [Fl::scrollbar_size\(\)](#), which is the default.

Parameters

in	<i>newSize</i>	Sets the scrollbar size in pixels. If 0 (default), scrollbar size tracks the global Fl::scrollbar_size()
----	----------------	---

See also

[Fl::scrollbar_size\(\)](#)

31.118.3.9 xposition()

```
int Fl_Scroll::xposition ( ) const [inline]
```

Gets the current horizontal scrolling position.

31.118.3.10 yposition()

```
int Fl_Scroll::yposition ( ) const [inline]
```

Gets the current vertical scrolling position.

The documentation for this class was generated from the following files:

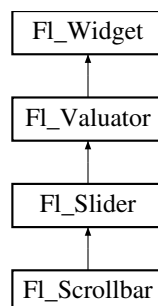
- `Fl_Scroll.H`
- `Fl_Scroll.cxx`

31.119 Fl_Scrollbar Class Reference

The [Fl_Scrollbar](#) widget displays a slider with arrow buttons at the ends of the scrollbar.

```
#include <Fl_Scrollbar.H>
```

Inheritance diagram for `Fl_Scrollbar`:



Public Member Functions

- [Fl_Scrollbar](#) (int X, int Y, int W, int H, const char *L=0)
Creates a new [Fl_Scrollbar](#) widget with given position, size, and label.
- int [handle](#) (int)
Handles the specified event.
- int [linesize](#) () const
Get the size of step, in lines, that the arrow keys move.
- void [linesize](#) (int i)
This number controls how big the steps are that the arrow keys do.
- int [value](#) () const
Gets the integer value (position) of the slider in the scrollbar.
- int [value](#) (int p)
Sets the value (position) of the slider in the scrollbar.
- int [value](#) (int pos, int windowSize, int first, int total)
Sets the position, size and range of the slider in the scrollbar.

- `~Fl_Scrollbar ()`

Destroys the Scrollbar.

Protected Member Functions

- `void draw ()`

Draws the widget.

Additional Inherited Members

31.119.1 Detailed Description

The `Fl_Scrollbar` widget displays a slider with arrow buttons at the ends of the scrollbar.

Clicking on the arrows move up/left and down/right by `linesize()`. Scrollbars also accept `FL_SHORTCUT` events: the arrows move by `linesize()`, and vertical scrollbars take Page Up/Down (they move by the page size minus `linesize()`) and Home/End (they jump to the top or bottom).

Scrollbars have `step(1)` preset (they always return integers). If desired you can set the `step()` to non-integer values. You will then have to use casts to get at the floating-point versions of `value()` from `Fl_Slider`.

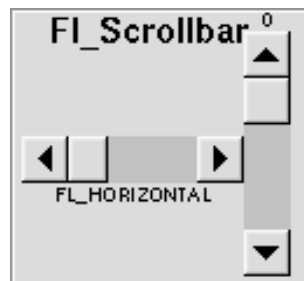


Figure 31.30 `Fl_Scrollbar`

31.119.2 Constructor & Destructor Documentation

31.119.2.1 `Fl_Scrollbar()`

```
Fl_Scrollbar::Fl_Scrollbar (
    int X,
    int Y,
    int W,
    int H,
    const char * L = 0 )
```

Creates a new `Fl_Scrollbar` widget with given position, size, and label. You need to do `type(FL_HORIZONTAL)` if you want a horizontal scrollbar.

31.119.3 Member Function Documentation

31.119.3.1 `draw()`

```
void Fl_Scrollbar::draw ( ) [protected], [virtual]
```

Draws the widget.

Never call this function directly. FLTK will schedule redrawing whenever needed. If your widget must be redrawn as soon as possible, call `redraw()` instead.

Override this function to draw your own widgets.

If you ever need to call another widget's draw method *from within your own `draw()` method*, e.g. for an embedded scrollbar, you can do it (because `draw()` is virtual) like this:

```
Fl_Widget *s = &scroll;           // scroll is an embedded Fl_Scrollbar
s->draw();                       // calls Fl_Scrollbar::draw()
```

Implements [Fl_Widget](#).

31.119.3.2 handle()

```
int Fl_Scrollbar::handle (
    int event ) [virtual]
```

Handles the specified event.

You normally don't call this method directly, but instead let FLTK do it when the user interacts with the widget. When implemented in a widget, this function must return 0 if the widget does not use the event or 1 otherwise. Most of the time, you want to call the inherited [handle\(\)](#) method in your overridden method so that you don't short-circuit events that you don't handle. In this last case you should return the callee retval.

Parameters

in	<i>event</i>	the kind of event received
----	--------------	----------------------------

Return values

0	if the event was not used or understood
1	if the event was used and can be deleted

See also

[Fl_Event](#)

Reimplemented from [Fl_Widget](#).

31.119.3.3 linesize()

```
void Fl_Scrollbar::linesize (
    int i ) [inline]
```

This number controls how big the steps are that the arrow keys do.

In addition page up/down move by the size last sent to [value\(\)](#) minus one [linesize\(\)](#). The default is 16.

31.119.3.4 value() [1/3]

```
int Fl_Scrollbar::value ( ) const [inline]
```

Gets the integer value (position) of the slider in the scrollbar.

You can get the floating point value with [Fl_Slider::value\(\)](#).

See also

[Fl_Scrollbar::value\(int p\)](#)

[Fl_Scrollbar::value\(int pos, int size, int first, int total\)](#)

31.119.3.5 value() [2/3]

```
int Fl_Scrollbar::value (
    int p ) [inline]
```

Sets the value (position) of the slider in the scrollbar.

See also

[Fl_Scrollbar::value\(\)](#)

[Fl_Scrollbar::value\(int pos, int size, int first, int total\)](#)

31.119.3.6 value() [3/3]

```
int Fl_Scrollbar::value (
    int pos,
    int windowSize,
    int first,
    int total ) [inline]
```

Sets the position, size and range of the slider in the scrollbar.

Parameters

in	<i>pos</i>	position, first line displayed
in	<i>windowSize</i>	number of lines displayed
in	<i>first</i>	number of first line
in	<i>total</i>	total number of lines

You should call this every time your window changes size, your data changes size, or your scroll position changes (even if in response to a callback from this scrollbar). All necessary calls to [redraw\(\)](#) are done.

Calls [Fl_Slider::scrollvalue\(int pos, int size, int first, int total\)](#).

The documentation for this class was generated from the following files:

- [Fl_Scrollbar.H](#)
- [Fl_Scrollbar.cxx](#)

31.120 Fl_Scroll::Fl_Scrollbar_Data Struct Reference

A local struct to manage a scrollbar's xywh region and tab values.

```
#include <Fl_Scroll.H>
```

Public Attributes

- int **first**
scrollbar tab's "number of first line"
- int **h**
- int **pos**
scrollbar tab's "position of first line displayed"
- int **size**
scrollbar tab's "size of window in lines"
- int **total**
scrollbar tab's "total number of lines"
- int **w**
- int **x**
- int **y**

31.120.1 Detailed Description

A local struct to manage a scrollbar's xywh region and tab values.

The documentation for this struct was generated from the following file:

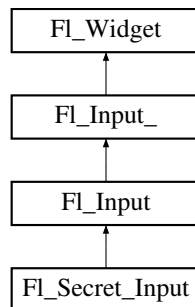
- [Fl_Scroll.H](#)

31.121 FI_Secret_Input Class Reference

The [FI_Secret_Input](#) class is a subclass of [FI_Input](#) that displays its input as a string of placeholders.

```
#include <Fl_Secret_Input.H>
```

Inheritance diagram for [FI_Secret_Input](#):



Public Member Functions

- [FI_Secret_Input](#) (int X, int Y, int W, int H, const char *l=0)
Creates a new [FI_Secret_Input](#) widget using the given position, size, and label string.
- int [handle](#) (int)
Handles the specified event.

Additional Inherited Members

31.121.1 Detailed Description

The [FI_Secret_Input](#) class is a subclass of [FI_Input](#) that displays its input as a string of placeholders.

Depending on the platform this placeholder is either the asterisk (*) or the Unicode bullet character (U+2022).

This subclass is usually used to receive passwords and other "secret" information.

31.121.2 Constructor & Destructor Documentation

31.121.2.1 FI_Secret_Input()

```
Fl_Secret_Input::Fl_Secret_Input (
    int X,
    int Y,
    int W,
    int H,
    const char * l = 0 )
```

Creates a new [FI_Secret_Input](#) widget using the given position, size, and label string.

The default boxtype is `FL_DOWN_BOX`.

Inherited destructor destroys the widget and any value associated with it.

31.121.3 Member Function Documentation

31.121.3.1 handle()

```
int Fl_Secret_Input::handle (
    int event ) [virtual]
```

Handles the specified event.

You normally don't call this method directly, but instead let FLTK do it when the user interacts with the widget.

When implemented in a widget, this function must return 0 if the widget does not use the event or 1 otherwise. Most of the time, you want to call the inherited [handle\(\)](#) method in your overridden method so that you don't short-circuit events that you don't handle. In this last case you should return the callee retval.

Parameters

<code>in</code>	<code>event</code>	the kind of event received
-----------------	--------------------	----------------------------

Return values

<code>0</code>	if the event was not used or understood
<code>1</code>	if the event was used and can be deleted

See also

[Fl_Event](#)

Reimplemented from [Fl_Input](#).

The documentation for this class was generated from the following files:

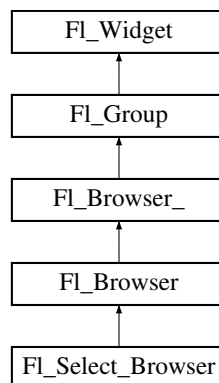
- [Fl_Secret_Input.H](#)
- [Fl_Input.cxx](#)

31.122 Fl_Select_Browser Class Reference

The class is a subclass of [Fl_Browser](#) which lets the user select a single item, or no items by clicking on the empty space.

```
#include <Fl_Select_Browser.H>
```

Inheritance diagram for [Fl_Select_Browser](#):



Public Member Functions

- [Fl_Select_Browser](#) (int X, int Y, int W, int H, const char *L=0)
Creates a new [Fl_Select_Browser](#) widget using the given position, size, and label string.

Additional Inherited Members

31.122.1 Detailed Description

The class is a subclass of [Fl_Browser](#) which lets the user select a single item, or no items by clicking on the empty space.

As long as the mouse button is held down on an unselected item it is highlighted. Normally the callback is done when the user presses the mouse, but you can change this with [when\(\)](#).

See [Fl_Browser](#) for methods to add and remove lines from the browser.

31.122.2 Constructor & Destructor Documentation

31.122.2.1 FI_Select_Browser()

```
Fl_Select_Browser::Fl_Select_Browser (
    int X,
    int Y,
    int W,
    int H,
    const char * L = 0 )
```

Creates a new [FI_Select_Browser](#) widget using the given position, size, and label string.

The default boxtype is FL_DOWN_BOX. The constructor specializes [FI_Browser\(\)](#) by setting the type to FL_SELECT_BROWSER. The destructor destroys the widget and frees all memory that has been allocated.

The documentation for this class was generated from the following files:

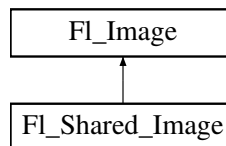
- FI_Select_Browser.H
- FI_Browser.cxx

31.123 FI_Shared_Image Class Reference

This class supports caching, loading, scaling, and drawing of image files.

```
#include <Fl_Shared_Image.H>
```

Inheritance diagram for FI_Shared_Image:



Public Member Functions

- virtual void [color_average](#) ([FI_Color](#) c, float i)
 - The [color_average\(\)](#) method averages the colors in the image with the FLTK color value c.*
- [FI_Image](#) * [copy](#) ()
- virtual [FI_Image](#) * [copy](#) (int W, int H)
 - The [copy\(\)](#) method creates a copy of the specified image.*
- virtual void [desaturate](#) ()
 - The [desaturate\(\)](#) method converts an image to grayscale.*
- void [draw](#) (int X, int Y)
- virtual void [draw](#) (int X, int Y, int W, int H, int cx, int cy)
 - Draws the image with a bounding box.*
- const char * [name](#) ()
 - Returns the filename of the shared image.*
- int [original](#) ()
 - Returns whether this is an original image.*
- int [refcount](#) ()
 - Returns the number of references of this shared image.*
- void [release](#) ()
 - Releases and possibly destroys (if refcount <= 0) a shared image.*
- void [reload](#) ()
 - Reloads the shared image from disk.*
- void [scale](#) (int width, int height, int proportional=1, int can_expand=0)

Sets the drawing size of the shared image.

- virtual void [uncache](#) ()

If the image has been cached for display, delete the cache data.

Static Public Member Functions

- static void **add_handler** (FI_Shared_Handler f)
Adds a shared image handler, which is basically a test function for adding new formats.
- static [FI_Shared_Image](#) * **find** (const char *name, int W=0, int H=0)
Finds a shared image from its name and size specifications.
- static [FI_Shared_Image](#) * **get** (const char *name, int W=0, int H=0)
Find or load an image that can be shared by multiple widgets.
- static [FI_Shared_Image](#) * **get** ([FI_RGB_Image](#) *rgb, int own_it=1)
Builds a shared image from a pre-existing [FI_RGB_Image](#).
- static [FI_Shared_Image](#) ** **images** ()
Returns the [FI_Shared_Image](#) array.*
- static int **num_images** ()
Returns the total number of shared images in the array.
- static void **remove_handler** (FI_Shared_Handler f)
Removes a shared image handler.
- static void **scaling_algorithm** ([FI_RGB_Scaling](#) algorithm)
Sets what algorithm is used when resizing a source image.

Protected Member Functions

- void **add** ()
Adds a shared image to the image cache.
- [FI_Shared_Image](#) ()
Creates an empty shared image.
- [FI_Shared_Image](#) (const char *n, [FI_Image](#) *img=0)
Creates a shared image from its filename and its corresponding [FI_Image](#) img.*
- void **update** ()
- virtual [~FI_Shared_Image](#) ()
The destructor frees all memory and server resources that are used by the image.

Static Protected Member Functions

- static int **compare** ([FI_Shared_Image](#) **i0, [FI_Shared_Image](#) **i1)
Compares two shared images.

Protected Attributes

- int **alloc_image_**
- [FI_Image](#) * **image_**
- const char * **name_**
- int **original_**
- int **refcount_**

Static Protected Attributes

- static int **alloc_handlers_** = 0
- static int **alloc_images_** = 0
- static FI_Shared_Handler * **handlers_** = 0
- static [FI_Shared_Image](#) ** **images_** = 0
- static int **num_handlers_** = 0
- static int **num_images_** = 0

Friends

- class `FI_JPEG_Image`
- class `FI_PNG_Image`

Additional Inherited Members

31.123.1 Detailed Description

This class supports caching, loading, scaling, and drawing of image files.

Most applications will also want to link against the `fltk_images` library and call the `fl_register_images()` function to support standard image formats such as BMP, GIF, JPEG, and PNG.

Images can be requested (loaded) with `FI_Shared_Image::get()`, `find()`, and some other methods. All images are cached in an internal list of shared images and should be released when they are no longer needed. A `refcount` is used to determine if a released image is to be destroyed with `delete`.

See also

[FI_Shared_Image::get\(\)](#)

[FI_Shared_Image::find\(\)](#)

[FI_Shared_Image::release\(\)](#)

31.123.2 Constructor & Destructor Documentation

31.123.2.1 FI_Shared_Image() [1/2]

```
FI_Shared_Image::FI_Shared_Image ( ) [protected]
```

Creates an empty shared image.

The constructors create a new shared image record in the image cache.

The constructors are protected and cannot be used directly from a program. Use the [get\(\)](#) method instead.

31.123.2.2 FI_Shared_Image() [2/2]

```
FI_Shared_Image::FI_Shared_Image (
    const char * n,
    FI_Image * img = 0 ) [protected]
```

Creates a shared image from its filename and its corresponding `FI_Image*` `img`.

The constructors create a new shared image record in the image cache.

The constructors are protected and cannot be used directly from a program. Use the [get\(\)](#) method instead.

31.123.2.3 ~FI_Shared_Image()

```
FI_Shared_Image::~FI_Shared_Image ( ) [protected], [virtual]
```

The destructor frees all memory and server resources that are used by the image.

The destructor is protected and cannot be used directly from a program. Use the [FI_Shared_Image::release\(\)](#) method instead.

31.123.3 Member Function Documentation

31.123.3.1 add()

```
void FI_Shared_Image::add ( ) [protected]
```

Adds a shared image to the image cache.

This **protected** method adds an image to the cache, an ordered list of shared images. The cache is searched for a matching image whenever one is requested, for instance with [FI_Shared_Image::get\(\)](#) or [FI_Shared_Image::find\(\)](#).

31.123.3.2 color_average()

```
void Fl_Shared_Image::color_average (
    Fl_Color c,
    float i ) [virtual]
```

The `color_average()` method averages the colors in the image with the FLTK color value `c`.

The `i` argument specifies the amount of the original image to combine with the color, so a value of 1.0 results in no color blend, and a value of 0.0 results in a constant image of the specified color.

An internal copy is made of the original image before changes are applied, to avoid modifying the original image.

Reimplemented from [Fl_Image](#).

31.123.3.3 compare()

```
int Fl_Shared_Image::compare (
    Fl_Shared_Image ** i0,
    Fl_Shared_Image ** i1 ) [static], [protected]
```

Compares two shared images.

The order of comparison is:

1. Image name, usually the filename used to load it
2. Image width
3. Image height

A special case is considered if the width of one of the images is zero and the other image is marked `original`. In this case the images match, i.e. the comparison returns success (0).

An image is marked `original` if it was directly loaded from a file or from memory as opposed to copied and resized images.

This comparison is used in [Fl_Shared_Image::find\(\)](#) to find an image that matches the requested one or to find the position where a new image should be entered into the sorted list of shared images.

It is usually used in two steps:

1. search with exact width and height
2. if not found, search again with width = 0 (and height = 0)

The first step will only return a match if the image exists with the same width and height. The second step will match if there is an image marked `original` with the same name, regardless of width and height.

Returns

Whether the images match or their relative sort order (see text).

Return values

<code>0</code>	the images match
<code><0</code>	Image <code>i0</code> is <i>less</i> than image <code>i1</code>
<code>>0</code>	Image <code>i0</code> is <i>greater</i> than image <code>i1</code>

31.123.3.4 copy()

```
Fl_Image * Fl_Shared_Image::copy (
    int W,
    int H ) [virtual]
```

The `copy()` method creates a copy of the specified image.

If the width and height are provided, the image is resized to the specified size. The image should be deleted (or in

the case of [Fl_Shared_Image](#), released) when you are done with it.
Reimplemented from [Fl_Image](#).

31.123.3.5 desaturate()

```
void Fl_Shared_Image::desaturate ( ) [virtual]
```

The [desaturate\(\)](#) method converts an image to grayscale.

If the image contains an alpha channel (depth = 4), the alpha channel is preserved.

An internal copy is made of the original image before changes are applied, to avoid modifying the original image.

Reimplemented from [Fl_Image](#).

31.123.3.6 draw()

```
void Fl_Shared_Image::draw (
    int X,
    int Y,
    int W,
    int H,
    int cx,
    int cy ) [virtual]
```

Draws the image with a bounding box.

Arguments X, Y, W, H specify a bounding box for the image, with the origin (upper-left corner) of the image offset by the cx and cy arguments.

In other words: `fl_push_clip(X, Y, W, H)` is applied, the image is drawn with its upper-left corner at X-cx, Y-cy and its own width and height, `fl_pop_clip()` is applied.

Reimplemented from [Fl_Image](#).

31.123.3.7 find()

```
Fl_Shared_Image * Fl_Shared_Image::find (
    const char * name,
    int W = 0,
    int H = 0 ) [static]
```

Finds a shared image from its name and size specifications.

This uses a binary search in the image cache.

If the image name exists with the exact width W and height H, then it is returned.

If W == 0 and the image name exists with another size, then the **original** image with that name is returned.

In either case the refcount of the returned image is increased. The found image should be released with [Fl_Shared_Image::release\(\)](#) when no longer needed.

31.123.3.8 get() [1/2]

```
Fl_Shared_Image * Fl_Shared_Image::get (
    const char * name,
    int W = 0,
    int H = 0 ) [static]
```

Find or load an image that can be shared by multiple widgets.

If the image exists with the requested size, this image will be returned.

If the image exists, but only with another size, then a new copy with the requested size (width W and height H) will be created as a resized copy of the original image. The new image is added to the internal list of shared images.

If the image does not yet exist, then a new image of the proper dimension is created from the filename name. The original image from filename name is always added to the list of shared images in its original size. If the requested size differs, then the resized copy with width W and height H is also added to the list of shared images.

Note

If the sizes differ, then *two* images are created as mentioned above. This is intentional so the original image is cached and preserved. If you request the same image with another size later, then the **original** image will be found, copied, resized, and returned.

Shared JPEG and PNG images can also be created from memory by using their named memory access constructor. You should [release\(\)](#) the image when you're done with it.

Parameters

<i>name</i>	name of the image
<i>W,H</i>	desired size

See also

[FI_Shared_Image::find\(const char *name, int W, int H\)](#)

[FI_Shared_Image::release\(\)](#)

[FI_JPEG_Image::FI_JPEG_Image\(const char *name, const unsigned char *data\)](#)

[FI_PNG_Image::FI_PNG_Image](#) (const char *name_png, const unsigned char *buffer, int maxsize)

31.123.3.9 get() [2/2]

```
FI_Shared_Image * FI_Shared_Image::get (
    FI_RGB_Image * rgb,
    int own_it = 1 ) [static]
```

Builds a shared image from a pre-existing [FI_RGB_Image](#).

Parameters

in	<i>rgb</i>	an FI_RGB_Image used to build a new shared image.
in	<i>own↔ _it</i>	1 if the shared image should delete <i>rgb</i> when it is itself deleted, 0 otherwise

Version

1.3.4

31.123.3.10 original()

```
int FI_Shared_Image::original ( ) [inline]
```

Returns whether this is an original image.

Images loaded from a file or from memory are marked `original` as opposed to images created as a copy of another image with different size (width or height).

Note

This is useful for debugging (rarely used in user code).

Since

FLTK 1.4.0

31.123.3.11 refcount()

```
int Fl_Shared_Image::refcount ( ) [inline]
```

Returns the number of references of this shared image.

When reference is below 1, the image is deleted.

31.123.3.12 release()

```
void Fl_Shared_Image::release ( )
```

Releases and possibly destroys (if `refcount <= 0`) a shared image.

In the latter case, it will reorganize the shared image array so that no hole will occur.

31.123.3.13 scale()

```
void Fl_Shared_Image::scale (
    int width,
    int height,
    int proportional = 1,
    int can_expand = 0 )
```

Sets the drawing size of the shared image.

This function gives the shared image its own size, independently from the size of the original image that is typically larger. This can be useful to draw a shared image on a drawing surface whose resolution is higher than the drawing unit for this surface: all pixels of the original image become available to fill an area of the drawing surface sized at `width, height`. Examples of such drawing surfaces: laser printers, PostScript files, PDF printers, retina displays on Apple hardware.

Parameters

<i>width,height</i>	maximum width and height (in drawing units) to use when drawing the shared image
<i>proportional</i>	if not null, keep the width and height of the shared image proportional to those of its original image
<i>can_expand</i>	if null, the width and height of the shared image will not exceed those of the original image

Version

1.3.4 and requires compiling with `FLTK_ABI_VERSION = 10304`

Example code: scale an image to fit in a box

```
Fl_Box *b = ... // a box
Fl_Shared_Image *shared = Fl_Shared_Image::get("/path/to/picture.jpeg"); // read a picture file
shared->scale(b->w(), b->h(), 1); // set the drawing size of the shared image to the size of the box
b->image(shared); // use the shared image as the box image
b->align(FL_ALIGN_INSIDE | FL_ALIGN_CENTER | FL_ALIGN_CLIP); // the image is to be drawn centered in the box
```

31.123.3.14 scaling_algorithm()

```
static void Fl_Shared_Image::scaling_algorithm (
    Fl_RGB_Scaling algorithm ) [inline], [static]
```

Sets what algorithm is used when resizing a source image.

The default algorithm is `FL_RGB_SCALING_BILINEAR`. Drawing an [Fl_Shared_Image](#) is sometimes performed by first resizing the source image and then drawing the resized copy. This occurs, e.g., when drawing to screen under Linux or MSWindows after having called [Fl_Shared_Image::scale\(\)](#). This function controls what method is used when the image to be resized is an [Fl_RGB_Image](#).

Version

1.3.4 and requires compiling with `FLTK_ABI_VERSION = 10304`

31.123.3.15 uncache()

```
void Fl_Shared_Image::uncache ( ) [virtual]
```

If the image has been cached for display, delete the cache data.

This allows you to change the data used for the image and then redraw it without recreating an image object.

Reimplemented from [Fl_Image](#).

The documentation for this class was generated from the following files:

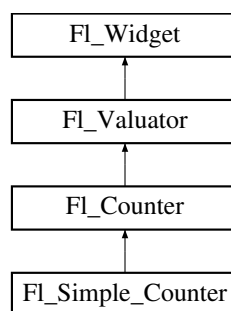
- [Fl_Shared_Image.H](#)
- [Fl_Shared_Image.cxx](#)

31.124 Fl_Simple_Counter Class Reference

This widget creates a counter with only 2 arrow buttons.

```
#include <Fl_Simple_Counter.H>
```

Inheritance diagram for Fl_Simple_Counter:



Public Member Functions

- [Fl_Simple_Counter](#) (int X, int Y, int W, int H, const char *L=0)

Additional Inherited Members

31.124.1 Detailed Description

This widget creates a counter with only 2 arrow buttons.

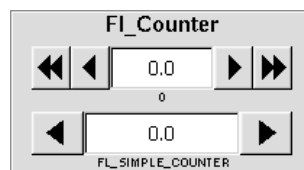


Figure 31.31 Fl_Simple_Counter

The documentation for this class was generated from the following files:

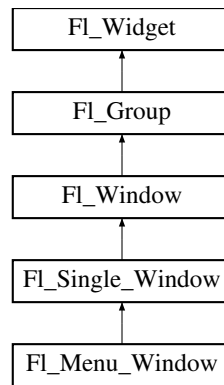
- [Fl_Simple_Counter.H](#)
- [Fl_Counter.cxx](#)

31.125 Fl_Single_Window Class Reference

This is the same as [Fl_Window](#).

```
#include <Fl_Single_Window.H>
```

Inheritance diagram for Fl_Single_Window:



Public Member Functions

- **FI_Single_Window** (int W, int H, const char *l=0)
Creates a new [FI_Single_Window](#) widget using the given size, and label (title) string.
- **FI_Single_Window** (int X, int Y, int W, int H, const char *l=0)
Creates a new [FI_Single_Window](#) widget using the given position, size, and label (title) string.
- void **flush** ()
Forces the window to be drawn, this window is also made current and calls [draw\(\)](#).
- int **make_current** ()
- void **show** ()
Puts the window on the screen.
- void **show** (int a, char **b)

Additional Inherited Members

31.125.1 Detailed Description

This is the same as [FI_Window](#).

However, it is possible that some implementations will provide double-buffered windows by default. This subclass can be used to force single-buffering. This may be useful for modifying existing programs that use incremental update, or for some types of image data, such as a movie flipbook.

31.125.2 Member Function Documentation

31.125.2.1 flush()

```
void FI_Single_Window::flush ( ) [virtual]
```

Forces the window to be drawn, this window is also made current and calls [draw\(\)](#).

Reimplemented from [FI_Window](#).

31.125.2.2 show()

```
void FI_Single_Window::show ( ) [virtual]
```

Puts the window on the screen.

Usually (on X) this has the side effect of opening the display.

If the window is already shown then it is restored and raised to the top. This is really convenient because your program can call [show\(\)](#) at any time, even if the window is already up. It also means that [show\(\)](#) serves the purpose of [raise\(\)](#) in other toolkits.

[FI_Window::show\(int argc, char **argv\)](#) is used for top-level windows and allows standard arguments to be parsed from the command-line.

Note

For some obscure reasons `FL_Window::show()` resets the current group by calling `FL_Group::current(0)`. The comments in the code say "get rid of very common user bug: forgot end()". Although this is true it may have unwanted side effects if you `show()` an unrelated window (maybe for an error message or warning) while building a window or any other group widget.

Todo Check if we can remove resetting the current group in a later FLTK version (after 1.3.x). This may break "already broken" programs though if they rely on this "feature".

See also

[FL_Window::show\(int argc, char **argv\)](#)

Reimplemented from [FL_Window](#).

The documentation for this class was generated from the following files:

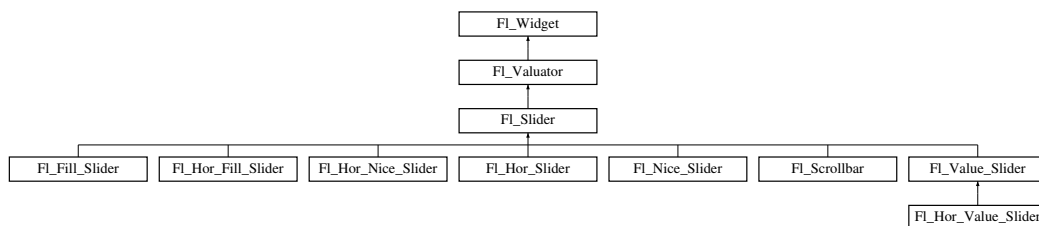
- `FL_Single_Window.H`
- `FL_Single_Window.cxx`

31.126 FL_Slider Class Reference

The [FL_Slider](#) widget contains a sliding knob inside a box.

```
#include <FL_Slider.H>
```

Inheritance diagram for [FL_Slider](#):



Public Member Functions

- void `bounds` (double a, double b)
Sets the minimum (a) and maximum (b) values for the valuator widget.
- `FL_Slider` (int X, int Y, int W, int H, const char *L=0)
Creates a new [FL_Slider](#) widget using the given position, size, and label string.
- `FL_Slider` (uchar t, int X, int Y, int W, int H, const char *L)
Creates a new [FL_Slider](#) widget using the given type, position, size, and label string.
- int `handle` (int)
Handles the specified event.
- int `scrollvalue` (int pos, int size, int first, int total)
Sets the size and position of the sliding knob in the box.
- `FL_Boxtype slider` () const
Gets the slider box type.
- void `slider` (FL_Boxtype c)
Sets the slider box type.
- float `slider_size` () const
Get the dimensions of the moving piece of slider.
- void `slider_size` (double v)
Set the dimensions of the moving piece of slider.

Protected Member Functions

- void `draw` ()
Draws the widget.
- void `draw` (int, int, int, int)
- int `handle` (int, int, int, int, int)

Additional Inherited Members

31.126.1 Detailed Description

The `Fl_Slider` widget contains a sliding knob inside a box.

It is often used as a scrollbar. Moving the box all the way to the top/left sets it to the `minimum()`, and to the bottom/right to the `maximum()`. The `minimum()` may be greater than the `maximum()` to reverse the slider direction. Use `void Fl_Widget::type(int)` to set how the slider is drawn, which can be one of the following:

- `FL_VERTICAL` - Draws a vertical slider (this is the default).
- `FL_HORIZONTAL` - Draws a horizontal slider.
- `FL_VERT_FILL_SLIDER` - Draws a filled vertical slider, useful as a progress or value meter.
- `FL_HOR_FILL_SLIDER` - Draws a filled horizontal slider, useful as a progress or value meter.
- `FL_VERT_NICE_SLIDER` - Draws a vertical slider with a nice looking control knob.
- `FL_HOR_NICE_SLIDER` - Draws a horizontal slider with a nice looking control knob.

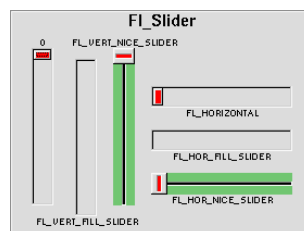


Figure 31.32 `Fl_Slider`

31.126.2 Constructor & Destructor Documentation

31.126.2.1 `Fl_Slider()`

```
Fl_Slider::Fl_Slider (
    int X,
    int Y,
    int W,
    int H,
    const char * L = 0 )
```

Creates a new `Fl_Slider` widget using the given position, size, and label string. The default boxtype is `FL_DOWN_BOX`.

31.126.3 Member Function Documentation

31.126.3.1 bounds()

```
void Fl_Slider::bounds (
    double a,
    double b )
```

Sets the minimum (a) and maximum (b) values for the valuator widget. if at least one of the values is changed, a partial redraw is asked.

31.126.3.2 draw()

```
void Fl_Slider::draw ( ) [protected], [virtual]
```

Draws the widget.

Never call this function directly. FLTK will schedule redrawing whenever needed. If your widget must be redrawn as soon as possible, call [redraw\(\)](#) instead.

Override this function to draw your own widgets.

If you ever need to call another widget's draw method *from within your own draw() method*, e.g. for an embedded scrollbar, you can do it (because [draw\(\)](#) is virtual) like this:

```
Fl_Widget *s = &scroll; // scroll is an embedded Fl_Scrollbar
s->draw(); // calls Fl_Scrollbar::draw()
```

Implements [Fl_Widget](#).

Reimplemented in [Fl_Value_Slider](#).

31.126.3.3 handle()

```
int Fl_Slider::handle (
    int event ) [virtual]
```

Handles the specified event.

You normally don't call this method directly, but instead let FLTK do it when the user interacts with the widget.

When implemented in a widget, this function must return 0 if the widget does not use the event or 1 otherwise.

Most of the time, you want to call the inherited [handle\(\)](#) method in your overridden method so that you don't short-circuit events that you don't handle. In this last case you should return the callee retval.

Parameters

in	<i>event</i>	the kind of event received
----	--------------	----------------------------

Return values

0	if the event was not used or understood
1	if the event was used and can be deleted

See also

[Fl_Event](#)

Reimplemented from [Fl_Widget](#).

Reimplemented in [Fl_Value_Slider](#).

31.126.3.4 scrollvalue()

```
int Fl_Slider::scrollvalue (
    int pos,
    int size,
    int first,
    int total )
```

Sets the size and position of the sliding knob in the box.

Parameters

in	<i>pos</i>	position of first line displayed
in	<i>size</i>	size of window in lines
in	<i>first</i>	number of first line
in	<i>total</i>	total number of lines Returns <code>Fl_Valuator::value(p)</code>

31.126.3.5 slider_size()

```
void Fl_Slider::slider_size (
    double v )
```

Set the dimensions of the moving piece of slider.

This is the fraction of the size of the entire widget. If you set this to 1 then the slider cannot move. The default value is .08.

For the "fill" sliders this is the size of the area around the end that causes a drag effect rather than causing the slider to jump to the mouse.

The documentation for this class was generated from the following files:

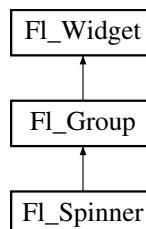
- `Fl_Slider.H`
- `Fl_Slider.cxx`

31.127 Fl_Spinner Class Reference

This widget is a combination of the input widget and repeat buttons.

```
#include <Fl_Spinner.H>
```

Inheritance diagram for `Fl_Spinner`:

**Public Member Functions**

- `Fl_Color color () const`
Return the background color of the spinner widget's input field.
- `void color (Fl_Color v)`
Change the background color of the spinner widget's input field.
- `Fl_Spinner (int X, int Y, int W, int H, const char *L=0)`
Creates a new `Fl_Spinner` widget using the given position, size, and label string.
- `const char * format ()`
Sets or returns the format string for the value.
- `void format (const char *f)`
Sets or returns the format string for the value.
- `int handle (int event)`
Handles the specified event.
- `double maximum () const`
Gets the maximum value of the widget.
- `void maximum (double m)`

- Sets the maximum value of the widget.*

 - double **maximum** () const

Spelling mistakes retained for source compatibility.
- double **minimum** () const

Gets the minimum value of the widget.
- void **minimum** (double m)

Sets the minimum value of the widget.
- double **mininum** () const

Spelling mistakes retained for source compatibility.
- void **range** (double a, double b)

Sets the minimum and maximum values for the widget.
- void **resize** (int X, int Y, int W, int H)

Resizes the [FI_Group](#) widget and all of its children.
- [FI_Color](#) **selection_color** () const

Return the selection color of the spinner widget's input field.
- void **selection_color** ([FI_Color](#) val)

Change the selection color of the spinner widget's input field.
- double **step** () const

Sets or returns the amount to change the value when the user clicks a button.
- void **step** (double s)

See double [FI_Spinner::step\(\)](#) const.
- [FI_Color](#) **textcolor** () const

Gets the color of the text in the input field.
- void **textcolor** ([FI_Color](#) c)

Sets the color of the text in the input field.
- [FI_Font](#) **textfont** () const

Gets the font of the text in the input field.
- void **textfont** ([FI_Font](#) f)

Sets the font of the text in the input field.
- [FI_Fontsize](#) **textsize** () const

Gets the size of the text in the input field.
- void **textsize** ([FI_Fontsize](#) s)

Sets the size of the text in the input field.
- [uchar](#) **type** () const

Gets the numeric representation in the input field.
- void **type** ([uchar](#) v)

Sets the numeric representation in the input field.
- double **value** () const

Gets the current value of the widget.
- void **value** (double v)

Sets the current value of the widget.

Protected Attributes

- [FI_Repeat_Button](#) **down_button_**
- [FI_Input](#) **input_**
- [FI_Repeat_Button](#) **up_button_**

Additional Inherited Members

31.127.1 Detailed Description

This widget is a combination of the input widget and repeat buttons. The user can either type into the input area or use the buttons to change the value.

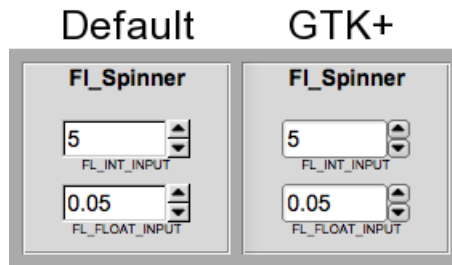


Figure 31.33 Fl_Spinner widget

31.127.2 Constructor & Destructor Documentation

31.127.2.1 Fl_Spinner()

```
Fl_Spinner::Fl_Spinner (
    int X,
    int Y,
    int W,
    int H,
    const char * L = 0 )
```

Creates a new [Fl_Spinner](#) widget using the given position, size, and label string. Inherited destructor Destroys the widget and any value associated with it.

31.127.3 Member Function Documentation

31.127.3.1 handle()

```
int Fl_Spinner::handle (
    int event ) [inline], [virtual]
```

Handles the specified event.

You normally don't call this method directly, but instead let FLTK do it when the user interacts with the widget.

When implemented in a widget, this function must return 0 if the widget does not use the event or 1 otherwise.

Most of the time, you want to call the inherited [handle\(\)](#) method in your overridden method so that you don't short-circuit events that you don't handle. In this last case you should return the callee retval.

Parameters

in	<i>event</i>	the kind of event received
----	--------------	----------------------------

Return values

0	if the event was not used or understood
1	if the event was used and can be deleted

See also

[Fl_Event](#)

Reimplemented from [Fl_Group](#).

31.127.3.2 maximum()

```
double Fl_Spinner::maximum ( ) const [inline]
```

Spelling mistakes retained for source compatibility.

Deprecated

31.127.3.3 minimum()

```
double Fl_Spinner::minimum ( ) const [inline]
```

Spelling mistakes retained for source compatibility.

Deprecated

31.127.3.4 resize()

```
void Fl_Spinner::resize (
    int X,
    int Y,
    int W,
    int H ) [inline], [virtual]
```

Resizes the [Fl_Group](#) widget and all of its children.

The [Fl_Group](#) widget first resizes itself, and then it moves and resizes all its children according to the rules documented for [Fl_Group::resizable\(Fl_Widget*\)](#)

See also

[Fl_Group::resizable\(Fl_Widget*\)](#)

[Fl_Group::resizable\(\)](#)

[Fl_Widget::resize\(int,int,int,int\)](#)

Reimplemented from [Fl_Group](#).

31.127.3.5 step()

```
double Fl_Spinner::step ( ) const [inline]
```

Sets or returns the amount to change the value when the user clicks a button.

Before setting step to a non-integer value, the spinner [type\(\)](#) should be changed to floating point.

31.127.3.6 type() [1/2]

```
uchar Fl_Spinner::type ( ) const [inline]
```

Gets the numeric representation in the input field.

See also

[Fl_Spinner::type\(uchar\)](#)

31.127.3.7 type() [2/2]

```
void Fl_Spinner::type (
    uchar v ) [inline]
```

Sets the numeric representation in the input field.

Valid values are FL_INT_INPUT and FL_FLOAT_INPUT. Also changes the [format\(\)](#) template. Setting a new spinner type via a superclass pointer will not work.

Note

type is not a virtual function.

31.127.3.8 value()

```
void Fl_Spinner::value (
    double v ) [inline]
```

Sets the current value of the widget.

Before setting value to a non-integer value, the spinner [type\(\)](#) should be changed to floating point.

The documentation for this class was generated from the following files:

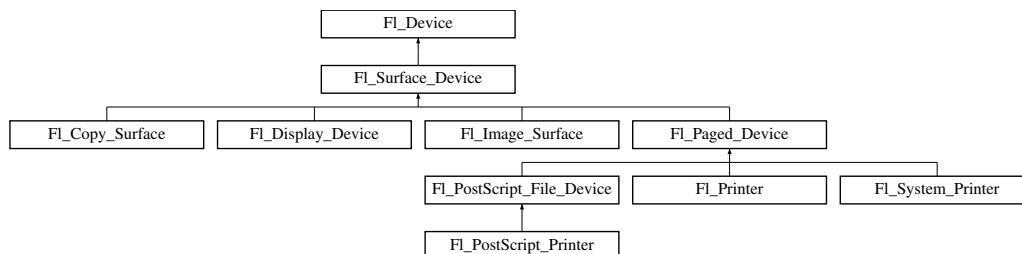
- Fl_Spinner.H
- Fl_Group.cxx

31.128 Fl_Surface_Device Class Reference

A drawing surface that's susceptible to receive graphical output.

```
#include <Fl_Device.H>
```

Inheritance diagram for Fl_Surface_Device:



Public Member Functions

- const char * [class_name](#) ()
Returns the name of the class of this object.
- [Fl_Graphics_Driver](#) * [driver](#) ()
Returns the graphics driver of this drawing surface.
- void [driver](#) ([Fl_Graphics_Driver](#) *graphics_driver)
Sets the graphics driver of this drawing surface.
- virtual void [set_current](#) (void)
Make this surface the current drawing surface.
- virtual ~[Fl_Surface_Device](#) ()
The destructor.

Static Public Member Functions

- static [Fl_Surface_Device](#) * [surface](#) ()
The current drawing surface.

Static Public Attributes

- static const char * `class_id` = "Fl_Surface_Device"

Protected Member Functions

- `Fl_Surface_Device` (`Fl_Graphics_Driver` *graphics_driver)
Constructor that sets the graphics driver to use for the created surface.

31.128.1 Detailed Description

A drawing surface that's susceptible to receive graphical output.

Any FLTK application has at any time a current drawing surface to which all drawing requests are directed. The current surface is given by `Fl_Surface_Device::surface()`. When `main()` begins running, the current drawing surface has been set to the computer's display, an instance of the `Fl_Display_Device` class.

A drawing surface other than the computer's display, is typically used as follows:

1. Create `surface`, an object from a particular `Fl_Surface_Device` derived class (e.g., `Fl_Copy_Surface`, `Fl_Printer`).
2. Memorize what is the current drawing surface with `Fl_Surface_Device *old_current = Fl_Surface_Device::surface();`
3. Call `surface->set_current();` to redirect all graphics requests to `surface` which becomes the new current drawing surface (not necessary with class `Fl_Printer` because it is done by `Fl_Printer::start_job()`).
4. At this point any of the [Drawing functions](#) (e.g., `fl_rect()`) or the [Color & Font functions](#) or [Drawing Images functions](#) (e.g., `fl_draw_image()`, `Fl_Image::draw()`) operates on the new current drawing surface. Certain drawing surfaces allow additional ways to draw to them (e.g., `Fl_Printer::print_widget()`, `Fl_Image_Surface::draw()`).
5. After all drawing requests have been performed, redirect graphics requests back to their previous destination with `old_current->set_current();`
6. Delete `surface`.

31.128.2 Member Function Documentation

31.128.2.1 class_name()

```
const char * Fl_Surface_Device::class_name ( ) [inline], [virtual]
```

Returns the name of the class of this object.

Use of the `class_name()` function is discouraged because it will be removed from future FLTK versions.

The class of an instance of an `Fl_Device` subclass can be checked with code such as:

```
if ( instance->class_name() == Fl_Printer::class_id ) { ... }
```

Reimplemented from `Fl_Device`.

Reimplemented in `Fl_Display_Device`, `Fl_Image_Surface`, `Fl_Paged_Device`, `Fl_PostScript_File_Device`, `Fl_System_Printer`, `Fl_PostScript_Printer`, and `Fl_Printer`.

31.128.2.2 set_current()

```
void Fl_Surface_Device::set_current (
    void ) [virtual]
```

Make this surface the current drawing surface.

This surface will receive all future graphics requests.

Reimplemented in `Fl_Copy_Surface`, `Fl_Image_Surface`, and `Fl_Printer`.

31.128.2.3 surface()

```
static Fl_Surface_Device * Fl_Surface_Device::surface ( ) [inline], [static]
```

The current drawing surface.

In other words, the [Fl_Surface_Device](#) object that currently receives all graphics output

The documentation for this class was generated from the following files:

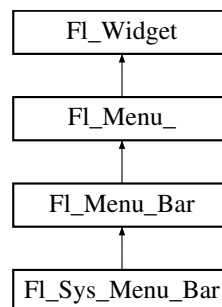
- [Fl_Device.H](#)
- [Fl_Device.cxx](#)

31.129 Fl_Sys_Menu_Bar Class Reference

A class to create, modify and delete menus that appear on Mac OS X in the menu bar at the top of the screen.

```
#include <Fl_Sys_Menu_Bar.H>
```

Inheritance diagram for [Fl_Sys_Menu_Bar](#):



Public Member Functions

- int [add](#) (const char *label, const char *shortcut, [Fl_Callback](#) *cb, void *user_data=0, int flags=0)
Adds a new menu item.
- int [add](#) (const char *label, int shortcut, [Fl_Callback](#) *, void *user_data=0, int flags=0)
Add a new menu item to the system menu bar.
- int [add](#) (const char *str)
Forms-compatible procedure to add items to the system menu bar.
- void [clear](#) ()
Set the [Fl_Menu_Item](#) array pointer to null, indicating a zero-length menu.
- int [clear_submenu](#) (int index)
Clears the specified submenu pointed to by index of all menu items.
- [Fl_Sys_Menu_Bar](#) (int x, int y, int w, int h, const char *l=0)
The constructor.
- void [global](#) ()
Make the shortcuts for this menu work no matter what window has the focus when you type it.
- int [insert](#) (int index, const char *label, const char *shortcut, [Fl_Callback](#) *cb, void *user_data=0, int flags=0)
Insert a new menu item.
- int [insert](#) (int index, const char *label, int shortcut, [Fl_Callback](#) *cb, void *user_data=0, int flags=0)
insert in the system menu bar a new menu item
- const [Fl_Menu_Item](#) * [menu](#) () const
Return the system menu's array of [Fl_Menu_Item](#)'s.
- void [menu](#) (const [Fl_Menu_Item](#) *m)
create a system menu bar using the given list of menu structs
- int [mode](#) (int i) const
Gets the flags of item i.
- void [mode](#) (int i, int fl)

- Sets the flags of item i.*
- void **remove** (int n)
 - remove an item from the system menu bar*
- void **replace** (int index, const char *name)
 - rename an item from the system menu bar*
- void **setonly** (FI_Menu_Item *item)
 - Turns the radio item "on" for the menu item and turns "off" adjacent radio items of the same group.*
- void **shortcut** (int i, int s)
 - Changes the shortcut of item i to n.*
- void **update** ()
 - Updates the system menu after any change to its items.*
- **~FI_Sys_Menu_Bar** ()
 - The destructor.*

Protected Member Functions

- void **draw** ()
 - Draws the widget.*

Additional Inherited Members

31.129.1 Detailed Description

A class to create, modify and delete menus that appear on Mac OS X in the menu bar at the top of the screen.

On other than Mac OS X platforms, [FI_Sys_Menu_Bar](#) is a synonym of class [FI_Menu_Bar](#).

To use this class, just replace [FI_Menu_Bar](#) by [FI_Sys_Menu_Bar](#), and, on the Mac platform, a system menu at the top of the screen will be available. This menu will match an array of [FI_Menu_Item](#)'s exactly as with standard FLTK menus.

Changes to the menu state are immediately visible in the menubar when they are made using member functions of the [FI_Sys_Menu_Bar](#) class. Other changes (e.g., by a call to [FI_Menu_Item::set\(\)](#)) should be followed by a call to [FI_Sys_Menu_Bar::update\(\)](#) to be visible in the menubar across all platforms.

A few FLTK features are not supported by the Mac System menu:

- no symbolic labels
- no embossed labels
- no font sizes

You can configure a callback for the 'About' menu item to invoke your own code with [fl_mac_set_about\(\)](#).

31.129.2 Constructor & Destructor Documentation

31.129.2.1 FI_Sys_Menu_Bar()

```
FI_Sys_Menu_Bar::FI_Sys_Menu_Bar (
    int x,
    int y,
    int w,
    int h,
    const char * l = 0 )
```

The constructor.

On Mac OS X, all arguments are unused. On other platforms they are used as by [FI_Menu_Bar::FI_Menu_Bar\(\)](#).

31.129.3 Member Function Documentation

31.129.3.1 add() [1/3]

```
int Fl_Sys_Menu_Bar::add (
    const char * label,
    const char * shortcut,
    Fl_Callback * cb,
    void * user_data = 0,
    int flags = 0 ) [inline]
```

Adds a new menu item.

See also

[Fl_Menu_::add\(const char* label, int shortcut, Fl_Callback*, void *user_data=0, int flags=0\)](#)

31.129.3.2 add() [2/3]

```
int Fl_Sys_Menu_Bar::add (
    const char * label,
    int shortcut,
    Fl_Callback * cb,
    void * user_data = 0,
    int flags = 0 )
```

Add a new menu item to the system menu bar.

Parameters

<i>label</i>	- new menu item's label
<i>shortcut</i>	- new menu item's integer shortcut (can be 0 for none, or e.g. FL_ALT+'x')
<i>cb</i>	- callback to be invoked when item selected (can be 0 for none, in which case the menubar's callback() can be used instead)
<i>user_data</i>	- argument to the callback
<i>flags</i>	- item's flags, e.g. FL_MENU_TOGGLE , etc.

Returns

the index into the [menu\(\)](#) array, where the entry was added

See also

[Fl_Menu_::add\(const char* label, int shortcut, Fl_Callback *cb, void *user_data, int flags\)](#)

31.129.3.3 add() [3/3]

```
int Fl_Sys_Menu_Bar::add (
    const char * str )
```

Forms-compatible procedure to add items to the system menu bar.

Returns

the index into the [menu\(\)](#) array, where the entry was added

See also

[Fl_Menu_::add\(const char* str\)](#)

31.129.3.4 clear()

```
void Fl_Sys_Menu_Bar::clear ( )
```

Set the [Fl_Menu_Item](#) array pointer to null, indicating a zero-length menu.

See also

[Fl_Menu_::clear\(\)](#)

31.129.3.5 clear_submenu()

```
int Fl_Sys_Menu_Bar::clear_submenu (
    int index )
```

Clears the specified submenu pointed to by index of all menu items.

See also

[Fl_Menu_::clear_submenu\(int index\)](#)

31.129.3.6 draw()

```
void Fl_Sys_Menu_Bar::draw ( ) [protected], [virtual]
```

Draws the widget.

Never call this function directly. FLTK will schedule redrawing whenever needed. If your widget must be redrawn as soon as possible, call [redraw\(\)](#) instead.

Override this function to draw your own widgets.

If you ever need to call another widget's draw method *from within your own [draw\(\)](#) method*, e.g. for an embedded scrollbar, you can do it (because [draw\(\)](#) is virtual) like this:

```
Fl_Widget *s = &scroll; // scroll is an embedded Fl_Scrollbar
s->draw(); // calls Fl_Scrollbar::draw()
```

Reimplemented from [Fl_Menu_Bar](#).

31.129.3.7 insert() [1/2]

```
int Fl_Sys_Menu_Bar::insert (
    int index,
    const char * label,
    const char * shortcut,
    Fl_Callback * cb,
    void * user_data = 0,
    int flags = 0 ) [inline]
```

Insert a new menu item.

See also

[Fl_Menu_::insert\(int index, const char* label, const char* shortcut, Fl_Callback *cb, void *user_data=0, int flags=0\)](#)

31.129.3.8 insert() [2/2]

```
int Fl_Sys_Menu_Bar::insert (
    int index,
    const char * label,
    int shortcut,
    Fl_Callback * cb,
    void * user_data = 0,
    int flags = 0 )
```

insert in the system menu bar a new menu item

Insert in the system menu bar a new menu item, with a title string, shortcut int, callback, argument to the callback, and flags.

Returns

the index into the `menu()` array, where the entry was inserted

See also

[Fl_Menu_::insert\(int index, const char* label, int shortcut, Fl_Callback *cb, void *user_data, int flags\)](#)

31.129.3.9 menu()

```
void Fl_Sys_Menu_Bar::menu (
    const Fl_Menu_Item * m )
```

create a system menu bar using the given list of menu structs

Author

Matthias Melcher

Parameters

<i>m</i>	list of Fl_Menu_Item
----------	--------------------------------------

31.129.3.10 mode()

```
void Fl_Sys_Menu_Bar::mode (
    int i,
    int fl ) [inline]
```

Sets the flags of item *i*.

See also

[Fl_Menu_::mode\(int i, int fl\)](#)

31.129.3.11 remove()

```
void Fl_Sys_Menu_Bar::remove (
    int index )
```

remove an item from the system menu bar

Parameters

<i>index</i>	the index of the item to remove
--------------	---------------------------------

31.129.3.12 replace()

```
void Fl_Sys_Menu_Bar::replace (
    int index,
    const char * name )
```

rename an item from the system menu bar

Parameters

<i>index</i>	the index of the item to rename
<i>name</i>	the new item name as a UTF8 string

The documentation for this class was generated from the following files:

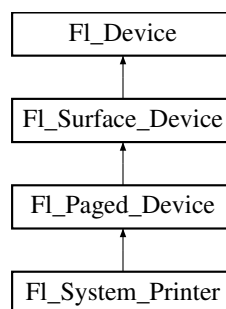
- FI_Sys_Menu_Bar.H
- FI_Sys_Menu_Bar.mm

31.130 FI_System_Printer Class Reference

Print support under MSWindows and Mac OS.

```
#include <FI_Printer.H>
```

Inheritance diagram for FI_System_Printer:



Public Member Functions

- const char * [class_name](#) ()
Returns the name of the class of this object.
- void [end_job](#) (void)
To be called at the end of a print job.
- int [end_page](#) (void)
To be called at the end of each page.
- void [margins](#) (int *left, int *top, int *right, int *bottom)
Computes the dimensions of margins that lie between the printable page area and the full page.
- void [origin](#) (int *x, int *y)
Computes the page coordinates of the current origin of graphics functions.
- void [origin](#) (int x, int y)
Sets the position in page coordinates of the origin of graphics functions.
- int [printable_rect](#) (int *w, int *h)
Computes the width and height of the printable area of the page.
- void [rotate](#) (float angle)
Rotates the graphics operations relatively to paper.
- void [scale](#) (float scale_x, float scale_y=0.)
Changes the scaling of page coordinates.
- int [start_job](#) (int pagecount, int *frompage=NULL, int *topage=NULL)
Starts a print job.
- int [start_page](#) (void)
Starts a new printed page.
- void [translate](#) (int x, int y)
Translates the current graphics origin accounting for the current rotation.

- void [untranslate](#) (void)
Undoes the effect of a previous [translate\(\)](#) call.
- [~FI_System_Printer](#) (void)
The destructor.

Static Public Attributes

- static const char * [class_id](#) = FI_Printer::class_id

Protected Member Functions

- [FI_System_Printer](#) (void)
The constructor.

Friends

- class [FI_Printer](#)

Additional Inherited Members

31.130.1 Detailed Description

Print support under MSWindows and Mac OS.

Class [Fl_System_Printer](#) is implemented only on the MSWindows and Mac OS platforms. It has no public constructor. Use [FI_Printer](#) instead that is cross-platform and has the same API.

31.130.2 Member Function Documentation

31.130.2.1 [class_name\(\)](#)

```
const char * Fl_System_Printer::class_name ( ) [inline], [virtual]
```

Returns the name of the class of this object.

Use of the [class_name\(\)](#) function is discouraged because it will be removed from future FLTK versions.

The class of an instance of an [Fl_Device](#) subclass can be checked with code such as:

```
if ( instance->class_name() == FI_Printer::class_id ) { ... }
```

Reimplemented from [FI_Paged_Device](#).

31.130.2.2 [end_job\(\)](#)

```
void Fl_System_Printer::end_job (
    void ) [virtual]
```

To be called at the end of a print job.

Reimplemented from [FI_Paged_Device](#).

31.130.2.3 [end_page\(\)](#)

```
int Fl_System_Printer::end_page (
    void ) [virtual]
```

To be called at the end of each page.

Returns

0 if OK, non-zero if any error.

Reimplemented from [FI_Paged_Device](#).

31.130.2.4 margins()

```
void Fl_System_Printer::margins (
    int * left,
    int * top,
    int * right,
    int * bottom ) [virtual]
```

Computes the dimensions of margins that lie between the printable page area and the full page. Values are in the same unit as that used by FLTK drawing functions. They are changed by [scale\(\)](#) calls.

Parameters

out	<i>left</i>	If non-null, *left is set to the left margin size.
out	<i>top</i>	If non-null, *top is set to the top margin size.
out	<i>right</i>	If non-null, *right is set to the right margin size.
out	<i>bottom</i>	If non-null, *bottom is set to the bottom margin size.

Reimplemented from [Fl_Paged_Device](#).

31.130.2.5 origin() [1/2]

```
void Fl_System_Printer::origin (
    int * x,
    int * y ) [virtual]
```

Computes the page coordinates of the current origin of graphics functions.

Parameters

out	<i>x</i>	If non-null, *x is set to the horizontal page offset of graphics origin.
out	<i>y</i>	Same as above, vertically.

Reimplemented from [Fl_Paged_Device](#).

31.130.2.6 origin() [2/2]

```
void Fl_System_Printer::origin (
    int x,
    int y ) [virtual]
```

Sets the position in page coordinates of the origin of graphics functions.

Arguments should be expressed relatively to the result of a previous [printable_rect\(\)](#) call. That is, `printable_rect(&w, &h); origin(w/2, 0);` sets the graphics origin at the top center of the page printable area. [Origin\(\)](#) calls are not affected by [rotate\(\)](#) calls. Successive [origin\(\)](#) calls don't combine their effects.

Parameters

in	<i>x</i>	Horizontal position in page coordinates of the desired origin of graphics functions.
in	<i>y</i>	Same as above, vertically.

Reimplemented from [Fl_Paged_Device](#).

31.130.2.7 printable_rect()

```
int Fl_System_Printer::printable_rect (
    int * w,
```

```
int * h ) [virtual]
```

Computes the width and height of the printable area of the page.

Values are in the same unit as that used by FLTK drawing functions, are unchanged by calls to [origin\(\)](#), but are changed by [scale\(\)](#) calls. Values account for the user-selected paper type and print orientation.

Returns

0 if OK, non-zero if any error

Reimplemented from [Fl_Paged_Device](#).

31.130.2.8 rotate()

```
void Fl_System_Printer::rotate (
    float angle ) [virtual]
```

Rotates the graphics operations relatively to paper.

The rotation is centered on the current graphics origin. Successive [rotate\(\)](#) calls don't combine their effects.

Parameters

<i>angle</i>	Rotation angle in counter-clockwise degrees.
--------------	--

Reimplemented from [Fl_Paged_Device](#).

31.130.2.9 scale()

```
void Fl_System_Printer::scale (
    float scale_x,
    float scale_y = 0. ) [virtual]
```

Changes the scaling of page coordinates.

This function also resets the origin of graphics functions at top left of printable page area. After a [scale\(\)](#) call, do a [printable_rect\(\)](#) call to get the new dimensions of the printable page area. Successive [scale\(\)](#) calls don't combine their effects.

Parameters

<i>scale</i> <i>_x</i>	Horizontal dimensions of plot are multiplied by this quantity.
<i>scale</i> <i>_y</i>	Same as above, vertically. The value 0. is equivalent to setting <code>scale_y = scale_x</code> . Thus, <code>scale(factor);</code> is equivalent to <code>scale(factor, factor);</code>

Reimplemented from [Fl_Paged_Device](#).

31.130.2.10 start_job()

```
int Fl_System_Printer::start_job (
    int pagecount,
    int * frompage = NULL,
    int * topage = NULL ) [virtual]
```

Starts a print job.

Parameters

in	<i>pagecount</i>	the total number of pages of the job (or 0 if you don't know the number of pages)
out	<i>frompage</i>	if non-null, *frompage is set to the first page the user wants printed
out	<i>topage</i>	if non-null, *topage is set to the last page the user wants printed

Returns

0 if OK, non-zero if any error

Reimplemented from [Fl_Paged_Device](#).

31.130.2.11 start_page()

```
int Fl_System_Printer::start_page (
    void ) [virtual]
```

Starts a new printed page.

The page coordinates are initially in points, i.e., 1/72 inch, and with origin at the top left of the printable page area.

Returns

0 if OK, non-zero if any error

Reimplemented from [Fl_Paged_Device](#).

31.130.2.12 translate()

```
void Fl_System_Printer::translate (
    int x,
    int y ) [virtual]
```

Translates the current graphics origin accounting for the current rotation.

This function is only useful after a [rotate\(\)](#) call. Each [translate\(\)](#) call must be matched by an [untranslate\(\)](#) call. Successive [translate\(\)](#) calls add up their effects.

Reimplemented from [Fl_Paged_Device](#).

31.130.2.13 untranslate()

```
void Fl_System_Printer::untranslate (
    void ) [virtual]
```

Undoes the effect of a previous [translate\(\)](#) call.

Reimplemented from [Fl_Paged_Device](#).

The documentation for this class was generated from the following files:

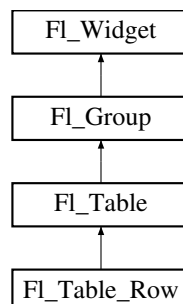
- [Fl_Printer.H](#)
- [Fl_Printer.cxx](#)

31.131 Fl_Table Class Reference

A table of widgets or other content.

```
#include <Fl_Table.H>
```

Inheritance diagram for `Fl_Table`:



Public Types

- enum `TableContext` {
`CONTEXT_NONE` = 0 , `CONTEXT_STARTPAGE` = 0x01 , `CONTEXT_ENDPAGE` = 0x02 , `CONTEXT_ROW_HEADER`
= 0x04 ,
`CONTEXT_COL_HEADER` = 0x08 , `CONTEXT_CELL` = 0x10 , `CONTEXT_TABLE` = 0x20 , `CONTEXT_RC_RESIZE`
= 0x40 }

The context bit flags for `FI_Table` related callbacks.

Public Member Functions

- void `add` (`FI_Widget` &wgt)
- void `add` (`FI_Widget` *wgt)
- `FI_Widget` *const * `array` ()
- void `begin` ()
- void `callback` (`FI_Widget` *, void *)
Callbacks will be called depending on the setting of `FI_Widget::when()`.
- int `callback_col` ()
Returns the current column the event occurred on.
- `TableContext` `callback_context` ()
Returns the current 'table context'.
- int `callback_row` ()
Returns the current row the event occurred on.
- `FI_Widget` * `child` (int n) const
Returns the child widget by an index.
- int `children` () const
Returns the number of children in the table.
- virtual void `clear` ()
Clears the table to zero rows (`rows(0)`), zero columns (`cols(0)`), and clears any widgets (`table->clear()`) that were added with `begin()/end()` or `add()/insert()/etc`.
- int `col_header` ()
Returns if column headers are enabled or not.
- void `col_header` (int flag)
Enable or disable column headers.
- `FI_Color` `col_header_color` ()
Gets the color for column headers.
- void `col_header_color` (`FI_Color` val)
Sets the color for column headers and redraws the table.
- int `col_header_height` ()
Gets the column header height.
- void `col_header_height` (int height)
Sets the height in pixels for column headers and redraws the table.
- int `col_position` ()
Returns the current column scroll position as a column number.
- void `col_position` (int col)
Sets the column scroll position to column 'col', and causes the screen to redraw.
- int `col_resize` ()
Returns if column resizing by the user is allowed.
- void `col_resize` (int flag)
Allows/disallows column resizing by the user.
- int `col_resize_min` ()
Returns the current column minimum resize value.
- void `col_resize_min` (int val)

- Sets the current column minimum resize value.*
- int **col_width** (int col)
 - Returns the current width of the specified column in pixels.*
- void **col_width** (int col, int width)
 - Sets the width of the specified column in pixels, and the table is redrawn.*
- void **col_width_all** (int width)
 - Convenience method to set the width of all columns to the same value, in pixels.*
- int **cols** ()
 - Get the number of columns in the table.*
- virtual void **cols** (int val)
 - Set the number of columns in the table and redraw.*
- void **do_callback** (TableContext context, int row, int col)
- void **draw** (void)
 - Draws the widget.*
- void **end** ()
- int **find** (const FI_Widget &wgt) const
- int **find** (const FI_Widget *wgt) const
- FI_Table (int X, int Y, int W, int H, const char *l=0)
 - The constructor for the FI_Table.*
- void **get_selection** (int &row_top, int &col_left, int &row_bot, int &col_right)
 - Gets the region of cells selected (highlighted).*
- void **init_sizes** ()
- void **insert** (FI_Widget &wgt, FI_Widget *w2)
- void **insert** (FI_Widget &wgt, int n)
- int **is_interactive_resize** ()
 - Returns 1 if someone is interactively resizing a row or column.*
- int **is_selected** (int r, int c)
 - See if the cell at row r and column c is selected.*
- int **move_cursor** (int R, int C)
- int **move_cursor** (int R, int C, int shiftselect)
- void **remove** (FI_Widget &wgt)
- void **resize** (int X, int Y, int W, int H)
 - Changes the size of the FI_Table, causing it to redraw.*
- int **row_header** ()
 - Returns if row headers are enabled or not.*
- void **row_header** (int flag)
 - Enables/disables showing the row headers.*
- FI_Color **row_header_color** ()
 - Returns the current row header color.*
- void **row_header_color** (FI_Color val)
 - Sets the row header color and causes the screen to redraw.*
- int **row_header_width** ()
 - Returns the current row header width (in pixels).*
- void **row_header_width** (int width)
 - Sets the row header width to n and causes the screen to redraw.*
- int **row_height** (int row)
 - Returns the current height of the specified row as a value in pixels.*
- void **row_height** (int row, int height)
 - Sets the height of the specified row in pixels, and the table is redrawn.*
- void **row_height_all** (int height)
 - Convenience method to set the height of all rows to the same value, in pixels.*

- int **row_position** ()
Returns the current row scroll position as a row number.
- void **row_position** (int row)
Sets the row scroll position to 'row', and causes the screen to redraw.
- int **row_resize** ()
Returns if row resizing by the user is allowed.
- void **row_resize** (int flag)
Allows/disallows row resizing by the user.
- int **row_resize_min** ()
Returns the current row minimum resize value.
- void **row_resize_min** (int val)
Sets the current row minimum resize value.
- int **rows** ()
Returns the number of rows in the table.
- virtual void **rows** (int val)
Sets the number of rows in the table, and the table is redrawn.
- int **scrollbar_size** () const
Gets the current size of the scrollbars' troughs, in pixels.
- void **scrollbar_size** (int newSize)
*Sets the pixel size of the scrollbars' troughs to *newSize*, in pixels.*
- void **set_selection** (int row_top, int col_left, int row_bot, int col_right)
Sets the region of cells to be selected (highlighted).
- int **tab_cell_nav** () const
Get state of table's 'Tab' key cell navigation flag.
- void **tab_cell_nav** (int val)
Flag to control if Tab navigates table cells or not.
- void **table_box** (FI_Boxtype val)
Sets the kind of box drawn around the data table, the default being FL_NO_BOX.
- FI_Boxtype **table_box** (void)
Returns the current box type used for the data table.
- int **top_row** ()
Returns the current top row shown in the table.
- void **top_row** (int row)
Sets which row should be at the top of the table, scrolling as necessary, and the table is redrawn.
- void **visible_cells** (int &r1, int &r2, int &c1, int &c2)
Returns the range of row and column numbers for all visible and partially visible cells in the table.
- void **when** (FI_When flags)
*The *FI_Widget::when()* function is used to set a group of flags, determining when the widget callback is called:*
- ~FI_Table ()
*The destructor for the *FI_Table*.*

Protected Types

- enum **ResizeFlag** {
RESIZE_NONE = 0 , **RESIZE_COL_LEFT** = 1 , **RESIZE_COL_RIGHT** = 2 , **RESIZE_ROW_ABOVE** = 3 ,
RESIZE_ROW_BELOW = 4 }

Protected Member Functions

- void **change_cursor** ([Fl_Cursor](#) newcursor)
- long **col_scroll_position** (int col)
- [TableContext](#) **cursor2rowcol** (int &R, int &C, ResizeFlag &resizeflag)
- void **damage_zone** (int r1, int c1, int r2, int c2, int r3=0, int c3=0)
- virtual void **draw_cell** ([TableContext](#) context, int R=0, int C=0, int X=0, int Y=0, int W=0, int H=0)
Subclass should override this method to handle drawing the cells.
- int **find_cell** ([TableContext](#) context, int R, int C, int &X, int &Y, int &W, int &H)
- void **get_bounds** ([TableContext](#) context, int &X, int &Y, int &W, int &H)
- int **handle** (int e)
Handles the specified event.
- int **is_fltk_container** ()
- void **recalc_dimensions** ()
- void **redraw_range** (int topRow, int botRow, int leftCol, int rightCol)
- int **row_col_clamp** ([TableContext](#) context, int &R, int &C)
- long **row_scroll_position** (int row)
- void **table_resized** ()
- void **table_scrolled** ()

Static Protected Member Functions

- static void **scroll_cb** ([Fl_Widget](#) *, void *)

Protected Attributes

- int **botrow**
- int **current_col**
- int **current_row**
- [Fl_Scrollbar](#) * **hscrollbar**
- int **leftcol**
- int **leftcol_scrollpos**
- int **rightcol**
- int **select_col**
- int **select_row**
- [Fl_Scroll](#) * **table**
- int **table_h**
- int **table_w**
- int **tih**
- int **tiw**
- int **tix**
- int **tiy**
- int **toh**
- int **toprow**
- int **toprow_scrollpos**
- int **tow**
- int **tox**
- int **toy**
- [Fl_Scrollbar](#) * **vscrollbar**
- int **wih**
- int **wiw**
- int **wix**
- int **wiy**

Additional Inherited Members

31.131.1 Detailed Description

A table of widgets or other content.

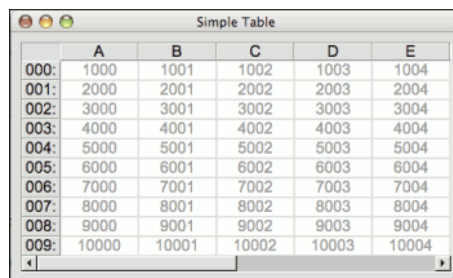
This is the base class for table widgets.

To be useful it must be subclassed and several virtual functions defined. Normally applications use widgets derived from this widget, and do not use this widget directly; this widget is usually too low level to be used directly by applications.

This widget does *not* handle the data in the table. The `draw_cell()` method must be overridden by a subclass to manage drawing the contents of the cells.

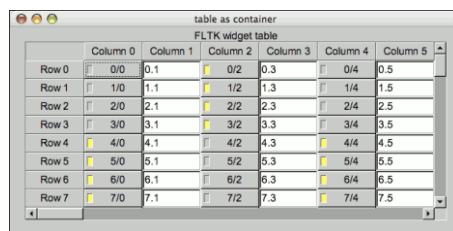
This widget can be used in several ways:

- As a custom widget; see examples/table-simple.cxx and test/table.cxx. Very optimal for even extremely large tables.
- As a table made up of a single FLTK widget instanced all over the table, simulating a numeric spreadsheet. See examples/table-spreadsheet.cxx and examples/table-spreadsheet-with-keyboard-nav.cxx. Optimal for large tables.
- As a regular container of FLTK widgets, one widget per cell. See examples/table-as-container.cxx. *Not* recommended for large tables.



	A	B	C	D	E
000:	1000	1001	1002	1003	1004
001:	2000	2001	2002	2003	2004
002:	3000	3001	3002	3003	3004
003:	4000	4001	4002	4003	4004
004:	5000	5001	5002	5003	5004
005:	6000	6001	6002	6003	6004
006:	7000	7001	7002	7003	7004
007:	8000	8001	8002	8003	8004
008:	9000	9001	9002	9003	9004
009:	10000	10001	10002	10003	10004

Figure 31.34 table-simple example



	Column 0	Column 1	Column 2	Column 3	Column 4	Column 5
Row 0	0/0	0.1	0/2	0.3	0/4	0.5
Row 1	1/0	1.1	1/2	1.3	1/4	1.5
Row 2	2/0	2.1	2/2	2.3	2/4	2.5
Row 3	3/0	3.1	3/2	3.3	3/4	3.5
Row 4	4/0	4.1	4/2	4.3	4/4	4.5
Row 5	5/0	5.1	5/2	5.3	5/4	5.5
Row 6	6/0	6.1	6/2	6.3	6/4	6.5
Row 7	7/0	7.1	7/2	7.3	7/4	7.5

Figure 31.35 table-as-container example

When acting as part of a custom widget, events on the cells and/or headings generate callbacks when they are clicked by the user. You control when events are generated based on the setting for `Fl_Table::when()`.

When acting as a container for FLTK widgets, the FLTK widgets maintain themselves. Although the `draw_cell()` method must be overridden, its contents can be very simple. See the `draw_cell()` code in examples/table-simple.cxx.

The following variables are available to classes deriving from `Fl_Table`:

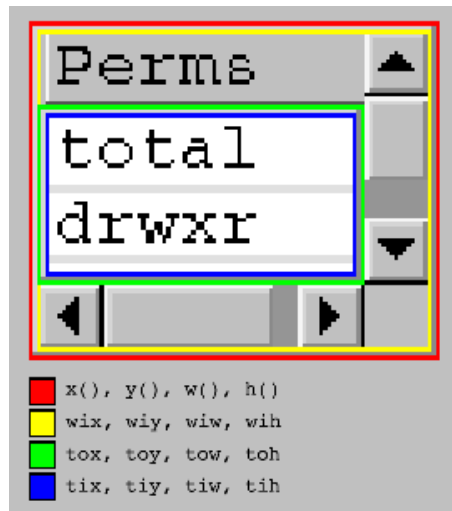


Figure 31.36 Fl_Table Dimensions

<code>x()/y()/w()/h()</code>	Fl_Table widget's outer dimension. The outer edge of the border of the Fl_Table. (Red in the diagram above)
<code>wix/wiy/wiw/wih</code>	Fl_Table widget's inner dimension. The inner edge of the border of the Fl_Table. eg. if the Fl_Table's <code>box()</code> is FL_NO_BOX, these values are the same as <code>x()/y()/w()/h()</code> . (Yellow in the diagram above)
<code>tox/toy/tow/toh</code>	The table's outer dimension. The outer edge of the border around the cells, but inside the row/col headings and scrollbars. (Green in the diagram above)
<code>tix/tiy/tiw/tih</code>	The table's inner dimension. The inner edge of the border around the cells, but inside the row/col headings and scrollbars. AKA the table's clip region. eg. if the <code>table_box()</code> is FL_↔_NO_BOX, these values are the same as <code>tox/toy/tow/toh</code> . (Blue in the diagram above)

CORE DEVELOPERS

- Greg Ercolano : 12/16/2002 - initial implementation 12/16/02. [Fl_Table](#), [Fl_Table_Row](#), docs.
- Jean-Marc Lienher : 02/22/2004 - added keyboard nav + mouse selection, and ported [Fl_Table](#) into fltk-utf8-1.1.4

OTHER CONTRIBUTORS

- Inspired by the Feb 2000 version of FLVW's Flvw_Table widget. Mucho thanks to those folks.
- Mister Satan : 04/07/2003 - MinGW porting mods, and singleinput.cxx; a cool [Fl_Input](#) oriented spreadsheet example
- Marek Paliwoda : 01/08/2003 - Porting mods for Borland
- Ori Berger : 03/16/2006 - Optimizations for >500k rows/cols

LICENSE

Greg added the following license to the original distribution of [Fl_Table](#). He kindly gave his permission to integrate [Fl_Table](#) and [Fl_Table_Row](#) into FLTK, allowing FLTK license to apply while his widgets are part of the library. If used on its own, this is the license that applies:

```
Fl_Table License
December 16, 2002
```

```
The Fl_Table library and included programs are provided under the terms
of the GNU Library General Public License (LGPL) with the following
exceptions:
```

1. Modifications to the Fl_Table configure script, config header file, and makefiles by themselves to support a specific platform do not constitute a modified or derivative work.

The authors do request that such modifications be contributed to the Fl_Table project - send all contributions to "erco at seriss dot com".

2. Widgets that are subclassed from Fl_Table widgets do not constitute a derivative work.

3. Static linking of applications and widgets to the Fl_Table library does not constitute a derivative work and does not require the author to provide source code for the application or widget, use the shared Fl_Table libraries, or link their applications or widgets against a user-supplied version of Fl_Table.

If you link the application or widget to a modified version of Fl_Table, then the changes to Fl_Table must be provided under the terms of the LGPL in sections 1, 2, and 4.

4. You do not have to provide a copy of the Fl_Table license with programs that are linked to the Fl_Table library, nor do you have to identify the Fl_Table license in your program or documentation as required by section 6 of the LGPL.

However, programs must still identify their use of Fl_Table. The following example statement can be included in user documentation to satisfy this requirement:

```
[program/widget] is based in part on the work of
the Fl_Table project http://seriss.com/people/erco/fttk/Fl\_Table/
```

31.131.2 Member Enumeration Documentation

31.131.2.1 TableContext

enum `Fl_Table::TableContext`

The context bit flags for `Fl_Table` related callbacks.

Used in `draw_cell()`, `callback()`, etc.

Enumerator

CONTEXT_NONE	no known context
CONTEXT_STARTPAGE	before a page is redrawn
CONTEXT_ENDPAGE	after a page is redrawn
CONTEXT_ROW_HEADER	in the row header
CONTEXT_COL_HEADER	in the col header
CONTEXT_CELL	in one of the cells
CONTEXT_TABLE	in a dead zone of table
CONTEXT_RC_RESIZE	column or row being resized

31.131.3 Constructor & Destructor Documentation

31.131.3.1 Fl_Table()

```
Fl_Table::Fl_Table (
    int X,
    int Y,
    int W,
    int H,
    const char * I = 0 )
```

The constructor for the [Fl_Table](#).

This creates an empty table with no rows or columns, with headers and row/column resize behavior disabled.

31.131.3.2 ~Fl_Table()

```
Fl_Table::~Fl_Table ( )
```

The destructor for the [Fl_Table](#).

Destroys the table and its associated widgets.

31.131.4 Member Function Documentation

31.131.4.1 callback()

```
void Fl_Table::callback (
    Fl_Widget * ,
    void * )
```

Callbacks will be called depending on the setting of [Fl_Widget::when\(\)](#).

Callback functions should use the following functions to determine the context/row/column:

- [Fl_Table::callback_row\(\)](#) returns current row
- [Fl_Table::callback_col\(\)](#) returns current column
- [Fl_Table::callback_context\(\)](#) returns current table context

[callback_row\(\)](#) and [callback_col\(\)](#) will be set to the row and column number the event occurred on. If someone clicked on a row header, `col` will be 0.

If someone clicked on a column header, `row` will be 0.

[callback_context\(\)](#) will return one of the following:

Fl_Table::CONTEXT_ROW_HEADER	Someone clicked on a row header. Excludes resizing.
Fl_Table::CONTEXT_COL_HEADER	Someone clicked on a column header. Excludes resizing.
Fl_Table::CONTEXT_CELL	Someone clicked on a cell. To receive callbacks for FL_RELEASE events, you must set <code>when(FL_WHEN_RELEASE)</code> .
Fl_Table::CONTEXT_RC_RESIZE	Someone is resizing rows/columns either interactively, or via the col_width() or row_height() API. Use is_interactive_resize() to determine interactive resizing. If resizing a column, <code>R=0</code> and <code>C=column</code> being resized. If resizing a row, <code>C=0</code> and <code>R=row</code> being resized. NOTE: To receive resize events, you must set <code>when(FL_WHEN↵_CHANGED)</code> .

```
class MyTable : public Fl_Table {
[..]
private:
    // Handle events that happen on the table
    void event_callback2() {
        int R = callback_row(),           // row where event occurred
            C = callback_col();           // column where event occurred
        TableContext context = callback_context(); // which part of table
        fprintf(stderr, "callback: Row=%d Col=%d Context=%d Event=%d\n",
            R, C, (int)context, (int)Fl::event());
    }
};
```

```

}
// Actual static callback
static void event_callback(Fl_Widget*, void* data) {
    MyTable *o = (MyTable*)data;
    o->event_callback2();
}

public:
// Constructor
MyTable() {
    [...]
    table.callback(&event_callback, (void*)this); // setup callback
    table.when(FL_WHEN_CHANGED|FL_WHEN_RELEASE); // when to call it
}
};

```

31.131.4.2 callback_col()

```
int Fl_Table::callback_col ( ) [inline]
```

Returns the current column the event occurred on.

This function should only be used from within the user's callback function.

31.131.4.3 callback_context()

```
TableContext Fl_Table::callback_context ( ) [inline]
```

Returns the current 'table context'.

This function should only be used from within the user's callback function.

31.131.4.4 callback_row()

```
int Fl_Table::callback_row ( ) [inline]
```

Returns the current row the event occurred on.

This function should only be used from within the user's callback function.

31.131.4.5 child()

```
Fl_Widget * Fl_Table::child (
    int n ) const [inline]
```

Returns the child widget by an index.

When using the [Fl_Table](#) as a container for FLTK widgets, this method returns the widget pointer from the internal array of widgets in the container.

Typically used in loops, eg:

```
for ( int i=0; i<children(); i++ ) {
    Fl_Widget *w = child(i);
    [...]
}

```

31.131.4.6 children()

```
int Fl_Table::children ( ) const [inline]
```

Returns the number of children in the table.

When using the [Fl_Table](#) as a container for FLTK widgets, this method returns how many child widgets the table has.

See also

[child\(int\)](#)

31.131.4.7 clear()

```
virtual void Fl_Table::clear ( ) [inline], [virtual]
```

Clears the table to zero rows (`rows(0)`), zero columns (`cols(0)`), and clears any widgets (`table->clear()`) that were added with `begin()/end()` or `add()/insert()/etc.`

See also

[rows\(int\)](#), [cols\(int\)](#)

Reimplemented in [Fl_Table_Row](#).

31.131.4.8 col_header()

```
void Fl_Table::col_header (
    int flag ) [inline]
```

Enable or disable column headers.
If changed, the table is redrawn.

31.131.4.9 col_resize()

```
void Fl_Table::col_resize (
    int flag ) [inline]
```

Allows/disallows column resizing by the user.

1=allow interactive resizing, 0=disallow interactive resizing. Since interactive resizing is done via the column headers, [col_header\(\)](#) must also be enabled to allow resizing.

31.131.4.10 col_resize_min()

```
void Fl_Table::col_resize_min (
    int val ) [inline]
```

Sets the current column minimum resize value.

This is used to prevent the user from interactively resizing any column to be smaller than 'pixels'. Must be a value ≥ 1 .

31.131.4.11 col_width()

```
void Fl_Table::col_width (
    int col,
    int width )
```

Sets the width of the specified column in pixels, and the table is redrawn.

[callback\(\)](#) will be invoked with CONTEXT_RC_RESIZE if the column's width was actually changed, and [when\(\)](#) is FL_WHEN_CHANGED.

31.131.4.12 col_width_all()

```
void Fl_Table::col_width_all (
    int width ) [inline]
```

Convenience method to set the width of all columns to the same value, in pixels.

The screen is redrawn.

31.131.4.13 draw()

```
void Fl_Table::draw (
    void ) [virtual]
```

Draws the widget.

Never call this function directly. FLTK will schedule redrawing whenever needed. If your widget must be redrawn as soon as possible, call [redraw\(\)](#) instead.

Override this function to draw your own widgets.

If you ever need to call another widget's draw method *from within your own [draw\(\)](#) method*, e.g. for an embedded scrollbar, you can do it (because [draw\(\)](#) is virtual) like this:

```
Fl_Widget *s = &scroll; // scroll is an embedded Fl_Scrollbar
s->draw(); // calls Fl_Scrollbar::draw()
```

Reimplemented from [Fl_Group](#).

31.131.4.14 draw_cell()

```
virtual void Fl_Table::draw_cell (
    TableContext context,
    int R = 0,
    int C = 0,
    int X = 0,
    int Y = 0,
    int W = 0,
    int H = 0 ) [inline], [protected], [virtual]
```

Subclass should override this method to handle drawing the cells.

This method will be called whenever the table is redrawn, once per cell.

Only cells that are completely (or partially) visible will be told to draw.

context will be one of the following:

<code>Fl_Table::CONTEXT_STARTPAGE</code>	When table, or parts of the table, are about to be redrawn. Use to initialize static data, such as font selections. R/C will be zero, X/Y/W/H will be the dimensions of the table's entire data area. (Useful for locking a database before accessing; see also visible_cells())
<code>Fl_Table::CONTEXT_ENDPAGE</code>	When table has completed being redrawn. R/C will be zero, X/Y/W/H dimensions of table's data area. (Useful for unlocking a database after accessing)
<code>Fl_Table::CONTEXT_ROW_HEADER</code>	Whenever a row header cell needs to be drawn. R will be the row number of the header being redrawn, C will be zero, X/Y/W/H will be the fltk drawing area of the row header in the window
<code>Fl_Table::CONTEXT_COL_HEADER</code>	Whenever a column header cell needs to be drawn. R will be zero, C will be the column number of the header being redrawn, X/Y/W/H will be the fltk drawing area of the column header in the window
<code>Fl_Table::CONTEXT_CELL</code>	Whenever a data cell in the table needs to be drawn. R/C will be the row/column of the cell to be drawn, X/Y/W/H will be the fltk drawing area of the cell in the window
<code>Fl_Table::CONTEXT_RC_RESIZE</code>	Whenever table or row/column is resized or scrolled, either interactively or via col_width() or row_height() . R/C/X/Y/W/H will all be zero. Useful for fltk containers that need to resize or move the child fltk widgets.

row and col will be set to the row and column number of the cell being drawn. In the case of row headers, col will be 0. In the case of column headers, row will be 0.

x/y/w/h will be the position and dimensions of where the cell should be drawn.

In the case of custom widgets, a minimal [draw_cell\(\)](#) override might look like the following. With custom widgets it is up to the caller to handle drawing everything within the dimensions of the cell, including handling the selection color. Note all clipping must be handled as well; this allows drawing outside the dimensions of the cell if so desired for 'custom effects'.

```
// This is called whenever Fl_Table wants you to draw a cell
void MyTable::draw_cell(TableContext context, int R=0, int C=0, int X=0, int Y=0, int W=0, int H=0) {
    static char s[40];
    sprintf(s, "%d/%d", R, C); // text for each cell
    switch ( context ) {
        case CONTEXT_STARTPAGE: // Fl_Table telling us it's starting to draw page
            fl_font(FL_HELVETICA, 16);
            return;
        case CONTEXT_ROW_HEADER: // Fl_Table telling us to draw row/col headers
        case CONTEXT_COL_HEADER:
            fl_push_clip(X, Y, W, H);
            {
```



```

    fl_draw_box(FL_THIN_UP_BOX, X, Y, W, H, color());
    fl_color(FL_BLACK);
    fl_draw(s, X, Y, W, H, FL_ALIGN_CENTER);
}
fl_pop_clip();
return;

case CONTEXT_CELL: // Fl_Table telling us to draw cells
    fl_push_clip(X, Y, W, H);
    {
        // BG COLOR
        fl_color( row_selected(R) ? selection_color() : FL_WHITE);
        fl_rectf(X, Y, W, H);

        // TEXT
        fl_color(FL_BLACK);
        fl_draw(s, X, Y, W, H, FL_ALIGN_CENTER);

        // BORDER
        fl_color(FL_LIGHT2);
        fl_rect(X, Y, W, H);
    }
    fl_pop_clip();
    return;

default:
    return;
}
//NOTREACHED
}

```

31.131.4.15 get_selection()

```

void Fl_Table::get_selection (
    int & row_top,
    int & col_left,
    int & row_bot,
    int & col_right )

```

Gets the region of cells selected (highlighted).

Parameters

in	<i>row_top</i>	Returns the top row of selection area
in	<i>col_left</i>	Returns the left column of selection area
in	<i>row_bot</i>	Returns the bottom row of selection area
in	<i>col_right</i>	Returns the right column of selection area

31.131.4.16 handle()

```

int Fl_Table::handle (
    int event ) [protected], [virtual]

```

Handles the specified event.

You normally don't call this method directly, but instead let FLTK do it when the user interacts with the widget.

When implemented in a widget, this function must return 0 if the widget does not use the event or 1 otherwise.

Most of the time, you want to call the inherited [handle\(\)](#) method in your overridden method so that you don't short-circuit events that you don't handle. In this last case you should return the callee retval.

Parameters

in	<i>event</i>	the kind of event received
----	--------------	----------------------------

Return values

0	if the event was not used or understood
---	---

Return values

1	if the event was used and can be deleted
---	--

See also

[Fl_Event](#)

Reimplemented from [Fl_Group](#).

Reimplemented in [Fl_Table_Row](#).

31.131.4.17 is_interactive_resize()

```
int Fl_Table::is_interactive_resize ( ) [inline]
```

Returns 1 if someone is interactively resizing a row or column.

You can currently call this only from within your [callback\(\)](#).

31.131.4.18 is_selected()

```
int Fl_Table::is_selected (
    int r,
    int c )
```

See if the cell at row *r* and column *c* is selected.

Returns

1 if the cell is selected, 0 if not.

31.131.4.19 resize()

```
void Fl_Table::resize (
    int X,
    int Y,
    int W,
    int H ) [virtual]
```

Changes the size of the [Fl_Table](#), causing it to redraw.

Reimplemented from [Fl_Group](#).

31.131.4.20 row_header()

```
void Fl_Table::row_header (
    int flag ) [inline]
```

Enables/disables showing the row headers.

1=enabled, 0=disabled. If changed, the table is redrawn.

31.131.4.21 row_height()

```
void Fl_Table::row_height (
    int row,
    int height )
```

Sets the height of the specified row in pixels, and the table is redrawn.

[callback\(\)](#) will be invoked with `CONTEXT_RC_RESIZE` if the row's height was actually changed, and [when\(\)](#) is `FL_WHEN_CHANGED`.

31.131.4.22 row_height_all()

```
void Fl_Table::row_height_all (
    int height ) [inline]
```

Convenience method to set the height of all rows to the same value, in pixels. The screen is redrawn.

31.131.4.23 row_resize()

```
void Fl_Table::row_resize (
    int flag ) [inline]
```

Allows/disallows row resizing by the user.

1=allow interactive resizing, 0=disallow interactive resizing. Since interactive resizing is done via the row headers, [row_header\(\)](#) must also be enabled to allow resizing.

31.131.4.24 row_resize_min()

```
void Fl_Table::row_resize_min (
    int val ) [inline]
```

Sets the current row minimum resize value.

This is used to prevent the user from interactively resizing any row to be smaller than 'pixels'. Must be a value ≥ 1 .

31.131.4.25 rows()

```
void Fl_Table::rows (
    int val ) [virtual]
```

Sets the number of rows in the table, and the table is redrawn.

Reimplemented in [Fl_Table_Row](#).

31.131.4.26 scrollbar_size() [1/2]

```
int Fl_Table::scrollbar_size ( ) const [inline]
```

Gets the current size of the scrollbars' troughs, in pixels.

If this value is zero (default), this widget will use the [Fl::scrollbar_size\(\)](#) value as the scrollbar's width.

Returns

Scrollbar size in pixels, or 0 if the global [Fl::scrollbar_size\(\)](#) is being used.

See also

[Fl::scrollbar_size\(int\)](#)

31.131.4.27 scrollbar_size() [2/2]

```
void Fl_Table::scrollbar_size (
    int newSize ) [inline]
```

Sets the pixel size of the scrollbars' troughs to `newSize`, in pixels.

Normally you should not need this method, and should use [Fl::scrollbar_size\(int\)](#) instead to manage the size of ALL your widgets' scrollbars. This ensures your application has a consistent UI, is the default behavior, and is normally what you want.

Only use THIS method if you really need to override the global scrollbar size. The need for this should be rare.

Setting `newSize` to the special value of 0 causes the widget to track the global [Fl::scrollbar_size\(\)](#), which is the default.

Parameters

in	<i>newSize</i>	Sets the scrollbar size in pixels. If 0 (default), scrollbar size tracks the global Fl::scrollbar_size()
----	----------------	---

See also

[Fl::scrollbar_size\(\)](#)

31.131.4.28 set_selection()

```
void Fl_Table::set_selection (
    int row_top,
    int col_left,
    int row_bot,
    int col_right )
```

Sets the region of cells to be selected (highlighted).

So for instance, `set_selection(0,0,0,0)` selects the top/left cell in the table. And `set_selection(0,0,1,1)` selects the four cells in rows 0 and 1, column 0 and 1.

Parameters

in	<i>row_top</i>	Top row of selection area
in	<i>col_left</i>	Left column of selection area
in	<i>row_bot</i>	Bottom row of selection area
in	<i>col_right</i>	Right column of selection area

31.131.4.29 tab_cell_nav() [1/2]

```
int Fl_Table::tab_cell_nav ( ) const [inline]
```

Get state of table's 'Tab' key cell navigation flag.

Returns

1 if Tab configured to navigate cells in table
0 to navigate widget focus (default)

See also

[tab_cell_nav\(int\)](#)

31.131.4.30 tab_cell_nav() [2/2]

```
void Fl_Table::tab_cell_nav (
    int val ) [inline]
```

Flag to control if Tab navigates table cells or not.

If on, Tab key navigates table cells. If off, Tab key navigates fltk widget focus. (default)

As of fltk 1.3, the default behavior of the Tab key is to navigate focus off of the current widget, and on to the next one. But in some applications, it's useful for Tab to be used to navigate cells in the [Fl_Table](#).

Parameters

in	val	If val is 1, Tab key navigates cells in table, not fltk widgets. If val is 0, Tab key will advance focus to the next fltk widget (default), and does not navigate cells in table.
----	-----	--

31.131.4.31 table_box()

```
void Fl_Table::table_box (
    Fl_Boxtype val ) [inline]
```

Sets the kind of box drawn around the data table, the default being FL_NO_BOX. Changing this value will cause the table to redraw.

31.131.4.32 top_row() [1/2]

```
int Fl_Table::top_row ( ) [inline]
```

Returns the current top row shown in the table. This row may be partially obscured.

31.131.4.33 top_row() [2/2]

```
void Fl_Table::top_row (
    int row ) [inline]
```

Sets which row should be at the top of the table, scrolling as necessary, and the table is redrawn. If the table cannot be scrolled that far, it is scrolled as far as possible.

31.131.4.34 visible_cells()

```
void Fl_Table::visible_cells (
    int & r1,
    int & r2,
    int & c1,
    int & c2 ) [inline]
```

Returns the range of row and column numbers for all visible and partially visible cells in the table. These values can be used e.g. by your [draw_cell\(\)](#) routine during CONTEXT_STARTPAGE to figure out what cells are about to be redrawn for the purposes of locking the data from a database before it's drawn.

```

leftcol      rightcol
:
:
toprow .. :-----:
|         |         |
|  V I S I B L E  |
|         |         |
|   T A B L E   |
|         |         |
botrow .. :-----:

```

e.g. in a table where the visible rows are 5-20, and the visible columns are 100-120, then those variables would be:

- toprow = 5
- botrow = 20
- leftcol = 100
- rightcol = 120

31.131.4.35 when()

```
void Fl_Table::when (
    Fl_When flags )
```

The [Fl_Widget::when\(\)](#) function is used to set a group of flags, determining when the widget callback is called:

FL_WHEN_CHANGED	<code>callback()</code> will be called when rows or columns are resized (interactively or via <code>col_width()</code> or <code>row_height()</code>), passing <code>CONTEXT_RC_RESIZE</code> via <code>callback_context()</code> .
FL_WHEN_RELEASE	<code>callback()</code> will be called during <code>FL_RELEASE</code> events, such as when someone releases a mouse button somewhere on the table.

The `callback()` routine is sent a `TableContext` that indicates the context the event occurred in, such as in a cell, in a header, or elsewhere on the table. When an event occurs in a cell or header, `callback_row()` and `callback_col()` can be used to determine the row and column. The callback can also look at the regular fltk event values (ie. `Fl::event()` and `Fl::event_button()`) to determine what kind of event is occurring.

The documentation for this class was generated from the following files:

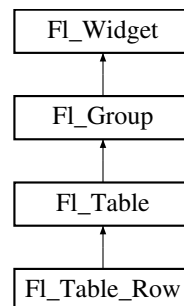
- `Fl_Table.H`
- `Fl_Table.cxx`

31.132 Fl_Table_Row Class Reference

A table with row selection capabilities.

```
#include <Fl_Table_Row.H>
```

Inheritance diagram for `Fl_Table_Row`:



Public Types

- enum `TableRowSelectMode` { `SELECT_NONE` , `SELECT_SINGLE` , `SELECT_MULTI` }

Public Member Functions

- void `clear` ()

Clears the table to zero rows (`rows(0)`), zero columns (`cols(0)`), and clears any widgets (`table->clear()`) that were added with `begin()/end()` or `add()/insert()/etc`.
- `Fl_Table_Row` (int X, int Y, int W, int H, const char *l=0)

The constructor for the `Fl_Table_Row`.
- int `row_selected` (int row)

Checks to see if 'row' is selected.
- int `rows` ()
- void `rows` (int val)

Sets the number of rows in the table, and the table is redrawn.
- void `select_all_rows` (int flag=1)

This convenience function changes the selection state for all rows based on 'flag'.
- int `select_row` (int row, int flag=1)

Changes the selection state for 'row', depending on the value of 'flag'.
- `TableRowSelectMode type` () const
- void `type` (TableRowSelectMode val)

Sets the table selection mode.
- `~Fl_Table_Row` ()

The destructor for the `Fl_Table_Row`.

Protected Member Functions

- int `find_cell` ([TableContext](#) context, int R, int C, int &X, int &Y, int &W, int &H)
- int `handle` (int event)

Handles the specified event.

Additional Inherited Members

31.132.1 Detailed Description

A table with row selection capabilities.

This class implements a simple table with the ability to select rows. This widget is similar to an [FI_Browser](#) with columns. Most methods of importance will be found in the [FI_Table](#) widget, such as [FI_Table::rows\(\)](#) and [FI_Table::cols\(\)](#).

To be useful it must be subclassed and at minimum the [draw_cell\(\)](#) method must be overridden to provide the content of the cells. This widget does *not* manage the cell's data content; it is up to the parent class's [draw_cell\(\)](#) method override to provide this.

Events on the cells and/or headings generate callbacks when they are clicked by the user. You control when events are generated based on the values you supply for [FI_Table::when\(\)](#).

31.132.2 Constructor & Destructor Documentation

31.132.2.1 FI_Table_Row()

```
FI_Table_Row::FI_Table_Row (
    int X,
    int Y,
    int W,
    int H,
    const char * I = 0 ) [inline]
```

The constructor for the [FI_Table_Row](#).

This creates an empty table with no rows or columns, with headers and row/column resize behavior disabled.

31.132.2.2 ~FI_Table_Row()

```
FI_Table_Row::~FI_Table_Row ( ) [inline]
```

The destructor for the [FI_Table_Row](#).

Destroys the table and its associated widgets.

31.132.3 Member Function Documentation

31.132.3.1 clear()

```
void FI_Table_Row::clear ( ) [inline], [virtual]
```

Clears the table to zero rows (`rows(0)`), zero columns (`cols(0)`), and clears any widgets (`table->clear()`) that were added with `begin()/end()` or `add()/insert()/etc.`

See also

[rows\(int\)](#), [cols\(int\)](#)

Reimplemented from [FI_Table](#).

31.132.3.2 handle()

```
int Fl_Table_Row::handle (
    int event ) [protected], [virtual]
```

Handles the specified event.

You normally don't call this method directly, but instead let FLTK do it when the user interacts with the widget. When implemented in a widget, this function must return 0 if the widget does not use the event or 1 otherwise. Most of the time, you want to call the inherited [handle\(\)](#) method in your overridden method so that you don't short-circuit events that you don't handle. In this last case you should return the callee retval.

Parameters

in	<i>event</i>	the kind of event received
----	--------------	----------------------------

Return values

0	if the event was not used or understood
1	if the event was used and can be deleted

See also

[Fl_Event](#)

Reimplemented from [Fl_Table](#).

31.132.3.3 row_selected()

```
int Fl_Table_Row::row_selected (
    int row )
```

Checks to see if 'row' is selected.

Returns 1 if selected, 0 if not. You can change the selection of a row by clicking on it, or by using [select_row\(row, flag\)](#)

31.132.3.4 rows()

```
void Fl_Table_Row::rows (
    int val ) [virtual]
```

Sets the number of rows in the table, and the table is redrawn.

Reimplemented from [Fl_Table](#).

31.132.3.5 select_all_rows()

```
void Fl_Table_Row::select_all_rows (
    int flag = 1 )
```

This convenience function changes the selection state for *all* rows based on 'flag'. 0=deselect, 1=select, 2=toggle existing state.

31.132.3.6 select_row()

```
int Fl_Table_Row::select_row (
    int row,
    int flag = 1 )
```

Changes the selection state for 'row', depending on the value of 'flag'. 0=deselected, 1=select, 2=toggle existing state.

31.132.3.7 type()

```
void Fl_Table_Row::type (
    TableRowSelectMode val )
```

Sets the table selection mode.

- `Fl_Table_Row::SELECT_NONE` - No selection allowed
- `Fl_Table_Row::SELECT_SINGLE` - Only single rows can be selected
- `Fl_Table_Row::SELECT_MULTI` - Multiple rows can be selected

The documentation for this class was generated from the following files:

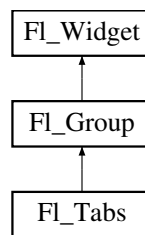
- `Fl_Table_Row.H`
- `Fl_Table_Row.cxx`

31.133 Fl_Tabs Class Reference

The `Fl_Tabs` widget is the "file card tabs" interface that allows you to put lots and lots of buttons and switches in a panel, as popularized by many toolkits.

```
#include <Fl_Tabs.H>
```

Inheritance diagram for `Fl_Tabs`:



Public Member Functions

- void `client_area` (int &rx, int &ry, int &rw, int &rh, int tabh=0)
Returns the position and size available to be used by its children.
- `Fl_Tabs` (int, int, int, int, const char *=0)
Creates a new `Fl_Tabs` widget using the given position, size, and label string.
- int `handle` (int)
Handles the specified event.
- `Fl_Widget *` `push` () const
Returns the tab group for the tab the user has currently down-clicked on and remains over until `FL_RELEASE`.
- int `push` (`Fl_Widget *`)
This is called by the tab widget's `handle()` method to set the tab group widget the user last `FL_PUSH`'ed on.
- `Fl_Widget *` `value` ()
Gets the currently visible widget/tab.
- int `value` (`Fl_Widget *`)
Sets the widget to become the current visible widget/tab.
- `Fl_Widget *` `which` (int event_x, int event_y)
Return the widget of the tab the user clicked on at `event_x` / `event_y`.

Protected Member Functions

- void `draw` ()
Draws the widget.
- void `redraw_tabs` ()

Additional Inherited Members

31.133.1 Detailed Description

The `Fl_Tabs` widget is the "file card tabs" interface that allows you to put lots and lots of buttons and switches in a panel, as popularized by many toolkits.

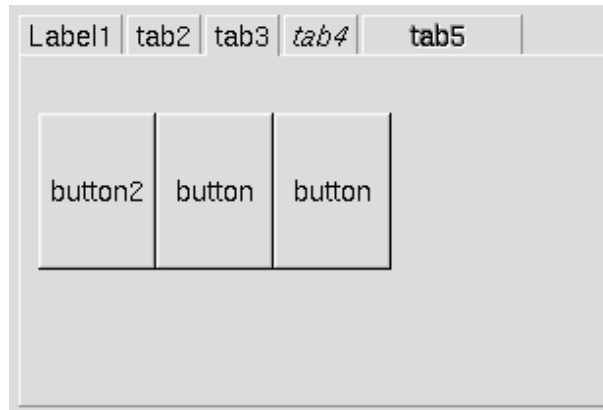


Figure 31.37 `Fl_Tabs`

Clicking the tab makes a child `visible()` by calling `show()` on it, and all other children are made invisible by calling `hide()` on them. Usually the children are `Fl_Group` widgets containing several widgets themselves.

Each child makes a card, and its `label()` is printed on the card tab, including the label font and style. The selection color of that child is used to color the tab, while the color of the child determines the background color of the pane.

The size of the tabs is controlled by the bounding box of the children (there should be some space between the children and the edge of the `Fl_Tabs`), and the tabs may be placed "inverted" on the bottom - this is determined by which gap is larger. It is easiest to lay this out in fluid, using the fluid browser to select each child group and resize them until the tabs look the way you want them to.

The background area behind and to the right of the tabs is "transparent", exposing the background detail of the parent. The value of `Fl_Tabs::box()` does not affect this area. So if `Fl_Tabs` is resized by itself without the parent, force the appropriate parent (visible behind the tabs) to `redraw()` to prevent artifacts.

See "Resizing Caveats" below on how to keep tab heights constant. See "Callback's Use Of `when()`" on how to control the details of how clicks invoke the `callback()`.

A typical use of the `Fl_Tabs` widget:

```
// Typical use of Fl_Tabs
Fl_Tabs *tabs = new Fl_Tabs(10,10,300,200);
{
    Fl_Group *grp1 = new Fl_Group(20,30,280,170,"Tab1");
    {
        ..widgets that go in tab#1..
    }
    grp1->end();
    Fl_Group *grp2 = new Fl_Group(20,30,280,170,"Tab2");
    {
        ..widgets that go in tab#2..
    }
    grp2->end();
}
tabs->end();
```

Default Appearance

The appearance of each "tab" is taken from the `label()` and `color()` of the child group corresponding to that "tab" and panel. Where the "tabs" appear depends on the position and size of the child groups that make up the panels within the `Fl_Tab`, i.e. whether there is more space above or below them. The height of the "tabs" depends on how much free space is available.

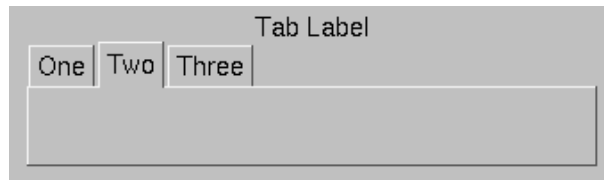


Figure 31.38 Fl_Tabs Default Appearance

Highlighting The Selected Tab

The selected "tab" can be highlighted further by setting the `selection_color()` of the Fl_Tab itself, e.g.

```
..
tabs = new Fl_Tabs(..);
tabs->selection_color(FL_DARK3);
..
```

The result of the above looks like:

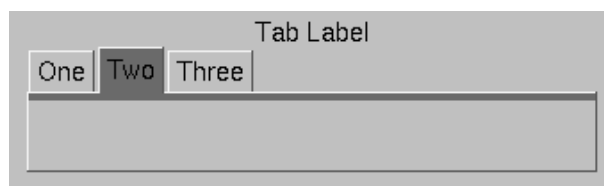


Figure 31.39 Highlighting the selected tab

Uniform Tab and Panel Appearance

In order to have uniform tab and panel appearance, not only must the `color()` and `selection_color()` for each child group be set, but also the `selection_color()` of the Fl_Tab itself any time a new "tab" is selected. This can be achieved within the Fl_Tab callback, e.g.

```
void MyTabCallback(Fl_Widget *w, void*) {
    Fl_Tabs *tabs = (Fl_Tabs*)w;
    // When tab changed, make sure it has same color as its group
    tabs->selection_color( (tab->value()->color() ) );
}
..
int main(..) {
    // Define tabs widget
    tabs = new Fl_Tabs(..);
    tabs->callback(MyTabCallback);
    // Create three tabs each colored differently
    grp1 = new Fl_Group(.. "One");
    grp1->color(9);
    grp1->selection_color(9);
    grp1->end();
    grp2 = new Fl_Group(.. "Two");
    grp2->color(10);
    grp2->selection_color(10);
    grp2->end();
    grp3 = new Fl_Group(.. "Three");
    grp3->color(14);
    grp3->selection_color(14);
    grp3->end();
    ..
    // Make sure default tab has same color as its group
    tabs->selection_color( (tab->value()->color() ) );
    ..
    return Fl::run();
}
```

The result of the above looks like:

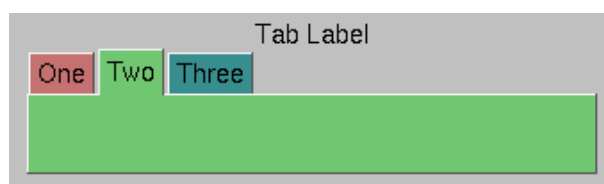


Figure 31.40 Fl_Tabs with uniform colors

Resizing Caveats

When `Fl_Tabs` is resized vertically, the default behavior scales the tab's height as well as its children. To keep the tab height constant during resizing, set the tab widget's `resizable()` to one of the tab's child groups, i.e.

```
tabs = new Fl_Tabs(..);
grp1 = new Fl_Group(..);
..
grp2 = new Fl_Group(..);
..
tabs->end();
tabs->resizable(grp1);           // keeps tab height constant
```

Callback's Use Of `when()`

As of FLTK 1.3.3, `Fl_Tabs()` supports the following flags for `when()`:

- `FL_WHEN_NEVER` – callback never invoked (all flags off)
- `FL_WHEN_CHANGED` – if flag set, invokes callback when a tab has been changed (on click or keyboard navigation)
- `FL_WHEN_NOT_CHANGED` – if flag set, invokes callback when the tabs remain unchanged (on click or keyboard navigation)
- `FL_WHEN_RELEASE` – if flag set, invokes callback on RELEASE of mouse button or keyboard navigation

Notes:

1. The above flags can be logically OR-ed (`|`) or added (`+`) to combine behaviors.
2. The default value for `when()` is `FL_WHEN_RELEASE` (inherited from `Fl_Widget`).
3. If `FL_WHEN_RELEASE` is the *only* flag specified, the behavior will be as if `(FL_WHEN_RELEASE|FL_WHEN_CHANGED)` was specified.
4. The value of `changed()` will be valid during the callback.
5. If both `FL_WHEN_CHANGED` and `FL_WHEN_NOT_CHANGED` are specified, the callback is invoked whether the tab has been changed or not. The `changed()` method can be used to determine the cause.
6. `FL_WHEN_NOT_CHANGED` can happen if someone clicks on an already selected tab, or if a keyboard navigation attempt results in no change to the tabs, such as using the arrow keys while at the left or right end of the tabs.

31.133.2 Constructor & Destructor Documentation

31.133.2.1 `Fl_Tabs()`

```
Fl_Tabs::Fl_Tabs (
    int X,
    int Y,
    int W,
    int H,
    const char * l = 0 )
```

Creates a new `Fl_Tabs` widget using the given position, size, and label string.

The default boxtype is `FL_THIN_UP_BOX`.

Use `add(Fl_Widget*)` to add each child, which are usually `Fl_Group` widgets. The children should be sized to stay away from the top or bottom edge of the `Fl_Tabs` widget, which is where the tabs will be drawn.

All children of `Fl_Tabs` should have the same size and exactly fit on top of each other. They should only leave space above or below where the tabs will go, but not on the sides. If the first child of `Fl_Tabs` is set to `"resizable()"`, the riders will not resize when the tabs are resized.

The destructor *also deletes all the children*. This allows a whole tree to be deleted at once, without having to keep a pointer to all the children in the user code. A kludge has been done so the `Fl_Tabs` and all of its children can be automatic (local) variables, but you must declare the `Fl_Tabs` widget *first* so that it is destroyed last.

31.133.3 Member Function Documentation

31.133.3.1 client_area()

```
void Fl_Tabs::client_area (
    int & rx,
    int & ry,
    int & rw,
    int & rh,
    int tabh = 0 )
```

Returns the position and size available to be used by its children.

If there isn't any child yet the `tabh` parameter will be used to calculate the return values. This assumes that the children's labelsize is the same as the `Fl_Tabs`' labelsize and adds a small border.

If there are already children, the values of `child(0)` are returned, and `tabh` is ignored.

Note

Children should always use the same positions and sizes.

`tabh` can be one of

- 0: calculate label size, tabs on top
- -1: calculate label size, tabs on bottom
- > 0: use given `tabh` value, tabs on top (height = `tabh`)
- < -1: use given `tabh` value, tabs on bottom (height = `-tabh`)

Parameters

in	<i>tabh</i>	position and optional height of tabs (see above)
out	<i>rx,ry,rw,rh</i>	(x,y,w,h) of client area for children

Since

FLTK 1.3.0

31.133.3.2 draw()

```
void Fl_Tabs::draw ( ) [protected], [virtual]
```

Draws the widget.

Never call this function directly. FLTK will schedule redrawing whenever needed. If your widget must be redrawn as soon as possible, call `redraw()` instead.

Override this function to draw your own widgets.

If you ever need to call another widget's draw method *from within your own `draw()` method*, e.g. for an embedded scrollbar, you can do it (because `draw()` is virtual) like this:

```
Fl_Widget *s = &scroll;           // scroll is an embedded Fl_Scrollbar
s->draw();                          // calls Fl_Scrollbar::draw()
```

Reimplemented from `Fl_Group`.

31.133.3.3 handle()

```
int Fl_Tabs::handle (
    int event ) [virtual]
```

Handles the specified event.

You normally don't call this method directly, but instead let FLTK do it when the user interacts with the widget. When implemented in a widget, this function must return 0 if the widget does not use the event or 1 otherwise. Most of the time, you want to call the inherited [handle\(\)](#) method in your overridden method so that you don't short-circuit events that you don't handle. In this last case you should return the callee retval.

Parameters

<code>in</code>	<code>event</code>	the kind of event received
-----------------	--------------------	----------------------------

Return values

<code>0</code>	if the event was not used or understood
<code>1</code>	if the event was used and can be deleted

See also

[Fl_Event](#)

Reimplemented from [Fl_Group](#).

31.133.3.4 push() [1/2]

```
Fl_Widget * Fl_Tabs::push ( ) const [inline]
```

Returns the tab group for the tab the user has currently down-clicked on and remains over until FL_RELEASE. Otherwise, returns NULL.

While the user is down-clicked on a tab, the return value is the tab group for that tab. But as soon as the user releases, or drags off the tab with the button still down, the return value will be NULL.

See also

[push\(Fl_Widget*\)](#).

31.133.3.5 push() [2/2]

```
int Fl_Tabs::push (
    Fl_Widget * o )
```

This is called by the tab widget's [handle\(\)](#) method to set the tab group widget the user last FL_PUSH'ed on. Set back to zero on FL_RELEASE.

As of this writing, the value is mainly used by [draw_tab\(\)](#) to determine whether or not to draw a 'down' box for the tab when it's clicked, and to turn it off if the user drags off it.

See also

[push\(\)](#).

31.133.3.6 value() [1/2]

```
Fl_Widget * Fl_Tabs::value ( )
```

Gets the currently visible widget/tab.

The [value\(\)](#) is the first visible child (or the last child if none are visible) and this also hides any other children. This allows the tabs to be deleted, moved to other groups, and [show\(\)/hide\(\)](#) called without it screwing up.

31.133.3.7 value() [2/2]

```
int Fl_Tabs::value (
    Fl_Widget * newvalue )
```

Sets the widget to become the current visible widget/tab.

Setting the value hides all other children, and makes this one visible, if it is really a child.

Returns

- 1 if there was a change (new value different from previous),
- 0 if there was no change (new value already set)

31.133.3.8 which()

```
Fl_Widget * Fl_Tabs::which (
    int event_x,
    int event_y )
```

Return the widget of the tab the user clicked on at `event_x / event_y`.

This is used for event handling (clicks) and by fluid to pick tabs.

Returns

- The child widget of the tab the user clicked on, or
- 0 if there are no children or if the event is outside of the tabs area.

The documentation for this class was generated from the following files:

- Fl_Tabs.H
- Fl_Tabs.cxx

31.134 Fl_Text_Buffer Class Reference

This class manages Unicode text displayed in one or more [Fl_Text_Display](#) widgets.

```
#include <Fl_Text_Buffer.H>
```

Public Member Functions

- void [add_modify_callback](#) (Fl_Text_Modify_Cb bufModifiedCB, void *cbArg)
Adds a callback function that is called whenever the text buffer is modified.
- void [add_predelete_callback](#) (Fl_Text_Predelete_Cb bufPredelCB, void *cbArg)
Adds a callback routine to be called before text is deleted from the buffer.
- char * [address](#) (int pos)
Convert a byte offset in buffer into a memory address.
- const char * [address](#) (int pos) const
Convert a byte offset in buffer into a memory address.
- void [append](#) (const char *t)
Appends the text string to the end of the buffer.
- int [appendfile](#) (const char *file, int buflen=128 *1024)
Appends the named file to the end of the buffer.
- char [byte_at](#) (int pos) const
Returns the raw byte at the specified position pos in the buffer.
- void [call_modify_callbacks](#) ()
Calls all modify callbacks that have been registered using the [add_modify_callback\(\)](#) method.
- void [call_predelete_callbacks](#) ()
Calls the stored pre-delete callback procedure(s) for this buffer to update the changed area(s) on the screen and any other listeners.

- void **canUndo** (char flag=1)
Lets the undo system know if we can undo changes.
- unsigned int **char_at** (int pos) const
Returns the character at the specified position `pos` in the buffer.
- void **copy** (FI_Text_Buffer *fromBuf, int fromStart, int fromEnd, int toPos)
Copies text from another FI_Text_Buffer to this one.
- int **count_displayed_characters** (int lineStartPos, int targetPos) const
Count the number of displayed characters between buffer position `lineStartPos` and `targetPos`.
- int **count_lines** (int startPos, int endPos) const
Counts the number of newlines between `startPos` and `endPos` in buffer.
- int **findchar_backward** (int startPos, unsigned int searchChar, int *foundPos) const
Search backwards in buffer `buf` for character `searchChar`, starting with the character before `startPos`, returning the result in `foundPos`.
- int **findchar_forward** (int startPos, unsigned searchChar, int *foundPos) const
Finds the next occurrence of the specified character.
- FI_Text_Buffer (int requestedSize=0, int preferredGapSize=1024)
Create an empty text buffer of a pre-determined size.
- int **highlight** ()
Returns the highlighted text.
- void **highlight** (int start, int end)
Highlights the specified text within the buffer.
- int **highlight_position** (int *start, int *end)
Highlights the specified text between `start` and `end` within the buffer.
- const FI_Text_Selection * **highlight_selection** () const
Returns the current highlight selection.
- char * **highlight_text** ()
Returns the highlighted text.
- void **insert** (int pos, const char *text)
Inserts null-terminated string `text` at position `pos`.
- int **insertfile** (const char *file, int pos, int buflen=128 *1024)
Inserts a file at the specified position.
- int **length** () const
Returns the number of bytes in the buffer.
- int **line_end** (int pos) const
Finds and returns the position of the end of the line containing position `pos` (which is either a pointer to the newline character ending the line or a pointer to one character beyond the end of the buffer).
- int **line_start** (int pos) const
Returns the position of the start of the line containing position `pos`.
- char * **line_text** (int pos) const
Returns the text from the entire line containing the specified character position.
- int **loadfile** (const char *file, int buflen=128 *1024)
Loads a text file into the buffer.
- int **next_char** (int ix) const
Returns the index of the next character.
- int **next_char_clipped** (int ix) const
- int **outputfile** (const char *file, int start, int end, int buflen=128 *1024)
Writes the specified portions of the text buffer to a file.
- int **prev_char** (int ix) const
Returns the index of the previous character.
- int **prev_char_clipped** (int ix) const
- FI_Text_Selection * **primary_selection** ()

- Returns the primary selection.*

 - const `FI_Text_Selection * primary_selection ()` const

Returns the primary selection.
- void `remove (int start, int end)`

Deletes a range of characters in the buffer.
- void `remove_modify_callback (FI_Text_Modify_Cb bufModifiedCB, void *cbArg)`

Removes a modify callback.
- void `remove_predelete_callback (FI_Text_Predelete_Cb preDelCB, void *cbArg)`

Removes a callback routine `bufPreDeleteCB` associated with argument `cbArg` to be called before text is deleted from the buffer.
- void `remove_secondary_selection ()`

Removes the text from the buffer corresponding to the secondary text selection object.
- void `remove_selection ()`

Removes the text in the primary selection.
- void `replace (int start, int end, const char *text)`

Deletes the characters between `start` and `end`, and inserts the null-terminated string `text` in their place in the buffer.
- void `replace_secondary_selection (const char *text)`

Replaces the text from the buffer corresponding to the secondary text selection object with the new string `text`.
- void `replace_selection (const char *text)`

Replaces the text in the primary selection.
- int `rewind_lines (int startPos, int nLines)`

Finds and returns the position of the first character of the line `nLines` backwards from `startPos` (not counting the character pointed to by `startPos` if that is a newline) in the buffer.
- int `savefile (const char *file, int buflen=128 *1024)`

Saves a text file from the current buffer.
- int `search_backward (int startPos, const char *searchString, int *foundPos, int matchCase=0)` const

Search backwards in buffer for string `searchString`, starting with the character at `startPos`, returning the result in `foundPos`.
- int `search_forward (int startPos, const char *searchString, int *foundPos, int matchCase=0)` const

Search forwards in buffer for string `searchString`, starting with the character `startPos`, and returning the result in `foundPos`.
- void `secondary_select (int start, int end)`

Selects a range of characters in the secondary selection.
- int `secondary_selected ()`

Returns a non-zero value if text has been selected in the secondary text selection, 0 otherwise.
- const `FI_Text_Selection * secondary_selection ()` const

Returns the secondary selection.
- int `secondary_selection_position (int *start, int *end)`

Returns the current selection in the secondary text selection object.
- char * `secondary_selection_text ()`

Returns the text in the secondary selection.
- void `secondary_unselect ()`

Clears any selection in the secondary text selection object.
- void `select (int start, int end)`

Selects a range of characters in the buffer.
- int `selected ()` const

Returns a non-zero value if text has been selected, 0 otherwise.
- int `selection_position (int *start, int *end)`

Gets the selection position.
- char * `selection_text ()`

Returns the currently selected text.

- int [skip_displayed_characters](#) (int lineStartPos, int nChars)
Count forward from buffer position `startPos` in displayed characters.
- int **skip_lines** (int startPos, int nLines)
Finds the first character of the line `nLines` forward from `startPos` in the buffer and returns its position.
- int [tab_distance](#) () const
Gets the tab width.
- void **tab_distance** (int tabDist)
Set the hardware tab distance (width) used by all displays for this buffer, and used in computing offsets for rectangular selection operations.
- char * [text](#) () const
Get a copy of the entire contents of the text buffer.
- void [text](#) (const char *text)
Replaces the entire contents of the text buffer.
- char * [text_range](#) (int start, int end) const
Get a copy of a part of the text buffer.
- int **undo** (int *cp=0)
Undo text modification according to the undo variables or insert text from the undo buffer.
- void **unhighlight** ()
Unhighlights text in the buffer.
- void **unselect** ()
Cancel any previous selection on the primary text selection object.
- int **utf8_align** (int) const
Align an index into the buffer to the current or previous UTF-8 boundary.
- int [word_end](#) (int pos) const
Returns the position corresponding to the end of the word.
- int [word_start](#) (int pos) const
Returns the position corresponding to the start of the word.
- ~**FI_Text_Buffer** ()
Frees a text buffer.

Public Attributes

- int **input_file_was_transcoded**
true if the loaded file has been transcoded to UTF-8.
- void(* [transcoding_warning_action](#))(FI_Text_Buffer *)
Pointer to a function called after reading a non UTF-8 encoded file.

Static Public Attributes

- static const char * [file_encoding_warning_message](#)
This message may be displayed using the `fl_alert()` function when a file which was not UTF-8 encoded is input.

Protected Member Functions

- void **call_modify_callbacks** (int pos, int nDeleted, int nInserted, int nRestyled, const char *deletedText) const
Calls the stored modify callback procedure(s) for this buffer to update the changed area(s) on the screen and any other listeners.
- void **call_predelete_callbacks** (int pos, int nDeleted) const
Calls the stored pre-delete callback procedure(s) for this buffer to update the changed area(s) on the screen and any other listeners.
- int [insert_](#) (int pos, const char *text)
Internal (non-redisplaying) version of [insert\(\)](#).

- void **move_gap** (int pos)
Move the gap to start at a new position.
- void **realloc_with_gap** (int newGapStart, int newGapLen)
Reallocates the text storage in the buffer to have a gap starting at `newGapStart` and a gap size of `newGapLen`, preserving the buffer's current contents.
- void **redisplay_selection** (FI_Text_Selection *oldSelection, FI_Text_Selection *newSelection) const
Calls the stored redisplay procedure(s) for this buffer to update the screen for a change in a selection.
- void **remove_** (int start, int end)
Internal (non-redisplaying) version of `remove()`.
- void **remove_selection_** (FI_Text_Selection *sel)
Removes the text from the buffer corresponding to `sel`.
- void **replace_selection_** (FI_Text_Selection *sel, const char *text)
Replaces the `text` in selection `sel`.
- char * **selection_text_** (FI_Text_Selection *sel) const
- void **update_selections** (int pos, int nDeleted, int nInserted)
Updates all of the selections in the buffer for changes in the buffer's text.

Protected Attributes

- char * **mBuf**
allocated memory where the text is stored
- char **mCanUndo**
if this buffer is used for attributes, it must not do any undo calls
- void ** **mCbArgs**
caller arguments for modifyProcs above
- int **mCursorPosHint**
hint for reasonable cursor position after a buffer modification operation
- int **mGapEnd**
points to the first character after the gap
- int **mGapStart**
points to the first character of the gap
- FI_Text_Selection **mHighlight**
highlighted areas
- int **mLength**
length of the text in the buffer (the length of the buffer itself must be calculated: `gapEnd - gapStart + length`)
- FI_Text_Modify_Cb * **mModifyProcs**
procedures to call when buffer is modified to redisplay contents
- int **mNModifyProcs**
number of modify-redisplay procs attached
- int **mNPredeleteProcs**
number of pre-delete procs attached
- void ** **mPredeleteCbArgs**
caller argument for pre-delete proc above
- FI_Text_Predelete_Cb * **mPredeleteProcs**
procedure to call before text is deleted from the buffer; at most one is supported.
- int **mPreferredGapSize**
the default allocation for the text gap is 1024 bytes and should only be increased if frequent and large changes in buffer size are expected
- FI_Text_Selection **mPrimary**
highlighted areas
- FI_Text_Selection **mSecondary**
highlighted areas
- int **mTabDist**
equiv.

31.134.1 Detailed Description

This class manages Unicode text displayed in one or more [Fl_Text_Display](#) widgets.

All text in [Fl_Text_Buffer](#) must be encoded in UTF-8. All indices used in the function calls must be aligned to the start of a UTF-8 sequence. All indices and pointers returned will be aligned. All functions that return a single character will return that in an unsigned int in UCS-4 encoding.

The [Fl_Text_Buffer](#) class is used by the [Fl_Text_Display](#) and [Fl_Text_Editor](#) to manage complex text data and is based upon the excellent NEdit text editor engine - see <http://www.nedit.org/>.

31.134.2 Constructor & Destructor Documentation

31.134.2.1 Fl_Text_Buffer()

```
Fl_Text_Buffer::Fl_Text_Buffer (
    int requestedSize = 0,
    int preferredGapSize = 1024 )
```

Create an empty text buffer of a pre-determined size.

Parameters

<i>requestedSize</i>	use this to avoid unnecessary re-allocation if you know exactly how much the buffer will need to hold
<i>preferredGapSize</i>	Initial size for the buffer gap (empty space in the buffer where text might be inserted if the user is typing sequential characters)

31.134.3 Member Function Documentation

31.134.3.1 add_modify_callback()

```
void Fl_Text_Buffer::add_modify_callback (
    Fl_Text_Modify_Cb bufModifiedCB,
    void * cbArg )
```

Adds a callback function that is called whenever the text buffer is modified.

The callback function is declared as follows:

```
typedef void (*Fl_Text_Modify_Cb)(int pos, int nInserted, int nDeleted,
    int nRestyled, const char* deletedText,
    void* cbArg);
```

31.134.3.2 address() [1/2]

```
char * Fl_Text_Buffer::address (
    int pos ) [inline]
```

Convert a byte offset in buffer into a memory address.

Parameters

<i>pos</i>	byte offset into buffer
------------	-------------------------

Returns

byte offset converted to a memory address

31.134.3.3 address() [2/2]

```
const char * Fl_Text_Buffer::address (
    int pos ) const [inline]
```

Convert a byte offset in buffer into a memory address.

Parameters

<i>pos</i>	byte offset into buffer
------------	-------------------------

Returns

byte offset converted to a memory address

31.134.3.4 append()

```
void Fl_Text_Buffer::append (
    const char * t ) [inline]
```

Appends the text string to the end of the buffer.

Parameters

<i>t</i>	UTF-8 encoded and nul terminated text
----------	---------------------------------------

31.134.3.5 appendfile()

```
int Fl_Text_Buffer::appendfile (
    const char * file,
    int buflen = 128*1024 ) [inline]
```

Appends the named file to the end of the buffer.

See also [insertfile\(\)](#).

31.134.3.6 byte_at()

```
char Fl_Text_Buffer::byte_at (
    int pos ) const
```

Returns the raw byte at the specified position *pos* in the buffer.

Positions start at 0.

Parameters

<i>pos</i>	byte offset into buffer
------------	-------------------------

Returns

unencoded raw byte

31.134.3.7 char_at()

```
unsigned int Fl_Text_Buffer::char_at (
    int pos ) const
```

Returns the character at the specified position *pos* in the buffer.

Positions start at 0.

Parameters

<i>pos</i>	byte offset into buffer, <i>pos</i> must be at a UTF-8 character boundary
------------	---

Returns

Unicode UCS-4 encoded character

31.134.3.8 copy()

```
void Fl_Text_Buffer::copy (
    Fl_Text_Buffer * fromBuf,
    int fromStart,
    int fromEnd,
    int toPos )
```

Copies text from another [Fl_Text_Buffer](#) to this one.

Parameters

<i>fromBuf</i>	source text buffer, may be the same as this
<i>fromStart</i>	byte offset into buffer
<i>fromEnd</i>	byte offset into buffer
<i>toPos</i>	destination byte offset into buffer

31.134.3.9 count_displayed_characters()

```
int Fl_Text_Buffer::count_displayed_characters (
    int lineStartPos,
    int targetPos ) const
```

Count the number of displayed characters between buffer position *lineStartPos* and *targetPos*. Displayed characters are the characters shown on the screen to represent characters in the buffer, where tabs and control characters are expanded.

31.134.3.10 count_lines()

```
int Fl_Text_Buffer::count_lines (
    int startPos,
    int endPos ) const
```

Counts the number of newlines between *startPos* and *endPos* in buffer. The character at position *endPos* is not counted.

31.134.3.11 findchar_backward()

```
int Fl_Text_Buffer::findchar_backward (
    int startPos,
    unsigned int searchChar,
    int * foundPos ) const
```

Search backwards in buffer *buf* for character *searchChar*, starting with the character *before* *startPos*, returning the result in *foundPos*.

Returns 1 if found, 0 if not. The difference between this and [search_backward\(\)](#) is that it's optimized for single characters. The overall performance of the text widget is dependent on its ability to count lines quickly, hence searching for a single character: newline.

Parameters

<i>startPos</i>	byte offset to start position
<i>searchChar</i>	UCS-4 character that we want to find
<i>foundPos</i>	byte offset where the character was found

Returns

1 if found, 0 if not

31.134.3.12 findchar_forward()

```
int Fl_Text_Buffer::findchar_forward (
    int startPos,
    unsigned searchChar,
    int * foundPos ) const
```

Finds the next occurrence of the specified character.

Search forwards in buffer for character *searchChar*, starting with the character *startPos*, and returning the result in *foundPos*. Returns 1 if found, 0 if not. The difference between this and [search_forward\(\)](#) is that it's optimized for single characters. The overall performance of the text widget is dependent on its ability to count lines quickly, hence searching for a single character: `newline`.

Parameters

<i>startPos</i>	byte offset to start position
<i>searchChar</i>	UCS-4 character that we want to find
<i>foundPos</i>	byte offset where the character was found

Returns

1 if found, 0 if not

31.134.3.13 highlight()

```
int Fl_Text_Buffer::highlight ( ) [inline]
```

Returns the highlighted text.

When you are done with the text, free it using the `free()` function.

31.134.3.14 highlight_text()

```
char * Fl_Text_Buffer::highlight_text ( )
```

Returns the highlighted text.

When you are done with the text, free it using the `free()` function.

31.134.3.15 insert()

```
void Fl_Text_Buffer::insert (
    int pos,
    const char * text )
```

Inserts null-terminated string *text* at position *pos*.

Parameters

<i>pos</i>	insertion position as byte offset (must be UTF-8 character aligned)
<i>text</i>	UTF-8 encoded and nul terminated text

31.134.3.16 insert_()

```
int Fl_Text_Buffer::insert_ (
    int pos,
    const char * text ) [protected]
```

Internal (non-redisplaying) version of [insert\(\)](#).

Returns the length of text inserted (this is just `strlen(text)`, however this calculation can be expensive and the length will be required by any caller who will continue on to call `redisplay`). `pos` must be contiguous with the existing text in the buffer (i.e. not past the end).

Returns

the number of bytes inserted

31.134.3.17 insertfile()

```
int Fl_Text_Buffer::insertfile (
    const char * file,
    int pos,
    int buflen = 128*1024 )
```

Inserts a file at the specified position.

Returns

- 0 on success
- non-zero on error (`strerror()` contains reason)
- 1 indicates open for read failed (no data loaded)
- 2 indicates error occurred while reading data (data was partially loaded)

File can be UTF-8 or CP1252 encoded. If the input file is not UTF-8 encoded, the `Fl_Text_Buffer` widget will contain data transcoded to UTF-8. By default, the message `Fl_Text_Buffer::file_encoding_warning_message` will warn the user about this.

See also

[input_file_was_transcoded](#) and [transcoding_warning_action](#).

31.134.3.18 length()

```
int Fl_Text_Buffer::length ( ) const [inline]
```

Returns the number of bytes in the buffer.

Returns

size of text in bytes

31.134.3.19 line_end()

```
int Fl_Text_Buffer::line_end (
    int pos ) const
```

Finds and returns the position of the end of the line containing position `pos` (which is either a pointer to the newline character ending the line or a pointer to one character beyond the end of the buffer).

Parameters

<i>pos</i>	byte index into buffer
------------	------------------------

Returns

byte offset to line end

31.134.3.20 line_start()

```
int Fl_Text_Buffer::line_start (  
    int pos ) const
```

Returns the position of the start of the line containing position *pos*.

Parameters

<i>pos</i>	byte index into buffer
------------	------------------------

Returns

byte offset to line start

31.134.3.21 line_text()

```
char * Fl_Text_Buffer::line_text (  
    int pos ) const
```

Returns the text from the entire line containing the specified character position. When you are done with the text, free it using the `free()` function.

Parameters

<i>pos</i>	byte index into buffer
------------	------------------------

Returns

copy of UTF-8 text, must be free'd

31.134.3.22 loadfile()

```
int Fl_Text_Buffer::loadfile (  
    const char * file,  
    int buflen = 128*1024 ) [inline]
```

Loads a text file into the buffer.

See also [insertfile\(\)](#).

31.134.3.23 next_char()

```
int Fl_Text_Buffer::next_char (  
    int ix ) const
```

Returns the index of the next character.

Parameters

<i>ix</i>	index to the current character
-----------	--------------------------------

31.134.3.24 outputfile()

```
int Fl_Text_Buffer::outputfile (
    const char * file,
    int start,
    int end,
    int buflen = 128*1024 )
```

Writes the specified portions of the text buffer to a file.

Returns

- 0 on success
- non-zero on error (strerror() contains reason)
- 1 indicates open for write failed (no data saved)
- 2 indicates error occurred while writing data (data was partially saved)

See also

[savefile\(const char *file, int buflen\)](#)

31.134.3.25 prev_char()

```
int Fl_Text_Buffer::prev_char (
    int ix ) const
```

Returns the index of the previous character.

Parameters

<i>ix</i>	index to the current character
-----------	--------------------------------

31.134.3.26 remove()

```
void Fl_Text_Buffer::remove (
    int start,
    int end )
```

Deletes a range of characters in the buffer.

Parameters

<i>start</i>	byte offset to first character to be removed
<i>end</i>	byte offset to character after last character to be removed

31.134.3.27 remove_()

```
void Fl_Text_Buffer::remove_ (
    int start,
```

```
int end ) [protected]
```

Internal (non-redisplaying) version of [remove\(\)](#).

Removes the contents of the buffer between `start` and `end` (and moves the gap to the site of the delete).

31.134.3.28 `replace()`

```
void Fl_Text_Buffer::replace (
    int start,
    int end,
    const char * text )
```

Deletes the characters between `start` and `end`, and inserts the null-terminated string `text` in their place in the buffer.

Parameters

<i>start</i>	byte offset to first character to be removed and new insert position
<i>end</i>	byte offset to character after last character to be removed
<i>text</i>	UTF-8 encoded and nul terminated text

31.134.3.29 `rewind_lines()`

```
int Fl_Text_Buffer::rewind_lines (
    int startPos,
    int nLines )
```

Finds and returns the position of the first character of the line `nLines` backwards from `startPos` (not counting the character pointed to by `startpos` if that is a newline) in the buffer.

`nLines == 0` means find the beginning of the line.

31.134.3.30 `savefile()`

```
int Fl_Text_Buffer::savefile (
    const char * file,
    int buflen = 128*1024 ) [inline]
```

Saves a text file from the current buffer.

Returns

- 0 on success
- non-zero on error (`strerror()` contains reason)
- 1 indicates open for write failed (no data saved)
- 2 indicates error occurred while writing data (data was partially saved)

See also

[outfile\(const char *file, int start, int end, int buflen\)](#)

31.134.3.31 `search_backward()`

```
int Fl_Text_Buffer::search_backward (
    int startPos,
    const char * searchString,
    int * foundPos,
    int matchCase = 0 ) const
```

Search backwards in buffer for string `searchString`, starting with the character *at* `startPos`, returning the result in `foundPos`.

Returns 1 if found, 0 if not.

Parameters

<i>startPos</i>	byte offset to start position
<i>searchString</i>	UTF-8 string that we want to find
<i>foundPos</i>	byte offset where the string was found
<i>matchCase</i>	if set, match character case

Returns

1 if found, 0 if not

31.134.3.32 search_forward()

```
int Fl_Text_Buffer::search_forward (
    int startPos,
    const char * searchString,
    int * foundPos,
    int matchCase = 0 ) const
```

Search forwards in buffer for string `searchString`, starting with the character `startPos`, and returning the result in `foundPos`.

Returns 1 if found, 0 if not.

Parameters

<i>startPos</i>	byte offset to start position
<i>searchString</i>	UTF-8 string that we want to find
<i>foundPos</i>	byte offset where the string was found
<i>matchCase</i>	if set, match character case

Returns

1 if found, 0 if not

31.134.3.33 secondary_selection_text()

```
char * Fl_Text_Buffer::secondary_selection_text ( )
```

Returns the text in the secondary selection.

When you are done with the text, free it using the `free()` function.

31.134.3.34 selection_text()

```
char * Fl_Text_Buffer::selection_text ( )
```

Returns the currently selected text.

When you are done with the text, free it using the `free()` function.

31.134.3.35 skip_displayed_characters()

```
int Fl_Text_Buffer::skip_displayed_characters (
    int lineStartPos,
    int nChars )
```

Count forward from buffer position `startPos` in displayed characters.

Displayed characters are the characters shown on the screen to represent characters in the buffer, where tabs and control characters are expanded.

Parameters

<i>lineStartPos</i>	byte offset into buffer
<i>nChars</i>	number of bytes that are sent to the display

Returns

byte offset in input after all output bytes are sent

31.134.3.36 tab_distance()

```
int Fl_Text_Buffer::tab_distance ( ) const [inline]
```

Gets the tab width.

The tab width is measured in characters. The pixel position is calculated using an average character width.

31.134.3.37 text() [1/2]

```
char * Fl_Text_Buffer::text ( ) const
```

Get a copy of the entire contents of the text buffer.

Memory is allocated to contain the returned string, which the caller must free.

Returns

newly allocated text buffer - must be free'd, text is UTF-8

31.134.3.38 text() [2/2]

```
void Fl_Text_Buffer::text (
    const char * text )
```

Replaces the entire contents of the text buffer.

Parameters

<i>text</i>	Text must be valid UTF-8. If null, an empty string is substituted.
-------------	--

31.134.3.39 text_range()

```
char * Fl_Text_Buffer::text_range (
    int start,
    int end ) const
```

Get a copy of a part of the text buffer.

Return a copy of the text between *start* and *end* character positions from text buffer *buf*. Positions start at 0, and the range does not include the character pointed to by *end*. When you are done with the text, free it using the `free()` function.

Parameters

<i>start</i>	byte offset to first character
<i>end</i>	byte offset after last character in range

Returns

newly allocated text buffer - must be free'd, text is UTF-8

31.134.3.40 word_end()

```
int Fl_Text_Buffer::word_end (
    int pos ) const
```

Returns the position corresponding to the end of the word.

Parameters

<i>pos</i>	byte index into buffer
------------	------------------------

Returns

byte offset to word end

31.134.3.41 word_start()

```
int Fl_Text_Buffer::word_start (
    int pos ) const
```

Returns the position corresponding to the start of the word.

Parameters

<i>pos</i>	byte index into buffer
------------	------------------------

Returns

byte offset to word start

31.134.4 Member Data Documentation**31.134.4.1 file_encoding_warning_message**

```
const char * Fl_Text_Buffer::file_encoding_warning_message [static]
```

Initial value:

```
=
"Displayed text contains the UTF-8 transcoding\n"
"of the input file which was not UTF-8 encoded.\n"
"Some changes may have occurred."
```

This message may be displayed using the [fl_alert\(\)](#) function when a file which was not UTF-8 encoded is input.

31.134.4.2 mTabDist

```
int Fl_Text_Buffer::mTabDist [protected]
equiv.
```

number of characters in a tab

31.134.4.3 transcoding_warning_action

```
void(* Fl_Text_Buffer::transcoding_warning_action) (Fl_Text_Buffer *)
```

Pointer to a function called after reading a non UTF-8 encoded file.

This function is called after reading a file if the file content was transcoded to UTF-8. Its default implementation calls `fl_alert()` with the text of `file_encoding_warning_message`. No warning message is displayed if this pointer is set to NULL. Use `input_file_was_transcoded` to be informed if file input required transcoding to UTF-8.

The documentation for this class was generated from the following files:

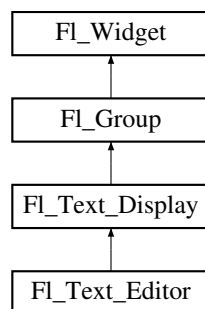
- `Fl_Text_Buffer.H`
- `Fl_Text_Buffer.cxx`

31.135 Fl_Text_Display Class Reference

Rich text display widget.

```
#include <Fl_Text_Display.H>
```

Inheritance diagram for Fl_Text_Display:



Classes

- struct [Style_Table_Entry](#)

This structure associates the color, font, and font size of a string to draw with an attribute mask matching attr.

Public Types

- enum { `NORMAL_CURSOR`, `CARET_CURSOR`, `DIM_CURSOR`, `BLOCK_CURSOR`, `HEAVY_CURSOR`, `SIMPLE_CURSOR` }
text display cursor shapes enumeration
- enum { `CURSOR_POS`, `CHARACTER_POS` }
the character position is the left edge of a character, whereas the cursor is thought to be between the centers of two consecutive characters.
- enum { `DRAG_NONE` = -2, `DRAG_START_DND` = -1, `DRAG_CHAR` = 0, `DRAG_WORD` = 1, `DRAG_LINE` = 2 }
drag types - they match `Fl::event_clicks()` so that single clicking to start a collection selects by character, double clicking selects by word and triple clicking selects by line.
- enum { `WRAP_NONE`, `WRAP_AT_COLUMN`, `WRAP_AT_PIXEL`, `WRAP_AT_BOUNDS` }
wrap types - used in `wrap_mode()`
- typedef void(* `Unfinished_Style_Cb`) (int, void *)

Public Member Functions

- `Fl_Text_Buffer * buffer () const`
Gets the current text buffer associated with the text widget.
- void `buffer (Fl_Text_Buffer &buf)`
Sets the current text buffer associated with the text widget.
- void `buffer (Fl_Text_Buffer *buf)`

- Attach a text buffer to display, replacing the current buffer (if any)*

 - double `col_to_x` (double col) const

Convert a column number into an x pixel position.
 - int `count_lines` (int start, int end, bool start_pos_is_line_start) const

Count the number of lines between two positions.
 - `FI_Color` `cursor_color` () const

Gets the text cursor color.
 - void `cursor_color` (`FI_Color` n)

Sets the text cursor color.
 - void `cursor_style` (int style)

Sets the text cursor style.
 - `FI_Text_Display` (int X, int Y, int W, int H, const char *l=0)

Creates a new text display widget.
 - virtual int `handle` (int e)

Event handling.
 - void `hide_cursor` ()

Hides the text cursor.
 - void `highlight_data` (`FI_Text_Buffer` *styleBuffer, const `Style_Table_Entry` *styleTable, int nStyles, char unfinishedStyle, Unfinished_Style_Cb unfinishedHighlightCB, void *cbArg)

Attach (or remove) highlight information in text display and redisplay.
 - int `in_selection` (int x, int y) const

Check if a pixel position is within the primary selection.
 - void `insert` (const char *text)

Inserts "text" at the current cursor location.
 - int `insert_position` () const

Gets the position of the text insertion cursor for text display.
 - void `insert_position` (int newPos)

Sets the position of the text insertion cursor for text display.
 - int `line_end` (int startPos, bool startPosIsLineStart) const

Returns the end of a line.
 - int `line_start` (int pos) const

Return the beginning of a line.
 - `FI_Align` `linenumber_align` () const

Returns the alignment used for line numbers (if enabled).
 - void `linenumber_align` (`FI_Align` val)

Set alignment for line numbers (if enabled).
 - `FI_Color` `linenumber_bgcolor` () const

Returns the background color used for line numbers (if enabled).
 - void `linenumber_bgcolor` (`FI_Color` val)

Set the background color used for line numbers (if enabled).
 - `FI_Color` `linenumber_fgcolor` () const

Return the foreground color used for line numbers (if enabled).
 - void `linenumber_fgcolor` (`FI_Color` val)

Set the foreground color used for line numbers (if enabled).
 - `FI_Font` `linenumber_font` () const

Return the font used for line numbers (if enabled).
 - void `linenumber_font` (`FI_Font` val)

Set the font used for line numbers (if enabled).
 - const char * `linenumber_format` () const

Returns the line number printf() format string.
 - void `linenumber_format` (const char *val)

- Sets the printf() style format string used for line numbers.*

 - `FI_Fontsize` **linenumber_size** () const
 - Return the font size used for line numbers (if enabled).*
 - void `linenumber_size` (FI_Fontsize val)
 - Set the font size used for line numbers (if enabled).*
 - int **linenumber_width** () const
 - Return the screen area width provided for line numbers.*
 - void `linenumber_width` (int width)
 - Set width of screen area for line numbers.*
 - int `move_down` ()
 - Moves the current insert position down one line.*
 - int `move_left` ()
 - Moves the current insert position left one character.*
 - int `move_right` ()
 - Moves the current insert position right one character.*
 - int `move_up` ()
 - Moves the current insert position up one line.*
 - void **next_word** (void)
 - Moves the current insert position right one word.*
 - void `overstrike` (const char *text)
 - Replaces text at the current insert position.*
 - int `position_style` (int lineStartPos, int lineLen, int lineIndex) const
 - Find the correct style for a character.*
 - int `position_to_xy` (int pos, int *x, int *y) const
 - Convert a character index into a pixel position.*
 - void **previous_word** (void)
 - Moves the current insert position left one word.*
 - void `redisplay_range` (int start, int end)
 - Marks text from start to end as needing a redraw.*
 - virtual void `resize` (int X, int Y, int W, int H)
 - Change the size of the displayed text area.*
 - int `rewind_lines` (int startPos, int nLines)
 - Skip a number of lines back.*
 - void `scroll` (int topLineNum, int horizOffset)
 - Scrolls the current buffer to start at the specified line and column.*
 - `FI_Align` `scrollbar_align` () const
 - Gets the scrollbar alignment type.*
 - void `scrollbar_align` (FI_Align a)
 - Sets the scrollbar alignment type.*
 - int `scrollbar_width` () const
 - Gets the width/height of the scrollbars.*
 - void `scrollbar_width` (int W)
 - Sets the width/height of the scrollbars.*
 - int `shortcut` () const
 - void `shortcut` (int s)
 - void `show_cursor` (int b=1)
 - Shows the text cursor.*
 - void `show_insert_position` ()
 - Scrolls the text buffer to show the current insert position.*
 - int `skip_lines` (int startPos, int nLines, bool startPosIsLineStart)
 - Skip a number of lines forward.*

- [FI_Color textcolor](#) () const
Gets the default color of text in the widget.
- void [textcolor](#) ([FI_Color](#) n)
Sets the default color of text in the widget.
- [FI_Font textfont](#) () const
Gets the default font used when drawing text in the widget.
- void [textfont](#) ([FI_Font](#) s)
Sets the default font used when drawing text in the widget.
- [FI_Fontsize textsize](#) () const
Gets the default size of text in the widget.
- void [textsize](#) ([FI_Fontsize](#) s)
Sets the default size of text in the widget.
- int [word_end](#) (int pos) const
Moves the insert position to the end of the current word.
- int [word_start](#) (int pos) const
Moves the insert position to the beginning of the current word.
- void [wrap_mode](#) (int wrap, int wrap_margin)
Set the new text wrap mode.
- int [wrapped_column](#) (int row, int column) const
Nobody knows what this function does.
- int [wrapped_row](#) (int row) const
Nobody knows what this function does.
- double [x_to_col](#) (double x) const
Convert an x pixel position into a column number.
- [~FI_Text_Display](#) ()
Free a text display and release its associated memory.

Protected Types

- enum { [DRAW_LINE](#) , [FIND_INDEX](#) , [FIND_INDEX_FROM_ZERO](#) , [GET_WIDTH](#) }

Protected Member Functions

- void [absolute_top_line_number](#) (int oldFirstChar)
Line numbering stuff, currently unused.
- void [calc_last_char](#) ()
Update last display character index.
- void [calc_line_starts](#) (int startLine, int endLine)
Update the line start arrays.
- void [clear_rect](#) (int style, int x, int y, int width, int height) const
Clear a rectangle with the appropriate background color for style.
- void [display_insert](#) ()
Scroll the display to bring insertion cursor into view.
- virtual void [draw](#) ()
Draw the widget.
- void [draw_cursor](#) (int, int)
Draw a cursor with top center at X, Y.
- void [draw_line_numbers](#) (bool clearAll)
Refresh the line number area.
- void [draw_range](#) (int start, int end)
Draw a range of text.
- void [draw_string](#) (int style, int x, int y, int toX, const char *string, int nChars) const

- Draw a text segment in a single style.*

 - void `draw_text` (int X, int Y, int W, int H)
- Refresh a rectangle of the text display.*

 - void `draw_vline` (int visLineNum, int leftClip, int rightClip, int leftCharIndex, int rightCharIndex)
- Draw a single line of text.*

 - int `empty_vlines` () const
- Return true if there are lines visible with no corresponding buffer text.*

 - void `extend_range_for_styles` (int *start, int *end)
- I don't know what this does!*

 - void `find_line_end` (int pos, bool start_pos_is_line_start, int *lineEnd, int *nextLineStart) const
- Finds both the end of the current line and the start of the next line.*

 - void `find_wrap_range` (const char *deletedText, int pos, int nInserted, int nDeleted, int *modRangeStart, int *modRangeEnd, int *linesInserted, int *linesDeleted)
- Wrapping calculations.*

 - int `find_x` (const char *s, int len, int style, int x) const
- Find the index of the character that lies at the given x position.*

 - int `get_absolute_top_line_number` () const
- Line numbering stuff, currently unused.*

 - int `handle_vline` (int mode, int lineStart, int lineLen, int leftChar, int rightChar, int topClip, int bottomClip, int leftClip, int rightClip) const
- Universal pixel machine.*

 - int `longest_vline` () const
- Find the longest line of all visible lines.*

 - void `maintain_absolute_top_line_number` (int state)
- Line numbering stuff, currently unused.*

 - int `maintaining_absolute_top_line_number` () const
- Line numbering stuff, currently unused.*

 - void `measure_deleted_lines` (int pos, int nDeleted)
- Wrapping calculations.*

 - double `measure_proportional_character` (const char *s, int colNum, int pos) const
- Wrapping calculations.*

 - int `measure_vline` (int visLineNum) const
- Returns the width in pixels of the displayed line pointed to by "visLineNum".*

 - void `offset_line_starts` (int newTopLineNum)
- Offset line start counters for a new vertical scroll position.*

 - int `position_to_line` (int pos, int *lineNum) const
- Convert a position index into a line number offset.*

 - int `position_to_linecol` (int pos, int *lineNum, int *column) const
- Find the line and column number of position pos.*

 - void `reset_absolute_top_line_number` ()
- Line numbering stuff, probably unused.*

 - int `scroll_` (int topLineNum, int horizOffset)
- Scrolls the current buffer to start at the specified line and column.*

 - double `string_width` (const char *string, int length, int style) const
- Find the width of a string in the font of a particular style.*

 - void `update_h_scrollbar` ()
- Update horizontal scrollbar.*

 - void `update_line_starts` (int pos, int charsInserted, int charsDeleted, int linesInserted, int linesDeleted, int *scrolled)
- Update line start arrays and variables.*

 - void `update_v_scrollbar` ()

- *Update vertical scrollbar.*
- int [vline_length](#) (int visLineNum) const
Count number of bytes in a visible line.
- int [wrap_uses_character](#) (int lineEndPos) const
Check if the line break is caused by a \n or by line wrapping.
- void [wrapped_line_counter](#) ([FI_Text_Buffer](#) *buf, int startPos, int maxPos, int maxLines, bool startPosIs↵
LineStart, int styleBufOffset, int *retPos, int *retLines, int *retLineStart, int *retLineEnd, bool countLast↵
LineMissingNewLine=true) const
Wrapping calculations.
- int [xy_to_position](#) (int x, int y, int PosType=CHARACTER_POS) const
Translate a pixel position into a character index.
- void [xy_to_rowcol](#) (int x, int y, int *row, int *column, int PosType=CHARACTER_POS) const
Translate pixel coordinates into row and column.

Static Protected Member Functions

- static void [buffer_modified_cb](#) (int pos, int nInserted, int nDeleted, int nRestyled, const char *deletedText, void *cbArg)
This is called whenever the buffer is modified.
- static void [buffer_predelete_cb](#) (int pos, int nDeleted, void *cbArg)
This is called before any characters are deleted.
- static void [h_scrollbar_cb](#) ([FI_Scrollbar](#) *w, [FI_Text_Display](#) *d)
Callbacks for drag or valueChanged on horizontal scrollbar.
- static void [scroll_timer_cb](#) (void *)
Timer callback for scroll events.
- static void [v_scrollbar_cb](#) ([FI_Scrollbar](#) *w, [FI_Text_Display](#) *d)
Callbacks for drag or valueChanged on vertical scrollbar.

Protected Attributes

- int **damage_range1_end**
- int **damage_range1_start**
- int **damage_range2_end**
- int **damage_range2_start**
- int **display_insert_position_hint**
- int **dragging**
- int **dragPos**
- int **dragType**
- [FI_Align](#) **linenumber_align_**
- [FI_Color](#) **linenumber_bgcolor_**
- [FI_Color](#) **linenumber_fgcolor_**
- [FI_Font](#) **linenumber_font_**
- const char * **linenumber_format_**
- [FI_Fontsize](#) **linenumber_size_**
- int **mAbsTopLineNum**
- [FI_Text_Buffer](#) * **mBuffer**
- double **mColumnScale**
- int **mContinuousWrap**
- [FI_Color](#) **mCursor_color**
- int **mCursorOldY**
- int **mCursorOn**
- int **mCursorPos**
- int **mCursorPreferredXPos**
- int **mCursorStyle**

- int **mCursorToHint**
- int **mFirstChar**
- void * **mHighlightCBArg**
- int **mHorizOffset**
- int **mHorizOffsetHint**
- [Fl_Scrollbar](#) * **mHScrollBar**
- int **mLastChar**
- int **mLineNumLeft**
- int **mLineNumWidth**
- int * **mLineStarts**
- int **mMaxsize**
- int **mModifyingTabDistance**
- int **mNBufferLines**
- int **mNeedAbsTopLineNum**
- int **mNLinesDeleted**
- int **mNStyles**
- int **mNVisibleLines**
- [Fl_Text_Buffer](#) * **mStyleBuffer**
- const [Style_Table_Entry](#) * **mStyleTable**
- int **mSuppressResync**
- int **mTopLineNum**
- int **mTopLineNumHint**
- Unfinished_Style_Cb **mUnfinishedHighlightCB**
- char **mUnfinishedStyle**
- [Fl_Scrollbar](#) * **mVScrollBar**
- int **mWrapMarginPix**
- [Fl_Align](#) **scrollbar_align_**
- int **scrollbar_width_**
- int **shortcut_**
-

```
struct {
    int h
    int w
    int x
    int y
} text_area
```

- [Fl_Color](#) **textcolor_**
- [Fl_Font](#) **textfont_**
- [Fl_Fontsize](#) **textsize_**

Friends

- void **fl_text_drag_me** (int pos, [Fl_Text_Display](#) *d)

Additional Inherited Members

31.135.1 Detailed Description

Rich text display widget.

This is the FLTK text display widget. It allows the user to view multiple lines of text and supports highlighting, word wrap, mixes of font faces and colors, line numbers and scrolling. The buffer that is displayed in the widget is managed by the [Fl_Text_Buffer](#) class. A single Text Buffer can be displayed by multiple Text Displays.

```

//
// "$Id: editor.cxx,v 1.2.2.3.2.5 2001/12/09 12:52:13
//
// A simple text editor program for the Fast Light Tool
//
// This program is described in Chapter 4 of the FLTK
//
// Copyright 1998-2001 by Bill Spitzak and others.
//
// This library is free software; you can redistribute
// modify it under the terms of the GNU Library Genera
// License as published by the Free Software Foundatio
// version 2 of the License, or (at your option) any l
//
// This library is distributed in the hope that it wil
// but WITHOUT ANY WARRANTY; without even the implied
// MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE
// Library General Public License for more details.
//
// You should have received a copy of the GNU Library
// License along with this library; if not, write to t
// Foundation, Inc., 59 Temple Place, Suite 330, Bosto
// USA.
//

```

Figure 31.41 Fl_Text_Display widget

```

1 //
2 // "$Id: editor.cxx,v 1.2.2.3.2.5 2001/12/09 12:52:13
3 //
4 // A simple text editor program for the Fast Light Too
5 //
6 // This program is described in Chapter 4 of the FLTK
7 //
8 // Copyright 1998-2001 by Bill Spitzak and others.
9 //
10 // This program is free software; you can redistribute
11 // modify it under the terms of the GNU Library Genera
12 // License as published by the Free Software Foundatio
13 // version 2 of the License, or (at your option) any l
14 //
15 // This program is distributed in the hope that it wil
16 // but WITHOUT ANY WARRANTY; without even the implied
17 // MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE
18 // Library General Public License for more details.
19 //
20 // You should have received a copy of the GNU General
21 // License along with this program; if not, write to t
22 // Foundation, Inc., 51 Franklin Street, Fifth Floor,
23 // USA.
24 //

```

Figure 31.42 Fl_Text_Display widget with line numbers enabled

Example Use

```

#include <FL/FL_Text_Display.H>
..
int main() {
..
    Fl_Text_Buffer *buff = new Fl_Text_Buffer();
    Fl_Text_Display *disp = new Fl_Text_Display(10, 10, 640, 480);
    disp->buffer(buff); // attach text buffer to display widget
    buff->text("line one\nline two"); // add some text to buffer
..
}

```

Features

- Word wrap: [wrap_mode\(\)](#), [wrapped_column\(\)](#), [wrapped_row\(\)](#)
- Font control: [textfont\(\)](#), [textsize\(\)](#), [textcolor\(\)](#)
- Font styling: [highlight_data\(\)](#)
- Cursor: [cursor_style\(\)](#), [show_cursor\(\)](#), [hide_cursor\(\)](#), [cursor_color\(\)](#)
- Line numbers: [linenumber_width\(\)](#), [linenumber_font\(\)](#), [linenumber_size\(\)](#), [linenumber_fgcolor\(\)](#), [linenumber_bgcolor\(\)](#), [linenumber_align\(\)](#), [linenumber_format\(\)](#)

Note that other features may be available via [Fl_Text_Editor](#) and [Fl_Text_Buffer](#) classes.

Note

Line numbers were added in 1.3.3. To avoid breaking ABI, many of its options are read only. To adjust these features in 1.3.x, you must build FLTK with `FLTK_ABI_VERSION` set to 10303 or higher.

31.135.2 Member Enumeration Documentation

31.135.2.1 anonymous enum

anonymous enum

text display cursor shapes enumeration

Enumerator

NORMAL_CURSOR	I-beam.
CARET_CURSOR	caret under the text
DIM_CURSOR	dim I-beam
BLOCK_CURSOR	unfilled box under the current character
HEAVY_CURSOR	thick I-beam
SIMPLE_CURSOR	as cursor as Fl_Input cursor

31.135.2.2 anonymous enum

anonymous enum

wrap types - used in [wrap_mode\(\)](#)

Enumerator

WRAP_NONE	don't wrap text at all
WRAP_AT_COLUMN	wrap text at the given text column
WRAP_AT_PIXEL	wrap text at a pixel position
WRAP_AT_BOUNDS	wrap text so that it fits into the widget width

31.135.3 Constructor & Destructor Documentation**31.135.3.1 Fl_Text_Display()**

```
Fl_Text_Display::Fl_Text_Display (
    int X,
    int Y,
    int W,
    int H,
    const char * l = 0 )
```

Creates a new text display widget.

Parameters

<i>X, Y, W, H</i>	position and size of widget
<i>l</i>	label text, defaults to none

31.135.3.2 ~Fl_Text_Display()

```
Fl_Text_Display::~Fl_Text_Display ( )
```

Free a text display and release its associated memory.

Note, the text BUFFER that the text display displays is a separate entity and is not freed, nor are the style buffer or style table.

31.135.4 Member Function Documentation

31.135.4.1 absolute_top_line_number()

```
void Fl_Text_Display::absolute_top_line_number (
    int oldFirstChar ) [protected]
```

Line numbering stuff, currently unused.

Re-calculate absolute top line number for a change in scroll position.

31.135.4.2 buffer() [1/3]

```
Fl_Text_Buffer * Fl_Text_Display::buffer ( ) const [inline]
```

Gets the current text buffer associated with the text widget.

Multiple text widgets can be associated with the same text buffer.

Returns

current text buffer

31.135.4.3 buffer() [2/3]

```
void Fl_Text_Display::buffer (
    Fl_Text_Buffer & buf ) [inline]
```

Sets the current text buffer associated with the text widget.

Multiple text widgets can be associated with the same text buffer.

Parameters

<i>buf</i>	new text buffer
------------	-----------------

31.135.4.4 buffer() [3/3]

```
void Fl_Text_Display::buffer (
    Fl_Text_Buffer * buf )
```

Attach a text buffer to display, replacing the current buffer (if any)

Parameters

<i>buf</i>	attach this text buffer
------------	-------------------------

31.135.4.5 buffer_modified_cb()

```
void Fl_Text_Display::buffer_modified_cb (
    int pos,
    int nInserted,
    int nDeleted,
    int nRestyled,
    const char * deletedText,
    void * cbArg ) [static], [protected]
```

This is called whenever the buffer is modified.

Callback attached to the text buffer to receive modification information

This callback can be used to adjust the display or update other setting. It is not advisable to change any buffers or

text in this callback, or line counting may get out of sync.

Parameters

<i>pos</i>	starting index of modification
<i>nInserted</i>	number of bytes we inserted (must be UTF-8 aligned!)
<i>nDeleted</i>	number of bytes deleted (must be UTF-8 aligned!)
<i>nRestyled</i>	??
<i>deletedText</i>	this is what was removed, must not be NULL if nDeleted is set
<i>cbArg</i>	"this" pointer for static callback function

31.135.4.6 buffer_predelete_cb()

```
void Fl_Text_Display::buffer_predelete_cb (
    int pos,
    int nDeleted,
    void * cbArg ) [static], [protected]
```

This is called before any characters are deleted.

Callback attached to the text buffer to receive delete information before the modifications are actually made.

This callback can be used to adjust the display or update other setting. It is not advisable to change any buffers or text in this callback, or line counting may get out of sync.

Parameters

<i>pos</i>	starting index of deletion
<i>nDeleted</i>	number of bytes we will delete (must be UTF-8 aligned!)
<i>cbArg</i>	"this" pointer for static callback function

31.135.4.7 calc_last_char()

```
void Fl_Text_Display::calc_last_char ( ) [protected]
```

Update last display character index.

Given a [Fl_Text_Display](#) with a complete, up-to-date lineStarts array, update the lastChar entry to point to the last buffer position displayed.

31.135.4.8 calc_line_starts()

```
void Fl_Text_Display::calc_line_starts (
    int startLine,
    int endLine ) [protected]
```

Update the line start arrays.

Scan through the text in the "textD"'s buffer and recalculate the line starts array values beginning at index "startLine" and continuing through (including) "endLine". It assumes that the line starts entry preceding "startLine" (or mFirstChar if startLine is 0) is good, and re-counts newlines to fill in the requested entries. Out of range values for "startLine" and "endLine" are acceptable.

Parameters

<i>startLine,endLine</i>	range of lines to scan as line numbers
--------------------------	--

31.135.4.9 clear_rect()

```
void Fl_Text_Display::clear_rect (
    int style,
    int X,
    int Y,
    int width,
    int height ) const [protected]
```

Clear a rectangle with the appropriate background color for `style`.

Parameters

<code>style</code>	index into style table
<code>X,Y,width,height</code>	size and position of background area

31.135.4.10 col_to_x()

```
double Fl_Text_Display::col_to_x (
    double col ) const
```

Convert a column number into an x pixel position.

Parameters

<code>col</code>	an approximate column number based on the main font
------------------	---

Returns

number of pixels from the left margin to the left of an average sized character

31.135.4.11 count_lines()

```
int Fl_Text_Display::count_lines (
    int startPos,
    int endPos,
    bool startPosIsLineStart ) const
```

Count the number of lines between two positions.

Same as `Fl_Text_Buffer::count_lines()`, but takes into account wrapping if wrapping is turned on. If the caller knows that `startPos` is at a line start, it can pass `startPosIsLineStart` as `True` to make the call more efficient by avoiding the additional step of scanning back to the last newline.

Parameters

<code>startPos</code>	index to first character
<code>endPos</code>	index after last character
<code>startPosIsLineStart</code>	avoid scanning back to the line start

Returns

number of lines

31.135.4.12 cursor_color() [1/2]

```
Fl_Color Fl_Text_Display::cursor_color ( ) const [inline]
```

Gets the text cursor color.

Returns

cursor color

31.135.4.13 cursor_color() [2/2]

```
void Fl_Text_Display::cursor_color (
    Fl_Color n ) [inline]
```

Sets the text cursor color.

Parameters

<i>n</i>	new cursor color
----------	------------------

31.135.4.14 cursor_style()

```
void Fl_Text_Display::cursor_style (
    int style )
```

Sets the text cursor style.

Sets the text cursor style to one of the following:

- [Fl_Text_Display::NORMAL_CURSOR](#) - Shows an I beam.
- [Fl_Text_Display::CARET_CURSOR](#) - Shows a caret under the text.
- [Fl_Text_Display::DIM_CURSOR](#) - Shows a dimmed I beam.
- [Fl_Text_Display::BLOCK_CURSOR](#) - Shows an unfilled box around the current character.
- [Fl_Text_Display::HEAVY_CURSOR](#) - Shows a thick I beam.

This call also switches the cursor on and may trigger a redraw.

Parameters

<i>style</i>	new cursor style
--------------	------------------

31.135.4.15 display_insert()

```
void Fl_Text_Display::display_insert ( ) [protected]
```

Scroll the display to bring insertion cursor into view.

Note: it would be nice to be able to do this without counting lines twice ([scroll_\(\)](#) counts them too) and/or to count from the most efficient starting point, but the efficiency of this routine is not as important to the overall performance of the text display.

Todo Unicode?

31.135.4.16 draw()

```
void Fl_Text_Display::draw (
    void ) [protected], [virtual]
```

Draw the widget.

This function tries to limit drawing to smaller areas if possible.

Reimplemented from [Fl_Group](#).

31.135.4.17 draw_cursor()

```
void Fl_Text_Display::draw_cursor (
    int X,
    int Y ) [protected]
```

Draw a cursor with top center at X, Y.

Parameters

<i>X, Y</i>	cursor position in pixels
-------------	---------------------------

31.135.4.18 draw_line_numbers()

```
void Fl_Text_Display::draw_line_numbers (
    bool clearAll ) [protected]
```

Refresh the line number area.

Parameters

<i>clearAll</i>	– (currently unused) If False, only draws the line number text, does not clear the area behind it. If True, clears the area and redraws the text. Use False to avoid a 'flash' for single buffered windows.
-----------------	---

31.135.4.19 draw_range()

```
void Fl_Text_Display::draw_range (
    int startpos,
    int endpos ) [protected]
```

Draw a range of text.

Refresh all of the text between buffer positions *startpos* and *endpos* not including the character at the position *endpos*.

If *endpos* points beyond the end of the buffer, refresh the whole display after *startpos*, including blank lines which are not technically part of any range of characters.

Parameters

<i>startpos</i>	index of first character to draw
<i>endpos</i>	index after last character to draw

31.135.4.20 draw_string()

```
void Fl_Text_Display::draw_string (
    int style,
    int X,
    int Y,
    int toX,
    const char * string,
    int nChars ) const [protected]
```

Draw a text segment in a single style.

Draw a string or blank area according to parameter *style*, using the appropriate colors and drawing method for that style, with top left corner at X, Y. If style says to draw text, use *string* as source of characters, and draw *nChars*, if style is FILL, erase rectangle where text would have drawn from X to *toX* and from Y to the maximum y extent of the current font(s).

Parameters

<i>style</i>	index into style lookup table
<i>X,Y</i>	drawing origin
<i>toX</i>	rightmost position if this is a fill operation
<i>string</i>	text if this is a drawing operation
<i>nChars</i>	number of characters to draw

31.135.4.21 draw_text()

```
void Fl_Text_Display::draw_text (
    int left,
    int top,
    int width,
    int height ) [protected]
```

Refresh a rectangle of the text display.

Parameters

<i>left,top</i>	are in coordinates of the text drawing window.
<i>width,height</i>	size in pixels

31.135.4.22 draw_vline()

```
void Fl_Text_Display::draw_vline (
    int visLineNum,
    int leftClip,
    int rightClip,
    int leftCharIndex,
    int rightCharIndex ) [protected]
```

Draw a single line of text.

Draw the text on a single line represented by *visLineNum* (the number of lines down from the top of the display), limited by *leftClip* and *rightClip* window coordinates and *leftCharIndex* and *rightCharIndex* character positions (not including the character at position *rightCharIndex*).

Parameters

<i>visLineNum</i>	index of line in the visible line number lookup
<i>leftClip,rightClip</i>	pixel position of clipped area
<i>leftCharIndex,rightCharIndex</i>	index into line of segment that we want to draw

31.135.4.23 empty_vlines()

```
int Fl_Text_Display::empty_vlines ( ) const [protected]
```

Return true if there are lines visible with no corresponding buffer text.

Returns

1 if there are empty lines

31.135.4.24 `extend_range_for_styles()`

```
void Fl_Text_Display::extend_range_for_styles (
    int * startpos,
    int * endpos ) [protected]
```

I don't know what this does!

Extend the range of a redraw request (from **start* to **end*) with additional redraw requests resulting from changes to the attached style buffer (which contains auxiliary information for coloring or styling text).

Parameters

<i>startpos</i>	??
<i>endpos</i>	??

Todo Unicode?

31.135.4.25 `find_line_end()`

```
void Fl_Text_Display::find_line_end (
    int startPos,
    bool startPosIsLineStart,
    int * lineEnd,
    int * nextLineStart ) const [protected]
```

Finds both the end of the current line and the start of the next line.

Why? In continuous wrap mode, if you need to know both, figuring out one from the other can be expensive or error prone. The problem comes when there's a trailing space or tab just before the end of the buffer. To translate an end of line value to or from the next lines start value, you need to know whether the trailing space or tab is being used as a line break or just a normal character, and to find that out would otherwise require counting all the way back to the beginning of the line.

Parameters

	<i>startPos</i>	
	<i>startPosIsLineStart</i>	
out	<i>lineEnd</i>	
out	<i>nextLineStart</i>	

31.135.4.26 `find_wrap_range()`

```
void Fl_Text_Display::find_wrap_range (
    const char * deletedText,
    int pos,
    int nInserted,
    int nDeleted,
    int * modRangeStart,
    int * modRangeEnd,
```

```
int * linesInserted,
int * linesDeleted ) [protected]
```

Wrapping calculations.

When continuous wrap is on, and the user inserts or deletes characters, wrapping can happen before and beyond the changed position. This routine finds the extent of the changes, and counts the deleted and inserted lines over that range. It also attempts to minimize the size of the range to what has to be counted and re-displayed, so the results can be useful both for delimiting where the line starts need to be recalculated, and for deciding what part of the text to redisplay.

Parameters

<i>deletedText</i>	
<i>pos</i>	
<i>nInserted</i>	
<i>nDeleted</i>	
<i>modRangeStart</i>	
<i>modRangeEnd</i>	
<i>linesInserted</i>	
<i>linesDeleted</i>	

31.135.4.27 find_x()

```
int Fl_Text_Display::find_x (
    const char * s,
    int len,
    int style,
    int x ) const [protected]
```

Find the index of the character that lies at the given x position.

Parameters

<i>s</i>	UTF-8 text string
<i>len</i>	length of string
<i>style</i>	index into style lookup table
<i>x</i>	position in pixels

Returns

index into buffer

31.135.4.28 get_absolute_top_line_number()

```
int Fl_Text_Display::get_absolute_top_line_number ( ) const [protected]
```

Line numbering stuff, currently unused.

Returns the absolute (non-wrapped) line number of the first line displayed. Returns 0 if the absolute top line number is not being maintained.

31.135.4.29 handle()

```
int Fl_Text_Display::handle (
    int e ) [virtual]
```

Event handling.

Reimplemented from [Fl_Group](#).

Reimplemented in [Fl_Text_Editor](#).

31.135.4.30 `handle_vline()`

```
int Fl_Text_Display::handle_vline (
    int mode,
    int lineStartPos,
    int lineLen,
    int leftChar,
    int rightChar,
    int Y,
    int bottomClip,
    int leftClip,
    int rightClip ) const [protected]
```

Universal pixel machine.

We use a single function that handles all line layout, measuring, and drawing

- draw a text range
- return the width of a text range in pixels
- return the index of a character that is at a pixel position

Parameters

in	<i>mode</i>	DRAW_LINE, GET_WIDTH, FIND_INDEX
in	<i>lineStartPos</i>	index of first character
in	<i>lineLen</i>	size of string in bytes
in	<i>leftChar, rightChar</i>	
in	<i>Y</i>	drawing position
in	<i>bottomClip, leftClip, rightClip</i>	stop work when we reach the clipped area. rightClip is the X position that we search in FIND_INDEX.

Return values

<i>DRAW_LINE</i>	index of last drawn character
<i>GET_WIDTH</i>	width in pixels of text segment if we would draw it
<i>FIND_INDEX</i>	index of character at given x position in window coordinates
<i>FIND_INDEX_FROM_ZERO</i>	index of character at given x position without scrolling and widget offsets

Todo we need to handle hidden hyphens and tabs here!

we handle all styles and selections

we must provide code to get pixel positions of the middle of a character as well

31.135.4.31 `highlight_data()`

```
void Fl_Text_Display::highlight_data (
    Fl_Text_Buffer * styleBuffer,
    const Style_Table_Entry * styleTable,
    int nStyles,
    char unfinishedStyle,
```



```
Unfinished_Style_Cb unfinishedHighlightCB,
void * cbArg )
```

Attach (or remove) highlight information in text display and redisplay.

Highlighting information consists of a style buffer which parallels the normal text buffer, but codes font and color information for the display; a style table which translates style buffer codes (indexed by buffer character - 'A') into fonts and colors; and a callback mechanism for as-needed highlighting, triggered by a style buffer entry of "unfinished↵ Style". Style buffer can trigger additional redisplay during a normal buffer modification if the buffer contains a primary [Fl_Text_Selection](#) (see `extendRangeForStyleMods` for more information on this protocol).

Style buffers, tables and their associated memory are managed by the caller.

Styles are ranged from 65 ('A') to 126.

Parameters

<i>styleBuffer</i>	this buffer works in parallel to the text buffer. For every character in the text buffer, the style buffer has a byte at the same offset that contains an index into an array of possible styles.
<i>styleTable</i>	a list of styles indexed by the style buffer
<i>nStyles</i>	number of styles in the style table
<i>unfinishedStyle</i>	if this style is found, the callback below is called
<i>unfinishedHighlightCB</i>	if a character with an unfinished style is found, this callback will be called
<i>cbArg</i>	and optional argument for the callback above, usually a pointer to the Text Display.

31.135.4.32 in_selection()

```
int Fl_Text_Display::in_selection (
    int X,
    int Y ) const
```

Check if a pixel position is within the primary selection.

Parameters

<i>X, Y</i>	pixel position to test
-------------	------------------------

Returns

1 if position (X, Y) is inside of the primary [Fl_Text_Selection](#)

31.135.4.33 insert()

```
void Fl_Text_Display::insert (
    const char * text )
```

Inserts "text" at the current cursor location.

This has the same effect as inserting the text into the buffer using `Buflnsert` and then moving the insert position after the newly inserted text, except that it's optimized to do less redrawing.

Parameters

<i>text</i>	new text in UTF-8 encoding.
-------------	-----------------------------

31.135.4.34 insert_position() [1/2]

```
int Fl_Text_Display::insert_position ( ) const [inline]
```

Gets the position of the text insertion cursor for text display.

Returns

insert position index into text buffer

31.135.4.35 insert_position() [2/2]

```
void Fl_Text_Display::insert_position (
    int newPos )
```

Sets the position of the text insertion cursor for text display.

Move the insertion cursor in front of the character at `newPos`. This function may trigger a redraw.

Parameters

<i>newPos</i>	new caret position
---------------	--------------------

31.135.4.36 line_end()

```
int Fl_Text_Display::line_end (
    int startPos,
    bool startPosIsLineStart ) const
```

Returns the end of a line.

Same as `BufEndOfLine`, but takes into account line breaks when wrapping is turned on. If the caller knows that `startPos` is at a line start, it can pass "startPosIsLineStart" as `True` to make the call more efficient by avoiding the additional step of scanning back to the last newline.

Note that the definition of the end of a line is less clear when continuous wrap is on. With continuous wrap off, it's just a pointer to the newline that ends the line. When it's on, it's the character beyond the last **displayable** character on the line, where a whitespace character which has been "converted" to a newline for wrapping is not considered displayable. Also note that a line can be wrapped at a non-whitespace character if the line had no whitespace. In this case, this routine returns a pointer to the start of the next line. This is also consistent with the model used by `visLineLength`.

Parameters

<i>startPos</i>	index to starting character
<i>startPosIsLineStart</i>	avoid scanning back to the line start

Returns

new position as index

31.135.4.37 line_start()

```
int Fl_Text_Display::line_start (
    int pos ) const
```

Return the beginning of a line.

Same as `BufStartOfLine`, but returns the character after last wrap point rather than the last newline.

Parameters

<i>pos</i>	index to starting character
------------	-----------------------------

Returns

new position as index

31.135.4.38 `linenumber_align()`

```
void Fl_Text_Display::linenumber_align (
    Fl_Align val )
```

Set alignment for line numbers (if enabled).

Valid values are FL_ALIGN_LEFT, FL_ALIGN_CENTER or FL_ALIGN_RIGHT.

Version

1.3.3 ABI feature (ignored in 1.3.x unless FLTK_ABI_VERSION is 10303 or higher)

31.135.4.39 `linenumber_bgcolor()`

```
void Fl_Text_Display::linenumber_bgcolor (
    Fl_Color val )
```

Set the background color used for line numbers (if enabled).

Version

1.3.3 ABI feature (ignored in 1.3.x unless FLTK_ABI_VERSION is 10303 or higher)

31.135.4.40 `linenumber_fgcolor()`

```
void Fl_Text_Display::linenumber_fgcolor (
    Fl_Color val )
```

Set the foreground color used for line numbers (if enabled).

Version

1.3.3 ABI feature (ignored in 1.3.x unless FLTK_ABI_VERSION is 10303 or higher)

31.135.4.41 `linenumber_font()`

```
void Fl_Text_Display::linenumber_font (
    Fl_Font val )
```

Set the font used for line numbers (if enabled).

Version

1.3.3 ABI feature (ignored in 1.3.x unless FLTK_ABI_VERSION is 10303 or higher)

31.135.4.42 `linenumber_format()`

```
void Fl_Text_Display::linenumber_format (
    const char * val )
```

Sets the `printf()` style format string used for line numbers.

Default is "%d" for normal unpadded decimal integers.

An internal copy of `val` is allocated and managed; it is automatically freed whenever a new value is assigned, or when the widget is destroyed.

The value of `val` must *not* be NULL.

Example values:

```
- "%d"   -- For normal line numbers without padding (Default)
- "%03d" -- For 000 padding
- "%x"   -- For hexadecimal line numbers
- "%o"   -- For octal line numbers
```

Version

1.3.3 ABI feature (ignored in 1.3.x unless `FLTK_ABI_VERSION` is 10303 or higher)

31.135.4.43 `linenumber_size()`

```
void Fl_Text_Display::linenumber_size (
    Fl_Fontsize val )
```

Set the font size used for line numbers (if enabled).

Version

1.3.3 ABI feature (ignored in 1.3.x unless `FLTK_ABI_VERSION` is 10303 or higher)

31.135.4.44 `linenumber_width()`

```
void Fl_Text_Display::linenumber_width (
    int width )
```

Set width of screen area for line numbers.

Use to also enable/disable line numbers. A value of 0 disables line numbering, values >0 enable the line number display.

Parameters

<i>width</i>	The new width of the area for line numbers to appear, in pixels. 0 disables line numbers (default)
--------------	--

31.135.4.45 `longest_vline()`

```
int Fl_Text_Display::longest_vline ( ) const [protected]
```

Find the longest line of all visible lines.

Returns

the width of the longest visible line in pixels

31.135.4.46 `maintain_absolute_top_line_number()`

```
void Fl_Text_Display::maintain_absolute_top_line_number (
    int state ) [protected]
```

Line numbering stuff, currently unused.

In continuous wrap mode, internal line numbers are calculated after wrapping. A separate non-wrapped line count is maintained when line numbering is turned on. There is some performance cost to maintaining this line count, so normally absolute line numbers are not tracked if line numbering is off. This routine allows callers to specify that they still want this line count maintained (for use via `TextDPosToLineAndCol`). More specifically, this allows the line number reported in the statistics line to be calibrated in absolute lines, rather than post-wrapped lines.

31.135.4.47 `maintaining_absolute_top_line_number()`

```
int Fl_Text_Display::maintaining_absolute_top_line_number ( ) const [protected]
```

Line numbering stuff, currently unused.

Return true if a separate absolute top line number is being maintained (for displaying line numbers or showing in the statistics line).

31.135.4.48 `measure_deleted_lines()`

```
void Fl_Text_Display::measure_deleted_lines (
    int pos,
    int nDeleted ) [protected]
```

Wrapping calculations.

This is a stripped-down version of the `findWrapRange()` function above, intended to be used to calculate the number of "deleted" lines during a buffer modification. It is called *before* the modification takes place.

This function should only be called in continuous wrap mode with a non-fixed font width. In that case, it is impossible to calculate the number of deleted lines, because the necessary style information is no longer available *after* the modification. In other cases, we can still perform the calculation afterwards (possibly even more efficiently).

Parameters

<i>pos</i>	
<i>nDeleted</i>	

31.135.4.49 `measure_proportional_character()`

```
double Fl_Text_Display::measure_proportional_character (
    const char * s,
    int xPix,
    int pos ) const [protected]
```

Wrapping calculations.

Measure the width in pixels of the first character of string "s" at a particular column "colNum" and buffer position "pos". This is for measuring characters in proportional or mixed-width highlighting fonts.

A note about proportional and mixed-width fonts: the mixed width and proportional font code in `nedit` does not get much use in general editing, because `nedit` doesn't allow per-language-mode fonts, and editing programs in a proportional font is usually a bad idea, so very few users would choose a proportional font as a default. There are still probably mixed-width syntax highlighting cases where things don't redraw properly for insertion/deletion, though static display and wrapping and resizing should now be solid because they are now used for online help display.

Parameters

<i>s</i>	text string
<i>xPix</i>	x pixel position needed for calculating tab widths
<i>pos</i>	offset within string

Returns

width of character in pixels

31.135.4.50 measure_vline()

```
int Fl_Text_Display::measure_vline (
    int visLineNum ) const [protected]
```

Returns the width in pixels of the displayed line pointed to by "visLineNum".

Parameters

<i>visLineNum</i>	index into visible lines array
-------------------	--------------------------------

Returns

width of line in pixels

31.135.4.51 move_down()

```
int Fl_Text_Display::move_down ( )
```

Moves the current insert position down one line.

Returns

1 if the cursor moved, 0 if the beginning of the text was reached

31.135.4.52 move_left()

```
int Fl_Text_Display::move_left ( )
```

Moves the current insert position left one character.

Returns

1 if the cursor moved, 0 if the beginning of the text was reached

31.135.4.53 move_right()

```
int Fl_Text_Display::move_right ( )
```

Moves the current insert position right one character.

Returns

1 if the cursor moved, 0 if the end of the text was reached

31.135.4.54 move_up()

```
int Fl_Text_Display::move_up ( )
```

Moves the current insert position up one line.

Returns

1 if the cursor moved, 0 if the beginning of the text was reached

31.135.4.55 offset_line_starts()

```
void Fl_Text_Display::offset_line_starts (
    int newTopLineNum ) [protected]
```

Offset line start counters for a new vertical scroll position.

Offset the line starts array, mTopLineNum, mFirstChar and lastChar, for a new vertical scroll position given by new↵TopLineNum. If any currently displayed lines will still be visible, salvage the line starts values, otherwise, count lines from the nearest known line start (start or end of buffer, or the closest value in the mLineStarts array)

Parameters

<i>newTopLineNum</i>	index into buffer
----------------------	-------------------

31.135.4.56 overstrike()

```
void Fl_Text_Display::overstrike (
    const char * text )
```

Replaces text at the current insert position.

Parameters

<i>text</i>	new text in UTF-8 encoding
-------------	----------------------------

Todo Unicode? Find out exactly what we do here and simplify.

31.135.4.57 position_style()

```
int Fl_Text_Display::position_style (
    int lineStartPos,
    int lineLen,
    int lineIndex ) const
```

Find the correct style for a character.

Determine the drawing method to use to draw a specific character from "buf". lineStartPos gives the character index where the line begins, lineIndex, the number of characters past the beginning of the line, and line↵Index the number of displayed characters past the beginning of the line. Passing lineStartPos of -1 returns the drawing style for "no text".

Why not just: position_style(pos)? Because style applies to blank areas of the window beyond the text boundaries, and because this routine must also decide whether a position is inside of a rectangular [Fl_Text_Selection](#), and do so efficiently, without re-counting character positions from the start of the line.

Note that style is a somewhat incorrect name, drawing method would be more appropriate.

Parameters

<i>lineStartPos</i>	beginning of this line
<i>lineLen</i>	number of bytes in line
<i>lineIndex</i>	position of character within line

Returns

style for the given character

31.135.4.58 position_to_line()

```
int Fl_Text_Display::position_to_line (
    int pos,
    int * lineNum ) const [protected]
```

Convert a position index into a line number offset.

Find the line number of position `pos` relative to the first line of displayed text. Returns 0 if the line is not displayed.

Parameters

	<i>pos</i>	??
out	<i>lineNum</i>	??

Returns

??

Todo What does this do?

31.135.4.59 position_to_linecol()

```
int Fl_Text_Display::position_to_linecol (
    int pos,
    int * lineNum,
    int * column ) const [protected]
```

Find the line and column number of position `pos`.

This only works for displayed lines. If the line is not displayed, the function returns 0 (without the `mLineStarts` array it could turn in to very long calculation involving scanning large amounts of text in the buffer). If continuous wrap mode is on, returns the absolute line number (as opposed to the wrapped line number which is used for scrolling).

Parameters

	<i>pos</i>	character index
out	<i>lineNum</i>	absolute (unwrapped) line number
out	<i>column</i>	character offset to the beginning of the line

Returns

0 if `pos` is off screen, line number otherwise

Todo a column number makes little sense in the UTF-8/variable font width environment. We will have to further define what exactly we want to return. Please check the functions that call this particular function.

31.135.4.60 position_to_xy()

```
int Fl_Text_Display::position_to_xy (
    int pos,
    int * X,
    int * Y ) const
```

Convert a character index into a pixel position.

Translate a buffer text position to the XY location where the top left of the cursor would be positioned to point to that character. Returns 0 if the position is not displayed because it is **vertically out** of view. If the position is horizontally out of view, returns the X coordinate where the position would be if it were visible.

Parameters

	<i>pos</i>	character index
out	<i>X, Y</i>	pixel position of character on screen

Returns

0 if character vertically out of view, X & Y positions otherwise

31.135.4.61 redisplay_range()

```
void Fl_Text_Display::redisplay_range (
    int startpos,
    int endpos )
```

Marks text from start to end as needing a redraw.

This function will trigger a damage event and later a redraw of parts of the widget.

Parameters

<i>startpos</i>	index of first character needing redraw
<i>endpos</i>	index after last character needing redraw

31.135.4.62 reset_absolute_top_line_number()

```
void Fl_Text_Display::reset_absolute_top_line_number ( ) [protected]
```

Line numbering stuff, probably unused.

Count lines from the beginning of the buffer to reestablish the absolute (non-wrapped) top line number. If mode is not continuous wrap, or the number is not being maintained, does nothing.

31.135.4.63 resize()

```
void Fl_Text_Display::resize (
    int X,
    int Y,
    int W,
    int H ) [virtual]
```

Change the size of the displayed text area.

Calling this function will trigger a recalculation of all lines visible and of all scrollbar sizes.

Parameters

<i>X, Y, W, H</i>	new position and size of this widget
-------------------	--------------------------------------

Reimplemented from [Fl_Group](#).

31.135.4.64 rewind_lines()

```
int Fl_Text_Display::rewind_lines (
    int startPos,
    int nLines )
```

Skip a number of lines back.

Same as BufCountBackwardNLines, but takes into account line breaks when wrapping is turned on.

Parameters

<i>startPos</i>	index to starting character
<i>nLines</i>	number of lines to skip back

Returns

new position as index

31.135.4.65 scroll()

```
void Fl_Text_Display::scroll (
    int topLineNum,
    int horizOffset )
```

Scrolls the current buffer to start at the specified line and column.

Parameters

<i>topLineNum</i>	top line number
<i>horizOffset</i>	column number

Todo Column numbers make little sense here.

31.135.4.66 scroll_()

```
int Fl_Text_Display::scroll_ (
    int topLineNum,
    int horizOffset ) [protected]
```

Scrolls the current buffer to start at the specified line and column.

Parameters

<i>topLineNum</i>	top line number
<i>horizOffset</i>	in pixels

Returns

0 if nothing changed, 1 if we scrolled

31.135.4.67 scroll_timer_cb()

```
void Fl_Text_Display::scroll_timer_cb (
    void * user_data ) [static], [protected]
```

Timer callback for scroll events.

This timer event scrolls the text view proportionally to how far the mouse pointer has left the text area. This allows for smooth scrolling without "wiggeling" the mouse.

31.135.4.68 scrollbar_align() [1/2]

```
Fl_Align Fl_Text_Display::scrollbar_align ( ) const [inline]
```

Gets the scrollbar alignment type.

Returns

scrollbar alignment

31.135.4.69 scrollbar_align() [2/2]

```
void Fl_Text_Display::scrollbar_align (
    Fl_Align a ) [inline]
```

Sets the scrollbar alignment type.

Parameters

<i>a</i>	new scrollbar alignment
----------	-------------------------

31.135.4.70 scrollbar_width() [1/2]

```
int Fl_Text_Display::scrollbar_width ( ) const [inline]
```

Gets the width/height of the scrollbars.

Returns

width of scrollbars

31.135.4.71 scrollbar_width() [2/2]

```
void Fl_Text_Display::scrollbar_width (
    int W ) [inline]
```

Sets the width/height of the scrollbars.

Parameters

<i>W</i>	width of scrollbars
----------	---------------------

31.135.4.72 shortcut() [1/2]

```
int Fl_Text_Display::shortcut ( ) const [inline]
```

Todo FIXME : get set methods pointing on shortcut_ have no effects as shortcut_ is unused in this class and derived!

Returns

the current shortcut key

31.135.4.73 shortcut() [2/2]

```
void Fl_Text_Display::shortcut (
    int s ) [inline]
```

Todo FIXME : get set methods pointing on shortcut_ have no effects as shortcut_ is unused in this class and derived!

Parameters

<i>s</i>	the new shortcut key
----------	----------------------

31.135.4.74 show_cursor()

```
void Fl_Text_Display::show_cursor (
    int b = 1 )
```

Shows the text cursor.

This function may trigger a redraw.

Parameters

<i>b</i>	show(1) or hide(0) the text cursor (caret).
----------	---

31.135.4.75 show_insert_position()

```
void Fl_Text_Display::show_insert_position ( )
```

Scrolls the text buffer to show the current insert position.

This function triggers a complete recalculation, ending in a call to [Fl_Text_Display::display_insert\(\)](#)

31.135.4.76 skip_lines()

```
int Fl_Text_Display::skip_lines (
    int startPos,
    int nLines,
    bool startPosIsLineStart )
```

Skip a number of lines forward.

Same as `BufCountForwardNLines`, but takes into account line breaks when wrapping is turned on. If the caller knows that `startPos` is at a line start, it can pass "startPosIsLineStart" as `True` to make the call more efficient by avoiding the additional step of scanning back to the last newline.

Parameters

<i>startPos</i>	index to starting character
<i>nLines</i>	number of lines to skip ahead
<i>startPosIsLineStart</i>	avoid scanning back to the line start

Returns

new position as index

31.135.4.77 string_width()

```
double Fl_Text_Display::string_width (
    const char * string,
    int length,
    int style ) const [protected]
```

Find the width of a string in the font of a particular style.

Parameters

<i>string</i>	the text
<i>length</i>	number of bytes in string
<i>style</i>	index into style table

Returns

width of text segment in pixels

31.135.4.78 textcolor() [1/2]

```
Fl_Color Fl_Text_Display::textcolor ( ) const [inline]
```

Gets the default color of text in the widget.

Returns

text color unless overridden by a style

31.135.4.79 textcolor() [2/2]

```
void Fl_Text_Display::textcolor (
    Fl_Color n ) [inline]
```

Sets the default color of text in the widget.

Parameters

<i>n</i>	new text color
----------	----------------

31.135.4.80 textfont() [1/2]

```
Fl_Font Fl_Text_Display::textfont ( ) const [inline]
```

Gets the default font used when drawing text in the widget.

Returns

current text font face unless overridden by a style

31.135.4.81 textfont() [2/2]

```
void Fl_Text_Display::textfont (
    Fl_Font s ) [inline]
```

Sets the default font used when drawing text in the widget.

Parameters

<i>s</i>	default text font face
----------	------------------------

31.135.4.82 `textsize()` [1/2]

`Fl_Fontsize` `Fl_Text_Display::textsize () const` [inline]
Gets the default size of text in the widget.

Returns

current text height unless overridden by a style

31.135.4.83 `textsize()` [2/2]

`void Fl_Text_Display::textsize (`
 `Fl_Fontsize s)` [inline]
Sets the default size of text in the widget.

Parameters

<code>s</code>	new text size
----------------	---------------

31.135.4.84 `update_h_scrollbar()`

`void Fl_Text_Display::update_h_scrollbar ()` [protected]
Update horizontal scrollbar.
Update the minimum, maximum, slider size, page increment, and value for the horizontal scrollbar.

31.135.4.85 `update_line_starts()`

`void Fl_Text_Display::update_line_starts (`
 `int pos,`
 `int charsInserted,`
 `int charsDeleted,`
 `int linesInserted,`
 `int linesDeleted,`
 `int * scrolled)` [protected]

Update line start arrays and variables.

Update the line starts array, `mTopLineNum`, `mFirstChar` and `lastChar` for this text display after a modification to the text buffer, given by the position `pos` where the change began, and the numbers of characters and lines inserted and deleted.

Parameters

	<code>pos</code>	index into buffer of recent changes
	<code>charsInserted</code>	number of bytes(!) inserted
	<code>charsDeleted</code>	number of bytes(!) deleted
	<code>linesInserted</code>	number of lines
	<code>linesDeleted</code>	number of lines
out	<code>scrolled</code>	set to 1 if the text display needs to be scrolled

31.135.4.86 `update_v_scrollbar()`

`void Fl_Text_Display::update_v_scrollbar ()` [protected]
Update vertical scrollbar.
Update the minimum, maximum, slider size, page increment, and value for the vertical scrollbar.

31.135.4.87 vline_length()

```
int Fl_Text_Display::vline_length (
    int visLineNum ) const [protected]
```

Count number of bytes in a visible line.

Return the length of a line (number of bytes) by examining entries in the line starts array rather than by scanning for newlines.

Parameters

<i>visLineNum</i>	index of line in visible line array
-------------------	-------------------------------------

Returns

number of bytes in this line

31.135.4.88 word_end()

```
int Fl_Text_Display::word_end (
    int pos ) const [inline]
```

Moves the insert position to the end of the current word.

Parameters

<i>pos</i>	start calculation at this index
------------	---------------------------------

Returns

index of first character after the end of the word

31.135.4.89 word_start()

```
int Fl_Text_Display::word_start (
    int pos ) const [inline]
```

Moves the insert position to the beginning of the current word.

Parameters

<i>pos</i>	start calculation at this index
------------	---------------------------------

Returns

beginning of the words

31.135.4.90 wrap_mode()

```
void Fl_Text_Display::wrap_mode (
    int wrap,
    int wrapMargin )
```

Set the new text wrap mode.

If `wrap` mode is not zero, this call enables automatic word wrapping at column `wrapMargin`. Word-wrapping does not change the text buffer itself, only the way the text is displayed. Different Text Displays can have different wrap modes, even if they share the same Text Buffer.

Parameters

<i>wrap</i>	new wrap mode is WRAP_NONE (don't wrap text at all), WRAP_AT_COLUMN (wrap text at the given text column), WRAP_AT_PIXEL (wrap text at a pixel position), or WRAP_AT_BOUNDS (wrap text so that it fits into the widget width)
<i>wrapMargin</i>	in WRAP_AT_COLUMN mode, text will wrap at the n'th character. For variable width fonts, an average character width is calculated. The column width is calculated using the current textfont or the first style when this function is called. If the font size changes, this function must be called again. In WRAP_AT_PIXEL mode, this is the pixel position.

Todo we need new wrap modes to wrap at the window edge and based on pixel width or average character width.

31.135.4.91 wrap_uses_character()

```
int Fl_Text_Display::wrap_uses_character (
    int lineEndPos ) const [protected]
```

Check if the line break is caused by a `\n` or by line wrapping.

Line breaks in continuous wrap mode usually happen at newlines or whitespace. This line-terminating character is not included in line width measurements and has a special status as a non-visible character. However, lines with no whitespace are wrapped without the benefit of a line terminating character, and this distinction causes endless trouble with all of the text display code which was originally written without continuous wrap mode and always expects to wrap at a newline character.

Given the position of the end of the line, as returned by `TextEndOfLine` or `BufEndOfLine`, this returns true if there is a line terminating character, and false if there's not. On the last character in the buffer, this function can't tell for certain whether a trailing space was used as a wrap point, and just guesses that it wasn't. So if an exact accounting is necessary, don't use this function.

Parameters

<i>lineEndPos</i>	index of character where the line wraps
-------------------	---

Returns

1 if a `\n` character causes the line wrap

31.135.4.92 wrapped_column()

```
int Fl_Text_Display::wrapped_column (
    int row,
    int column ) const
```

Nobody knows what this function does.

Correct a column number based on an unconstrained position (as returned by `TextDXYToUnconstrainedPosition`) to be relative to the last actual newline in the buffer before the row and column position given, rather than the last line start created by line wrapping. This is an adapter for rectangular selections and code written before continuous wrap mode, which thinks that the unconstrained column is the number of characters from the last newline. Obviously this is time consuming, because it involves character re-counting.

Parameters

<i>row</i>	
<i>column</i>	

Returns

something unknown

Todo What does this do and how is it useful? Column numbers mean little in this context. Which functions depend on this one?

Todo Unicode?

31.135.4.93 wrapped_line_counter()

```
void Fl_Text_Display::wrapped_line_counter (
    Fl_Text_Buffer * buf,
    int startPos,
    int maxPos,
    int maxLines,
    bool startPosIsLineStart,
    int styleBufOffset,
    int * retPos,
    int * retLines,
    int * retLineStart,
    int * retLineEnd,
    bool countLastLineMissingNewLine = true ) const [protected]
```

Wrapping calculations.

Count forward from startPos to either maxPos or maxLines (whichever is reached first), and return all relevant positions and line count. The provided textBuffer may differ from the actual text buffer of the widget. In that case it must be a (partial) copy of the actual text buffer and the styleBufOffset argument must indicate the starting position of the copy, to take into account the correct style information.

Parameters

in	<i>buf</i>	The text buffer to operate on
in	<i>startPos</i>	Starting index position into the buffer
in	<i>maxPos</i>	Maximum index position into the buffer we'll reach
in	<i>maxLines</i>	Maximum number of lines we'll reach
in	<i>startPosIsLineStart</i>	Flag indicating if startPos is start of line. (If set, prevents our having to find the line start)
in	<i>styleBufOffset</i>	Offset index position into style buffer.
out	<i>retPos</i>	Position where counting ended. When counting lines, the position returned is the start of the line "maxLines" lines beyond "startPos".
out	<i>retLines</i>	Number of line breaks counted
out	<i>retLineStart</i>	Start of the line where counting ended
out	<i>retLineEnd</i>	End position of the last line traversed
out	<i>countLastLineMissingNewLine</i>	

31.135.4.94 wrapped_row()

```
int Fl_Text_Display::wrapped_row (
    int row ) const
```

Nobody knows what this function does.

Correct a row number from an unconstrained position (as returned by TextDXYToUnconstrainedPosition) to a straight number of newlines from the top line of the display. Because rectangular selections are based on newlines, rather than display wrapping, and anywhere a rectangular selection needs a row, it needs it in terms of un-wrapped lines.

Parameters

<i>row</i>	
------------	--

Returns

something unknown

Todo What does this do and how is it useful? Column numbers mean little in this context. Which functions depend on this one?

31.135.4.95 x_to_col()

```
double Fl_Text_Display::x_to_col (
    double x ) const
```

Convert an x pixel position into a column number.

Parameters

<i>x</i>	number of pixels from the left margin
----------	---------------------------------------

Returns

an approximate column number based on the main font

31.135.4.96 xy_to_position()

```
int Fl_Text_Display::xy_to_position (
    int X,
    int Y,
    int posType = CHARACTER_POS ) const [protected]
```

Translate a pixel position into a character index.

Translate window coordinates to the nearest (insert cursor or character cell) text position. The parameter *posType* specifies how to interpret the position: `CURSOR_POS` means translate the coordinates to the nearest cursor position, and `CHARACTER_POS` means return the position of the character closest to (X, Y).

Parameters

<i>X, Y</i>	pixel position
<i>posType</i>	<code>CURSOR_POS</code> or <code>CHARACTER_POS</code>

Returns

index into text buffer

31.135.4.97 xy_to_rowcol()

```
void Fl_Text_Display::xy_to_rowcol (
    int X,
    int Y,
    int * row,
    int * column,
    int posType = CHARACTER_POS ) const [protected]
```

Translate pixel coordinates into row and column.

Translate window coordinates to the nearest row and column number for positioning the cursor. This, of course, makes no sense when the font is proportional, since there are no absolute columns. The parameter `posType` specifies how to interpret the position: `CURSOR_POS` means translate the coordinates to the nearest position between characters, and `CHARACTER_POS` means translate the position to the nearest character cell.

Parameters

	<i>X, Y</i>	pixel coordinates
<i>out</i>	<i>row, column</i>	nearest row and column
	<i>posType</i>	<code>CURSOR_POS</code> or <code>CHARACTER_POS</code>

The documentation for this class was generated from the following files:

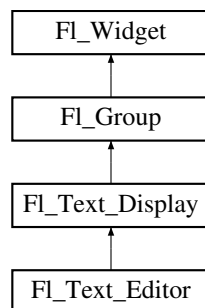
- `Fl_Text_Display.H`
- `Fl_Text_Display.cxx`

31.136 Fl_Text_Editor Class Reference

This is the FLTK text editor widget.

```
#include <Fl_Text_Editor.H>
```

Inheritance diagram for `Fl_Text_Editor`:



Classes

- struct [Key_Binding](#)
Simple linked list item associating a key/state to a function.

Public Types

- typedef `int(* Key_Func)(int key, Fl_Text_Editor *editor)`
Key function binding callback type.

Public Member Functions

- void **add_default_key_bindings** ([Key_Binding](#) **list)
Adds all of the default editor key bindings to the specified key binding list.
- void **add_key_binding** (int key, int state, [Key_Func](#) f)
Adds a key of state state with the function f.
- void **add_key_binding** (int key, int state, [Key_Func](#) f, [Key_Binding](#) **list)
Adds a key of state state with the function function to an arbitrary key binding list list.
- [Key_Func](#) **bound_key_function** (int key, int state) const
Returns the function associated with a key binding.
- [Key_Func](#) **bound_key_function** (int key, int state, [Key_Binding](#) *list) const

- Returns the function associated with a key binding.*
- void **default_key_function** ([Key_Func](#) f)
 - Sets the default key function for unassigned keys.*
 - **FI_Text_Editor** (int X, int Y, int W, int H, const char *l=0)
 - The constructor creates a new text editor widget.*
 - virtual int **handle** (int e)
 - Event handling.*
 - int **insert_mode** ()
 - Gets the current insert mode; if non-zero, new text is inserted before the current cursor position.*
 - void **insert_mode** (int b)
 - Sets the current insert mode; if non-zero, new text is inserted before the current cursor position.*
 - void **remove_all_key_bindings** ()
 - Removes all of the key bindings associated with the text editor or list.*
 - void **remove_all_key_bindings** ([Key_Binding](#) **list)
 - Removes all of the key bindings associated with the text editor or list.*
 - void **remove_key_binding** (int key, int state)
 - Removes the key binding associated with the key "key" of state "state".*
 - void **remove_key_binding** (int key, int state, [Key_Binding](#) **list)
 - Removes the key binding associated with the key key of state state from the [Key_Binding](#) list list.*
 - int **tab_nav** () const
 - Check if Tab focus navigation is enabled.*
 - void **tab_nav** (int val)
 - Enables or disables Tab key focus navigation.*

Static Public Member Functions

- static int **kf_backspace** (int c, [FI_Text_Editor](#) *e)
 - Does a backspace for key 'c' in the current buffer of editor 'e'.*
 - static int **kf_c_s_move** (int c, [FI_Text_Editor](#) *e)
 - Extends the current selection in the direction indicated by control key 'c' in editor 'e'.*
 - static int **kf_copy** (int c, [FI_Text_Editor](#) *e)
 - Does a copy of selected text or the current character in the current buffer of editor 'e'.*
 - static int **kf_ctrl_move** (int c, [FI_Text_Editor](#) *e)
 - Moves the current text cursor in the direction indicated by control key 'c' in editor 'e'.*
 - static int **kf_cut** (int c, [FI_Text_Editor](#) *e)
 - Does a cut of selected text in the current buffer of editor 'e'.*
 - static int **kf_default** (int c, [FI_Text_Editor](#) *e)
 - Inserts the text associated with key 'c' in editor 'e'.*
 - static int **kf_delete** (int c, [FI_Text_Editor](#) *e)
 - Does a delete of selected text or the current character in the current buffer of editor 'e'.*
 - static int **kf_down** (int c, [FI_Text_Editor](#) *e)
 - Moves the text cursor one line down for editor 'e'.*
 - static int **kf_end** (int c, [FI_Text_Editor](#) *e)
 - Moves the text cursor to the end of the current line in editor 'e'.*
 - static int **kf_enter** (int c, [FI_Text_Editor](#) *e)
 - Inserts a newline for key 'c' at the current cursor position in editor 'e'.*
 - static int **kf_home** (int, [FI_Text_Editor](#) *e)
 - Moves the text cursor to the beginning of the current line in editor 'e'.*
 - static int **kf_ignore** (int c, [FI_Text_Editor](#) *e)
 - Ignores the key 'c' in editor 'e'.*
 - static int **kf_insert** (int c, [FI_Text_Editor](#) *e)

- Toggles the insert mode for editor 'e'.*

 - static int `kf_left` (int c, `Fl_Text_Editor *e`)

Moves the text cursor one character to the left in editor 'e'.
- static int `kf_m_s_move` (int c, `Fl_Text_Editor *e`)

Extends the current selection in the direction indicated by meta key 'c' in editor 'e'.
- static int `kf_meta_move` (int c, `Fl_Text_Editor *e`)

Moves the current text cursor in the direction indicated by meta key 'c' in editor 'e'.
- static int `kf_move` (int c, `Fl_Text_Editor *e`)

Moves the text cursor in the direction indicated by key 'c' in editor 'e'.
- static int `kf_page_down` (int c, `Fl_Text_Editor *e`)

Moves the text cursor down one page for editor 'e'.
- static int `kf_page_up` (int c, `Fl_Text_Editor *e`)

Moves the text cursor up one page for editor 'e'.
- static int `kf_paste` (int c, `Fl_Text_Editor *e`)

Does a paste of selected text in the current buffer of editor 'e'.
- static int `kf_right` (int c, `Fl_Text_Editor *e`)

Moves the text cursor one character to the right for editor 'e'.
- static int `kf_select_all` (int c, `Fl_Text_Editor *e`)

Selects all text in the current buffer in editor 'e'.
- static int `kf_shift_move` (int c, `Fl_Text_Editor *e`)

Extends the current selection in the direction of key 'c' in editor 'e'.
- static int `kf_undo` (int c, `Fl_Text_Editor *e`)

Undo last edit in the current buffer of editor 'e'.
- static int `kf_up` (int c, `Fl_Text_Editor *e`)

Moves the text cursor one line up for editor 'e'.

Protected Member Functions

- int `handle_key` ()
- Handles a key press in the editor.*
- void `maybe_do_callback` ()
- does or does not a callback according to `changed()` and `when()` settings*

Static Protected Attributes

- static `Key_Binding * global_key_bindings`
- Global key binding list.*

Additional Inherited Members

31.136.1 Detailed Description

This is the FLTK text editor widget.

It allows the user to edit multiple lines of text and supports highlighting and scrolling. The buffer that is displayed in the widget is managed by the `Fl_Text_Buffer` class.

31.136.2 Member Function Documentation

31.136.2.1 add_key_binding()

```
void Fl_Text_Editor::add_key_binding (
    int key,
    int state,
    Key_Func function,
    Key_Binding ** list )
```

Adds a key of state `state` with the function `function` to an arbitrary key binding list `list`.

This can be used in derived classes to add global key bindings by using the global (static) [Key_Binding](#) list `Fl_Text_Editor::global_key_bindings`.

31.136.2.2 handle()

```
int Fl_Text_Editor::handle (
    int e ) [virtual]
```

Event handling.

Reimplemented from [Fl_Text_Display](#).

31.136.2.3 insert_mode() [1/2]

```
int Fl_Text_Editor::insert_mode ( ) [inline]
```

Gets the current insert mode; if non-zero, new text is inserted before the current cursor position.

Otherwise, new text replaces text at the current cursor position.

31.136.2.4 insert_mode() [2/2]

```
void Fl_Text_Editor::insert_mode (
    int b ) [inline]
```

Sets the current insert mode; if non-zero, new text is inserted before the current cursor position.

Otherwise, new text replaces text at the current cursor position.

31.136.2.5 kf_backspace()

```
int Fl_Text_Editor::kf_backspace (
    int c,
    Fl_Text_Editor * e ) [static]
```

Does a backspace for key '`c`' in the current buffer of editor '`e`'.

Any current selection is deleted. Otherwise, the character left is deleted and the cursor moved. The key value '`c`' is currently unused.

31.136.2.6 kf_c_s_move()

```
int Fl_Text_Editor::kf_c_s_move (
    int c,
    Fl_Text_Editor * e ) [static]
```

Extends the current selection in the direction indicated by control key '`c`' in editor '`e`'.

See also

[kf_ctrl_move\(\)](#).

31.136.2.7 kf_copy()

```
int Fl_Text_Editor::kf_copy (
    int c,
    Fl_Text_Editor * e ) [static]
```

Does a copy of selected text or the current character in the current buffer of editor '`e`'.

The key value '`c`' is currently unused.

31.136.2.8 kf_ctrl_move()

```
int Fl_Text_Editor::kf_ctrl_move (
    int c,
    Fl_Text_Editor * e ) [static]
```

Moves the current text cursor in the direction indicated by control key 'c' in editor 'e'.

Supported values for 'c' are currently:

```
FL_Home    -- moves the cursor to the beginning of the document
FL_End     -- moves the cursor to the end of the document
FL_Left    -- moves the cursor left one word
FL_Right   -- moves the cursor right one word
FL_Up      -- scrolls up one line, without moving cursor
FL_Down    -- scrolls down one line, without moving cursor
FL_Page_Up -- moves the cursor to the beginning of the top line on the current page
FL_Page_Down -- moves the cursor to the beginning of the last line on the current page
```

31.136.2.9 kf_cut()

```
int Fl_Text_Editor::kf_cut (
    int c,
    Fl_Text_Editor * e ) [static]
```

Does a cut of selected text in the current buffer of editor 'e'.

The key value 'c' is currently unused.

31.136.2.10 kf_default()

```
int Fl_Text_Editor::kf_default (
    int c,
    Fl_Text_Editor * e ) [static]
```

Inserts the text associated with key 'c' in editor 'e'.

Honors the current selection and insert/overstrike mode.

31.136.2.11 kf_delete()

```
int Fl_Text_Editor::kf_delete (
    int c,
    Fl_Text_Editor * e ) [static]
```

Does a delete of selected text or the current character in the current buffer of editor 'e'.

The key value 'c' is currently unused.

31.136.2.12 kf_down()

```
int Fl_Text_Editor::kf_down (
    int c,
    Fl_Text_Editor * e ) [static]
```

Moves the text cursor one line down for editor 'e'.

Same as kf_move(FL_Down, e). The key value 'c' is currently unused.

31.136.2.13 kf_end()

```
int Fl_Text_Editor::kf_end (
    int c,
    Fl_Text_Editor * e ) [static]
```

Moves the text cursor to the end of the current line in editor 'e'.

Same as kf_move(FL_End, e). The key value 'c' is currently unused.

31.136.2.14 kf_enter()

```
int Fl_Text_Editor::kf_enter (
    int c,
    Fl_Text_Editor * e ) [static]
```

Inserts a newline for key 'c' at the current cursor position in editor 'e'.
The key value 'c' is currently unused.

31.136.2.15 kf_home()

```
int Fl_Text_Editor::kf_home (
    int ,
    Fl_Text_Editor * e ) [static]
```

Moves the text cursor to the beginning of the current line in editor 'e'.
Same as kf_move(FL_Home, e). The key value 'c' is currently unused.

31.136.2.16 kf_ignore()

```
int Fl_Text_Editor::kf_ignore (
    int c,
    Fl_Text_Editor * e ) [static]
```

Ignores the key 'c' in editor 'e'.

This method can be used as a keyboard binding to disable a key that might otherwise be handled or entered as text.
An example would be disabling FL_Escape, so that it isn't added to the buffer when invoked by the user.

31.136.2.17 kf_insert()

```
int Fl_Text_Editor::kf_insert (
    int c,
    Fl_Text_Editor * e ) [static]
```

Toggles the insert mode for editor 'e'.

The key value 'c' is currently unused.

31.136.2.18 kf_left()

```
int Fl_Text_Editor::kf_left (
    int c,
    Fl_Text_Editor * e ) [static]
```

Moves the text cursor one character to the left in editor 'e'.

Same as kf_move(FL_Left, e). The key value 'c' is currently unused.

31.136.2.19 kf_m_s_move()

```
int Fl_Text_Editor::kf_m_s_move (
    int c,
    Fl_Text_Editor * e ) [static]
```

Extends the current selection in the direction indicated by meta key 'c' in editor 'e'.

See also

[kf_meta_move\(\)](#).

31.136.2.20 kf_meta_move()

```
int Fl_Text_Editor::kf_meta_move (
    int c,
    Fl_Text_Editor * e ) [static]
```

Moves the current text cursor in the direction indicated by meta key 'c' in editor 'e'.

Supported values for 'c' are currently:

```
FL_Up      -- moves cursor to the beginning of the current document
FL_Down    -- moves cursor to the end of the current document
FL_Left    -- moves the cursor to the beginning of the current line
FL_Right   -- moves the cursor to the end of the current line
```


31.136.2.21 kf_move()

```
int Fl_Text_Editor::kf_move (
    int c,
    Fl_Text_Editor * e ) [static]
```

Moves the text cursor in the direction indicated by key 'c' in editor 'e'.

Supported values for 'c' are currently:

```
FL_Home    -- moves the cursor to the beginning of the current line
FL_End     -- moves the cursor to the end of the current line
FL_Left    -- moves the cursor left one character
FL_Right   -- moves the cursor right one character
FL_Up      -- moves the cursor up one line
FL_Down    -- moves the cursor down one line
FL_Page_Up -- moves the cursor up one page
FL_Page_Down -- moves the cursor down one page
```

31.136.2.22 kf_page_down()

```
int Fl_Text_Editor::kf_page_down (
    int c,
    Fl_Text_Editor * e ) [static]
```

Moves the text cursor down one page for editor 'e'.

Same as kf_move(FL_Page_Down, e). The key value 'c' is currently unused.

31.136.2.23 kf_page_up()

```
int Fl_Text_Editor::kf_page_up (
    int c,
    Fl_Text_Editor * e ) [static]
```

Moves the text cursor up one page for editor 'e'.

Same as kf_move(FL_Page_Up, e). The key value 'c' is currently unused.

31.136.2.24 kf_paste()

```
int Fl_Text_Editor::kf_paste (
    int c,
    Fl_Text_Editor * e ) [static]
```

Does a paste of selected text in the current buffer of editor 'e'.

Any current selection is replaced with the pasted content. The key value 'c' is currently unused.

31.136.2.25 kf_right()

```
int Fl_Text_Editor::kf_right (
    int c,
    Fl_Text_Editor * e ) [static]
```

Moves the text cursor one character to the right for editor 'e'.

Same as kf_move(FL_Right, e). The key value 'c' is currently unused.

31.136.2.26 kf_select_all()

```
int Fl_Text_Editor::kf_select_all (
    int c,
    Fl_Text_Editor * e ) [static]
```

Selects all text in the current buffer in editor 'e'.

The key value 'c' is currently unused.

31.136.2.27 kf_shift_move()

```
int Fl_Text_Editor::kf_shift_move (
    int c,
    Fl_Text_Editor * e ) [static]
```

Extends the current selection in the direction of key 'c' in editor 'e'.

See also

[kf_move\(\)](#)

31.136.2.28 kf_undo()

```
int Fl_Text_Editor::kf_undo (
    int c,
    Fl_Text_Editor * e ) [static]
```

Undo last edit in the current buffer of editor 'e'.

Also deselects previous selection. The key value 'c' is currently unused.

31.136.2.29 kf_up()

```
int Fl_Text_Editor::kf_up (
    int c,
    Fl_Text_Editor * e ) [static]
```

Moves the text cursor one line up for editor 'e'.

Same as `kf_move(FL_Up, e)`. The key value 'c' is currently unused.

31.136.2.30 remove_key_binding()

```
void Fl_Text_Editor::remove_key_binding (
    int key,
    int state,
    Key_Binding ** list )
```

Removes the key binding associated with the key `key` of state `state` from the `Key_Binding` list `list`.

This can be used in derived classes to remove global key bindings by using the global (static) `Key_Binding` list `Fl_Text_Editor::global_key_bindings`.

31.136.2.31 tab_nav() [1/2]

```
int Fl_Text_Editor::tab_nav ( ) const
```

Check if Tab focus navigation is enabled.

If disabled (default), hitting Tab inserts a tab character into the editor buffer.

If enabled, hitting Tab navigates focus to the next widget, and Shift-Tab navigates focus to the previous widget.

Returns

if Tab inserts tab characters or moves the focus

Return values

0	Tab inserts tab characters (default)
1	Tab navigation is enabled.

See also

[tab_nav\(int\), Fl::OPTION_ARROW_FOCUS](#).

Version

1.3.4 ABI feature

31.136.2.32 tab_nav() [2/2]

```
void Fl_Text_Editor::tab_nav (
    int val )
```

Enables or disables Tab key focus navigation.

When disabled (default), tab characters are inserted into [Fl_Text_Editor](#). Only the mouse can change focus. This behavior is desirable when [Fl_Text_Editor](#) is used, e.g. in a source code editor.

When enabled, Tab navigates focus to the next widget, and Shift-Tab navigates focus to the previous widget. This behavior is desirable when [Fl_Text_Editor](#) is used e.g. in a database input form.

Currently, this method is implemented as a convenience method that adjusts the key bindings for the Tab key. This implementation detail may change in the future. Know that changing the editor's key bindings for Tab and Shift-Tab may affect tab navigation.

Parameters

in	val	
		If val is 0, Tab inserts a tab character (default). If val is 1, Tab navigates widget focus.

See also

[tab_nav\(\)](#), [Fl::OPTION_ARROW_FOCUS](#).

Version

1.3.4 ABI feature

31.136.3 Member Data Documentation**31.136.3.1 global_key_bindings**

```
Key_Binding* Fl_Text_Editor::global_key_bindings [static], [protected]
```

Global key binding list.

Derived classes can add key bindings for all [Fl_Text_Editor](#) widgets by adding a [Key_Binding](#) to this list.

See also

[add_key_binding\(int key, int state, Key_Func f, Key_Binding** list\);](#)

The documentation for this class was generated from the following files:

- [Fl_Text_Editor.H](#)
- [Fl_Text_Editor.cxx](#)

31.137 Fl_Text_Selection Class Reference

This is an internal class for [Fl_Text_Buffer](#) to manage text selections.

```
#include <Fl_Text_Buffer.H>
```

Public Member Functions

- int [end](#) () const
Return the byte offset to the character after the last selected character.
- int [includes](#) (int pos) const
Return true if position pos with indentation dispIndex is in the [Fl_Text_Selection](#).
- int [position](#) (int *start, int *end) const
Return the positions of this selection.
- bool [selected](#) () const

- Returns true if any text is selected.*
- void `selected` (bool b)
 - Modify the 'selected' flag.*
- void `set` (int start, int end)
 - Set the selection range.*
- int `start` () const
 - Return the byte offset to the first selected character.*
- void `update` (int pos, int nDeleted, int nInserted)
 - Updates a selection after text was modified.*

Protected Attributes

- int `mEnd`
 - byte offset to the character after the last selected character*
- bool `mSelected`
 - this flag is set if any text is selected*
- int `mStart`
 - byte offset to the first selected character*

Friends

- class `Fl_Text_Buffer`

31.137.1 Detailed Description

This is an internal class for `Fl_Text_Buffer` to manage text selections.

This class works correctly with UTF-8 strings assuming that the parameters for all calls are on character boundaries.

31.137.2 Member Function Documentation

31.137.2.1 `end()`

```
int Fl_Text_Selection::end ( ) const [inline]
```

Return the byte offset to the character after the last selected character.

Returns

byte offset

31.137.2.2 `position()`

```
int Fl_Text_Selection::position (
    int * start,
    int * end ) const
```

Return the positions of this selection.

Parameters

<i>start</i>	return byte offset to first selected character
<i>end</i>	return byte offset pointing after last selected character

Returns

true if selected

31.137.2.3 selected() [1/2]

```
bool Fl_Text_Selection::selected ( ) const [inline]
```

Returns true if any text is selected.

Returns

a non-zero number if any text has been selected, or 0 if no text is selected.

31.137.2.4 selected() [2/2]

```
void Fl_Text_Selection::selected (
    bool b ) [inline]
```

Modify the 'selected' flag.

Parameters

<i>b</i>	new flag
----------	----------

31.137.2.5 set()

```
void Fl_Text_Selection::set (
    int start,
    int end )
```

Set the selection range.

Parameters

<i>start</i>	byte offset to first selected character
<i>end</i>	byte offset pointing after last selected character

31.137.2.6 start()

```
int Fl_Text_Selection::start ( ) const [inline]
```

Return the byte offset to the first selected character.

Returns

byte offset

31.137.2.7 update()

```
void Fl_Text_Selection::update (
    int pos,
    int nDeleted,
    int nInserted )
```

Updates a selection after text was modified.

Updates an individual selection for changes in the corresponding text

Parameters

<i>pos</i>	byte offset into text buffer at which the change occurred
<i>nDeleted</i>	number of bytes deleted from the buffer
<i>nInserted</i>	number of bytes inserted into the buffer

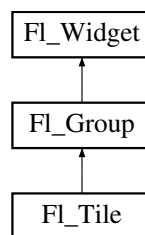
The documentation for this class was generated from the following files:

- Fl_Text_Buffer.H
- Fl_Text_Buffer.cxx

31.138 Fl_Tile Class Reference

The [Fl_Tile](#) class lets you resize its children by dragging the border between them.

Inheritance diagram for Fl_Tile:



Public Member Functions

- [Fl_Tile](#) (int X, int Y, int W, int H, const char *L=0)
Creates a new [Fl_Tile](#) widget using the given position, size, and label string.
- int [handle](#) (int event)
Handles the specified event.
- void [position](#) (int oldx, int oldy, int newx, int newy)
Drags the intersection at (oldx,oldy) to (newx,newy).
- void [resize](#) (int X, int Y, int W, int H)
Resizes the [Fl_Tile](#) widget and its children.

Additional Inherited Members

31.138.1 Detailed Description

The [Fl_Tile](#) class lets you resize its children by dragging the border between them.

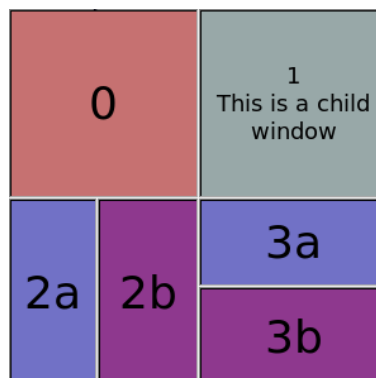


Figure 31.43 Fl_Tile

For the tiling to work correctly, the children of an [FL_Tile](#) must cover the entire area of the widget, but not overlap. This means that all children must touch each other at their edges, and no gaps can be left inside the [FL_Tile](#).

[FL_Tile](#) does not normally draw any graphics of its own. The "borders" which can be seen in the snapshot above are actually part of the children. Their boxtypes have been set to `FL_DOWN_BOX` creating the impression of "ridges" where the boxes touch. What you see are actually two adjacent `FL_DOWN_BOX`'s drawn next to each other. All neighboring widgets share the same edge - the widget's thick borders make it appear as though the widgets aren't actually touching, but they are. If the edges of adjacent widgets do not touch, then it will be impossible to drag the corresponding edges.

[FL_Tile](#) allows objects to be resized to zero dimensions. To prevent this you can use the [resizable\(\)](#) to limit where corners can be dragged to. For more information see note below.

Even though objects can be resized to zero sizes, they must initially have non-zero sizes so the [FL_Tile](#) can figure out their layout. If desired, call [position\(\)](#) after creating the children but before displaying the window to set the borders where you want.

Note on [resizable\(FL_Widget &w\)](#): The "resizable" child widget (which should be invisible) limits where the borders can be dragged to. All dragging will be limited inside the resizable widget's borders. If you don't set it, it will be possible to drag the borders right to the edges of the [FL_Tile](#) widget, and thus resize objects on the edges to zero width or height. When the entire [FL_Tile](#) widget is resized, the [resizable\(\)](#) widget will keep its border distance to all borders the same (this is normal resize behavior), so that you can effectively set a border width that will never change. To ensure correct event delivery to all child widgets the [resizable\(\)](#) widget must be the first child of the [FL_Tile](#) widget group. Otherwise some events (e.g. `FL_MOVE` and `FL_ENTER`) might be consumed by the [resizable\(\)](#) widget so that they are lost for widgets covered (overlapped) by the [resizable\(\)](#) widget.

Note

You can still resize widgets **inside** the [resizable\(\)](#) to zero width and/or height, i.e. box **2b** above to zero width and box **3a** to zero height.

See also

void [FL_Group::resizable\(FL_Widget &w\)](#)

Example for resizable with 20 pixel border distance:

```
int dx = 20, dy = dx;
FL_Tile tile(50,50,300,300);
// create resizable() box first
FL_Box r(tile.x()+dx,tile.y()+dy,tile.w()-2*dx,tile.h()-2*dy);
tile.resizable(r);
// ... create widgets inside tile (see test/tile.cxx) ...
tile.end();
```

See also the complete example program in `test/tile.cxx`.

31.138.2 Constructor & Destructor Documentation

31.138.2.1 FL_Tile()

```
FL_Tile::FL_Tile (
    int X,
    int Y,
    int W,
    int H,
    const char * L = 0 )
```

Creates a new [FL_Tile](#) widget using the given position, size, and label string.

The default boxtype is `FL_NO_BOX`.

The destructor *also deletes all the children*. This allows a whole tree to be deleted at once, without having to keep a pointer to all the children in the user code. A kludge has been done so the [FL_Tile](#) and all of its children can be automatic (local) variables, but you must declare the [FL_Tile](#) *first*, so that it is destroyed last.

See also

class [FL_Group](#)

31.138.3 Member Function Documentation

31.138.3.1 handle()

```
int Fl_Tile::handle (
    int event ) [virtual]
```

Handles the specified event.

You normally don't call this method directly, but instead let FLTK do it when the user interacts with the widget.

When implemented in a widget, this function must return 0 if the widget does not use the event or 1 otherwise.

Most of the time, you want to call the inherited [handle\(\)](#) method in your overridden method so that you don't short-circuit events that you don't handle. In this last case you should return the callee retval.

Parameters

in	<i>event</i>	the kind of event received
----	--------------	----------------------------

Return values

0	if the event was not used or understood
1	if the event was used and can be deleted

See also

[Fl_Event](#)

Reimplemented from [Fl_Group](#).

31.138.3.2 position()

```
void Fl_Tile::position (
    int oldx,
    int oldy,
    int newx,
    int newy )
```

Drags the intersection at (oldx,oldy) to (newx,newy).

This redraws all the necessary children.

Pass zero as oldx or oldy to disable drag in that direction.

31.138.3.3 resize()

```
void Fl_Tile::resize (
    int X,
    int Y,
    int W,
    int H ) [virtual]
```

Resizes the [Fl_Tile](#) widget and its children.

[Fl_Tile](#) implements its own [resize\(\)](#) method. It does not use [Fl_Group::resize\(\)](#) to resize itself and its children.

Enlarging works by just moving the lower-right corner and resizing the bottom and right border widgets accordingly. Shrinking the [Fl_Tile](#) works in the opposite way by shrinking the bottom and right border widgets, unless they are reduced to zero width or height, resp. or to their minimal sizes defined by the [resizable\(\)](#) widget. In this case other widgets will be shrunk as well.

See the [Fl_Tile](#) class documentation about how the [resizable\(\)](#) works.

Reimplemented from [Fl_Group](#).

The documentation for this class was generated from the following files:

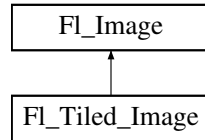
- [Fl_Tile.H](#)
- [Fl_Tile.cxx](#)

31.139 FI_Tiled_Image Class Reference

This class supports tiling of images over a specified area.

```
#include <Fl_Tiled_Image.H>
```

Inheritance diagram for FI_Tiled_Image:



Public Member Functions

- virtual void [color_average](#) ([FI_Color](#) c, float i)

The [color_average\(\)](#) method averages the colors in the image with the FLTK color value c.
- [FI_Image](#) * [copy](#) ()
- virtual [FI_Image](#) * [copy](#) (int W, int H)

The [copy\(\)](#) method creates a copy of the specified image.
- virtual void [desaturate](#) ()

The [desaturate\(\)](#) method converts an image to grayscale.
- void [draw](#) (int X, int Y)
- virtual void [draw](#) (int X, int Y, int W, int H, int cx, int cy)

Draws a tiled image.
- [FI_Tiled_Image](#) ([FI_Image](#) *i, int W=0, int H=0)
- [FI_Image](#) * [image](#) ()

Gets The image that is tiled.
- virtual ~[FI_Tiled_Image](#) ()

The destructor frees all memory and server resources that are used by the tiled image.

Protected Attributes

- int [alloc_image_](#)
- [FI_Image](#) * [image_](#)

Additional Inherited Members

31.139.1 Detailed Description

This class supports tiling of images over a specified area.

The source (tile) image is **not** copied unless you call the [color_average\(\)](#), [desaturate\(\)](#), or [inactive\(\)](#) methods.

31.139.2 Constructor & Destructor Documentation

31.139.2.1 FI_Tiled_Image()

```

Fl_Tiled_Image::Fl_Tiled_Image (
    FI\_Image * i,
    int W = 0,
    int H = 0 )
  
```

The constructors create a new tiled image containing the specified image.

Use a width and height of 0 to tile the whole window/widget.

Note

Due to implementation constraints in FLTK 1.3.3 and later width and height of 0 may not work as expected when used as background image in widgets other than windows. You may need to center and clip the image (label) and set the label type to `FL_NORMAL_LABEL`. Doing so will let the tiled image fill the whole widget as its background image. Other combinations of label flags may or may not work.

```
#include "bg.xpm"
Fl_Pixmap *bg_xpm = new Fl_Pixmap(bg_xpm);
Fl_Tiled_Image *bg_tiled = new Fl_Tiled_Image(bg_xpm,0,0);
Fl_Box *box = new Fl_Box(40,40,300,100,"");
box->box(FL_UP_BOX);
box->labeltype(FL_NORMAL_LABEL);
box->align(FL_ALIGN_INSIDE | FL_ALIGN_CENTER | FL_ALIGN_CLIP);
box->image(bg_tiled);
```

Note

Setting an image (label) for a window may not work as expected due to implementation constraints in FLTK 1.3.x and maybe later. The reason is the way `Fl::scheme()` initializes the window's label type and image. A possible workaround is to use another `Fl_Group` as the only child widget and to set the background image for this group as described above.

Todo Fix `Fl_Tiled_Image` as background image for widgets and windows and fix the implementation of `Fl::scheme(const char *)`.

31.139.3 Member Function Documentation**31.139.3.1 color_average()**

```
void Fl_Tiled_Image::color_average (
    Fl_Color c,
    float i ) [virtual]
```

The `color_average()` method averages the colors in the image with the FLTK color value `c`.

The `i` argument specifies the amount of the original image to combine with the color, so a value of 1.0 results in no color blend, and a value of 0.0 results in a constant image of the specified color.

An internal copy is made of the original image before changes are applied, to avoid modifying the original image.

Reimplemented from `Fl_Image`.

31.139.3.2 copy()

```
Fl_Image * Fl_Tiled_Image::copy (
    int W,
    int H ) [virtual]
```

The `copy()` method creates a copy of the specified image.

If the width and height are provided, the image is resized to the specified size. The image should be deleted (or in the case of `Fl_Shared_Image`, released) when you are done with it.

Reimplemented from `Fl_Image`.

31.139.3.3 desaturate()

```
void Fl_Tiled_Image::desaturate ( ) [virtual]
```

The `desaturate()` method converts an image to grayscale.

If the image contains an alpha channel (depth = 4), the alpha channel is preserved.

An internal copy is made of the original image before changes are applied, to avoid modifying the original image.

Reimplemented from `Fl_Image`.

31.139.3.4 draw()

```
void Fl_Tiled_Image::draw (
    int X,
    int Y,
    int W,
    int H,
    int cx,
    int cy ) [virtual]
```

Draws a tiled image.

Tiled images can be used as background images for widgets and windows. However, due to implementation constraints, you must take care when setting label types and alignment flags. Only certain combinations work as expected, others may yield unexpected results and undefined behavior.

This draw method can draw multiple copies of one image in an area given by X, Y, W, H.

The optional arguments `cx` and `cy` can be used to crop the image starting at offsets (`cx`, `cy`). `cx` and `cy` must be ≥ 0 (negative values are ignored). If one of the values is greater than the image width or height resp. (`cx` \geq `image()->w()` or `cy` \geq `image()->h()`) nothing is drawn, because the resulting image would be empty.

After calculating the resulting image size the image is drawn as often as necessary to fill the given area, starting at the top left corner.

If both `W` and `H` are 0 the image is repeated as often as necessary to fill the entire window, unless there is a valid clip region. If you want to fill only one particular widget's background, then you should either set a clip region in your `draw()` method or use the label alignment flags `FL_ALIGN_INSIDE|FL_ALIGN_CLIP` to make sure the image is clipped.

This may be improved in a later version of the library.

Reimplemented from [Fl_Image](#).

The documentation for this class was generated from the following files:

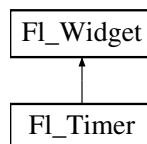
- `Fl_Tiled_Image.H`
- `Fl_Tiled_Image.cxx`

31.140 Fl_Timer Class Reference

This is provided only to emulate the Forms Timer widget.

```
#include <Fl_Timer.H>
```

Inheritance diagram for `Fl_Timer`:



Public Member Functions

- char `direction` () const
Gets or sets the direction of the timer.
- void `direction` (char d)
Gets or sets the direction of the timer.
- `Fl_Timer` (uchar t, int x, int y, int w, int h, const char *l)
Creates a new Fl_Timer widget using the given type, position, size, and label string.
- int `handle` (int)
Handles the specified event.
- char `suspended` () const
Gets or sets whether the timer is suspended.
- void `suspended` (char d)
Gets or sets whether the timer is suspended.

- double **value** () const
See void [Fl_Timer::value\(double\)](#)
- void **value** (double)
Sets the current timer value.
- **~Fl_Timer** ()
Destroys the timer and removes the timeout.

Protected Member Functions

- void **draw** ()
Draws the widget.

Additional Inherited Members

31.140.1 Detailed Description

This is provided only to emulate the Forms Timer widget. It works by making a timeout callback every 1/5 second. This is wasteful and inaccurate if you just want something to happen a fixed time in the future. You should directly call [Fl::add_timeout\(\)](#) instead.

31.140.2 Constructor & Destructor Documentation

31.140.2.1 Fl_Timer()

```
Fl_Timer::Fl_Timer (
    uchar t,
    int X,
    int Y,
    int W,
    int H,
    const char * l )
```

Creates a new [Fl_Timer](#) widget using the given type, position, size, and label string. The type parameter can be any of the following symbolic constants:

- `FL_NORMAL_TIMER` - The timer just does the callback and displays the string "Timer" in the widget.
- `FL_VALUE_TIMER` - The timer does the callback and displays the current timer value in the widget.
- `FL_HIDDEN_TIMER` - The timer just does the callback and does not display anything.

31.140.3 Member Function Documentation

31.140.3.1 direction() [1/2]

```
char Fl_Timer::direction ( ) const [inline]
```

Gets or sets the direction of the timer.

If the direction is zero then the timer will count up, otherwise it will count down from the initial [value\(\)](#).

31.140.3.2 direction() [2/2]

```
void Fl_Timer::direction (
    char d ) [inline]
```

Gets or sets the direction of the timer.

If the direction is zero then the timer will count up, otherwise it will count down from the initial [value\(\)](#).

31.140.3.3 draw()

```
void Fl_Timer::draw ( ) [protected], [virtual]
```

Draws the widget.

Never call this function directly. FLTK will schedule redrawing whenever needed. If your widget must be redrawn as soon as possible, call [redraw\(\)](#) instead.

Override this function to draw your own widgets.

If you ever need to call another widget's draw method *from within your own [draw\(\)](#) method*, e.g. for an embedded scrollbar, you can do it (because [draw\(\)](#) is virtual) like this:

```
Fl_Widget *s = &scroll;           // scroll is an embedded Fl_Scrollbar
s->draw();                          // calls Fl_Scrollbar::draw()
```

Implements [Fl_Widget](#).

31.140.3.4 handle()

```
int Fl_Timer::handle (
    int event ) [virtual]
```

Handles the specified event.

You normally don't call this method directly, but instead let FLTK do it when the user interacts with the widget.

When implemented in a widget, this function must return 0 if the widget does not use the event or 1 otherwise.

Most of the time, you want to call the inherited [handle\(\)](#) method in your overridden method so that you don't short-circuit events that you don't handle. In this last case you should return the callee retval.

Parameters

in	<i>event</i>	the kind of event received
----	--------------	----------------------------

Return values

0	if the event was not used or understood
1	if the event was used and can be deleted

See also

[Fl_Event](#)

Reimplemented from [Fl_Widget](#).

31.140.3.5 suspended()

```
char Fl_Timer::suspended ( ) const [inline]
```

Gets or sets whether the timer is suspended.

The documentation for this class was generated from the following files:

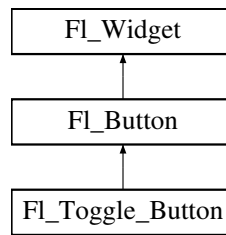
- [Fl_Timer.H](#)
- [forms_timer.cxx](#)

31.141 Fl_Toggle_Button Class Reference

The toggle button is a push button that needs to be clicked once to toggle on, and one more time to toggle off.

```
#include <Fl_Toggle_Button.H>
```

Inheritance diagram for [Fl_Toggle_Button](#):



Public Member Functions

- [Fl_Toggle_Button](#) (int X, int Y, int W, int H, const char *l=0)
Creates a new [Fl_Toggle_Button](#) widget using the given position, size, and label string.

Additional Inherited Members

31.141.1 Detailed Description

The toggle button is a push button that needs to be clicked once to toggle on, and one more time to toggle off. The [Fl_Toggle_Button](#) subclass displays the "on" state by drawing a pushed-in button. Buttons generate callbacks when they are clicked by the user. You control exactly when and how by changing the values for [type\(\)](#) and [when\(\)](#).

31.141.2 Constructor & Destructor Documentation

31.141.2.1 Fl_Toggle_Button()

```

Fl_Toggle_Button::Fl_Toggle_Button (
    int X,
    int Y,
    int W,
    int H,
    const char * L = 0 )
  
```

Creates a new [Fl_Toggle_Button](#) widget using the given position, size, and label string. The constructor creates the button using the given position, size, and label. The inherited destructor deletes the toggle button. The Button [type\(\)](#) is set to FL_TOGGLE_BUTTON.

Parameters

in	<i>X,Y,W,H</i>	position and size of the widget
in	<i>L</i>	widget label, default is no label

The documentation for this class was generated from the following files:

- [Fl_Toggle_Button.H](#)
- [Fl_Button.cxx](#)

31.142 FL_Tooltip Class Reference

The [Fl_Tooltip](#) class provides tooltip support for all FLTK widgets.

```
#include <Fl_Tooltip.H>
```

Static Public Member Functions

- static [Fl_Color color](#) ()

- Gets the background color for tooltips.*

 - static void `color` (`FI_Color` c)
- Sets the background color for tooltips.*

 - static `FI_Widget` * `current` ()
- Gets the current widget target.*

 - static void `current` (`FI_Widget` *)
- Sets the current widget target.*

 - static float `delay` ()
- Gets the tooltip delay.*

 - static void `delay` (float f)
- Sets the tooltip delay.*

 - static void `disable` ()
- Same as `enable(0)`, disables tooltips on all widgets.*

 - static void `enable` (int b=1)
- Enables tooltips on all widgets (or disables if b is false).*

 - static int `enabled` ()
- Returns non-zero if tooltips are enabled.*

 - static void `enter_area` (`FI_Widget` *w, int X, int Y, int W, int H, const char *tip)
- You may be able to use this to provide tooltips for internal pieces of your widget.*

 - static `FI_Font` font ()
- Gets the typeface for the tooltip text.*

 - static void `font` (`FI_Font` i)
- Sets the typeface for the tooltip text.*

 - static float `hoverdelay` ()
- Gets the tooltip hover delay, the delay between tooltips.*

 - static void `hoverdelay` (float f)
- Sets the tooltip hover delay, the delay between tooltips.*

 - static int `margin_height` ()
- Gets the amount of extra space above and below the tooltip's text.*

 - static void `margin_height` (int v)
- Sets the amount of extra space above and below the tooltip's text.*

 - static int `margin_width` ()
- Gets the amount of extra space left/right of the tooltip's text.*

 - static void `margin_width` (int v)
- Sets the amount of extra space left/right of the tooltip's text.*

 - static `FI_Fontsize` size ()
- Gets the size of the tooltip text.*

 - static void `size` (`FI_Fontsize` s)
- Sets the size of the tooltip text.*

 - static `FI_Color` `textcolor` ()
- Gets the color of the text in the tooltip.*

 - static void `textcolor` (`FI_Color` c)
- Sets the color of the text in the tooltip.*

 - static int `wrap_width` ()
- Gets the maximum width for tooltip's text before it word wraps.*

 - static void `wrap_width` (int v)
- Sets the maximum width for tooltip's text before it word wraps.*

Static Public Attributes

- static void(* `enter`)(`FI_Widget` *w) = nothing
- static void(* `exit`)(`FI_Widget` *w) = nothing

Friends

- void `Fl_Widget::copy_tooltip` (const char *)
- void `Fl_Widget::tooltip` (const char *)

31.142.1 Detailed Description

The `Fl_Tooltip` class provides tooltip support for all FLTK widgets. It contains only static methods.

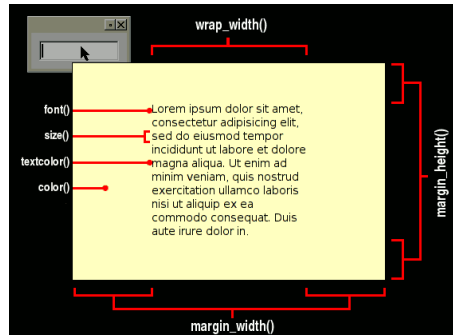


Figure 31.44 Fl_Tooltip Options

31.142.2 Member Function Documentation

31.142.2.1 color() [1/2]

```
static Fl_Color Fl_Tooltip::color ( ) [inline], [static]
```

Gets the background color for tooltips.

The default background color is a pale yellow.

31.142.2.2 color() [2/2]

```
static void Fl_Tooltip::color (
    Fl_Color c ) [inline], [static]
```

Sets the background color for tooltips.

The default background color is a pale yellow.

31.142.2.3 current()

```
void Fl_Tooltip::current (
    Fl_Widget * w ) [static]
```

Sets the current widget target.

Acts as though `enter(widget)` was done but does not pop up a tooltip. This is useful to prevent a tooltip from reappearing when a modal overlapping window is deleted. FLTK does this automatically when you click the mouse button.

31.142.2.4 delay() [1/2]

```
static float Fl_Tooltip::delay ( ) [inline], [static]
```

Gets the tooltip delay.

The default delay is 1.0 seconds.

31.142.2.5 delay() [2/2]

```
static void Fl_Tooltip::delay (
    float f ) [inline], [static]
```

Sets the tooltip delay.

The default delay is 1.0 seconds.

31.142.2.6 disable()

```
static void Fl_Tooltip::disable ( ) [inline], [static]
```

Same as enable(0), disables tooltips on all widgets.

31.142.2.7 enable()

```
static void Fl_Tooltip::enable (
    int b = 1 ) [inline], [static]
```

Enables tooltips on all widgets (or disables if *b* is false).

31.142.2.8 enabled()

```
static int Fl_Tooltip::enabled ( ) [inline], [static]
```

Returns non-zero if tooltips are enabled.

31.142.2.9 enter_area()

```
void Fl_Tooltip::enter_area (
    Fl_Widget * wid,
    int x,
    int y,
    int w,
    int h,
    const char * t ) [static]
```

You may be able to use this to provide tooltips for internal pieces of your widget.

Call this after setting [Fl::belowmouse\(\)](#) to your widget (because that calls the above enter() method). Then figure out what thing the mouse is pointing at, and call this with the widget (this pointer is used to remove the tooltip if the widget is deleted or hidden, and to locate the tooltip), the rectangle surrounding the area, relative to the top-left corner of the widget (used to calculate where to put the tooltip), and the text of the tooltip (which must be a pointer to static data as it is not copied).

31.142.2.10 font() [1/2]

```
static Fl_Font Fl_Tooltip::font ( ) [inline], [static]
```

Gets the typeface for the tooltip text.

31.142.2.11 font() [2/2]

```
static void Fl_Tooltip::font (
    Fl_Font i ) [inline], [static]
```

Sets the typeface for the tooltip text.

31.142.2.12 hoverdelay() [1/2]

```
static float Fl_Tooltip::hoverdelay ( ) [inline], [static]
```

Gets the tooltip hover delay, the delay between tooltips.

The default delay is 0.2 seconds.

31.142.2.13 hoverdelay() [2/2]

```
static void Fl_Tooltip::hoverdelay (
    float f ) [inline], [static]
```

Sets the tooltip hover delay, the delay between tooltips.

The default delay is 0.2 seconds.

31.142.2.14 margin_height() [1/2]

```
static int Fl_Tooltip::margin_height ( ) [inline], [static]
```

Gets the amount of extra space above and below the tooltip's text.

Default is 3.

31.142.2.15 margin_height() [2/2]

```
static void Fl_Tooltip::margin_height (
    int v ) [inline], [static]
```

Sets the amount of extra space above and below the tooltip's text.

Default is 3.

31.142.2.16 margin_width() [1/2]

```
static int Fl_Tooltip::margin_width ( ) [inline], [static]
```

Gets the amount of extra space left/right of the tooltip's text.

Default is 3.

31.142.2.17 margin_width() [2/2]

```
static void Fl_Tooltip::margin_width (
    int v ) [inline], [static]
```

Sets the amount of extra space left/right of the tooltip's text.

Default is 3.

31.142.2.18 size() [1/2]

```
static Fl_Fontsize Fl_Tooltip::size ( ) [inline], [static]
```

Gets the size of the tooltip text.

31.142.2.19 size() [2/2]

```
static void Fl_Tooltip::size (
    Fl_Fontsize s ) [inline], [static]
```

Sets the size of the tooltip text.

31.142.2.20 textcolor() [1/2]

```
static Fl_Color Fl_Tooltip::textcolor ( ) [inline], [static]
```

Gets the color of the text in the tooltip.

The default is black.

31.142.2.21 textcolor() [2/2]

```
static void Fl_Tooltip::textcolor (
    Fl_Color c ) [inline], [static]
```

Sets the color of the text in the tooltip.
The default is black.

31.142.2.22 wrap_width() [1/2]

```
static int Fl_Tooltip::wrap_width ( ) [inline], [static]
```

Gets the maximum width for tooltip's text before it word wraps.
Default is 400.

31.142.2.23 wrap_width() [2/2]

```
static void Fl_Tooltip::wrap_width (
    int v ) [inline], [static]
```

Sets the maximum width for tooltip's text before it word wraps.
Default is 400.

The documentation for this class was generated from the following files:

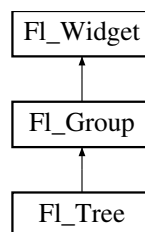
- Fl_Tooltip.H
- Fl.cxx
- Fl_Tooltip.cxx

31.143 Fl_Tree Class Reference

Tree widget.

```
#include <Fl_Tree.H>
```

Inheritance diagram for Fl_Tree:

**Public Member Functions**

- `Fl_Tree_Item * add` (const char *path, `Fl_Tree_Item *newitem=0`)
Adds a new item, given a menu style 'path'.
- `Fl_Tree_Item * add` (`Fl_Tree_Item *parent_item`, const char *name)
Add a new child item labeled 'name' to the specified 'parent_item'.
- void `calc_dimensions` ()
Recalculate widget dimensions and scrollbar visibility, normally managed automatically.
- void `calc_tree` ()
Recalculates the tree's sizes and scrollbar visibility, normally managed automatically.
- `Fl_Tree_Item * callback_item` ()
Gets the item that caused the callback.
- void `callback_item` (`Fl_Tree_Item *item`)
Sets the item that was changed for this callback.
- `Fl_Tree_Reason callback_reason` () const
Gets the reason for this callback.

- void `callback_reason` (`FI_Tree_Reason` reason)
 - Sets the reason for this callback.*
- void `clear` ()
 - Clear the entire tree's children, including the root.*
- void `clear_children` (`FI_Tree_Item` *item)
 - Clear all the children for 'item'.*
- int `close` (const char *path, int docallback=1)
 - Closes the item specified by 'path'.*
- int `close` (`FI_Tree_Item` *item, int docallback=1)
 - Closes the specified 'item'.*
- `FI_Image` * `closeicon` () const
 - Returns the icon to be used as the 'close' icon.*
- void `closeicon` (`FI_Image` *val)
 - Sets the icon to be used as the 'close' icon.*
- `FI_Color` `connectorcolor` () const
 - Get the connector color used for tree connection lines.*
- void `connectorcolor` (`FI_Color` val)
 - Set the connector color used for tree connection lines.*
- `FI_Tree_Connector` `connectorstyle` () const
 - Returns the line drawing style for inter-connecting items.*
- void `connectorstyle` (`FI_Tree_Connector` val)
 - Sets the line drawing style for inter-connecting items.*
- int `connectorwidth` () const
 - Gets the width of the horizontal connection lines (in pixels) that appear to the left of each tree item's label.*
- void `connectorwidth` (int val)
 - Sets the width of the horizontal connection lines (in pixels) that appear to the left of each tree item's label.*
- int `deselect` (const char *path, int docallback=1)
 - Deselect an item specified by 'path'.*
- int `deselect` (`FI_Tree_Item` *item, int docallback=1)
 - Deselect the specified item.*
- int `deselect_all` (`FI_Tree_Item` *item=0, int docallback=1)
 - Deselect 'item' and all its children.*
- void `display` (`FI_Tree_Item` *item)
 - Displays 'item', scrolling the tree as necessary.*
- int `displayed` (`FI_Tree_Item` *item)
 - See if 'item' is currently displayed on-screen (visible within the widget).*
- void `draw` ()
 - Standard FLTK `draw()` method, handles drawing the tree widget.*
- int `extend_selection` (`FI_Tree_Item` *from, `FI_Tree_Item` *to, int val=1, bool visible=false)
 - Extend a selection between 'from' and 'to' depending on 'visible'.*
- int `extend_selection_dir` (`FI_Tree_Item` *from, `FI_Tree_Item` *to, int dir, int val, bool visible)
 - Extend the selection between and including 'from' and 'to' depending on direction 'dir', 'val', and 'visible'.*
- `FI_Tree_Item` * `find_clicked` (int yonly=0)
 - Non-const version of `FI_Tree::find_clicked(int yonly)` const.*
- const `FI_Tree_Item` * `find_clicked` (int yonly=0) const
 - Find the item that was last clicked on.*
- `FI_Tree_Item` * `find_item` (const char *path)
 - Non-const version of `FI_Tree::find_item(const char *path)` const.*
- const `FI_Tree_Item` * `find_item` (const char *path) const
 - Find the item, given a menu style path, e.g.*

- [FI_Tree_Item](#) * [first](#) ()
Returns the first item in the tree, or 0 if none.
- [FI_Tree_Item](#) * [first_selected_item](#) ()
Returns the first selected item in the tree.
- [FI_Tree_Item](#) * [first_visible](#) ()
Returns the first [open\(\)](#), visible item in the tree, or 0 if none.
- [FI_Tree_Item](#) * [first_visible_item](#) ()
Returns the first [open\(\)](#), visible item in the tree, or 0 if none.
- [FI_Tree](#) (int X, int Y, int W, int H, const char *L=0)
Constructor.
- [FI_Tree_Item](#) * [get_item_focus](#) () const
Get the item that currently has keyboard focus.
- int [get_selected_items](#) ([FI_Tree_Item_Array](#) &ret_items)
Returns the currently selected items as an array of 'ret_items'.
- int [handle](#) (int e)
Standard FLTK event handler for this widget.
- int [hposition](#) () const
Returns the horizontal scroll position as a pixel offset.
- void [hposition](#) (int pos)
Sets the horizontal scroll offset to position 'pos'.
- [FI_Tree_Item](#) * [insert](#) ([FI_Tree_Item](#) *item, const char *name, int pos)
Insert a new item 'name' into 'item's children at position 'pos'.
- [FI_Tree_Item](#) * [insert_above](#) ([FI_Tree_Item](#) *above, const char *name)
Inserts a new item 'name' above the specified [FI_Tree_Item](#) 'above'.
- int [is_close](#) (const char *path) const
See if item specified by 'path' is closed.
- int [is_close](#) ([FI_Tree_Item](#) *item) const
See if the specified 'item' is closed.
- int [is_hscroll_visible](#) () const
See if the horizontal scrollbar is currently visible.
- int [is_open](#) (const char *path) const
See if item specified by 'path' is open.
- int [is_open](#) ([FI_Tree_Item](#) *item) const
See if 'item' is open.
- int [is_scrollbar](#) ([FI_Widget](#) *w)
See if widget 'w' is one of the [FI_Tree](#) widget's scrollbars.
- int [is_selected](#) (const char *path)
See if item specified by 'path' is selected.
- int [is_selected](#) ([FI_Tree_Item](#) *item) const
See if the specified 'item' is selected.
- int [is_vscroll_visible](#) () const
See if the vertical scrollbar is currently visible.
- [FI_Tree_Item](#) * [item_clicked](#) ()
Return the item that was last clicked.
- [FI_Tree_Item_Draw_Mode](#) [item_draw_mode](#) () const
Get the 'item draw mode' used for the tree.
- void [item_draw_mode](#) ([FI_Tree_Item_Draw_Mode](#) mode)
Set the 'item draw mode' used for the tree to 'mode'.
- void [item_draw_mode](#) (int mode)
Set the 'item draw mode' used for the tree to integer 'mode'.
- void [item_labelbgcolor](#) ([FI_Color](#) val)

- Set the default label background color used for creating new items.*

 - [FI_Color item_labelbgcolor](#) (void) const
- Get the default label background color used for creating new items.*

 - void [item_labelbgcolor](#) ([FI_Color](#) val)
- Set the default label foreground color used for creating new items.*

 - [FI_Color item_labelfgcolor](#) (void) const
- Get the default label foreground color used for creating new items.*

 - [FI_Font item_labelfont](#) () const
- Get the default font face used for creating new items.*

 - void [item_labelfont](#) ([FI_Font](#) val)
- Set the default font face used for creating new items.*

 - [FI_Fontsize item_labelsize](#) () const
- Get the default label fontsize used for creating new items.*

 - void [item_labelsize](#) ([FI_Fontsize](#) val)
- Set the default label font size used for creating new items.*

 - int [item_pathname](#) (char *pathname, int pathnamelen, const [FI_Tree_Item](#) *item) const
- Return 'pathname' of size 'pathnamelen' for the specified 'item'.*

 - [FI_Tree_Item_Reselect_Mode item_reselect_mode](#) () const
- Returns the current item re/selection mode.*

 - void [item_reselect_mode](#) ([FI_Tree_Item_Reselect_Mode](#) mode)
- Sets the item re/selection mode.*

 - int [labelmarginleft](#) () const
- Get the amount of white space (in pixels) that should appear to the left of the label text.*

 - void [labelmarginleft](#) (int val)
- Set the amount of white space (in pixels) that should appear to the left of the label text.*

 - [FI_Tree_Item](#) * [last](#) ()
- Returns the last item in the tree.*

 - [FI_Tree_Item](#) * [last_selected_item](#) ()
- Returns the last selected item in the tree.*

 - [FI_Tree_Item](#) * [last_visible](#) ()
- Returns the last [open\(\)](#), visible item in the tree.*

 - [FI_Tree_Item](#) * [last_visible_item](#) ()
- Returns the last [open\(\)](#), visible item in the tree.*

 - int [linespacing](#) () const
- Get the amount of white space (in pixels) that should appear between items in the tree.*

 - void [linespacing](#) (int val)
- Sets the amount of white space (in pixels) that should appear between items in the tree.*

 - void [load](#) (class [FI_Preferences](#) &)
- Load FLTK preferences.*

 - int [marginbottom](#) () const
- Get the amount of white space (in pixels) that should appear below the last visible item when the vertical scroller is scrolled to the bottom.*

 - void [marginbottom](#) (int val)
- Sets the amount of white space (in pixels) that should appear below the last visible item when the vertical scroller is scrolled to the bottom.*

 - int [marginleft](#) () const
- Get the amount of white space (in pixels) that should appear between the widget's left border and the tree's contents.*

 - void [marginleft](#) (int val)
- Set the amount of white space (in pixels) that should appear between the widget's left border and the left side of the tree's contents.*

 - int [margintop](#) () const

Get the amount of white space (in pixels) that should appear between the widget's top border and the top of the tree's contents.

- void **margintop** (int val)

Sets the amount of white space (in pixels) that should appear between the widget's top border and the top of the tree's contents.
- **Fl_Tree_Item** * **next** (**Fl_Tree_Item** *item=0)

Return the next item after 'item', or 0 if no more items.
- **Fl_Tree_Item** * **next_item** (**Fl_Tree_Item** *item, int dir=**FL_Down**, bool visible=false)

Returns next item after 'item' in direction 'dir' depending on 'visible'.
- **Fl_Tree_Item** * **next_selected_item** (**Fl_Tree_Item** *item=0, int dir=**FL_Down**)

Returns the next selected item above or below 'item', depending on 'dir'.
- **Fl_Tree_Item** * **next_visible_item** (**Fl_Tree_Item** *start, int dir)

Returns next **open()**, visible item above (dir==**FL_Up**) or below (dir==**FL_Down**) the specified 'item', or 0 if no more items.
- int **open** (const char *path, int docallback=1)

Opens the item specified by 'path'.
- int **open** (**Fl_Tree_Item** *item, int docallback=1)

Open the specified 'item'.
- void **open_toggle** (**Fl_Tree_Item** *item, int docallback=1)

Toggle the open state of 'item'.
- int **openchild_marginbottom** () const

Get the amount of white space (in pixels) that should appear below an open child tree's contents.
- void **openchild_marginbottom** (int val)

Set the amount of white space (in pixels) that should appear below an open child tree's contents.
- **Fl_Image** * **openicon** () const

Returns the icon to be used as the 'open' icon.
- void **openicon** (**Fl_Image** *val)

Sets the icon to be used as the 'open' icon.
- const **Fl_Tree_Prefs** & **prefs** () const
- **Fl_Tree_Item** * **prev** (**Fl_Tree_Item** *item=0)

Return the previous item before 'item', or 0 if no more items.
- void **recalc_tree** ()

Schedule tree to recalc the entire tree size.
- int **remove** (**Fl_Tree_Item** *item)

Remove the specified 'item' from the tree.
- void **resize** (int, int, int, int)

Resizes the **Fl_Group** widget and all of its children.
- **Fl_Tree_Item** * **root** ()

Returns the root item.
- void **root** (**Fl_Tree_Item** *newitem)

Sets the root item to 'newitem'.
- void **root_label** (const char *new_label)

Set the label for the root item to 'new_label'.
- int **scrollbar_size** () const

Gets the default size of scrollbars' troughs for this widget in pixels.
- void **scrollbar_size** (int size)

Sets the pixel size of the scrollbars' troughs to 'size' for this widget, in pixels.
- int **select** (const char *path, int docallback=1)

Select the item specified by 'path'.
- int **select** (**Fl_Tree_Item** *item, int docallback=1)

Select the specified 'item'.

- int `select_all` (`FI_Tree_Item *item=0`, `int docallback=1`)
Select 'item' and all its children.
- int `select_only` (`FI_Tree_Item *selitem`, `int docallback=1`)
Select only the specified item, deselecting all others that might be selected.
- void `select_toggle` (`FI_Tree_Item *item`, `int docallback=1`)
Toggle the select state of the specified 'item'.
- `FI_Boxtype selectbox` () const
Sets the style of box used to draw selected items.
- void `selectbox` (`FI_Boxtype val`)
Gets the style of box used to draw selected items.
- `FI_Tree_Select selectmode` () const
Gets the tree's current selection mode.
- void `selectmode` (`FI_Tree_Select val`)
Sets the tree's selection mode.
- void `set_item_focus` (`FI_Tree_Item *item`)
Set the item that currently should have keyboard focus.
- void `show_item` (`FI_Tree_Item *item`)
Adjust the vertical scrollbar to show 'item' at the top of the display IF it is currently off-screen (for instance `show_item_top()`).
- void `show_item` (`FI_Tree_Item *item`, `int yoff`)
Adjust the vertical scrollbar so that 'item' is visible 'yoff' pixels from the top of the `FI_Tree` widget's display.
- void `show_item_bottom` (`FI_Tree_Item *item`)
Adjust the vertical scrollbar so that 'item' is at the bottom of the display.
- void `show_item_middle` (`FI_Tree_Item *item`)
Adjust the vertical scrollbar so that 'item' is in the middle of the display.
- void `show_item_top` (`FI_Tree_Item *item`)
Adjust the vertical scrollbar so that 'item' is at the top of the display.
- void `show_self` ()
Print the tree as 'ascii art' to stdout.
- int `showcollapse` () const
Returns 1 if the collapse icon is enabled, 0 if not.
- void `showcollapse` (`int val`)
Set if we should show the collapse icon or not.
- int `showroot` () const
Returns 1 if the root item is to be shown, or 0 if not.
- void `showroot` (`int val`)
Set if the root item should be shown or not.
- `FI_Tree_Sort sortorder` () const
Set the default sort order used when items are added to the tree.
- void `sortorder` (`FI_Tree_Sort val`)
Gets the sort order used to add items to the tree.
- `FI_Image * usericon` () const
Returns the `FI_Image` being used as the default user icon for all newly created items.
- void `usericon` (`FI_Image *val`)
Sets the `FI_Image` to be used as the default user icon for all newly created items.
- int `usericonmarginleft` () const
Get the amount of white space (in pixels) that should appear to the left of the usericon.
- void `usericonmarginleft` (`int val`)
Set the amount of white space (in pixels) that should appear to the left of the usericon.
- int `vposition` () const
Returns the vertical scroll position as a pixel offset.

- void `vposition` (int pos)
Sets the vertical scroll offset to position 'pos'.
- int `widgetmarginleft` () const
Get the amount of white space (in pixels) that should appear to the left of the child fltk widget (if any).
- void `widgetmarginleft` (int val)
Set the amount of white space (in pixels) that should appear to the left of the child fltk widget (if any).
- `~FI_Tree` ()
Destructor.

Protected Member Functions

- void `do_callback_for_item` (FI_Tree_Item *item, FI_Tree_Reason reason)
Do the callback for the specified 'item' using 'reason', setting the `callback_item()` and `callback_reason()`.
- void `item_clicked` (FI_Tree_Item *val)
Set the item that was last clicked.

Protected Attributes

- `FI_Scrollbar * _hscroll`
Horizontal scrollbar.
- int `_tih`
Tree widget inner xywh dimension: inside borders + scrollbars.
- int `_tiw`
- int `_tix`
- int `_tiy`
- int `_toh`
Tree widget outer xywh dimension: outside scrollbars, inside widget border.
- int `_tow`
- int `_tox`
- int `_toy`
- int `_tree_h`
the calculated height of the entire tree hierarchy. See `calc_tree()`
- int `_tree_w`
the calculated width of the entire tree hierarchy. See `calc_tree()`
- `FI_Scrollbar * _vscroll`
Vertical scrollbar.

Friends

- class `FI_Tree_Item`

Additional Inherited Members

31.143.1 Detailed Description

Tree widget.

```
\image html tree-simple.png "Fl_Tree example program"
\image latex tree-simple.png "Fl_Tree example program" width=4cm

Fl_Tree                                     // Top level widget
|--- Fl_Tree_Item                           // Items in the tree
|--- Fl_Tree_Prefs                          // Preferences for the tree
|   |--- Fl_Tree_Connector (enum)           // Connection modes
|   |--- Fl_Tree_Select (enum)             // Selection modes
|   |--- Fl_Tree_Sort (enum)               // Sort behavior
```

Similar to `FI_Browser`, `FI_Tree` is a browser of `FI_Tree_Item`'s arranged in a parented hierarchy, or 'tree'. Subtrees can be expanded or closed. Items can be added, deleted, inserted, sorted and re-ordered.

The tree items may also contain other FLTK widgets, like buttons, input fields, or even "custom" widgets.

The `callback()` is invoked depending on the value of `when()`:

- FL_WHEN_RELEASE -- callback invoked when left mouse button is released on an item
- FL_WHEN_CHANGED -- callback invoked when left mouse changes selection state

The simple way to define a tree:

```
#include <FL/Fl_Tree.H>
[... ]
Fl_Tree tree(X, Y, W, H);
tree.begin();
    tree.add("Flintstones/Fred");
    tree.add("Flintstones/Wilma");
    tree.add("Flintstones/Pebbles");
    tree.add("Simpsons/Homer");
    tree.add("Simpsons/Marge");
    tree.add("Simpsons/Bart");
    tree.add("Simpsons/Lisa");
tree.end();
```

FEATURES

Items can be added with [add\(\)](#), removed with [remove\(\)](#), completely cleared with [clear\(\)](#), inserted with [insert\(\)](#) and [insert_above\(\)](#), selected/deselected with [select\(\)](#) and [deselect\(\)](#), open/closed with [open\(\)](#) and [close\(\)](#), positioned on the screen with [show_item_top\(\)](#), [show_item_middle\(\)](#) and [show_item_bottom\(\)](#), item children can be swapped around with [Fl_Tree_Item::swap_children\(\)](#), sorting can be controlled when items are [add\(\)](#)ed via [sortorder\(\)](#). You can walk the entire tree with [first\(\)](#) and [next\(\)](#). You can walk visible items with [first_visible_item\(\)](#) and [next_visible_item\(\)](#). You can walk selected items with [first_selected_item\(\)](#) and [next_selected_item\(\)](#). Items can be found by their pathname using [find_item\(const char*\)](#), and an item's pathname can be found with [item_pathname\(\)](#). The selected items' colors are controlled by [selection_color\(\)](#) (inherited from [Fl_Widget](#)). A hook is provided to allow you to redefine how item's labels are drawn via [Fl_Tree::item_draw_callback\(\)](#).

SELECTION OF ITEMS

The tree can have different selection behaviors controlled by [selectmode\(\)](#). The background color used for selected items is the [Fl_Tree::selection_color\(\)](#). The foreground color for selected items is controlled internally with [fl_contrast\(\)](#).

CHILD WIDGETS

FLTK widgets (including custom widgets) can be assigned to tree items via [Fl_Tree_Item::widget\(\)](#).

When an [Fl_Tree_Item::widget\(\)](#) is defined, the default behavior is for the [widget\(\)](#) to be shown in place of the item's label (if it has one). Only the [widget\(\)](#)'s width will be used; the [widget\(\)](#)'s [x\(\)](#) and [y\(\)](#) position will be managed by the tree, and the [h\(\)](#) will track the item's height. This default behavior can be altered (ABI 1.3.1): Setting [Fl_Tree::item_draw_mode\(\)](#)'s [FL_TREE_ITEM_DRAW_LABEL_AND_WIDGET](#) flag causes the label + widget to be displayed together in that order, and adding the [FL_TREE_ITEM_HEIGHT_FROM_WIDGET](#) flag causes widget's height to define the [widget\(\)](#)'s height.

ICONS

The tree's open/close icons can be redefined with [Fl_Tree::openicon\(\)](#), [Fl_Tree::closeicon\(\)](#). User icons can either be changed globally with [Fl_Tree::usericon\(\)](#), or on a per-item basis with [Fl_Tree_Item::usericon\(\)](#).

Various default preferences can be globally manipulated via [Fl_Tree_Prefs](#), including colors, margins, icons, connection lines, etc.

FONTS AND COLORS

When adding new items to the tree, the new items get the defaults for fonts and colors from:

- `Fl_Tree::item_labelfont()` – The default item label font (default: `FL_HELVETICA`)
- `Fl_Tree::item_labelsize()` – The default item label size (default: `FL_NORMAL_SIZE`)
- `Fl_Tree::item_labelfgcolor()` – The default item label foreground color (default: `FL_FOREGROUND_COLOR`)
- `Fl_Tree::item_labelbgcolor()` – The default item label background color (default: `0xffffffff`, which tree uses as 'transparent')

Each item (`Fl_Tree_Item`) inherits a copy of these font/color attributes when created, and each item has its own methods to let the app change these values on a per-item basis using methods of the same name:

- `Fl_Tree_Item::labelfont()` – The item's label font (default: `FL_HELVETICA`)
- `Fl_Tree_Item::labelsizesize()` – The item's label size (default: `FL_NORMAL_SIZE`)
- `Fl_Tree_Item::labelfgcolor()` – The item's label foreground color (default: `FL_FOREGROUND_COLOR`)
- `Fl_Tree_Item::labelbgcolor()` – The item's label background color (default: `0xffffffff`, which uses the tree's own bg color)

CALLBACKS

The tree's `callback()` will be invoked when items change state or are open/closed. `when()` controls when mouse/keyboard events invoke the callback. `callback_item()` and `callback_reason()` can be used to determine the cause of the callback. e.g.

```
void MyTreeCallback(Fl_Widget *w, void *data) {
    Fl_Tree *tree = (Fl_Tree*)w;
    Fl_Tree_Item *item = (Fl_Tree_Item*)tree->callback_item(); // get selected item
    switch ( tree->callback_reason() ) {
        case FL_TREE_REASON_SELECTED: [..]
        case FL_TREE_REASON_DESELECTED: [..]
        case FL_TREE_REASON_RESELECTED: [..]
        case FL_TREE_REASON_OPENED: [..]
        case FL_TREE_REASON_CLOSED: [..]
    }
}
```

SIMPLE EXAMPLES

To find all the selected items:

```
for ( Fl_Tree_Item *i=first_selected_item(); i; i=next_selected_item(i) )
    printf("Item %s is selected\n", i->label());
```

To get an item's full menu pathname, use `Fl_Tree::item_pathname()`, e.g.

```
char pathname[256] = "???" ;
tree->item_pathname(pathname, sizeof(pathname), item); // eg. "Parent/Child/Item"
```

To walk all the items of the tree from top to bottom:

```
// Walk all the items in the tree, and print their labels
for ( Fl_Tree_Item *item = tree->first(); item; item = tree->next(item) ) {
    printf("Item: %s\n", item->label());
}
```

To recursively walk all the children of a particular item, define a function that uses recursion:

```
// Find all of the item's children and print an indented report of their labels
void my_print_all_children(Fl_Tree_Item *item, int indent=0) {
    for ( int t=0; t<item->children(); t++ ) {
        printf("%*s Item: %s\n", indent, "", item->child(t)->label());
        my_print_all_children(item->child(t), indent+4); // recurse
    }
}
```

To change the default label font and color when creating new items:

```
tree = new Fl_Tree(..);
tree->item_labelfont(FL_COURIER); // Use Courier font for all new items
tree->item_labelcolor(FL_RED); // Use red color for labels of all new items
[..]
// Now create the items in the tree using the above defaults.
tree->add("Aaa");
tree->add("Bbb");
[..]
```

To change the font and color of all existing items in the tree:

```
// Change the font and color of all items currently in the tree
for ( Fl_Tree_Item *item = tree->first(); item; item = tree->next(item) ) {
    item->labelfont(FL_COURIER);
    item->labelcolor(FL_RED);
}
```

DISPLAY DESCRIPTION

The following image shows the tree's various visual elements and the methods that control them:

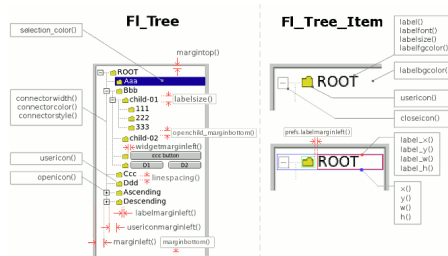


Figure 31.45 Fl_Tree elements

The following shows the protected dimension variables 'tree inner' (tix..) and 'tree outer' (tox..):

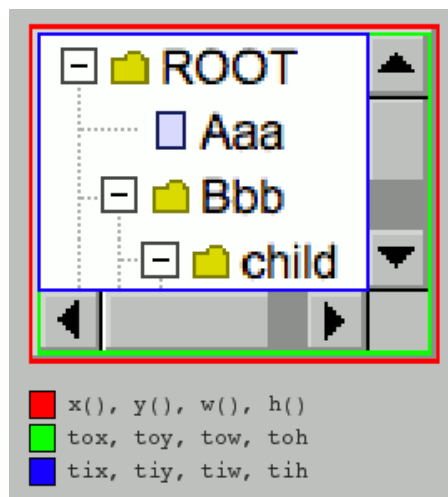


Figure 31.46 Fl_Tree inner/outer dimensions

KEYBOARD BINDINGS

The following table lists keyboard bindings for navigating the tree:

Table 31.466 Keyboard bindings.

Keyboard	FL_TREE_SELECT_↔ MULTI	FL_TREE_SELECT_↔ SINGLE	FL_TREE_SELECT_↔ NONE
Ctrl-A (Linux/Windows) Command-A (Mac)	Select all items.	N/A	N/A
Space	Selects item.	Selects item.	N/A
Ctrl-Space	Toggle item.	Toggle item.	N/A
Shift-Space	Extends selection from last item.	Selects item.	N/A
Enter, Ctrl-Enter, Shift-Enter	Toggles open/close	Toggles open/close	Toggles open/close
Right / Left	Open/Close item.	Open/Close item.	Open/Close item.
Up / Down	Move focus box up/down.	Move focus box up/down.	N/A
Shift-Up / Shift-Down	Extend selection up/down.	Move focus up/down.	N/A
Home / End	Move to top/bottom of tree.	Move to top/bottom of tree.	Move to top/bottom of tree.
PageUp / PageDown	Page up/down.	Page up/down.	Page up/down.

31.143.2 Member Function Documentation

31.143.2.1 add() [1/2]

```
Fl_Tree_Item * Fl_Tree::add (
    const char * path,
    Fl_Tree_Item * item = 0 )
```

Adds a new item, given a menu style 'path'.

Any parent nodes that don't already exist are created automatically. Adds the item based on the value of [sortorder\(\)](#). If 'item' is NULL, a new item is created.

To specify items or submenus that contain slashes ('/' or '\') use an escape character to protect them, e.g.

```
tree->add("/Holidays/Photos/12\\/25\\/2010"); // Adds item "12/25/2010"
tree->add("/Pathnames/c:\\\\Program Files\\\\MyApp"); // Adds item "c:\Program Files\MyApp"
```

Parameters

in	<i>path</i>	The path to the item, e.g. "Flintstone/Fred".
in	<i>item</i>	The new item to be added. If NULL, a new item is created with a name that is the last element in 'path'.

Returns

The new item added, or 0 on error.

Version

1.3.3

31.143.2.2 add() [2/2]

```
Fl_Tree_Item * Fl_Tree::add (
    Fl_Tree_Item * parent_item,
    const char * name )
```

Add a new child item labeled 'name' to the specified 'parent_item'.

Parameters

in	<i>parent_item</i>	The parent item the new child item will be added to. Must not be NULL.
in	<i>name</i>	The label for the new item

Returns

The new item added.

Version

1.3.0 release

31.143.2.3 calc_dimensions()

```
void Fl_Tree::calc_dimensions ( )
```

Recalculate widget dimensions and scrollbar visibility, normally managed automatically.

Low overhead way to update the tree widget's outer/inner dimensions and re-determine scrollbar visibility based on these changes without recalculating the entire size of the tree data.

Assumes that either the tree's size in `_tree_w/_tree_h` are correct so that scrollbar visibility can be calculated easily, or are both zero indicating scrollbar visibility can't be calculated yet.

This method is called when the widget is [resize\(\)](#)ed or if the scrollbar's sizes are changed (affects tree widget's inner dimensions `tix/y/w/h`), and also used by [calc_tree\(\)](#).

Version

1.3.3 ABI feature

31.143.2.4 calc_tree()

```
void Fl_Tree::calc_tree ( )
```

Recalculates the tree's sizes and scrollbar visibility, normally managed automatically.

On return:

- `_tree_w` will be the overall pixel width of the entire viewable tree
- `_tree_h` will be the overall pixel height "
- scrollbar visibility and pan sizes are updated
- internal `_tix/_tiy/_tiw/_tih` dimensions are updated

`_tree_w/_tree_h` include the tree's margins (e.g. [marginleft\(\)](#)), whether items are open or closed, label contents and font sizes, etc.

The tree hierarchy's size is managed separately from the widget's size as an optimization; this way [resize\(\)](#) on the widget doesn't involve recalculating the tree's hierarchy needlessly, as widget size has no bearing on the tree hierarchy.

The tree hierarchy's size only changes when items are added/removed, open/closed, label contents or font sizes changed, margins changed, etc.

This calculation involves walking the *entire* tree from top to bottom, potentially a slow calculation if the tree has many items (potentially hundreds of thousands), and should therefore be called sparingly.

For this reason, [recalc_tree\(\)](#) is used as a way to /schedule/ calculation when changes affect the tree hierarchy's size.

Apps may want to call this method directly if the app makes changes to the tree's geometry, then immediately needs to work with the tree's new dimensions before an actual redraw (and recalc) occurs. (This use by an app should only rarely be needed)

31.143.2.5 callback_item() [1/2]

```
Fl_Tree_Item * Fl_Tree::callback_item ( )
```

Gets the item that caused the callback.

The `callback()` can use this value to see which item changed.

31.143.2.6 callback_item() [2/2]

```
void Fl_Tree::callback_item (
    Fl_Tree_Item * item )
```

Sets the item that was changed for this callback.

Used internally to pass the item that invoked the callback.

31.143.2.7 callback_reason() [1/2]

```
Fl_Tree_Reason Fl_Tree::callback_reason ( ) const
```

Gets the reason for this callback.

The `callback()` can use this value to see why it was called. Example:

```
void MyTreeCallback(Fl_Widget *w, void *userdata) {
    Fl_Tree *tree = (Fl_Tree*)w;
    Fl_Tree_Item *item = tree->callback_item();    // the item changed (can be NULL if more than one item
    was changed!)
    switch ( tree->callback_reason() ) {           // reason callback was invoked
        case FL_TREE_REASON_OPENED:    ..item was opened..
        case FL_TREE_REASON_CLOSED:    ..item was closed..
        case FL_TREE_REASON_SELECTED:  ..item was selected..
        case FL_TREE_REASON_RESELECTED: ..item was reselected (double-clicked, etc)..
        case FL_TREE_REASON_DESELECTED: ..item was deselected..
    }
}
```

See also

[item_reselect_mode\(\)](#) – enables `FL_TREE_REASON_RESELECTED` events

31.143.2.8 callback_reason() [2/2]

```
void Fl_Tree::callback_reason (
    Fl_Tree_Reason reason )
```

Sets the reason for this callback.

Used internally to pass the reason the callback was invoked.

31.143.2.9 clear()

```
void Fl_Tree::clear ( )
```

Clear the entire tree's children, including the root.

The tree will be left completely empty.

31.143.2.10 clear_children()

```
void Fl_Tree::clear_children (
    Fl_Tree_Item * item )
```

Clear all the children for 'item'.

Item may not be NULL.

31.143.2.11 close() [1/2]

```
int Fl_Tree::close (
    const char * path,
    int docallback = 1 )
```

Closes the item specified by 'path'.

Invokes the callback depending on the value of optional parameter 'docallback'.

Handles calling `redraw()` if anything changed.

Items or submenus that themselves contain slashes ('/' or '\') should be escaped, e.g. `close("Holidays/12\25\2010")`. The callback can use `callback_item()` and `callback_reason()` respectively to determine the item changed and the reason the callback was called.

Parameters

in	<i>path</i>	– the tree item's pathname (e.g. "Flintstones/Fred")
in	<i>docallback</i>	– A flag that determines if the <code>callback()</code> is invoked or not: <ul style="list-style-type: none"> • 0 - <code>callback()</code> is not invoked • 1 - <code>callback()</code> is invoked if item changed (default), <code>callback_reason()</code> will be <code>FL_TREE_REASON_CLOSED</code>

Returns

- 1 – OK: item closed
- 0 – OK: item was already closed, no change
- -1 – ERROR: item was not found

See also

[open\(\)](#), [close\(\)](#), [is_open\(\)](#), [is_close\(\)](#), [callback_item\(\)](#), [callback_reason\(\)](#)

31.143.2.12 `close()` [2/2]

```
int Fl_Tree::close (
    Fl_Tree_Item * item,
    int docallback = 1 )
```

Closes the specified 'item'.

Invokes the callback depending on the value of optional parameter 'docallback'.

Handles calling `redraw()` if anything changed.

The callback can use `callback_item()` and `callback_reason()` respectively to determine the item changed and the reason the callback was called.

Parameters

in	<i>item</i>	– the item to be closed. Must not be NULL.
in	<i>docallback</i>	– A flag that determines if the <code>callback()</code> is invoked or not: <ul style="list-style-type: none"> • 0 - <code>callback()</code> is not invoked • 1 - <code>callback()</code> is invoked if item changed (default), <code>callback_reason()</code> will be <code>FL_TREE_REASON_CLOSED</code>

Returns

- 1 – item was closed
- 0 – item was already closed, no change

See also

[open\(\)](#), [close\(\)](#), [is_open\(\)](#), [is_close\(\)](#), [callback_item\(\)](#), [callback_reason\(\)](#)

31.143.2.13 closeicon() [1/2]

```
Fl_Image * Fl_Tree::closeicon ( ) const
```

Returns the icon to be used as the 'close' icon.

If none was set, the internal default is returned, a simple '[-]' icon.

31.143.2.14 closeicon() [2/2]

```
void Fl_Tree::closeicon (
    Fl_Image * val )
```

Sets the icon to be used as the 'close' icon.

This overrides the built in default '[-]' icon.

Parameters

in	<i>val</i>	– The new image, or zero to use the default '[-]' icon.
----	------------	---

31.143.2.15 connectorstyle()

```
void Fl_Tree::connectorstyle (
    Fl_Tree_Connector val )
```

Sets the line drawing style for inter-connecting items.

See [Fl_Tree_Connector](#) for possible values.

31.143.2.16 deselect() [1/2]

```
int Fl_Tree::deselect (
    const char * path,
    int docallback = 1 )
```

Deselect an item specified by 'path'.

Invokes the callback depending on the value of optional parameter 'docallback'.

Handles calling [redraw\(\)](#) if anything changed.

Items or submenus that themselves contain slashes ('/' or '\') should be escaped, e.g. `deselect("← Holidays/12\25\2010")`.

The callback can use [callback_item\(\)](#) and [callback_reason\(\)](#) respectively to determine the item changed and the reason the callback was called.

Parameters

in	<i>path</i>	– the tree item's pathname (e.g. "Flintstones/Fred")
in	<i>docallback</i>	– A flag that determines if the callback() is invoked or not: <ul style="list-style-type: none"> • 0 - the callback() is not invoked • 1 - the callback() is invoked if item changed state (default), callback_reason() will be <code>FL_TREE_REASON_DESELECTED</code>

Returns

- 1 - OK: item's state was changed
- 0 - OK: item was already deselected, no change was made
- -1 - ERROR: item was not found

31.143.2.17 deselect() [2/2]

```
int Fl_Tree::deselect (
```

```

    Fl_Tree_Item * item,
    int docallback = 1 )

```

Deselect the specified `item`.

Invokes the callback depending on the value of optional parameter '`docallback`'.

Handles calling `redraw()` if anything changed.

The callback can use `callback_item()` and `callback_reason()` respectively to determine the item changed and the reason the callback was called.

Parameters

in	<code>item</code>	– the item to be deselected. Must not be NULL.
in	<code>docallback</code>	– A flag that determines if the <code>callback()</code> is invoked or not: <ul style="list-style-type: none"> • 0 - the <code>callback()</code> is not invoked • 1 - the <code>callback()</code> is invoked if item changed state (default), <code>callback_reason()</code> will be <code>FL_TREE_REASON_DESELECTED</code>

Returns

- 0 - item was already deselected, no change was made
- 1 - item's state was changed

31.143.2.18 `deselect_all()`

```

int Fl_Tree::deselect_all (
    Fl_Tree_Item * item = 0,
    int docallback = 1 )

```

Deselect '`item`' and all its children.

If `item` is NULL, `first()` is used.

Invokes the callback depending on the value of optional parameter '`docallback`'.

Handles calling `redraw()` if anything changed.

The callback can use `callback_item()` and `callback_reason()` respectively to determine the item changed and the reason the callback was called.

Parameters

in	<code>item</code>	The item that will be deselected (along with all its children). If NULL, <code>first()</code> is used.
in	<code>docallback</code>	– A flag that determines if the <code>callback()</code> is invoked or not: <ul style="list-style-type: none"> • 0 - the <code>callback()</code> is not invoked • 1 - the <code>callback()</code> is invoked for each item that changed state (default), <code>callback_reason()</code> will be <code>FL_TREE_REASON_DESELECTED</code>

Returns

Count of how many items were actually changed to the deselected state.

31.143.2.19 `display()`

```

void Fl_Tree::display (
    Fl_Tree_Item * item )

```

Displays '`item`', scrolling the tree as necessary.

Parameters

in	<i>item</i>	The item to be displayed. If NULL, first() is used.
----	-------------	---

31.143.2.20 displayed()

```
int Fl_Tree::displayed (
    Fl_Tree_Item * item )
```

See if '*item*' is currently displayed on-screen (visible within the widget).

This can be used to detect if the item is scrolled off-screen. Checks to see if the item's vertical position is within the top and bottom edges of the display window. This does NOT take into account the [hide\(\)](#) / [show\(\)](#) or [open\(\)](#) / [close\(\)](#) status of the item.

Parameters

in	<i>item</i>	The item to be checked. If NULL, first() is used.
----	-------------	---

Returns

1 if displayed, 0 if scrolled off screen or no items are in tree.

31.143.2.21 draw()

```
void Fl_Tree::draw (
    void ) [virtual]
```

Standard FLTK [draw\(\)](#) method, handles drawing the tree widget.

Reimplemented from [Fl_Group](#).

31.143.2.22 extend_selection()

```
int Fl_Tree::extend_selection (
    Fl_Tree_Item * from,
    Fl_Tree_Item * to,
    int val = 1,
    bool visible = false )
```

Extend a selection between '*from*' and '*to*' depending on '*visible*'.

Similar to the more efficient [extend_selection_dir\(Fl_Tree_Item*,Fl_Tree_Item*,int dir,int val,bool vis\)](#) method, but direction (up or down) doesn't need to be known.

We're less efficient because we search the tree for to/from, then operate on items in between. The more efficient method avoids the "search", but necessitates a direction to be specified to find '*to*'.

Used by SHIFT-click to extend a selection between two items inclusive.

Handles calling [redraw\(\)](#) if anything changed.

Parameters

in	<i>from</i>	Starting item
in	<i>to</i>	Ending item
in	<i>val</i>	Select or deselect items (0=deselect, 1=select, 2=toggle)
in	<i>visible</i>	true=affect only open() , visible items, false=affect open or closed items (default)

Returns

The number of items whose selection states were changed, if any.

Version

1.3.3 ABI feature

31.143.2.23 extend_selection_dir()

```
int Fl_Tree::extend_selection_dir (
    Fl_Tree_Item * from,
    Fl_Tree_Item * to,
    int dir,
    int val,
    bool visible )
```

Extend the selection between and including 'from' and 'to' depending on direction 'dir', 'val', and 'visible'.

Efficient: does not walk entire tree; starts with 'from' and stops at 'to' while moving in direction 'dir'. Dir must be specified though. If dir cannot be known in advance, such as during SHIFT-click operations, the method [extend_selection\(Fl_Tree_Item*,Fl_Tree_Item*,int,bool\)](#) should be used. Handles calling [redraw\(\)](#) if anything changed.

Parameters

in	<i>from</i>	Starting item
in	<i>to</i>	Ending item
in	<i>dir</i>	Direction to extend selection (FL_Up or FL_Down)
in	<i>val</i>	0=deselect, 1=select, 2=toggle
in	<i>visible</i>	true=affect only open() , visible items, false=affect open or closed items (default)

Returns

The number of items whose selection states were changed, if any.

Version

1.3.3

31.143.2.24 find_clicked()

```
const Fl_Tree_Item * Fl_Tree::find_clicked (
    int yonly = 0 ) const
```

Find the item that was last clicked on.

You should use [callback_item\(\)](#) instead, which is fast, and is meant to be used within a callback to determine the item clicked.

This method walks the entire tree looking for the first item that is under the mouse. (The value of the 'yonly' flag affects whether both x and y events are checked, or just y)

Use this method /only/ if you've subclassed [Fl_Tree](#), and are receiving events before [Fl_Tree](#) has been able to process and update [callback_item\(\)](#).

Parameters

in	<i>yonly</i>	- 0: check both event's X and Y values. - 1: only check event's Y value, don't care about X.
----	--------------	--

Returns

The item clicked, or NULL if no item was under the current event.

Version

1.3.0

1.3.3 ABI feature: added yonly parameter

31.143.2.25 find_item()

```
const Fl_Tree_Item * Fl_Tree::find_item (
    const char * path ) const
```

Find the item, given a menu style path, e.g.

"/Parent/Child/item". There is both a const and non-const version of this method. Const version allows pure const methods to use this method to do lookups without causing compiler errors.

To specify items or submenus that contain slashes ('/' or '\') use an escape character to protect them, e.g.

```
tree->add("/Holidays/Photos/12\\25\\2010"); // Adds item "12/25/2010"
tree->add("/Pathnames/c:\\\\Program Files\\\\MyApp"); // Adds item "c:\Program Files\MyApp"
```

Parameters

in	<i>path</i>	– the tree item's pathname to be found (e.g. "Flintstones/Fred")
----	-------------	--

Returns

The item, or NULL if not found.

See also

[item_pathname\(\)](#)

31.143.2.26 first()

```
Fl_Tree_Item * Fl_Tree::first ( )
```

Returns the first item in the tree, or 0 if none.

Use this to walk the tree in the forward direction, e.g.

```
for ( Fl_Tree_Item *item = tree->first(); item; item = tree->next(item) )
    printf("Item: %s\n", item->label());
```

Returns

First item in tree, or 0 if none (tree empty).

See also

[first\(\)](#), [next\(\)](#), [last\(\)](#), [prev\(\)](#)

31.143.2.27 first_selected_item()

```
Fl_Tree_Item * Fl_Tree::first_selected_item ( )
```

Returns the first selected item in the tree.

Use this to walk the tree from top to bottom looking for all the selected items, e.g.

```
// Walk tree forward, from top to bottom
for ( Fl_Tree_Item *i=tree->first_selected_item(); i; i=tree->next_selected_item(i) )
    printf("Selected item: %s\n", i->label());
```

Returns

The first selected item, or 0 if none.

See also

[first_selected_item\(\)](#), [last_selected_item\(\)](#), [next_selected_item\(\)](#)

31.143.2.28 first_visible()

```
Fl_Tree_Item * Fl_Tree::first_visible ( )
```

Returns the first [open\(\)](#), visible item in the tree, or 0 if none.

Deprecated in 1.3.3 ABI – use [first_visible_item\(\)](#) instead.

31.143.2.29 first_visible_item()

```
Fl_Tree_Item * Fl_Tree::first_visible_item ( )
```

Returns the first [open\(\)](#), visible item in the tree, or 0 if none.

Returns

First visible item in tree, or 0 if none.

See also

[first_visible_item\(\)](#), [last_visible_item\(\)](#), [next_visible_item\(\)](#)

Version

1.3.3

31.143.2.30 get_selected_items()

```
int Fl_Tree::get_selected_items (
    Fl_Tree_Item_Array & ret_items )
```

Returns the currently selected items as an array of 'ret_items'.

Example:

```
// Get selected items as an array
Fl_Tree_Item_Array items;
tree->get_selected_items(items);
// Manipulate the returned array
for ( int t=0; t<items.total(); t++ ) {
    Fl_Tree_Item &item = items[t];
    ..do stuff with each selected item..
}
```

Parameters

out	<i>ret_items</i>	The returned array of selected items.
-----	------------------	---------------------------------------

Returns

The number of items in the returned array.

See also

[first_selected_item\(\)](#), [next_selected_item\(\)](#)

Version

1.3.3 ABI feature

31.143.2.31 handle()

```
int Fl_Tree::handle (
    int e ) [virtual]
```

Standard FLTK event handler for this widget.

Todo add [Fl_Widget_Tracker](#) (see [Fl_Browser_cxx::handle\(\)](#))

Reimplemented from [Fl_Group](#).

31.143.2.32 hposition() [1/2]

```
int Fl_Tree::hposition ( ) const
```

Returns the horizontal scroll position as a pixel offset.

The position returned is how many pixels of the tree are scrolled off the left edge of the screen.

See also

[hposition\(int\)](#), [vposition\(\)](#), [vposition\(int\)](#)

Note

Must be using FLTK ABI 1.3.3 or higher for this to be effective.

31.143.2.33 hposition() [2/2]

```
void Fl_Tree::hposition (
    int pos )
```

Sets the horizontal scroll offset to position 'pos'.

The position is how many pixels of the tree are scrolled off the left edge of the screen.

Parameters

<code>in</code>	<code>pos</code>	The vertical position (in pixels) to scroll the tree to.
-----------------	------------------	--

See also

[hposition\(\)](#), [vposition\(\)](#), [vposition\(int\)](#)

Note

Must be using FLTK ABI 1.3.3 or higher for this to be effective.

31.143.2.34 insert()

```
Fl_Tree_Item * Fl_Tree::insert (
    Fl_Tree_Item * item,
    const char * name,
    int pos )
```

Insert a new item 'name' into 'item's children at position 'pos'.

If pos is out of range the new item is

- prepended if `pos < 0` or
- appended if `pos > item->children()`.

Note: `pos == children()` is not considered out of range: the item is appended to the child list. Example:

```
tree->add("Aaa/000"); // "000" is index 0 in Aaa's children
tree->add("Aaa/111"); // "111" is index 1 in Aaa's children
tree->add("Aaa/222"); // "222" is index 2 in Aaa's children
..
// How to use insert() to insert a new item between Aaa/111 + Aaa/222
Fl_Tree_Item *item = tree->find_item("Aaa"); // get parent item Aaa
if (item) tree->insert(item, "New item", 2); // insert as a child of Aaa at index #2
```

Parameters

in	<i>item</i>	The existing item to insert new child into. Must not be NULL.
in	<i>name</i>	The label for the new item
in	<i>pos</i>	The position of the new item in the child list

Returns

The new item added.

See also

[insert_above\(\)](#)

31.143.2.35 insert_above()

```
Fl_Tree_Item * Fl_Tree::insert_above (
    Fl_Tree_Item * above,
    const char * name )
```

Inserts a new item 'name' above the specified [Fl_Tree_Item](#) 'above'.

Example:

```
tree->add("Aaa/000"); // "000" is index 0 in Aaa's children
tree->add("Aaa/111"); // "111" is index 1 in Aaa's children
tree->add("Aaa/222"); // "222" is index 2 in Aaa's children
..
// How to use insert_above() to insert a new item above Aaa/222
Fl_Tree_Item *item = tree->find_item("Aaa/222"); // get item Aaa/222
if (item) tree->insert_above(item, "New item"); // insert new item above it
```

Parameters

in	<i>above</i>	– the item above which to insert the new item. Must not be NULL.
in	<i>name</i>	– the name of the new item

Returns

The new item added, or 0 if 'above' could not be found.

See also

[insert\(\)](#)

31.143.2.36 is_close() [1/2]

```
int Fl_Tree::is_close (
    const char * path ) const
```

See if item specified by 'path' is closed.

Items or submenus that themselves contain slashes ('/' or '\') should be escaped, e.g. `is_close("← Holidays/12\25\2010")`.

Parameters

<code>in</code>	<code>path</code>	– the tree item's pathname (e.g. "Flintstones/Fred")
-----------------	-------------------	--

Returns

- 1 - OK: item is closed
- 0 - OK: item is open
- -1 - ERROR: item was not found

31.143.2.37 is_close() [2/2]

```
int Fl_Tree::is_close (
    Fl_Tree_Item * item ) const
```

See if the specified 'item' is closed.

Parameters

<code>in</code>	<code>item</code>	– the item to be tested. Must not be NULL.
-----------------	-------------------	--

Returns

- 1 : item is closed
- 0 : item is open

31.143.2.38 is_hscroll_visible()

```
int Fl_Tree::is_hscroll_visible ( ) const
```

See if the horizontal scrollbar is currently visible.

Returns

1 if scrollbar visible, 0 if not.

Note

Must be using FLTK ABI 1.3.3 or higher for this to be effective.

31.143.2.39 is_open() [1/2]

```
int Fl_Tree::is_open (
    const char * path ) const
```

See if item specified by 'path' is open.

Items or submenus that themselves contain slashes ('/' or '\') should be escaped, e.g. `is_open("← Holidays/12\25\2010")`.

Items that are 'open' are themselves not necessarily visible; one of the item's parents might be closed.

Parameters

<code>in</code>	<code>path</code>	– the tree item's pathname (e.g. "Flintstones/Fred")
-----------------	-------------------	--

Returns

- 1 - OK: item is open
- 0 - OK: item is closed
- -1 - ERROR: item was not found

See also

[Fl_Tree_Item::visible_r\(\)](#)

31.143.2.40 is_open() [2/2]

```
int Fl_Tree::is_open (
    Fl_Tree_Item * item ) const
```

See if 'item' is open.

Items that are 'open' are themselves not necessarily visible; one of the item's parents might be closed.

Parameters

in	<i>item</i>	– the item to be tested. Must not be NULL.
----	-------------	--

Returns

- 1 : item is open
- 0 : item is closed

31.143.2.41 is_scrollbar()

```
int Fl_Tree::is_scrollbar (
    Fl_Widget * w )
```

See if widget 'w' is one of the [Fl_Tree](#) widget's scrollbars.

Use this to skip over the scrollbars when walking the [child\(\)](#) array. Example:

```
for ( int i=0; i<tree->children(); i++ ) { // walk children
    Fl_Widget *w = tree->child(i);
    if ( tree->is_scrollbar(w) ) continue; // skip scrollbars
    ..do work here..
}
```

Parameters

in	<i>w</i>	Widget to test
----	----------	----------------

Returns

1 if w is a scrollbar, 0 if not.

Todo should be const

31.143.2.42 is_selected() [1/2]

```
int Fl_Tree::is_selected (
    const char * path )
```

See if item specified by 'path' is selected.

Items or submenus that themselves contain slashes (/ or \) should be escaped, e.g. `is_selected("← Holidays/12\25\2010")`.

Parameters

in	<i>path</i>	– the tree item's pathname (e.g. "Flintstones/Fred")
----	-------------	--

Returns

- 1 : item selected
- 0 : item deselected
- -1 : item was not found

31.143.2.43 is_selected() [2/2]

```
int Fl_Tree::is_selected (
    Fl_Tree_Item * item ) const
```

See if the specified '*item*' is selected.

Parameters

in	<i>item</i>	– the item to be tested. Must not be NULL.
----	-------------	--

Returns

- 1 : item selected
- 0 : item deselected

31.143.2.44 is_vscroll_visible()

```
int Fl_Tree::is_vscroll_visible ( ) const
```

See if the vertical scrollbar is currently visible.

Returns

1 if scrollbar visible, 0 if not.

31.143.2.45 item_clicked() [1/2]

```
Fl_Tree_Item * Fl_Tree::item_clicked ( )
```

Return the item that was last clicked.

Valid only from within the [callback\(\)](#).

Returns

The item clicked, or 0 if none. 0 may also be used to indicate several items were clicked/changed.

Deprecated in 1.3.3 ABI – use [callback_item\(\)](#) instead.

31.143.2.46 item_clicked() [2/2]

```
void Fl_Tree::item_clicked (
    Fl_Tree_Item * item ) [protected]
```

Set the item that was last clicked.

Should only be used by subclasses needing to change this value. Normally [Fl_Tree](#) manages this value.

Deprecated in 1.3.3 ABI – use [callback_item\(\)](#) instead.

31.143.2.47 item_draw_mode() [1/3]

```
Fl_Tree_Item_Draw_Mode Fl_Tree::item_draw_mode ( ) const
```

Get the 'item draw mode' used for the tree.

Version

1.3.1 ABI feature

31.143.2.48 item_draw_mode() [2/3]

```
void Fl_Tree::item_draw_mode (
    Fl_Tree_Item_Draw_Mode mode )
```

Set the 'item draw mode' used for the tree to 'mode'.

This affects how items in the tree are drawn, such as when a widget() is defined. See [Fl_Tree_Item_Draw_Mode](#) for possible values.

Version

1.3.1 ABI feature

31.143.2.49 item_draw_mode() [3/3]

```
void Fl_Tree::item_draw_mode (
    int mode )
```

Set the 'item draw mode' used for the tree to integer 'mode'.

This affects how items in the tree are drawn, such as when a widget() is defined. See [Fl_Tree_Item_Draw_Mode](#) for possible values.

Version

1.3.1 ABI feature

31.143.2.50 item_labelbgcolor() [1/2]

```
void Fl_Tree::item_labelbgcolor (
    Fl_Color val )
```

Set the default label background color used for creating new items.

A special case is made for color 0xffffffff (default) which is treated as 'transparent'. To change the background color on a per-item basis, use [Fl_Tree_Item::labelbgcolor\(Fl_Color\)](#)

31.143.2.51 item_labelbgcolor() [2/2]

```
Fl_Color Fl_Tree::item_labelbgcolor (
    void ) const
```

Get the default label background color used for creating new items.

If the color is 0xffffffff, it is 'transparent'.

31.143.2.52 item_labelfgcolor()

```
void Fl_Tree::item_labelfgcolor (
    Fl_Color val )
```

Set the default label foreground color used for creating new items.

To change the foreground color on a per-item basis, use [Fl_Tree_Item::labelfgcolor\(Fl_Color\)](#)

31.143.2.53 item_labelfont()

```
void Fl_Tree::item_labelfont (
    Fl_Font val )
```

Set the default font face used for creating new items.

To change the font face on a per-item basis, use [Fl_Tree_Item::labelfont\(Fl_Font\)](#)

31.143.2.54 item_labelsize()

```
void Fl_Tree::item_labelsize (
    Fl_Fontsize val )
```

Set the default label font size used for creating new items.

To change the font size on a per-item basis, use [Fl_Tree_Item::labelsizes\(Fl_Fontsize\)](#)

31.143.2.55 item_pathname()

```
int Fl_Tree::item_pathname (
    char * pathname,
    int pathnamelen,
    const Fl_Tree_Item * item ) const
```

Return 'pathname' of size 'pathnamelen' for the specified 'item'.

If 'item' is NULL, [root\(\)](#) is used.

The tree's root will be included in the pathname if [showroot\(\)](#) is on.

Menu items or submenus that contain slashes ('/' or '\') in their names will be escaped with a backslash. This is symmetrical with the [add\(\)](#) function which uses the same escape pattern to set names.

Parameters

out	<i>pathname</i>	The string to use to return the pathname
in	<i>pathnamelen</i>	The maximum length of the string (including NULL). Must not be zero.
in	<i>item</i>	The item whose pathname is to be returned.

Returns

- 0 : OK (*pathname* returns the item's pathname)
- -1 : item not found (*pathname*="")
- -2 : *pathname* not large enough (*pathname*="")

See also

[find_item\(\)](#)

31.143.2.56 item_reselect_mode() [1/2]

```
Fl_Tree_Item_Reselect_Mode Fl_Tree::item_reselect_mode ( ) const
```

Returns the current item re/selection mode.

Version

1.3.1 ABI feature

31.143.2.57 item_reselect_mode() [2/2]

```
void Fl_Tree::item_reselect_mode (
    Fl_Tree_Item_Reselect_Mode mode )
```

Sets the item re/selection mode.

See [Fl_Tree_Item_Reselect_Mode](#) for possible values.

Version

1.3.1 ABI feature

31.143.2.58 last()`Fl_Tree_Item * Fl_Tree::last ()`

Returns the last item in the tree.

This can be used to walk the tree in reverse, e.g.

```
for ( Fl_Tree_Item *item = tree->last(); item; item = tree->prev() )
    printf("Item: %s\n", item->label());
```

Returns

Last item in the tree, or 0 if none (tree empty).

See also[first\(\)](#), [next\(\)](#), [last\(\)](#), [prev\(\)](#)**31.143.2.59 last_selected_item()**`Fl_Tree_Item * Fl_Tree::last_selected_item ()`

Returns the last selected item in the tree.

Use this to walk the tree in reverse from bottom to top looking for all the selected items, e.g.

```
// Walk tree in reverse, from bottom to top
for ( Fl_Tree_Item *i=tree->last_selected_item(); i; i=tree->next_selected_item(i, FL_Up) )
    printf("Selected item: %s\n", i->label());
```

Returns

The last selected item, or 0 if none.

See also[first_selected_item\(\)](#), [last_selected_item\(\)](#), [next_selected_item\(\)](#)**Version**

1.3.3

31.143.2.60 last_visible()`Fl_Tree_Item * Fl_Tree::last_visible ()`Returns the last [open\(\)](#), visible item in the tree.**Deprecated** in 1.3.3 – use [last_visible_item\(\)](#) instead.**31.143.2.61 last_visible_item()**`Fl_Tree_Item * Fl_Tree::last_visible_item ()`Returns the last [open\(\)](#), visible item in the tree.**Returns**

Last visible item in the tree, or 0 if none.

See also

[first_visible_item\(\)](#), [last_visible_item\(\)](#), [next_visible_item\(\)](#)

Version

1.3.3

31.143.2.62 load()

```
void Fl_Tree::load (
    class Fl_Preferences & prefs )
```

Load FLTK preferences.

Read a preferences database into the tree widget.

A preferences database is a hierarchical collection of data which can be directly loaded into the tree view for inspection.

Parameters

in	<i>prefs</i>	the Fl_Preferences database
----	--------------	---

31.143.2.63 next()

```
Fl_Tree_Item * Fl_Tree::next (
    Fl_Tree_Item * item = 0 )
```

Return the next item after 'item', or 0 if no more items.

Use this code to walk the entire tree:

```
for ( Fl_Tree_Item *i = tree->first(); i; i = tree->next(i) )
    printf("Item: %s\n", i->label());
```

Parameters

in	<i>item</i>	The item to use to find the next item. If NULL, returns 0.
----	-------------	--

Returns

Next item in tree, or 0 if at last item.

See also

[first\(\)](#), [next\(\)](#), [last\(\)](#), [prev\(\)](#)

31.143.2.64 next_item()

```
Fl_Tree_Item * Fl_Tree::next_item (
    Fl_Tree_Item * item,
    int dir = FL_Down,
    bool visible = false )
```

Returns next item after 'item' in direction 'dir' depending on 'visible'.

Next item will be above (if dir==FL_Up) or below (if dir==FL_Down). If 'visible' is true, only items whose parents are [open\(\)](#) will be returned. If 'visible' is false, even items whose parents are [close\(\)](#)ed will be returned.

If item is 0, the return value will be the result of this truth table:

visible=true	visible=false
--------------	---------------

```

dir=FL_Up:      last_visible_item()  last()
dir=FL_Down:   first_visible_item() first()

```

Example use:

```

// Walk down the tree showing open(), visible items
for ( Fl_Tree_Item *i=tree->first_visible_item(); i; i=tree->next_item(i, FL_Down, true) )
    printf("Item:  %s\n", i->label());
// Walk up the tree showing open(), visible items
for ( Fl_Tree_Item *i=tree->last_visible_item(); i; i=tree->next_item(i, FL_Up, true) )
    printf("Item:  %s\n", i->label());
// Walk down the tree showing all items (open or closed)
for ( Fl_Tree_Item *i=tree->first(); i; i=tree->next_item(i, FL_Down, false) )
    printf("Item:  %s\n", i->label());
// Walk up the tree showing all items (open or closed)
for ( Fl_Tree_Item *i=tree->last(); i; i=tree->next_item(i, FL_Up, false) )
    printf("Item:  %s\n", i->label());

```

Parameters

in	<i>item</i>	The item to use to find the next item. If NULL, returns 0.
in	<i>dir</i>	Can be FL_Up or FL_Down (default=FL_Down or 'next')
in	<i>visible</i>	true=return only open() , visible items, false=return open or closed items (default)

Returns

Next item in tree in the direction and visibility specified, or 0 if no more items of specified visibility in that direction.

See also

[first\(\)](#), [last\(\)](#), [next\(\)](#),
[first_visible_item\(\)](#), [last_visible_item\(\)](#), [next_visible_item\(\)](#),
[first_selected_item\(\)](#), [last_selected_item\(\)](#), [next_selected_item\(\)](#)

Version

1.3.3

31.143.2.65 next_selected_item()

```

Fl_Tree_Item * Fl_Tree::next_selected_item (
    Fl_Tree_Item * item = 0,
    int dir = FL_Down )

```

Returns the next selected item above or below 'item', depending on 'dir'.

If 'item' is 0, search starts at either [first\(\)](#) or [last\(\)](#), depending on 'dir': [first\(\)](#) if 'dir' is FL_Down (default), [last\(\)](#) if 'dir' is FL_Up.

Use this to walk the tree looking for all the selected items, e.g.

```

// Walk down the tree (forwards)
for ( Fl_Tree_Item *i=tree->first_selected_item(); i; i=tree->next_selected_item(i, FL_Down) )
    printf("Item:  %s\n", i->label());
// Walk up the tree (backwards)
for ( Fl_Tree_Item *i=tree->last_selected_item(); i; i=tree->next_selected_item(i, FL_Up) )
    printf("Item:  %s\n", i->label());

```

Parameters

in	<i>item</i>	The item above or below which we'll find the next selected item. If NULL, first() is used if FL_Down, last() if FL_Up. (default=NULL)
in	<i>dir</i>	The direction to go. FL_Up for moving up the tree, FL_Down for down the tree (default)

Returns

The next selected item, or 0 if there are no more selected items.

See also

[first_selected_item\(\)](#), [last_selected_item\(\)](#), [next_selected_item\(\)](#)

Version

1.3.3

31.143.2.66 next_visible_item()

```
Fl_Tree_Item * Fl_Tree::next_visible_item (
    Fl_Tree_Item * item,
    int dir )
```

Returns next [open\(\)](#), visible item above (`dir==FL_Up`) or below (`dir==FL_Down`) the specified 'item', or 0 if no more items.

If 'item' is 0, returns [last\(\)](#) if 'dir' is FL_Up, or [first\(\)](#) if dir is FL_Down.

```
// Walk down the tree (forwards)
for ( Fl_Tree_Item *i=tree->first_visible_item(); i; i=tree->next_visible_item(i, FL_Down) )
    printf("Item: %s\n", i->label());
// Walk up the tree (backwards)
for ( Fl_Tree_Item *i=tree->last_visible_item(); i; i=tree->next_visible_item(i, FL_Up) )
    printf("Item: %s\n", i->label());
```

Parameters

in	<i>item</i>	The item above/below which we'll find the next visible item
in	<i>dir</i>	The direction to search. Can be FL_Up or FL_Down.

Returns

The item found, or 0 if there's no visible items above/below the specified item.

Version

1.3.3

31.143.2.67 open() [1/2]

```
int Fl_Tree::open (
    const char * path,
    int docallback = 1 )
```

Opens the item specified by 'path'.

This causes the item's children (if any) to be shown.

Invokes the callback depending on the value of optional parameter 'docallback'.

Handles calling [redraw\(\)](#) if anything changed.

Items or submenus that themselves contain slashes ('/' or '\') should be escaped, e.g. `open("Holidays/12\25\2010")`.

The callback can use [callback_item\(\)](#) and [callback_reason\(\)](#) respectively to determine the item changed and the reason the callback was called.

Parameters

in	<i>path</i>	– the tree item's pathname (e.g. "Flintstones/Fred")
----	-------------	--

Parameters

in	<i>docalcallback</i>	– A flag that determines if the callback() is invoked or not: <ul style="list-style-type: none"> • 0 - callback() is not invoked • 1 - callback() is invoked if item changed (default), callback_reason() will be <code>FL_TREE_REASON_OPENED</code>
----	----------------------	--

Returns

- 1 – OK: item opened
- 0 – OK: item was already open, no change
- -1 – ERROR: item was not found

See also

[open\(\)](#), [close\(\)](#), [is_open\(\)](#), [is_close\(\)](#), [callback_item\(\)](#), [callback_reason\(\)](#)

31.143.2.68 open() [2/2]

```
int Fl_Tree::open (
    Fl_Tree_Item * item,
    int docalcallback = 1 )
```

Open the specified 'item'.

This causes the item's children (if any) to be shown.

Invokes the callback depending on the value of optional parameter 'docalcallback'.

Handles calling [redraw\(\)](#) if anything changed.

The callback can use [callback_item\(\)](#) and [callback_reason\(\)](#) respectively to determine the item changed and the reason the callback was called.

Parameters

in	<i>item</i>	– the item to be opened. Must not be NULL.
in	<i>docalcallback</i>	– A flag that determines if the callback() is invoked or not: <ul style="list-style-type: none"> • 0 - callback() is not invoked • 1 - callback() is invoked if item changed (default), callback_reason() will be <code>FL_TREE_REASON_OPENED</code>

Returns

- 1 – item was opened
- 0 – item was already open, no change

See also

[open\(\)](#), [close\(\)](#), [is_open\(\)](#), [is_close\(\)](#), [callback_item\(\)](#), [callback_reason\(\)](#)

31.143.2.69 open_toggle()

```
void Fl_Tree::open_toggle (
    Fl_Tree_Item * item,
    int docalcallback = 1 )
```

Toggle the open state of 'item'.

Invokes the callback depending on the value of optional parameter 'docallback'.

Handles calling [redraw\(\)](#) if anything changed.

The callback can use [callback_item\(\)](#) and [callback_reason\(\)](#) respectively to determine the item changed and the reason the callback was called.

Parameters

in	<i>item</i>	– the item whose open state is to be toggled. Must not be NULL.
in	<i>docallback</i>	– A flag that determines if the callback() is invoked or not: <ul style="list-style-type: none"> • 0 - callback() is not invoked • 1 - callback() is invoked (default), callback_reason() will be either FL_TREE_REASON_OPENED or FL_TREE_REASON_CLOSED

See also

[open\(\)](#), [close\(\)](#), [is_open\(\)](#), [is_close\(\)](#), [callback_item\(\)](#), [callback_reason\(\)](#)

31.143.2.70 openicon() [1/2]

```
Fl_Image * Fl_Tree::openicon ( ) const
```

Returns the icon to be used as the 'open' icon.

If none was set, the internal default is returned, a simple '[+]' icon.

31.143.2.71 openicon() [2/2]

```
void Fl_Tree::openicon (
    Fl_Image * val )
```

Sets the icon to be used as the 'open' icon.

This overrides the built in default '[+]' icon.

Parameters

in	<i>val</i>	– The new image, or zero to use the default '[+]' icon.
----	------------	---

31.143.2.72 prev()

```
Fl_Tree_Item * Fl_Tree::prev (
    Fl_Tree_Item * item = 0 )
```

Return the previous item before 'item', or 0 if no more items.

This can be used to walk the tree in reverse, e.g.

```
for ( Fl_Tree_Item *item = tree->first(); item; item = tree->prev(item) )
    printf("Item: %s\n", item->label());
```

Parameters

in	<i>item</i>	The item to use to find the previous item. If NULL, returns 0.
----	-------------	--

Returns

Previous item in tree, or 0 if at first item.

See also

[first\(\)](#), [next\(\)](#), [last\(\)](#), [prev\(\)](#)

31.143.2.73 recalc_tree()

```
void Fl_Tree::recalc_tree ( )
```

Schedule tree to recalc the entire tree size.

Note

Must be using FLTK ABI 1.3.3 or higher for this to be effective.

31.143.2.74 remove()

```
int Fl_Tree::remove (
    Fl_Tree_Item * item )
```

Remove the specified 'item' from the tree.

item may not be NULL. If it has children, all those are removed too. If item being removed has focus, no item will have focus.

Returns

0 if done, -1 if 'item' not found.

31.143.2.75 resize()

```
void Fl_Tree::resize (
    int X,
    int Y,
    int W,
    int H ) [virtual]
```

Resizes the [Fl_Group](#) widget and all of its children.

The [Fl_Group](#) widget first resizes itself, and then it moves and resizes all its children according to the rules documented for [Fl_Group::resizable\(Fl_Widget*\)](#)

See also

[Fl_Group::resizable\(Fl_Widget*\)](#)

[Fl_Group::resizable\(\)](#)

[Fl_Widget::resize\(int,int,int,int\)](#)

Reimplemented from [Fl_Group](#).

31.143.2.76 root()

```
void Fl_Tree::root (
    Fl_Tree_Item * newitem )
```

Sets the root item to 'newitem'.

If a root item already exists, [clear\(\)](#) is called first to clear it before replacing it with newitem. Use this to install a custom item (derived from [Fl_Tree_Item](#)) as the root of the tree. This allows the derived class to implement custom drawing by overriding [Fl_Tree_Item::draw_item_content\(\)](#).

Version

1.3.3

31.143.2.77 root_label()

```
void Fl_Tree::root_label (
    const char * new_label )
```

Set the label for the root item to 'new_label'.
Makes an internally managed copy of 'new_label'.

31.143.2.78 scrollbar_size() [1/2]

```
int Fl_Tree::scrollbar_size ( ) const
```

Gets the default size of scrollbars' troughs for this widget in pixels.
If this value is zero (default), this widget will use the global [Fl::scrollbar_size\(\)](#) value as the scrollbar's width.

Returns

Scrollbar size in pixels, or 0 if the global [Fl::scrollbar_size\(\)](#) is being used.

See also

[Fl::scrollbar_size\(int\)](#)

31.143.2.79 scrollbar_size() [2/2]

```
void Fl_Tree::scrollbar_size (
    int size )
```

Sets the pixel size of the scrollbars' troughs to 'size' for this widget, in pixels.
Normally you should not need this method, and should use the global [Fl::scrollbar_size\(int\)](#) instead to manage the size of ALL your widgets' scrollbars. This ensures your application has a consistent UI, and is the default behavior. Normally this is what you want.
Only use this method if you really need to override just THIS instance of the widget's scrollbar size. (This need should be rare.)
Setting *size* to the special value of 0 causes the widget to track the global [Fl::scrollbar_size\(\)](#), which is the default.

Parameters

in	<i>size</i>	Sets the scrollbar size in pixels. If 0 (default), scrollbar size tracks the global Fl::scrollbar_size()
----	-------------	---

See also

[Fl::scrollbar_size\(\)](#)

31.143.2.80 select() [1/2]

```
int Fl_Tree::select (
    const char * path,
    int docallback = 1 )
```

Select the item specified by 'path'.
Invokes the callback depending on the value of optional parameter 'docallback'.
Handles calling [redraw\(\)](#) if anything changed.
Items or submenus that themselves contain slashes ('/' or '\') should be escaped, e.g. `select("← Holidays/12\25\2010")`.
The callback can use [callback_item\(\)](#) and [callback_reason\(\)](#) respectively to determine the item changed and the reason the callback was called.

Parameters

in	<i>path</i>	– the tree item's pathname (e.g. "Flintstones/Fred")
in	<i>docallback</i>	– A flag that determines if the callback() is invoked or not: <ul style="list-style-type: none"> • 0 - the callback() is not invoked • 1 - the callback() is invoked if item changed state (default), callback_reason() will be FL_TREE_REASON_SELECTED

Returns

- 1 : OK: item's state was changed
- 0 : OK: item was already selected, no change was made
- -1 : ERROR: item was not found

31.143.2.81 select() [2/2]

```
int Fl_Tree::select (
    Fl_Tree_Item * item,
    int docallback = 1 )
```

Select the specified 'item'.

Use '[deselect\(\)](#)' to deselect it.

Invokes the callback depending on the value of optional parameter *docallback*.

Handles calling [redraw\(\)](#) if anything changed.

The callback can use [callback_item\(\)](#) and [callback_reason\(\)](#) respectively to determine the item changed and the reason the callback was called.

Parameters

in	<i>item</i>	– the item to be selected. Must not be NULL.
in	<i>docallback</i>	– A flag that determines if the callback() is invoked or not: <ul style="list-style-type: none"> • 0 - the callback() is not invoked • 1 - the callback() is invoked if item changed state, callback_reason() will be FL_TREE_REASON_SELECTED

Returns

- 1 - item's state was changed
- 0 - item was already selected, no change was made

31.143.2.82 select_all()

```
int Fl_Tree::select_all (
    Fl_Tree_Item * item = 0,
    int docallback = 1 )
```

Select 'item' and all its children.

If item is NULL, [first\(\)](#) is used.

Invokes the callback depending on the value of optional parameter '*docallback*'.

Handles calling [redraw\(\)](#) if anything changed.

The callback can use [callback_item\(\)](#) and [callback_reason\(\)](#) respectively to determine the item changed and the reason the callback was called.

Parameters

in	<i>item</i>	The item that will be selected (along with all its children). If NULL, first() is used.
in	<i>docallback</i>	– A flag that determines if the callback() is invoked or not: <ul style="list-style-type: none"> • 0 - the callback() is not invoked • 1 - the callback() is invoked for each item that changed state (default), callback_reason() will be FL_TREE_REASON_SELECTED

Returns

Count of how many items were actually changed to the selected state.

31.143.2.83 select_only()

```
int Fl_Tree::select_only (
    Fl_Tree_Item * selitem,
    int docallback = 1 )
```

Select only the specified item, deselecting all others that might be selected.

If 'selitem' is 0, [first\(\)](#) is used.

Invokes the callback depending on the value of optional parameter 'docallback'.

Handles calling [redraw\(\)](#) if anything changed.

The callback can use [callback_item\(\)](#) and [callback_reason\(\)](#) respectively to determine the item changed and the reason the callback was called.

Parameters

in	<i>selitem</i>	The item to be selected. If NULL, first() is used.
in	<i>docallback</i>	– A flag that determines if the callback() is invoked or not: <ul style="list-style-type: none"> • 0 - the callback() is not invoked • 1 - the callback() is invoked for each item that changed state (default), callback_reason() will be either FL_TREE_REASON_SELECTED or FL_TREE_REASON_DESELECTED

Returns

The number of items whose selection states were changed, if any.

31.143.2.84 select_toggle()

```
void Fl_Tree::select_toggle (
    Fl_Tree_Item * item,
    int docallback = 1 )
```

Toggle the select state of the specified 'item'.

Invokes the callback depending on the value of optional parameter 'docallback'.

Handles calling [redraw\(\)](#) if anything changed.

The callback can use [callback_item\(\)](#) and [callback_reason\(\)](#) respectively to determine the item changed and the reason the callback was called.

Parameters

in	<i>item</i>	– the item to be selected. Must not be NULL.
----	-------------	--

Parameters

in	<i>docallback</i>	<p>– A flag that determines if the callback() is invoked or not:</p> <ul style="list-style-type: none"> • 0 - the callback() is not invoked • 1 - the callback() is invoked (default), callback_reason() will be either <code>FL_TREE_REASON_SELECTED</code> or <code>FL_TREE_REASON_DESELECTED</code>
----	-------------------	--

31.143.2.85 selectbox() [1/2]

```
Fl_Boxtype Fl_Tree::selectbox ( ) const
```

Sets the style of box used to draw selected items.

This is an fltk [Fl_Boxtype](#). The default is influenced by FLTK's current [Fl::scheme\(\)](#)

31.143.2.86 selectbox() [2/2]

```
void Fl_Tree::selectbox (
    Fl_Boxtype val )
```

Gets the style of box used to draw selected items.

This is an fltk [Fl_Boxtype](#). The default is influenced by FLTK's current [Fl::scheme\(\)](#)

31.143.2.87 selectmode() [1/2]

```
Fl_Tree_Select Fl_Tree::selectmode ( ) const
```

Gets the tree's current selection mode.

See [Fl_Tree_Select](#) for possible values.

31.143.2.88 selectmode() [2/2]

```
void Fl_Tree::selectmode (
    Fl_Tree_Select val )
```

Sets the tree's selection mode.

See [Fl_Tree_Select](#) for possible values.

31.143.2.89 set_item_focus()

```
void Fl_Tree::set_item_focus (
    Fl_Tree_Item * item )
```

Set the item that currently should have keyboard focus.

Handles calling [redraw\(\)](#) to update the focus box (if it is visible).

Parameters

in	<i>item</i>	The item that should take focus. If NULL, none will have focus.
----	-------------	---

31.143.2.90 show_item() [1/2]

```
void Fl_Tree::show_item (
    Fl_Tree_Item * item )
```

Adjust the vertical scrollbar to show '*item*' at the top of the display IF it is currently off-screen (for instance [show_item_top\(\)](#)).

If it is already on-screen, no change is made.

Parameters

in	<i>item</i>	The item to be shown. If NULL, first() is used.
----	-------------	---

See also

[show_item_top\(\)](#), [show_item_middle\(\)](#), [show_item_bottom\(\)](#)

31.143.2.91 show_item() [2/2]

```
void Fl_Tree::show_item (
    Fl_Tree_Item * item,
    int yoff )
```

Adjust the vertical scrollbar so that '*item*' is visible '*yoff*' pixels from the top of the [Fl_Tree](#) widget's display. For instance, *yoff*=0 will position the item at the top.

If *yoff* is larger than the vertical scrollbar's limit, the value will be clipped. So if *yoff*=100, but scrollbar's max is 50, then 50 will be used.

Parameters

in	<i>item</i>	The item to be shown. If NULL, first() is used.
in	<i>yoff</i>	The pixel offset from the top for the displayed position.

See also

[show_item_top\(\)](#), [show_item_middle\(\)](#), [show_item_bottom\(\)](#)

31.143.2.92 show_item_bottom()

```
void Fl_Tree::show_item_bottom (
    Fl_Tree_Item * item )
```

Adjust the vertical scrollbar so that '*item*' is at the bottom of the display.

Parameters

in	<i>item</i>	The item to be shown. If NULL, first() is used.
----	-------------	---

31.143.2.93 show_item_middle()

```
void Fl_Tree::show_item_middle (
    Fl_Tree_Item * item )
```

Adjust the vertical scrollbar so that '*item*' is in the middle of the display.

Parameters

in	<i>item</i>	The item to be shown. If NULL, first() is used.
----	-------------	---

31.143.2.94 show_item_top()

```
void Fl_Tree::show_item_top (
```

```
Fl_Tree_Item * item )
```

Adjust the vertical scrollbar so that 'item' is at the top of the display.

Parameters

in	<i>item</i>	The item to be shown. If NULL, first() is used.
----	-------------	---

31.143.2.95 show_self()

```
void Fl_Tree::show_self ( )
```

Print the tree as 'ascii art' to stdout.

Used mainly for debugging.

Todo should be const

Version

1.3.0

31.143.2.96 showcollapse() [1/2]

```
int Fl_Tree::showcollapse ( ) const
```

Returns 1 if the collapse icon is enabled, 0 if not.

See also

[showcollapse\(int\)](#)

31.143.2.97 showcollapse() [2/2]

```
void Fl_Tree::showcollapse (
    int val )
```

Set if we should show the collapse icon or not.

If collapse icons are disabled, the user will not be able to interactively collapse items in the tree, unless the application provides some other means via [open\(\)](#) and [close\(\)](#).

Parameters

in	<i>val</i>	1: shows collapse icons (default), 0: hides collapse icons.
----	------------	--

31.143.2.98 showroot()

```
void Fl_Tree::showroot (
    int val )
```

Set if the root item should be shown or not.

Parameters

in	<i>val</i>	1 – show the root item (default) 0 – hide the root item.
----	------------	---

31.143.2.99 sortorder()

```
Fl_Tree_Sort Fl_Tree::sortorder ( ) const
```

Set the default sort order used when items are added to the tree.

See [Fl_Tree_Sort](#) for possible values.

31.143.2.100 usericon() [1/2]

```
Fl_Image * Fl_Tree::usericon ( ) const
```

Returns the [Fl_Image](#) being used as the default user icon for all newly created items.

Returns zero if no icon has been set, which is the default.

31.143.2.101 usericon() [2/2]

```
void Fl_Tree::usericon (
    Fl_Image * val )
```

Sets the [Fl_Image](#) to be used as the default user icon for all newly created items.

If you want to specify user icons on a per-item basis, use [Fl_Tree_Item::usericon\(\)](#) instead.

Parameters

in	<i>val</i>	– The new image to be used, or zero to disable user icons.
----	------------	--

31.143.2.102 vposition() [1/2]

```
int Fl_Tree::vposition ( ) const
```

Returns the vertical scroll position as a pixel offset.

The position returned is how many pixels of the tree are scrolled off the top edge of the screen.

See also

[vposition\(int\)](#), [hposition\(\)](#), [hposition\(int\)](#)

31.143.2.103 vposition() [2/2]

```
void Fl_Tree::vposition (
    int pos )
```

Sets the vertical scroll offset to position '*pos*'.

The position is how many pixels of the tree are scrolled off the top edge of the screen.

Parameters

in	<i>pos</i>	The vertical position (in pixels) to scroll the tree to.
----	------------	--

See also

[vposition\(\)](#), [hposition\(\)](#), [hposition\(int\)](#)

The documentation for this class was generated from the following files:

- [Fl_Tree.H](#)
- [Fl_Tree.cxx](#)

31.144 FI_Tree_Item Class Reference

Tree widget item.

```
#include <Fl_Tree_Item.H>
```

Public Member Functions

- void **activate** (int val=1)
Change the item's activation state to the optionally specified 'val'.
- **FI_Tree_Item * add** (const **FI_Tree_Prefs** &prefs, char **arr)
Descend into the path specified by 'arr', and add a new child there.
- **FI_Tree_Item * add** (const **FI_Tree_Prefs** &prefs, char **arr, **FI_Tree_Item** *newitem)
Descend into path specified by 'arr' and add 'newitem' there.
- **FI_Tree_Item * add** (const **FI_Tree_Prefs** &prefs, const char *new_label)
Add a new child to this item with the name 'new_label' and defaults from 'prefs'.
- **FI_Tree_Item * add** (const **FI_Tree_Prefs** &prefs, const char *new_label, **FI_Tree_Item** *newitem)
Add 'item' as immediate child with 'new_label' and defaults from 'prefs'.
- **FI_Tree_Item * child** (int index)
Return the child item for the given 'index'.
- const **FI_Tree_Item * child** (int t) const
Return the const child item for the given 'index'.
- int **children** () const
Return the number of children this item has.
- void **clear_children** ()
Clear all the children for this item.
- void **close** ()
Close this item and all its children.
- void **deactivate** ()
Deactivate the item; the callback() won't be invoked when clicked.
- **FI_Tree_Item * deparent** (int index)
Deparent child at index position 'pos'.
- int **depth** () const
Returns how many levels deep this item is in the hierarchy.
- void **deselect** ()
Disable the item's selection state.
- int **deselect_all** ()
Deselect item and all its children.
- void **draw** (int X, int &Y, int W, **FI_Tree_Item** *itemfocus, int &tree_item_xmax, int lastchild=1, int render=1)
Draw this item and its children.
- virtual int **draw_item_content** (int render)
Draw the item content.
- int **event_on_collapse_icon** (const **FI_Tree_Prefs** &prefs) const
Was the event on the 'collapse' button of this item?
- int **event_on_label** (const **FI_Tree_Prefs** &prefs) const
Was event on the label() of this item?
- int **find_child** (const char *name)
Return the index of the immediate child of this item that has the label 'name'.
- int **find_child** (**FI_Tree_Item** *item)
Find the index number for the specified 'item' in the current item's list of children.
- **FI_Tree_Item * find_child_item** (char **arr)
*Non-const version of FI_Tree_Item::find_child_item(char **arr) const.*
- const **FI_Tree_Item * find_child_item** (char **arr) const

- Find child item by descending array 'arr' of names.*

 - [FI_Tree_Item](#) * **find_child_item** (const char *name)

*Non-const version of [FI_Tree_Item::find_child_item\(const char *name\)](#) const.*
- const [FI_Tree_Item](#) * **find_child_item** (const char *name) const

Return the /immediate/ child of current item that has the label 'name'.
- [FI_Tree_Item](#) * **find_clicked** (const [FI_Tree_Prefs](#) &prefs, int yonly=0)

Non-const version of [FI_Tree_Item::find_clicked\(const FI_Tree_Prefs&,int\)](#) const.
- const [FI_Tree_Item](#) * **find_clicked** (const [FI_Tree_Prefs](#) &prefs, int yonly=0) const

Find the item that the last event was over.
- [FI_Tree_Item](#) * **find_item** (char **arr)

*Non-const version of [FI_Tree_Item::find_item\(char **names\)](#) const.*
- const [FI_Tree_Item](#) * **find_item** (char **arr) const

Find item by descending array of 'names'.
- [FI_Tree_Item](#) (const [FI_Tree_Item](#) *o)

Copy constructor.
- [FI_Tree_Item](#) (const [FI_Tree_Prefs](#) &prefs)

Constructor.
- [FI_Tree_Item](#) ([FI_Tree](#) *tree)

Constructor.
- int **h** () const

The item's height.
- int **has_children** () const

See if this item has children.
- [FI_Tree_Item](#) * **insert** (const [FI_Tree_Prefs](#) &prefs, const char *new_label, int pos=0)

Insert a new item named 'new_label' into current item's children at a specified position 'pos'.
- [FI_Tree_Item](#) * **insert_above** (const [FI_Tree_Prefs](#) &prefs, const char *new_label)

Insert a new item named 'new_label' above this item.
- char **is_activated** () const

See if the item is activated.
- char **is_active** () const

See if the item is activated. Alias for [is_activated\(\)](#).
- int **is_close** () const

See if the item is 'closed'.
- int **is_open** () const

See if the item is 'open'.
- int **is_root** () const

Is this item the root of the tree?
- char **is_selected** () const

See if the item is selected.
- int **is_visible** () const

See if the item is visible.
- const char * **label** () const

Return the label.
- void **label** (const char *val)

Set the label to 'name'.
- int **label_h** () const

The item's label height.
- int **label_w** () const

The item's maximum label width to right edge of [FI_Tree](#)'s inner width within scrollbars.
- int **label_x** () const

The item's label x position relative to the window.

- int `label_y` () const
The item's label y position relative to the window.
- `FI_Color` `labelbgcolor` () const
Return item's label background text color.
- void `labelbgcolor` (`FI_Color` val)
Set item's label background color.
- `FI_Color` `labelcolor` () const
Return item's label text color. Alias for `labelfgcolor()` const).
- void `labelcolor` (`FI_Color` val)
Set item's label text color. Alias for `labelfgcolor(FI_Color)`.
- `FI_Color` `labelfgcolor` () const
Return item's label foreground text color.
- void `labelfgcolor` (`FI_Color` val)
Set item's label foreground text color.
- `FI_Font` `labelfont` () const
Get item's label font face.
- void `labelfont` (`FI_Font` val)
Set item's label font face.
- `FI_Fontsize` `labelsize` () const
Get item's label font size.
- void `labelsize` (`FI_Fontsize` val)
Set item's label font size.
- int `move` (`FI_Tree_Item` *item, int op=0, int pos=0)
Move the current item above/below/into the specified 'item', where 'op' determines the type of move:
- int `move` (int to, int from)
Move the item 'from' to sibling position of 'to'.
- int `move_above` (`FI_Tree_Item` *item)
Move the current item above the specified 'item'.
- int `move_below` (`FI_Tree_Item` *item)
Move the current item below the specified 'item'.
- int `move_into` (`FI_Tree_Item` *item, int pos=0)
Parent the current item as a child of the specified 'item'.
- `FI_Tree_Item` * `next` ()
Return the next item in the tree.
- `FI_Tree_Item` * `next_displayed` (`FI_Tree_Prefs` &prefs)
Same as `next_visible()`.
- `FI_Tree_Item` * `next_sibling` ()
Return this item's next sibling.
- `FI_Tree_Item` * `next_visible` (`FI_Tree_Prefs` &prefs)
Return the next `open()`, `visible()` item.
- void `open` ()
Open this item and all its children.
- void `open_toggle` ()
Toggle the item's open/closed state.
- `FI_Tree_Item` * `parent` ()
Return the parent for this item. Returns NULL if we are the root.
- const `FI_Tree_Item` * `parent` () const
Return the const parent for this item. Returns NULL if we are the root.
- void `parent` (`FI_Tree_Item` *val)
Set the parent for this item.
- const `FI_Tree_Prefs` & `prefs` () const

- Return the parent tree's prefs.*

 - `FI_Tree_Item * prev ()`
- Return the previous item in the tree.*

 - `FI_Tree_Item * prev_displayed (FI_Tree_Prefs &prefs)`

Same as `prev_visible()`.

 - `FI_Tree_Item * prev_sibling ()`

Return this item's previous sibling.

 - `FI_Tree_Item * prev_visible (FI_Tree_Prefs &prefs)`

Return the previous `open()`, `visible()` item.

- `int remove_child (const char *new_label)`
- Remove immediate child (and its children) by its label 'name'.*

 - `int remove_child (FI_Tree_Item *item)`

Remove 'item' from the current item's children.

- `int reparent (FI_Tree_Item *newchild, int index)`
- Reparent specified item as a child of ourself at position 'pos'.*

 - `FI_Tree_Item * replace (FI_Tree_Item *new_item)`

Replace the current item with a new item.

- `FI_Tree_Item * replace_child (FI_Tree_Item *olditem, FI_Tree_Item *newitem)`
- Replace existing child 'olditem' with 'newitem'.*

 - `void select (int val=1)`

Change the item's selection state to the optionally specified 'val'.

- `int select_all ()`
- Select item and all its children.*

 - `void select_toggle ()`

Toggle the item's selection state.

- `void show_self (const char *indent="") const`
- Print the tree as 'ascii art' to stdout.*

 - `int swap_children (FI_Tree_Item *a, FI_Tree_Item *b)`

Swap two of our immediate children, given item pointers.

- `void swap_children (int ax, int bx)`
- Swap two of our children, given two child index values 'ax' and 'bx'.*

 - `FI_Tree * tree ()`

Return the tree for this item.

- `const FI_Tree * tree () const`
- Return the tree for this item.*

 - `void update_prev_next (int index)`

Update our `_prev_sibling` and `_next_sibling` pointers to point to neighbors given `index` as being our current position in the parent's item array.

- `void * user_data () const`
- Retrieve the user-data value that has been assigned to the item.*

 - `void user_data (void *data)`

Set a user-data value for the item.

- `FI_Image * userdeicon () const`
- Return the deactivated version of the user icon, if any.*

 - `void userdeicon (FI_Image *val)`

Set the usericon to draw when the item is deactivated.

- `FI_Image * usericon () const`
- Get the item's user icon as an `FI_Image`. Returns '0' if disabled.*

 - `void usericon (FI_Image *val)`

Set the item's user icon to an `FI_Image`.

- `int visible () const`

- *See if the item is visible. Alias for [is_visible\(\)](#).*
- int [visible_r](#) () const
See if item and all its parents are [open\(\)](#) and [visible\(\)](#).
- int [w](#) () const
The entire item's width to right edge of [FL_Tree](#)'s inner width within scrollbars.
- [FL_Widget](#) * [widget](#) () const
Return FLTK widget assigned to this item.
- void [widget](#) ([FL_Widget](#) *val)
Assign an FLTK widget to this item.
- int [x](#) () const
The item's x position relative to the window.
- int [y](#) () const
The item's y position relative to the window.

Protected Member Functions

- void [_Init](#) (const [FL_Tree_Prefs](#) &prefs, [FL_Tree](#) *tree)
- int [calc_item_height](#) (const [FL_Tree_Prefs](#) &prefs) const
Return the item's 'visible' height.
- void [draw_horizontal_connector](#) (int x1, int x2, int y, const [FL_Tree_Prefs](#) &prefs)
Internal: Horizontal connector line based on preference settings.
- void [draw_vertical_connector](#) (int x, int y1, int y2, const [FL_Tree_Prefs](#) &prefs)
Internal: Vertical connector line based on preference settings.
- [FL_Color](#) [drawbgcolor](#) () const
Returns the recommended background color used for drawing this item.
- [FL_Color](#) [drawfgcolor](#) () const
Returns the recommended foreground color used for drawing this item.
- void [hide_widgets](#) ()
Internal: Hide the FLTK [widget\(\)](#) for this item and all children.
- int [is_flag](#) (unsigned short val) const
See if flag set. Returns 0 or 1.
- void [recalc_tree](#) ()
Call this when our geometry is changed.
- void [set_flag](#) (unsigned short flag, int val)
Set a flag to an on or off value. val is 0 or 1.
- void [show_widgets](#) ()
Internal: Show the FLTK [widget\(\)](#) for this item and all children.

31.144.1 Detailed Description

Tree widget item.

This class is a single tree item, and manages all of the item's attributes. [FL_Tree_Item](#) is used by [FL_Tree](#), which is comprised of many instances of [FL_Tree_Item](#).

[FL_Tree_Item](#) is hierarchical; it dynamically manages an [FL_Tree_Item_Array](#) of children that are themselves instances of [FL_Tree_Item](#). Each item can have zero or more children. When an item has children, [close\(\)](#) and [open\(\)](#) can be used to hide or show them.

Items have their own attributes; font size, face, color. Items maintain their own hierarchy of children.

When you make changes to items, you'll need to tell the tree to [redraw\(\)](#) for the changes to show up.

New 1.3.3 ABI feature: You can define custom items by either adding a custom widget to the item with [FL_Tree_Item::widget\(\)](#), or override the [draw_item_content\(\)](#) method if you want to just redefine how the label is drawn.

The following shows the [FL_Tree_Item](#)'s dimensions, useful when overriding the [draw_item_content\(\)](#) method:

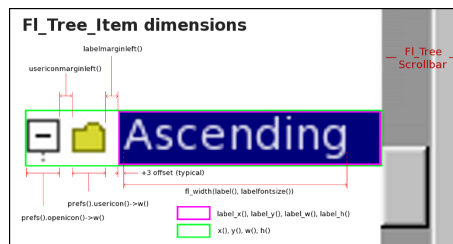


Figure 31.47 Fl_Tree_Item's internal dimensions.

31.144.2 Constructor & Destructor Documentation

31.144.2.1 Fl_Tree_Item() [1/2]

```
Fl_Tree_Item::Fl_Tree_Item (
    const Fl_Tree_Prefs & prefs )
```

Constructor.

Makes a new instance of [Fl_Tree_Item](#) using defaults from 'prefs'.

Deprecated in 1.3.3 ABI – you must use [Fl_Tree_Item\(Fl_Tree*\)](#) for proper horizontal scrollbar behavior.

31.144.2.2 Fl_Tree_Item() [2/2]

```
Fl_Tree_Item::Fl_Tree_Item (
    Fl_Tree * tree )
```

Constructor.

Makes a new instance of [Fl_Tree_Item](#) for 'tree'.

This must be used instead of the older, deprecated [Fl_Tree_Item\(Fl_Tree_Prefs\)](#) constructor for proper horizontal scrollbar calculation.

Version

1.3.3 ABI feature

31.144.3 Member Function Documentation

31.144.3.1 activate()

```
void Fl_Tree_Item::activate (
    int val = 1 ) [inline]
```

Change the item's activation state to the optionally specified 'val'.

When deactivated, the item will be 'grayed out'; the `callback()` won't be invoked if the user clicks on the label. If a [widget\(\)](#) is associated with the item, its activation state will be changed as well.

If 'val' is not specified, the item will be activated.

31.144.3.2 add() [1/4]

```
Fl_Tree_Item * Fl_Tree_Item::add (
    const Fl_Tree_Prefs & prefs,
    char ** arr )
```

Descend into the path specified by 'arr', and add a new child there.

Should be used only by [Fl_Tree](#)'s internals. Adds the item based on the value of `prefs.sortorder()`.

Returns

the item added.

Version

1.3.0 release

31.144.3.3 add() [2/4]

```
Fl_Tree_Item * Fl_Tree_Item::add (
    const Fl_Tree_Prefs & prefs,
    char ** arr,
    Fl_Tree_Item * newitem )
```

Descend into path specified by 'arr' and add 'newitem' there.

Should be used only by `Fl_Tree`'s internals. If item is NULL, a new item is created. Adds the item based on the value of `prefs.sortorder()`.

Returns

the item added.

Version

1.3.3 ABI feature

31.144.3.4 add() [3/4]

```
Fl_Tree_Item * Fl_Tree_Item::add (
    const Fl_Tree_Prefs & prefs,
    const char * new_label )
```

Add a new child to this item with the name 'new_label' and defaults from 'prefs'.

An internally managed copy is made of the label string. Adds the item based on the value of `prefs.sortorder()`.

Returns

the item added

Version

1.3.0 release

31.144.3.5 add() [4/4]

```
Fl_Tree_Item * Fl_Tree_Item::add (
    const Fl_Tree_Prefs & prefs,
    const char * new_label,
    Fl_Tree_Item * item )
```

Add 'item' as immediate child with 'new_label' and defaults from 'prefs'.

If 'item' is NULL, a new item is created. An internally managed copy is made of the label string. Adds the item based on the value of `prefs.sortorder()`.

Returns

the item added

Version

1.3.3

31.144.3.6 calc_item_height()

```
int Fl_Tree_Item::calc_item_height (
    const Fl_Tree_Prefs & prefs ) const [protected]
```

Return the item's 'visible' height.

Takes into account the item's:

- visibility (if `lis_visible()`, returns 0)
- `labelfont()` height: if `label()` != NULL
- `widget()` height: if `widget()` != NULL
- `openicon()` height (if not NULL)
- `usericon()` height (if not NULL) Does NOT include `Fl_Tree::linespacing()`;

Returns

maximum pixel height

31.144.3.7 child()

```
const Fl_Tree_Item * Fl_Tree_Item::child (
    int t ) const
```

Return the const child item for the given 'index'.

Return const child item for the specified 'index'.

31.144.3.8 deactivate()

```
void Fl_Tree_Item::deactivate ( ) [inline]
```

Deactivate the item; the `callback()` won't be invoked when clicked.

Same as `activate(0)`

31.144.3.9 deparent()

```
Fl_Tree_Item * Fl_Tree_Item::deparent (
    int pos )
```

Deparent child at index position 'pos'.

This creates an "orphaned" item that is still allocated, but has no parent or siblings. Normally the caller would want to immediately reparent the orphan elsewhere.

A successfully orphaned item will have its `parent()` and `prev_sibling()/next_sibling()` set to NULL.

Returns

- pointer to orphaned item on success
- NULL on error (could not deparent the item)

31.144.3.10 depth()

```
int Fl_Tree_Item::depth ( ) const
```

Returns how many levels deep this item is in the hierarchy.

For instance; root has a depth of zero, and its immediate children would have a depth of 1, and so on. Use e.g. for determining the horizontal indent of this item during drawing.

31.144.3.11 deselect_all()

```
int Fl_Tree_Item::deselect_all ( ) [inline]
```

Deselect item and all its children.

Returns count of how many items were in the 'selected' state, ie. how many items were "changed".

31.144.3.12 draw()

```
void Fl_Tree_Item::draw (
    int X,
    int & Y,
    int W,
    Fl_Tree_Item * itemfocus,
    int & tree_item_xmax,
    int lastchild = 1,
    int render = 1 )
```

Draw this item and its children.

Parameters

in	<i>X</i>	Horizontal position for item being drawn
in, out	<i>Y</i>	Vertical position for item being drawn, returns new position for next item
in	<i>W</i>	Recommended width for item
in	<i>itemfocus</i>	The tree's current focus item (if any)
in, out	<i>tree_item_xmax</i>	The tree's running xmax (right-most edge so far). Mainly used by parent tree when render==0 to calculate tree's max width.
in	<i>lastchild</i>	Is this item the last child in a subtree?
in	<i>render</i>	Whether or not to render the item: 0: no rendering, just calculate size w/out drawing. 1: render item as well as size calc

Version

1.3.3 ABI feature: modified parameters

31.144.3.13 draw_horizontal_connector()

```
void Fl_Tree_Item::draw_horizontal_connector (
    int x1,
    int x2,
    int y,
    const Fl_Tree_Prefs & prefs ) [protected]
```

Internal: Horizontal connector line based on preference settings.

Parameters

in	<i>x1</i>	The left hand X position of the horizontal connector
in	<i>x2</i>	The right hand X position of the horizontal connector
in	<i>y</i>	The vertical position of the horizontal connector
in	<i>prefs</i>	The Fl_Tree prefs

31.144.3.14 draw_item_content()

```
int Fl_Tree_Item::draw_item_content (
    int render ) [virtual]
```

Draw the item content.

This method can be overridden to implement custom drawing by filling the `label_[xywh]()` area with content.

A minimal example of how to override `draw_item_content()` and draw just a normal item's background and label ourselves:

```
class MyTreeItem : public Fl_Tree_Item {
public:
```

```

MyTreeItem() { }
~MyTreeItem() { }
// DRAW OUR CUSTOM CONTENT FOR THE ITEM
int draw_item_content(int render) {
    // Our item's dimensions + text content
    int X=label_x(), Y=label_y(), W=label_w(), H=label_h();
    const char *text = label() ? label() : "";
    // Rendering? Do any drawing that's needed
    if ( render ) {
        // Draw bg -- a filled rectangle
        fl_color(drawbgcolor()); fl_rectf(X,Y,W,H);
        // Draw label
        fl_font(labelfont(), labelsize()); // use item's label font/size
        fl_color(drawfgcolor()); // use recommended fg color
        fl_draw(text, X,Y,W,H, FL_ALIGN_LEFT); // draw the item's label
    }
    // Rendered or not, we must calculate content's max X position
    int lw=0, lh=0;
    fl_measure(text, lw, lh); // get width of label text
    return X + lw; // return X + label width
};

```

You can draw anything you want inside `draw_item_content()` using any of the `fl_draw.H` functions, as long as it's within the label's xywh area.

To add instances of your custom item to the tree, you can use:

```

// Example #1: using add()
MyTreeItem *bart = new MyTreeItem(..); // class derived from Fl_Tree_Item
tree->add("/Simpsons/Bart", bart); // Add item as /Simpsons/Bart

```

..or you can insert or replace existing items:

```

// Example #2: using replace()
MyTreeItem *marge = new MyTreeItem(..); // class derived from Fl_Tree_Item
item = tree->add("/Simpsons/Marge"); // create item
item->replace(mi); // replace it with our own

```

Parameters

in	<i>render</i>	Whether we should render content (1), or just tally the geometry (0). <code>Fl_Tree</code> may want only to find the widest item in the tree for scrollbar calculations.
----	---------------	--

Returns

the right-most X coordinate, or 'xmax' of content we drew, i.e. the "scrollable" content. The tree uses the largest xmax to determine the maximum width of the tree's content (needed for e.g. computing the horizontal scrollbar's size).

Version

1.3.3 ABI feature

31.144.3.15 draw_vertical_connector()

```

void Fl_Tree_Item::draw_vertical_connector (
    int x,
    int y1,
    int y2,
    const Fl_Tree_Prefs & prefs ) [protected]

```

Internal: Vertical connector line based on preference settings.

Parameters

in	<i>x</i>	The x position of the vertical connector
in	<i>y1</i>	The top of the vertical connector
in	<i>y2</i>	The bottom of the vertical connector
in	<i>prefs</i>	The <code>Fl_Tree</code> prefs

31.144.3.16 drawbgcolor()

```
Fl_Color Fl_Tree_Item::drawbgcolor ( ) const [protected]
```

Returns the recommended background color used for drawing this item.

See also

[draw_item_content\(\)](#)

Version

1.3.3 ABI

31.144.3.17 drawfgcolor()

```
Fl_Color Fl_Tree_Item::drawfgcolor ( ) const [protected]
```

Returns the recommended foreground color used for drawing this item.

See also

[draw_item_content\(\)](#)

Version

1.3.3 ABI ABI

31.144.3.18 find_child() [1/2]

```
int Fl_Tree_Item::find_child (
    const char * name )
```

Return the index of the immediate child of this item that has the label 'name'.

Returns

index of found item, or -1 if not found.

Version

1.3.0 release

31.144.3.19 find_child() [2/2]

```
int Fl_Tree_Item::find_child (
    Fl_Tree_Item * item )
```

Find the index number for the specified 'item' in the current item's list of children.

Returns

the index, or -1 if not found.

31.144.3.20 find_child_item() [1/2]

```
const Fl_Tree_Item * Fl_Tree_Item::find_child_item (
    char ** arr ) const
```

Find child item by descending array 'arr' of names.
Does not include self in search. Only [Fl_Tree](#) should need this method.

Returns

item, or 0 if not found

Version

1.3.0 release

31.144.3.21 find_child_item() [2/2]

```
const Fl_Tree_Item * Fl_Tree_Item::find_child_item (
    const char * name ) const
```

Return the /immediate/ child of current item that has the label 'name'.

Returns

const found item, or 0 if not found.

Version

1.3.3

31.144.3.22 find_clicked()

```
const Fl_Tree_Item * Fl_Tree_Item::find_clicked (
    const Fl_Tree_Prefs & prefs,
    int yonly = 0 ) const
```

Find the item that the last event was over.
If 'yonly' is 1, only check event's y value, don't care about x.

Parameters

in	<i>prefs</i>	The parent tree's Fl_Tree_Prefs
in	<i>yonly</i>	- 0: check both event's X and Y values. - 1: only check event's Y value, don't care about X.

Returns

pointer to clicked item, or NULL if none found

Version

1.3.3 ABI feature

31.144.3.23 find_item()

```
const Fl_Tree_Item * Fl_Tree_Item::find_item (
    char ** names ) const
```

Find item by descending array of 'names'.
Includes self in search. Only [Fl_Tree](#) should need this method. Use [Fl_Tree::find_item\(\)](#) instead.

Returns

const item, or 0 if not found

31.144.3.24 hide_widgets()

```
void Fl_Tree_Item::hide_widgets ( ) [protected]
```

Internal: Hide the FLTK [widget\(\)](#) for this item and all children.
Used by [close\(\)](#) to hide widgets.

31.144.3.25 insert()

```
Fl_Tree_Item * Fl_Tree_Item::insert (
    const Fl_Tree_Prefs & prefs,
    const char * new_label,
    int pos = 0 )
```

Insert a new item named 'new_label' into current item's children at a specified position 'pos'.
If pos is out of range the new item is

- prepended if pos < 0 or
- appended if pos > item->[children\(\)](#).

Returns

the new item inserted

See also

[Fl_Tree::insert\(\)](#)

31.144.3.26 insert_above()

```
Fl_Tree_Item * Fl_Tree_Item::insert_above (
    const Fl_Tree_Prefs & prefs,
    const char * new_label )
```

Insert a new item named 'new_label' above this item.

Returns

the new item inserted, or 0 if an error occurred.

31.144.3.27 label()

```
void Fl_Tree_Item::label (
    const char * name )
```

Set the label to 'name'.
Makes and manages an internal copy of 'name'.

31.144.3.28 label_h()

```
int Fl_Tree_Item::label_h ( ) const [inline]
```

The item's label height.

Version

1.3.3

31.144.3.29 label_w()

```
int Fl_Tree_Item::label_w ( ) const [inline]
```

The item's maximum label width to right edge of [Fl_Tree](#)'s inner width within scrollbars.

Version

1.3.3

31.144.3.30 label_x()

```
int Fl_Tree_Item::label_x ( ) const [inline]
```

The item's label x position relative to the window.

Version

1.3.3

31.144.3.31 label_y()

```
int Fl_Tree_Item::label_y ( ) const [inline]
```

The item's label y position relative to the window.

Version

1.3.3

31.144.3.32 labelbgcolor() [1/2]

```
Fl_Color Fl_Tree_Item::labelbgcolor ( ) const [inline]
```

Return item's label background text color.

If the color is 0xffffffff, the default behavior is the parent tree's bg color will be used. (An overloaded [draw_item_content\(\)](#) can override this behavior.)

31.144.3.33 labelbgcolor() [2/2]

```
void Fl_Tree_Item::labelbgcolor (
    Fl_Color val ) [inline]
```

Set item's label background color.

A special case is made for color 0xffffffff which uses the parent tree's bg color.

31.144.3.34 move() [1/2]

```
int Fl_Tree_Item::move (
    Fl_Tree_Item * item,
    int op = 0,
    int pos = 0 )
```

Move the current item above/below/into the specified 'item', where 'op' determines the type of move:

- 0: move above 'item' ('pos' ignored)
- 1: move below 'item' ('pos' ignored)
- 2: move into 'item' as a child (at optional position 'pos')

Returns

0 on success. a negative number on error:

- -1: one of the items has no parent
- -2: item's index could not be determined
- -3: bad 'op'
- -4: index range error
- -5: could not deparent
- -6: could not reparent at 'pos'
- (Other return values reserved for future use.)

31.144.3.35 move() [2/2]

```
int Fl_Tree_Item::move (
    int to,
    int from )
```

Move the item 'from' to sibling position of 'to'.

Returns

- 0: Success
- -1: range error (e.g. if 'to' or 'from' out of range).
- (Other return values reserved for future use)

31.144.3.36 move_above()

```
int Fl_Tree_Item::move_above (
    Fl_Tree_Item * item )
```

Move the current item above the specified 'item'.
This is the equivalent of calling `move(item,0,0)`.

Returns

0 on success.

On error returns a negative value; see [move\(Fl_Tree_Item*,int,int\)](#) for possible error codes.

31.144.3.37 move_below()

```
int Fl_Tree_Item::move_below (
    Fl_Tree_Item * item )
```

Move the current item below the specified 'item'.
This is the equivalent of calling `move(item,1,0)`.

Returns

0 on success.

On error returns a negative value; see [move\(Fl_Tree_Item*,int,int\)](#) for possible error codes.

31.144.3.38 move_into()

```
int Fl_Tree_Item::move_into (
    Fl_Tree_Item * item,
    int pos = 0 )
```

Parent the current item as a child of the specified 'item'.

This is the equivalent of calling `move(item,2,pos)`.

Returns

0 on success.

On error returns a negative value; see `move(Fl_Tree_Item*,int,int)` for possible error codes.

31.144.3.39 next()

```
Fl_Tree_Item * Fl_Tree_Item::next ( )
```

Return the next item in the tree.

This method can be used to walk the tree forward. For an example of how to use this method, see `Fl_Tree::first()`.

Returns

the next item in the tree, or 0 if there's no more items.

31.144.3.40 next_displayed()

```
Fl_Tree_Item * Fl_Tree_Item::next_displayed (
    Fl_Tree_Prefs & prefs )
```

Same as `next_visible()`.

Deprecated in 1.3.3 for confusing name, use `next_visible()` instead

31.144.3.41 next_sibling()

```
Fl_Tree_Item * Fl_Tree_Item::next_sibling ( )
```

Return this item's next sibling.

Moves to the next item below us at the same level (sibling). Use this to move down the tree without changing `depth()`. effectively skipping over this item's children/descendents.

Returns

item's next sibling, or 0 if none.

31.144.3.42 next_visible()

```
Fl_Tree_Item * Fl_Tree_Item::next_visible (
    Fl_Tree_Prefs & prefs )
```

Return the next `open()`, `visible()` item.

(If this item has children and is closed, children are skipped)

This method can be used to walk the tree forward, skipping items that are not currently open/visible to the user.

Returns

the next `open()` `visible()` item below us, or 0 if there's no more items.

Version

1.3.3

31.144.3.43 parent()

```
void Fl_Tree_Item::parent (
    Fl_Tree_Item * val ) [inline]
```

Set the parent for this item.
Should only be used by [Fl_Tree](#)'s internals.

31.144.3.44 prefs()

```
const Fl_Tree_Prefs & Fl_Tree_Item::prefs ( ) const
```

Return the parent tree's prefs.

Returns

a reference to the parent tree's [Fl_Tree_Prefs](#)

Version

1.3.3 ABI feature

31.144.3.45 prev()

```
Fl_Tree_Item * Fl_Tree_Item::prev ( )
```

Return the previous item in the tree.
This method can be used to walk the tree backwards. For an example of how to use this method, see [Fl_Tree::last\(\)](#).

Returns

the previous item in the tree, or 0 if there's no item above this one (hit the root).

31.144.3.46 prev_displayed()

```
Fl_Tree_Item * Fl_Tree_Item::prev_displayed (
    Fl_Tree_Prefs & prefs )
```

Same as [prev_visible\(\)](#).

Deprecated in 1.3.3 for confusing name, use [prev_visible\(\)](#)

31.144.3.47 prev_sibling()

```
Fl_Tree_Item * Fl_Tree_Item::prev_sibling ( )
```

Return this item's previous sibling.
Moves to the previous item above us at the same level (sibling). Use this to move up the tree without changing [depth\(\)](#).

Returns

This item's previous sibling, or 0 if none.

31.144.3.48 prev_visible()

```
Fl_Tree_Item * Fl_Tree_Item::prev_visible (
    Fl_Tree_Prefs & prefs )
```

Return the previous [open\(\)](#), [visible\(\)](#) item.
(If this item above us has children and is closed, its children are skipped)
This method can be used to walk the tree backward, skipping items that are not currently open/visible to the user.

Returns

the previous [open\(\)](#) [visible\(\)](#) item above us, or 0 if there's no more items.

31.144.3.49 recalc_tree()

```
void Fl_Tree_Item::recalc_tree ( ) [protected]
```

Call this when our geometry is changed.

(Font size, label contents, etc) Schedules tree to recalculate itself, as changes to us may affect tree widget's scrollbar visibility and tab sizes.

Version

1.3.3 ABI

31.144.3.50 remove_child() [1/2]

```
int Fl_Tree_Item::remove_child (
    const char * name )
```

Remove immediate child (and its children) by its label 'name'.

If more than one item matches 'name', only the first matching item is removed.

Parameters

in	<i>name</i>	The label name of the immediate child to remove
----	-------------	---

Returns

0 if removed, -1 if not found.

Version

1.3.3

31.144.3.51 remove_child() [2/2]

```
int Fl_Tree_Item::remove_child (
    Fl_Tree_Item * item )
```

Remove 'item' from the current item's children.

Returns

0 if removed, -1 if item not an immediate child.

31.144.3.52 reparent()

```
int Fl_Tree_Item::reparent (
    Fl_Tree_Item * newchild,
    int pos )
```

Reparent specified item as a child of ourself at position 'pos'.

Typically 'newchild' was recently orphaned with [deparent\(\)](#).

Returns

- 0: on success
- -1: on error (e.g. if 'pos' out of range) with no changes made.

31.144.3.53 replace()

```
Fl_Tree_Item * Fl_Tree_Item::replace (
    Fl_Tree_Item * newitem )
```

Replace the current item with a new item.

The current item is destroyed if successful. No checks are made to see if an item with the same name exists.

This method can be used to, for example, install 'custom' items into the tree derived from [Fl_Tree_Item](#); see [draw_item_content\(\)](#).

Parameters

in	<i>newitem</i>	The new item to replace the current item
----	----------------	--

Returns

newitem on success, NULL if could not be replaced.

See also

[Fl_Tree_Item::draw_item_content\(\)](#), [Fl_Tree::root\(Fl_Tree_Item*\)](#)

Version

1.3.3 ABI feature

31.144.3.54 replace_child()

```
Fl_Tree_Item * Fl_Tree_Item::replace_child (
    Fl_Tree_Item * olditem,
    Fl_Tree_Item * newitem )
```

Replace existing child 'olditem' with 'newitem'.

The 'olditem' is destroyed if successful. Can be used to put custom items (derived from [Fl_Tree_Item](#)) into the tree. No checks are made to see if an item with the same name exists.

Parameters

in	<i>olditem</i>	The item to be found and replaced
in	<i>newitem</i>	The new item to take the place of 'olditem'

Returns

newitem on success and 'olditem' is destroyed. NULL on error if 'olditem' was not found as an immediate child.

See also

[replace\(\)](#), [Fl_Tree_Item::draw\(\)](#)

Version

1.3.3 ABI feature

31.144.3.55 select()

```
void Fl_Tree_Item::select (
    int val = 1 ) [inline]
```

Change the item's selection state to the optionally specified 'val'.

If 'val' is not specified, the item will be selected.

31.144.3.56 select_all()

```
int Fl_Tree_Item::select_all ( ) [inline]
```

Select item and all its children.

Returns count of how many items were in the 'deselected' state, ie. how many items were "changed".

31.144.3.57 show_self()

```
void Fl_Tree_Item::show_self (
    const char * indent = "" ) const
```

Print the tree as 'ascii art' to stdout.

Used mainly for debugging.

31.144.3.58 show_widgets()

```
void Fl_Tree_Item::show_widgets ( ) [protected]
```

Internal: Show the FLTK [widget\(\)](#) for this item and all children.

Used by [open\(\)](#) to re-show widgets that were hidden by a previous [close\(\)](#)

31.144.3.59 swap_children() [1/2]

```
int Fl_Tree_Item::swap_children (
    Fl_Tree_Item * a,
    Fl_Tree_Item * b )
```

Swap two of our immediate children, given item pointers.

Use e.g. for sorting.

This method is SLOW because it involves linear lookups.

For speed, use [swap_children\(int,int\)](#) instead.

Parameters

in	<i>a,b</i>	The item ptrs of the two items to swap. Both must be immediate children of the current item.
----	------------	--

Returns

- 0 : OK
- -1 : failed: item 'a' or 'b' is not our child.

31.144.3.60 swap_children() [2/2]

```
void Fl_Tree_Item::swap_children (
    int ax,
    int bx )
```

Swap two of our children, given two child index values 'ax' and 'bx'.

Use e.g. for sorting.

This method is FAST, and does not involve lookups.

No range checking is done on either index value.

Parameters

in	<i>ax,bx</i>	the index of the items to swap
----	--------------	--------------------------------

31.144.3.61 tree() [1/2]

```
Fl_Tree * Fl_Tree_Item::tree ( ) [inline]
```

Return the tree for this item.

Version

1.3.4 (ABI feature)

31.144.3.62 tree() [2/2]

```
const Fl_Tree * Fl_Tree_Item::tree ( ) const [inline]
```

Return the tree for this item.

Version

1.3.3 (ABI feature)

31.144.3.63 update_prev_next()

```
void Fl_Tree_Item::update_prev_next (
    int index )
```

Update our `_prev_sibling` and `_next_sibling` pointers to point to neighbors given `index` as being our current position in the parent's item array.

Call this whenever items in the array are added/removed/moved/swapped/etc.

Parameters

<code>in</code>	<code>index</code>	Our index# in the parent. Special case if <code>index=-1</code> : become an orphan; null out all parent/sibling associations.
-----------------	--------------------	--

31.144.3.64 userdeicon() [1/2]

```
Fl_Image * Fl_Tree_Item::userdeicon ( ) const [inline]
```

Return the deactivated version of the user icon, if any.

Returns 0 if none.

31.144.3.65 userdeicon() [2/2]

```
void Fl_Tree_Item::userdeicon (
    Fl_Image * val ) [inline]
```

Set the usericon to draw when the item is deactivated.

Use '0' to disable. No internal copy is made; caller must manage icon's memory.

To create a typical 'grayed out' version of your usericon image, you can do the following:

```
// Create tree + usericon for items
Fl_Tree *tree = new Fl_Tree(..);
Fl_Image *usr_icon = new Fl_Pixmap(..); // your usericon
Fl_Image *de_icon = usr_icon->copy(); // make a copy, and..
de_icon->inactive(); // make it 'grayed out'
...
for ( .. ) { // item loop..
    item = tree->add("..."); // create new item
    item->usericon(usr_icon); // assign usericon to items
    item->userdeicon(de_icon); // assign userdeicon to items
    ..
}
```

In the above example, the app should 'delete' the two icons when they're no longer needed (e.g. after the tree is destroyed)

Version

1.3.4

31.144.3.66 usericon()

```
void Fl_Tree_Item::usericon (
    Fl_Image * val ) [inline]
```

Set the item's user icon to an [Fl_Image](#).

Use '0' to disable. No internal copy is made, caller must manage icon's memory.

Note, if you expect your items to be deactivated(), use [userdeicon\(Fl_Image*\)](#) to set up a 'grayed out' version of your icon to be used for display.

See also

[userdeicon\(Fl_Image*\)](#)

31.144.3.67 visible_r()

```
int Fl_Tree_Item::visible_r ( ) const
```

See if item and all its parents are [open\(\)](#) and [visible\(\)](#).

Returns

1 – item and its parents are [open\(\)](#) and [visible\(\)](#) 0 – item (or one of its parents) are invisible or [close\(\)](#)ed.

The documentation for this class was generated from the following files:

- [Fl_Tree_Item.H](#)
- [Fl_Tree_Item.cxx](#)

31.145 FI_Tree_Item_Array Class Reference

Manages an array of [Fl_Tree_Item](#) pointers.

```
#include <Fl_Tree_Item_Array.H>
```

Public Member Functions

- void [add](#) ([Fl_Tree_Item](#) *val)

Add an item to the end of the array.*
- void [clear](#) ()

Clear the entire array.
- int [deparent](#) (int pos)

Deparent item at 'pos' from our list of children.
- [FI_Tree_Item_Array](#) (const [FI_Tree_Item_Array](#) *o)

Copy constructor. Makes new copy of array, with new instances of each item.
- [FI_Tree_Item_Array](#) (int new_chunksize=10)

Constructor; creates an empty array.
- void [insert](#) (int pos, [Fl_Tree_Item](#) *new_item)

Insert an item at index position pos.
- int [manage_item_destroy](#) () const
- void [manage_item_destroy](#) (int val)

Option to control if [FI_Tree_Item_Array](#)'s destructor will also destroy the [FI_Tree_Item](#)'s.
- int [move](#) (int to, int from)

Move item at 'from' to new position 'to' in the array.
- [Fl_Tree_Item](#) * [operator\[\]](#) (int i)

Return the item and index i.
- const [Fl_Tree_Item](#) * [operator\[\]](#) (int i) const

Const version of [operator\[\]](#)(int i)
- int [remove](#) ([Fl_Tree_Item](#) *item)

- Remove the item from the array.*

 - void **remove** (int index)
- Remove the item at.*

 - int **reparent** (Fl_Tree_Item *item, Fl_Tree_Item *newparent, int pos)
- Reparent specified item as a child of ourself.*

 - void **replace** (int pos, Fl_Tree_Item *new_item)
- Replace the item at index with newitem.*

 - void **swap** (int ax, int bx)
- Swap the two items at index positions ax and bx.*

 - int **total** () const
- Return the total items in the array, or 0 if empty.*

 - ~**Fl_Tree_Item_Array** ()
- Destructor. Calls each item's destructor, destroys internal _items array.*

31.145.1 Detailed Description

Manages an array of [Fl_Tree_Item](#) pointers.

Because FLTK 1.x.x. has mandated that templates and STL not be used, we use this class to dynamically manage the arrays.

None of the methods do range checking on index values; the caller must be sure that index values are within the range $0 < \text{index} < \text{total}()$ (unless otherwise noted).

31.145.2 Constructor & Destructor Documentation

31.145.2.1 Fl_Tree_Item_Array()

```
Fl_Tree_Item_Array::Fl_Tree_Item_Array (
    int new_chunksize = 10 )
```

Constructor; creates an empty array.

The optional 'chunksize' can be specified to optimize memory allocation for potentially large arrays. Default chunksize is 10.

31.145.3 Member Function Documentation

31.145.3.1 add()

```
void Fl_Tree_Item_Array::add (
    Fl_Tree_Item * val )
```

Add an item* to the end of the array.

Assumes the item was created with 'new', and will remain allocated.. Fl_Tree_Item_Array will handle calling the item's destructor when the array is cleared or the item remove()'ed.

31.145.3.2 clear()

```
void Fl_Tree_Item_Array::clear ( )
```

Clear the entire array.

Each item will be deleted (destructors will be called), and the array will be cleared. total() will return 0.

31.145.3.3 deparent()

```
int Fl_Tree_Item_Array::deparent (
    int pos )
```

Deparent item at 'pos' from our list of children.

Similar to a [remove\(\)](#) without the destruction of the item. This creates an orphaned item (still allocated, has no parent) which soon after is typically reparented elsewhere.

\returns 0 on success, -1 on error (e.g. if \p 'pos' out of range)

31.145.3.4 insert()

```
void Fl_Tree_Item_Array::insert (
    int pos,
    Fl_Tree_Item * new_item )
```

Insert an item at index position pos.

Handles enlarging array if needed, total increased by 1.

If \p pos >= total(), the item is appended to the array.

If \p pos < 0, the item is prepended (works like pos == 0).

31.145.3.5 manage_item_destroy()

```
void Fl_Tree_Item_Array::manage_item_destroy (
    int val ) [inline]
```

Option to control if [Fl_Tree_Item_Array](#)'s destructor will also destroy the [Fl_Tree_Item](#)'s.

If set: items and item array is destroyed. If clear: only the item array is destroyed, not items themselves.

31.145.3.6 move()

```
int Fl_Tree_Item_Array::move (
    int to,
    int from )
```

Move item at 'from' to new position 'to' in the array.

Due to how the moving an item shuffles the array around, a positional 'move' implies things that may not be obvious:

- When 'from' moved lower in tree, appears BELOW item that was at 'to'.
- When 'from' moved higher in tree, appears ABOVE item that was at 'to'.

Returns

0 on success, -1 on range error (e.g. if 'to' or 'from' out of range)

31.145.3.7 remove() [1/2]

```
int Fl_Tree_Item_Array::remove (
    Fl_Tree_Item * item )
```

Remove the item from the array.

\returns 0 if removed, or -1 if the item was not in the array.

31.145.3.8 remove() [2/2]

```
void Fl_Tree_Item_Array::remove (
    int index )
```

Remove the item at.

Parameters

in	<i>index</i>	from the array. The item will be delete'd (if non-NULL), so its destructor will be called.
----	--------------	---

31.145.3.9 reparent()

```
int Fl_Tree_Item_Array::reparent (
    Fl_Tree_Item * item,
    Fl_Tree_Item * newparent,
    int pos )
```

Reparent specified item as a child of ourself.

Typically 'newchild' was recently orphaned with [deparent\(\)](#).

\returns 0 on success, -1 on error (e.g. if \p 'pos' out of range)

31.145.3.10 replace()

```
void Fl_Tree_Item_Array::replace (
    int index,
    Fl_Tree_Item * newitem )
```

Replace the item at `index` with `newitem`.

Old item at `index` position will be destroyed, and the new item will take it's place, and stitched into the linked list.

The documentation for this class was generated from the following files:

- [Fl_Tree_Item_Array.H](#)
- [Fl_Tree_Item_Array.cxx](#)

31.146 Fl_Tree_Prefs Class Reference

Tree widget's preferences.

```
#include <Fl_Tree_Prefs.H>
```

Public Member Functions

- [Fl_Image](#) * [closedeicon](#) () const
Return the deactivated version of the close icon, if any.
- [Fl_Image](#) * [closeicon](#) () const
Gets the default 'close' icon Returns the Fl_Image of the icon, or 0 if none.*
- void [closeicon](#) ([Fl_Image](#) *val)
Sets the icon to be used as the 'close' icon.
- [Fl_Color](#) [connectorcolor](#) () const
Get the connector color used for tree connection lines.
- void [connectorcolor](#) ([Fl_Color](#) val)
Set the connector color used for tree connection lines.
- [Fl_Tree_Connector](#) [connectorstyle](#) () const
Get the connector style.
- void [connectorstyle](#) ([Fl_Tree_Connector](#) val)
Set the connector style.
- void [connectorstyle](#) (int val)
Set the connector style [integer].

- int **connectorwidth** () const
Get the tree connection line's width.
- void **connectorwidth** (int val)
Set the tree connection line's width.
- void **do_item_draw_callback** (FI_Tree_Item *o) const
- **FI_Tree_Prefs** ()
FI_Tree_Prefs constructor.
- FI_Tree_Item_Draw_Callback * **item_draw_callback** () const
- void **item_draw_callback** (FI_Tree_Item_Draw_Callback *cb, void *data=0)
- **FI_Tree_Item_Draw_Mode** **item_draw_mode** () const
Get the 'item draw mode' used for the tree.
- void **item_draw_mode** (FI_Tree_Item_Draw_Mode val)
Set the 'item draw mode' used for the tree to val.
- void * **item_draw_user_data** () const
- **FI_Color** **item_labelbgcolor** () const
Get the default label background color.
- void **item_labelbgcolor** (FI_Color val)
Set the default label background color.
- **FI_Color** **item_labelfgcolor** () const
Get the default label foreground color.
- void **item_labelfgcolor** (FI_Color val)
Set the default label foreground color.
- **FI_Font** **item_labelfont** () const
Return the label's font.
- void **item_labelfont** (FI_Font val)
Set the label's font to val.
- **FI_Fontsize** **item_labelsize** () const
Return the label's size in pixels.
- void **item_labelsize** (FI_Fontsize val)
Set the label's size in pixels to val.
- **FI_Tree_Item_Reselect_Mode** **item_reselect_mode** () const
Returns the current item re/selection mode.
- void **item_reselect_mode** (FI_Tree_Item_Reselect_Mode mode)
Sets the item re/selection mode.
- **FI_Color** **labelbgcolor** () const
Obsolete: Get the default label background color. Please use [item_labelbgcolor\(\)](#) instead.
- void **labelbgcolor** (FI_Color val)
Obsolete: Set the default label background color. Please use [item_labelbgcolor\(FI_Color\)](#) instead.
- **FI_Color** **labelfgcolor** () const
Obsolete: Get the default label foreground color. Please use [item_labelfgcolor\(\)](#) instead.
- void **labelfgcolor** (FI_Color val)
Obsolete: Set the default label foreground color. Please use [item_labelfgcolor\(FI_Color\)](#) instead.
- **FI_Font** **labelfont** () const
Obsolete: Return the label's font. Please use [item_labelfont\(\)](#) instead.
- void **labelfont** (FI_Font val)
Obsolete: Set the label's font to val. Please use [item_labelfont\(FI_Font\)](#) instead.
- int **labelmarginleft** () const
Get the label's left margin value in pixels.
- void **labelmarginleft** (int val)
Set the label's left margin value in pixels.
- **FI_Fontsize** **labelsize** () const

- Obsolete: Return the label's size in pixels. Please use [item_labelsize\(\)](#) instead.*

 - void **labelsiz** ([FI_Fontsize](#) val)

Obsolete: Set the label's size in pixels to val. Please use [item_labelsize\(FI_Fontsize\)](#) instead.
- int **linespacing** () const
 - Get the line spacing value in pixels.*
- void **linespacing** (int val)
 - Set the line spacing value in pixels.*
- int **marginbottom** () const
 - Get the bottom margin's value in pixels.*
- void **marginbottom** (int val)
 - Set the bottom margin's value in pixels This is the extra distance the vertical scroller lets you travel.*
- int **marginleft** () const
 - Get the left margin's value in pixels.*
- void **marginleft** (int val)
 - Set the left margin's value in pixels.*
- int **margin** () const
 - Get the top margin's value in pixels.*
- void **margin** (int val)
 - Set the top margin's value in pixels.*
- int **openchild_marginbottom** () const
 - Get the margin below an open child in pixels.*
- void **openchild_marginbottom** (int val)
 - Set the margin below an open child in pixels.*
- [FI_Image](#) * **opendeicon** () const
 - Return the deactivated version of the open icon, if any.*
- [FI_Image](#) * **openicon** () const
 - Get the current default 'open' icon.*
- void **openicon** ([FI_Image](#) *val)
 - Sets the default icon to be used as the 'open' icon when items are add()ed to the tree.*
- [FI_Boxtype](#) **selectbox** () const
 - Get the default selection box's box drawing style as an [FI_Boxtype](#).*
- void **selectbox** ([FI_Boxtype](#) val)
 - Set the default selection box's box drawing style to val.*
- [FI_Tree_Select](#) **selectmode** () const
 - Get the selection mode used for the tree.*
- void **selectmode** ([FI_Tree_Select](#) val)
 - Set the selection mode used for the tree to val.*
- char **showcollapse** () const
 - Returns 1 if the collapse icon is enabled, 0 if not.*
- void **showcollapse** (int val)
 - Set if we should show the collapse icon or not.*
- int **showroot** () const
 - Returns 1 if the root item is to be shown, or 0 if not.*
- void **showroot** (int val)
 - Set if the root item should be shown or not.*
- [FI_Tree_Sort](#) **sortorder** () const
 - Get the default sort order value.*
- void **sortorder** ([FI_Tree_Sort](#) val)
 - Set the default sort order value.*
- [FI_Image](#) * **userdeicon** () const
 - Return the deactivated version of the user icon, if any.*

- `Fl_Image * usericon () const`
Gets the default 'user icon' (default is 0)
- `void usericon (Fl_Image *val)`
Sets the default 'user icon' Returns the Fl_Image of the icon, or 0 if none (default).*
- `int usericonmarginleft () const`
Get the user icon's left margin value in pixels.
- `void usericonmarginleft (int val)`
Set the user icon's left margin value in pixels.
- `int widgetmarginleft () const`
Get the widget()'s left margin value in pixels.
- `void widgetmarginleft (int val)`
Set the widget's left margin value in pixels.
- `~Fl_Tree_Prefs ()`
Fl_Tree_Prefs destructor.

31.146.1 Detailed Description

Tree widget's preferences.

[Fl_Tree](#)'s Preferences class.

This class manages the [Fl_Tree](#)'s defaults. You should probably be using the methods in [Fl_Tree](#) instead of trying to accessing tree's preferences settings directly.

31.146.2 Member Function Documentation

31.146.2.1 closedeicon()

```
Fl_Image * Fl_Tree_Prefs::closedeicon ( ) const [inline]
```

Return the deactivated version of the close icon, if any.

Returns 0 if none.

31.146.2.2 closeicon()

```
void Fl_Tree_Prefs::closeicon (
    Fl_Image * val )
```

Sets the icon to be used as the 'close' icon.

This overrides the built in default '[-]' icon.

Parameters

<code>in</code>	<code>val</code>	– The new image, or zero to use the default '[-]' icon.
-----------------	------------------	---

31.146.2.3 item_draw_mode()

```
void Fl_Tree_Prefs::item_draw_mode (
    Fl_Tree_Item_Draw_Mode val ) [inline]
```

Set the 'item draw mode' used for the tree to `val`.

This affects how items in the tree are drawn, such as when a `widget()` is defined. See [Fl_Tree_Item_Draw_Mode](#) for possible values.

31.146.2.4 item_labelbgcolor() [1/2]

```
Fl_Color Fl_Tree_Prefs::item_labelbgcolor (
    void ) const [inline]
```

Get the default label background color.
This returns the `Fl_Tree::color()` unless `item_labelbgcolor()` has been set explicitly.

31.146.2.5 `item_labelbgcolor()` [2/2]

```
void Fl_Tree_Prefs::item_labelbgcolor (
    Fl_Color val ) [inline]
```

Set the default label background color.
Once set, overrides the default behavior of using `Fl_Tree::color()`.

31.146.2.6 `marginbottom()`

```
int Fl_Tree_Prefs::marginbottom ( ) const [inline]
```

Get the bottom margin's value in pixels.
This is the extra distance the vertical scroller lets you travel.

31.146.2.7 `opendeicon()`

```
Fl_Image * Fl_Tree_Prefs::opendeicon ( ) const [inline]
```

Return the deactivated version of the open icon, if any.
Returns 0 if none.

31.146.2.8 `openicon()` [1/2]

```
Fl_Image * Fl_Tree_Prefs::openicon ( ) const [inline]
```

Get the current default 'open' icon.
Returns the `Fl_Image*` of the icon, or 0 if none.

31.146.2.9 `openicon()` [2/2]

```
void Fl_Tree_Prefs::openicon (
    Fl_Image * val )
```

Sets the default icon to be used as the 'open' icon when items are add()ed to the tree.
This overrides the built in default '[+]' icon.

Parameters

<code>in</code>	<code>val</code>	– The new image, or zero to use the default '[+]' icon.
-----------------	------------------	---

31.146.2.10 `selectmode()`

```
void Fl_Tree_Prefs::selectmode (
    Fl_Tree_Select val ) [inline]
```

Set the selection mode used for the tree to `val`.
This affects how items in the tree are selected when clicked on and dragged over by the mouse. See `Fl_Tree_Select` for possible values.

31.146.2.11 `showcollapse()`

```
void Fl_Tree_Prefs::showcollapse (
    int val ) [inline]
```

Set if we should show the collapse icon or not.
If collapse icons are disabled, the user will not be able to interactively collapse items in the tree, unless the application provides some other means via `open()` and `close()`.

Parameters

in	val	1: shows collapse icons (default), 0: hides collapse icons.
----	-----	--

31.146.2.12 showroot()

```
void Fl_Tree_Prefs::showroot (
    int val ) [inline]
```

Set if the root item should be shown or not.

Parameters

in	val	1 – show the root item (default) 0 – hide the root item.
----	-----	---

31.146.2.13 sortorder()

```
void Fl_Tree_Prefs::sortorder (
    Fl_Tree_Sort val ) [inline]
```

Set the default sort order value.

Defines the order new items appear when add()ed to the tree. See `Fl_Tree_Sort` for possible values.

31.146.2.14 userdeicon()

```
Fl_Image * Fl_Tree_Prefs::userdeicon ( ) const [inline]
```

Return the deactivated version of the user icon, if any.

Returns 0 if none.

The documentation for this class was generated from the following files:

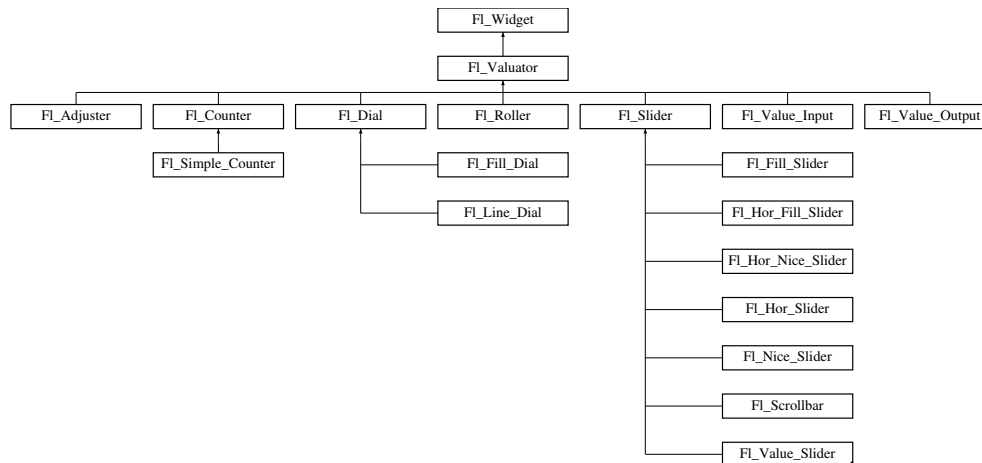
- [Fl_Tree_Prefs.H](#)
- [Fl_Tree_Prefs.cxx](#)

31.147 FI_Valuator Class Reference

The `Fl_Valuator` class controls a single floating-point value and provides a consistent interface to set the value, range, and step, and insures that callbacks are done the same for every object.

```
#include <Fl_Valuator.H>
```

Inheritance diagram for `Fl_Valuator`:



Public Member Functions

- void **bounds** (double a, double b)
Sets the minimum (a) and maximum (b) values for the valuator widget.
- double **clamp** (double)
Clamps the passed value to the valuator range.
- virtual int **format** (char *)
Uses internal rules to format the fields numerical value into the character array pointed to by the passed parameter.
- double **increment** (double, int)
Adds n times the step value to the passed value.
- double **maximum** () const
Gets the maximum value for the valuator.
- void **maximum** (double a)
Sets the maximum value for the valuator.
- double **minimum** () const
Gets the minimum value for the valuator.
- void **minimum** (double a)
Sets the minimum value for the valuator.
- void **precision** (int digits)
Sets the step value to $1.0 / 10^{\text{digits}}$.
- void **range** (double a, double b)
Sets the minimum and maximum values for the valuator.
- double **round** (double)
Round the passed value to the nearest step increment.
- double **step** () const
Gets or sets the step value.
- void **step** (double a, int b)
See double [FL_Valuator::step\(\)](#) const
- void **step** (double s)
See double [FL_Valuator::step\(\)](#) const.
- void **step** (int a)
See double [FL_Valuator::step\(\)](#) const
- double **value** () const
Gets the floating point(double) value.
- int **value** (double)
Sets the current value.

Protected Member Functions

- [FI_Valuator](#) (int X, int Y, int W, int H, const char *L)
Creates a new [FI_Valuator](#) widget using the given position, size, and label string.
- void **handle_drag** (double newvalue)
Called during a drag operation, after an `FL_WHEN_CHANGED` event is received and before the callback.
- void **handle_push** ()
Stores the current value in the previous value.
- void **handle_release** ()
Called after an `FL_WHEN_RELEASE` event is received and before the callback.
- int **horizontal** () const
Tells if the valuator is an `FL_HORIZONTAL` one.
- double **previous_value** () const
Gets the previous floating point value before an event changed it.
- void **set_value** (double v)
Sets the current floating point value.
- double **softclamp** (double)
Clamps the value, but accepts v if the previous value is not already out of range.
- virtual void **value_damage** ()
Asks for partial redraw.

Additional Inherited Members

31.147.1 Detailed Description

The [FI_Valuator](#) class controls a single floating-point value and provides a consistent interface to set the value, range, and step, and insures that callbacks are done the same for every object. There are probably more of these classes in FLTK than any others:

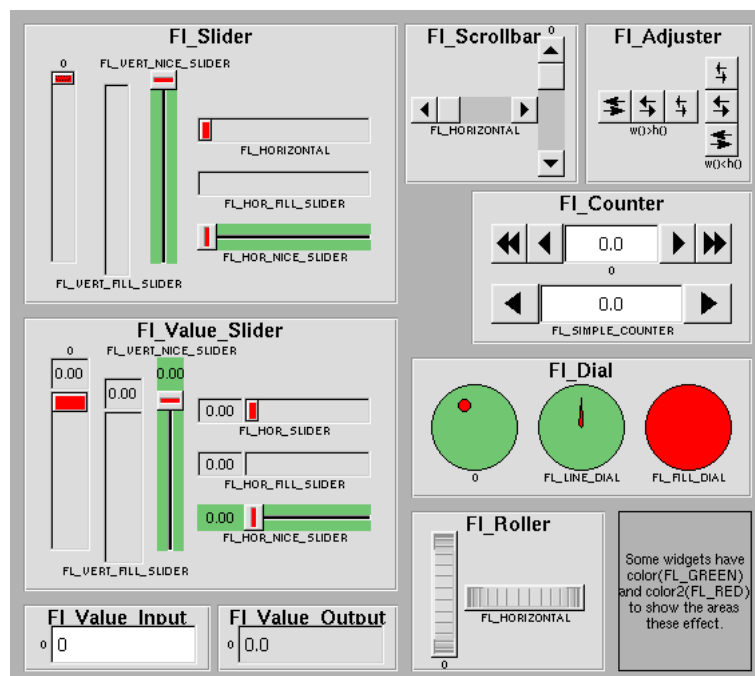


Figure 31.48 Valuators derived from [FI_Valuator](#)s

In the above diagram each box surrounds an actual subclass. These are further differentiated by setting the [type\(\)](#) of the widget to the symbolic value labeling the widget. The ones labelled "0" are the default versions with a [type\(0\)](#). For consistency the symbol `FL_VERTICAL` is defined as zero.

31.147.2 Constructor & Destructor Documentation

31.147.2.1 Fl_Valuator()

```
Fl_Valuator::Fl_Valuator (
    int X,
    int Y,
    int W,
    int H,
    const char * L ) [protected]
```

Creates a new [Fl_Valuator](#) widget using the given position, size, and label string. The default boxtype is FL_NO_BOX.

31.147.3 Member Function Documentation

31.147.3.1 format()

```
int Fl_Valuator::format (
    char * buffer ) [virtual]
```

Uses internal rules to format the fields numerical value into the character array pointed to by the passed parameter. The actual format used depends on the current step value. If the step value has been set to zero then a %g format is used. If the step value is non-zero, then a %.*f format is used, where the precision is calculated to show sufficient digits for the current step value. An integer step value, such as 1 or 1.0, gives a precision of 0, so the formatted value will appear as an integer.

This method is used by the `Fl_Valuator_...` group of widgets to format the current value into a text string. The return value is the length of the formatted text. The formatted value is written into `buffer`. `buffer` should have space for at least 128 bytes.

You may override this function to create your own text formatting.

31.147.3.2 increment()

```
double Fl_Valuator::increment (
    double v,
    int n )
```

Adds `n` times the step value to the passed value.

If step was set to zero it uses `fabs(maximum() - minimum()) / 100`.

31.147.3.3 maximum() [1/2]

```
double Fl_Valuator::maximum ( ) const [inline]
```

Gets the maximum value for the valuator.

31.147.3.4 maximum() [2/2]

```
void Fl_Valuator::maximum (
    double a ) [inline]
```

Sets the maximum value for the valuator.

31.147.3.5 minimum() [1/2]

```
double Fl_Valuator::minimum ( ) const [inline]
```

Gets the minimum value for the valuator.

31.147.3.6 minimum() [2/2]

```
void Fl_Valuator::minimum (
    double a ) [inline]
```

Sets the minimum value for the valuator.

31.147.3.7 precision()

```
void Fl_Valuator::precision (
    int digits )
```

Sets the step value to $1.0 / 10^{\text{digits}}$.

Precision `digits` is limited to 0...9 to avoid internal overflow errors. Values outside this range are clamped.

Note

For negative values of `digits` the step value is set to $A = 1.0$ and $B = 1$, i.e. $1.0/1 = 1$.

31.147.3.8 range()

```
void Fl_Valuator::range (
    double a,
    double b ) [inline]
```

Sets the minimum and maximum values for the valuator.

When the user manipulates the widget, the value is limited to this range. This clamping is done *after* rounding to the step value (this makes a difference if the range is not a multiple of the step).

The minimum may be greater than the maximum. This has the effect of "reversing" the object so the larger values are in the opposite direction. This also switches which end of the filled sliders is filled.

Some widgets consider this a "soft" range. This means they will stop at the range, but if the user releases and grabs the control again and tries to move it further, it is allowed.

The range may affect the display. You must [redraw\(\)](#) the widget after changing the range.

31.147.3.9 round()

```
double Fl_Valuator::round (
    double v )
```

Round the passed value to the nearest step increment.

Does nothing if step is zero.

31.147.3.10 step()

```
double Fl_Valuator::step ( ) const [inline]
```

Gets or sets the step value.

As the user moves the mouse the value is rounded to the nearest multiple of the step value. This is done *before* clamping it to the range. For most widgets the default step is zero.

For precision the step is stored as the ratio of a double A and an integer $B = A/B$. You can set these values directly. Currently setting a floating point value sets the nearest $A/1$ or $1/B$ value possible.

31.147.3.11 value() [1/2]

```
double Fl_Valuator::value ( ) const [inline]
```

Gets the floating point(double) value.

See int [value\(double\)](#)

31.147.3.12 value() [2/2]

```
int Fl_Valuator::value (
    double v )
```

Sets the current value.

The new value is *not* clamped or otherwise changed before storing it. Use [clamp\(\)](#) or [round\(\)](#) to modify the value before calling [value\(\)](#). The widget is redrawn if the new value is different than the current one. The initial value is zero.

[changed\(\)](#) will return true if the user has moved the slider, but it will be turned off by [value\(x\)](#) and just before doing a callback (the callback can turn it back on if desired).

31.147.3.13 value_damage()

```
void Fl_Valuator::value_damage ( ) [protected], [virtual]
```

Asks for partial redraw.

Reimplemented in [Fl_Adjuster](#).

The documentation for this class was generated from the following files:

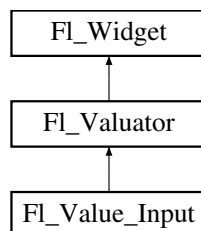
- [Fl_Valuator.H](#)
- [Fl_Valuator.cxx](#)

31.148 Fl_Value_Input Class Reference

The [Fl_Value_Input](#) widget displays a numeric value.

```
#include <Fl_Value_Input.H>
```

Inheritance diagram for [Fl_Value_Input](#):

**Public Member Functions**

- [Fl_Color](#) [cursor_color](#) () const
Gets the color of the text cursor.
- void [cursor_color](#) ([Fl_Color](#) n)
Sets the color of the text cursor.
- [Fl_Value_Input](#) (int x, int y, int w, int h, const char *l=0)
Creates a new [Fl_Value_Input](#) widget using the given position, size, and label string.
- int [handle](#) (int)
Handles the specified event.
- void [resize](#) (int, int, int, int)
Changes the size or position of the widget.
- int [shortcut](#) () const
Returns the current shortcut key for the Input.
- void [shortcut](#) (int s)
Sets the shortcut key to s.
- char [soft](#) () const
If "soft" is turned on, the user is allowed to drag the value outside the range.
- void [soft](#) (char s)
See void [Fl_Value_Input::soft\(char s\)](#)

- [FI_Color](#) `textcolor ()` const
Gets the color of the text in the value box.
- void `textcolor (FI_Color n)`
Sets the color of the text in the value box.
- [FI_Font](#) `textfont ()` const
Gets the typeface of the text in the value box.
- void `textfont (FI_Font s)`
Sets the typeface of the text in the value box.
- [FI_Fontsize](#) `textsize ()` const
Gets the size of the text in the value box.
- void `textsize (FI_Fontsize s)`
Sets the size of the text in the value box.

Public Attributes

- [FI_Input](#) `input`

Protected Member Functions

- void `draw ()`
Draws the widget.

Additional Inherited Members

31.148.1 Detailed Description

The [FI_Value_Input](#) widget displays a numeric value.

The user can click in the text field and edit it - there is in fact a hidden [FI_Input](#) widget with type(FL_FLOAT_INPUT) or type(FL_INT_INPUT) in there - and when they hit return or tab the value updates to what they typed and the callback is done.

If `step()` is non-zero and integral, then the range of numbers is limited to integers instead of floating point numbers. As well as displaying the value as an integer, typed input is also limited to integer values, even if the hidden [FI_Input](#) widget is of type(FL_FLOAT_INPUT).

If `step()` is non-zero, the user can also drag the mouse across the object and thus slide the value. The left button moves one `step()` per pixel, the middle by 10 `step()`, and the right button by 100 * `step()`. It is therefore impossible to select text by dragging across it, although clicking can still move the insertion cursor.

If `step()` is non-zero and integral, then the range of numbers are limited to integers instead of floating point values.

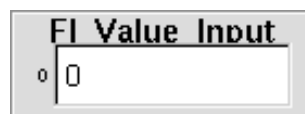


Figure 31.49 FI_Value_Input

31.148.2 Constructor & Destructor Documentation

31.148.2.1 FI_Value_Input()

```
Fl_Value_Input::Fl_Value_Input (
    int X,
    int Y,
    int W,
    int H,
    const char * l = 0 )
```

Creates a new [FI_Value_Input](#) widget using the given position, size, and label string. The default boxtype is FL_DOWN_BOX.

31.148.3 Member Function Documentation

31.148.3.1 `cursor_color()` [1/2]

```
Fl_Color Fl_Value_Input::cursor_color ( ) const [inline]
```

Gets the color of the text cursor.

The text cursor is black by default.

31.148.3.2 `cursor_color()` [2/2]

```
void Fl_Value_Input::cursor_color (
    Fl_Color n ) [inline]
```

Sets the color of the text cursor.

The text cursor is black by default.

31.148.3.3 `draw()`

```
void Fl_Value_Input::draw ( ) [protected], [virtual]
```

Draws the widget.

Never call this function directly. FLTK will schedule redrawing whenever needed. If your widget must be redrawn as soon as possible, call `redraw()` instead.

Override this function to draw your own widgets.

If you ever need to call another widget's draw method *from within your own `draw()` method*, e.g. for an embedded scrollbar, you can do it (because `draw()` is virtual) like this:

```
Fl_Widget *s = &scroll;           // scroll is an embedded Fl_Scrollbar
s->draw();                          // calls Fl_Scrollbar::draw()
```

Implements `Fl_Widget`.

31.148.3.4 `handle()`

```
int Fl_Value_Input::handle (
    int event ) [virtual]
```

Handles the specified event.

You normally don't call this method directly, but instead let FLTK do it when the user interacts with the widget.

When implemented in a widget, this function must return 0 if the widget does not use the event or 1 otherwise.

Most of the time, you want to call the inherited `handle()` method in your overridden method so that you don't short-circuit events that you don't handle. In this last case you should return the callee retval.

Parameters

in	<code>event</code>	the kind of event received
----	--------------------	----------------------------

Return values

0	if the event was not used or understood
1	if the event was used and can be deleted

See also

[Fl_Event](#)

Reimplemented from [Fl_Widget](#).

31.148.3.5 `resize()`

```
void Fl_Value_Input::resize (
```

```

    int x,
    int y,
    int w,
    int h ) [virtual]

```

Changes the size or position of the widget.

This is a virtual function so that the widget may implement its own handling of resizing. The default version does *not* call the [redraw\(\)](#) method, but instead relies on the parent widget to do so because the parent may know a faster way to update the display, such as scrolling from the old position.

Some window managers under X11 call [resize\(\)](#) a lot more often than needed. Please verify that the position or size of a widget did actually change before doing any extensive calculations.

position(X, Y) is a shortcut for [resize\(X, Y, w\(\), h\(\)\)](#), and size(W, H) is a shortcut for [resize\(x\(\), y\(\), W, H\)](#).

Parameters

in	<i>x,y</i>	new position relative to the parent window
in	<i>w,h</i>	new size

See also

[position\(int,int\)](#), [size\(int,int\)](#)

Reimplemented from [Fl_Widget](#).

31.148.3.6 shortcut() [1/2]

```
int Fl_Value_Input::shortcut ( ) const [inline]
```

Returns the current shortcut key for the Input.

See also

[Fl_Value_Input::shortcut\(int\)](#)

31.148.3.7 shortcut() [2/2]

```
void Fl_Value_Input::shortcut (
    int s ) [inline]
```

Sets the shortcut key to *s*.

Setting this overrides the use of '&' in the [label\(\)](#). The value is a bitwise OR of a key and a set of shift flags, for example `FL_ALT | 'a'`, `FL_ALT | (FL_F + 10)`, or just `'a'`. A value of 0 disables the shortcut.

The key can be any value returned by [Fl::event_key\(\)](#), but will usually be an ASCII letter. Use a lower-case letter unless you require the shift key to be held down.

The shift flags can be any set of values accepted by [Fl::event_state\(\)](#). If the bit is on that shift key must be pushed. Meta, Alt, Ctrl, and Shift must be off if they are not in the shift flags (zero for the other bits indicates a "don't care" setting).

31.148.3.8 soft()

```
char Fl_Value_Input::soft ( ) const [inline]
```

If "soft" is turned on, the user is allowed to drag the value outside the range.

If they drag the value to one of the ends, let go, then grab again and continue to drag, they can get to any value.

The default is true.

31.148.3.9 textcolor()

```
Fl_Color Fl_Value_Input::textcolor ( ) const [inline]
```

Gets the color of the text in the value box.

31.148.3.10 textfont() [1/2]

```
Fl_Font Fl_Value_Input::textfont ( ) const [inline]
```

Gets the typeface of the text in the value box.

31.148.3.11 textfont() [2/2]

```
void Fl_Value_Input::textfont (
    Fl_Font s ) [inline]
```

Sets the typeface of the text in the value box.

31.148.3.12 textsize() [1/2]

```
Fl_Fontsize Fl_Value_Input::textsize ( ) const [inline]
```

Gets the size of the text in the value box.

31.148.3.13 textsize() [2/2]

```
void Fl_Value_Input::textsize (
    Fl_Fontsize s ) [inline]
```

Sets the size of the text in the value box.

The documentation for this class was generated from the following files:

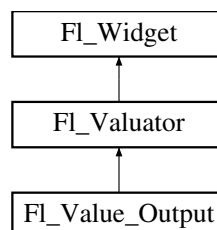
- Fl_Value_Input.H
- Fl_Value_Input.cxx

31.149 FI_Value_Output Class Reference

The `Fl_Value_Output` widget displays a floating point value.

```
#include <Fl_Value_Output.H>
```

Inheritance diagram for `Fl_Value_Output`:

**Public Member Functions**

- `Fl_Value_Output` (int *x*, int *y*, int *w*, int *h*, const char **l*=0)
 - Creates a new Fl_Value_Output widget using the given position, size, and label string.*
- int `handle` (int)
 - Handles the specified event.*
- uchar `soft` () const
 - If "soft" is turned on, the user is allowed to drag the value outside the range.*
- void `soft` (uchar *s*)

If "soft" is turned on, the user is allowed to drag the value outside the range.

- [FI_Color](#) `textcolor` () const
Sets the color of the text in the value box.
- void `textcolor` ([FI_Color](#) s)
Gets the color of the text in the value box.
- [FI_Font](#) `textfont` () const
Gets the typeface of the text in the value box.
- void `textfont` ([FI_Font](#) s)
Sets the typeface of the text in the value box.
- [FI_Fontsize](#) `textsize` () const
Gets the size of the text in the value box.
- void `textsize` ([FI_Fontsize](#) s)

Protected Member Functions

- void `draw` ()
Draws the widget.

Additional Inherited Members

31.149.1 Detailed Description

The [FI_Value_Output](#) widget displays a floating point value.

If `step()` is not zero, the user can adjust the value by dragging the mouse left and right. The left button moves one `step()` per pixel, the middle by $10 * \text{step}()$, and the right button by $100 * \text{step}()$.

This is much lighter-weight than [FI_Value_Input](#) because it contains no text editing code or character buffer.

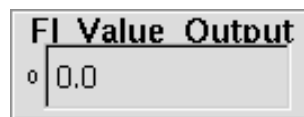


Figure 31.50 FI_Value_Output

31.149.2 Constructor & Destructor Documentation

31.149.2.1 FI_Value_Output()

```
FI_Value_Output::FI_Value_Output (
    int X,
    int Y,
    int W,
    int H,
    const char * l = 0 )
```

Creates a new [FI_Value_Output](#) widget using the given position, size, and label string.

The default boxtype is `FL_NO_BOX`.

Inherited destructor destroys the Valuator.

31.149.3 Member Function Documentation

31.149.3.1 draw()

```
void Fl_Value_Output::draw ( ) [protected], [virtual]
```

Draws the widget.

Never call this function directly. FLTK will schedule redrawing whenever needed. If your widget must be redrawn as soon as possible, call [redraw\(\)](#) instead.

Override this function to draw your own widgets.

If you ever need to call another widget's draw method *from within your own [draw\(\)](#) method*, e.g. for an embedded scrollbar, you can do it (because [draw\(\)](#) is virtual) like this:

```
Fl_Widget *s = &scroll;           // scroll is an embedded Fl_Scrollbar
s->draw();                         // calls Fl_Scrollbar::draw()
```

Implements [Fl_Widget](#).

31.149.3.2 handle()

```
int Fl_Value_Output::handle (
    int event ) [virtual]
```

Handles the specified event.

You normally don't call this method directly, but instead let FLTK do it when the user interacts with the widget.

When implemented in a widget, this function must return 0 if the widget does not use the event or 1 otherwise.

Most of the time, you want to call the inherited [handle\(\)](#) method in your overridden method so that you don't short-circuit events that you don't handle. In this last case you should return the callee retval.

Parameters

in	<i>event</i>	the kind of event received
----	--------------	----------------------------

Return values

0	if the event was not used or understood
1	if the event was used and can be deleted

See also

[Fl_Event](#)

Reimplemented from [Fl_Widget](#).

31.149.3.3 soft() [1/2]

```
uchar Fl_Value_Output::soft ( ) const [inline]
```

If "soft" is turned on, the user is allowed to drag the value outside the range.

If they drag the value to one of the ends, let go, then grab again and continue to drag, they can get to any value. Default is one.

31.149.3.4 soft() [2/2]

```
void Fl_Value_Output::soft (
    uchar s ) [inline]
```

If "soft" is turned on, the user is allowed to drag the value outside the range.

If they drag the value to one of the ends, let go, then grab again and continue to drag, they can get to any value. Default is one.

31.149.3.5 textcolor() [1/2]

```
Fl_Color Fl_Value_Output::textcolor ( ) const [inline]
```

Sets the color of the text in the value box.

31.149.3.6 textcolor() [2/2]

```
void Fl_Value_Output::textcolor (
    Fl_Color s ) [inline]
```

Gets the color of the text in the value box.

31.149.3.7 textfont() [1/2]

```
Fl_Font Fl_Value_Output::textfont ( ) const [inline]
```

Gets the typeface of the text in the value box.

31.149.3.8 textfont() [2/2]

```
void Fl_Value_Output::textfont (
    Fl_Font s ) [inline]
```

Sets the typeface of the text in the value box.

31.149.3.9 textsize()

```
Fl_Fontsize Fl_Value_Output::textsize ( ) const [inline]
```

Gets the size of the text in the value box.

The documentation for this class was generated from the following files:

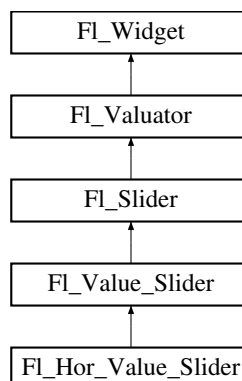
- Fl_Value_Output.H
- Fl_Value_Output.cxx

31.150 Fl_Value_Slider Class Reference

The [Fl_Value_Slider](#) widget is a [Fl_Slider](#) widget with a box displaying the current value.

```
#include <Fl_Value_Slider.H>
```

Inheritance diagram for [Fl_Value_Slider](#):



Public Member Functions

- [Fl_Value_Slider](#) (int *x*, int *y*, int *w*, int *h*, const char **l*=0)

Creates a new [Fl_Value_Slider](#) widget using the given position, size, and label string.
- int [handle](#) (int)

Handles the specified event.
- [Fl_Color](#) [textcolor](#) () const

Gets the color of the text in the value box.
- void [textcolor](#) ([Fl_Color](#) *s*)

Sets the color of the text in the value box.
- [Fl_Font](#) [textfont](#) () const

Gets the typeface of the text in the value box.
- void [textfont](#) ([Fl_Font](#) *s*)

Sets the typeface of the text in the value box.
- [Fl_Fontsize](#) [textsize](#) () const

Gets the size of the text in the value box.
- void [textsize](#) ([Fl_Fontsize](#) *s*)

Sets the size of the text in the value box.

Protected Member Functions

- void [draw](#) ()

Draws the widget.

Additional Inherited Members

31.150.1 Detailed Description

The [Fl_Value_Slider](#) widget is a [Fl_Slider](#) widget with a box displaying the current value.

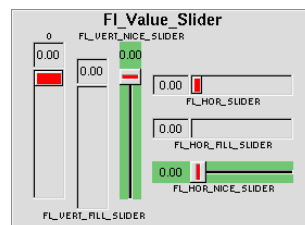


Figure 31.51 [Fl_Value_Slider](#)

31.150.2 Constructor & Destructor Documentation

31.150.2.1 [Fl_Value_Slider](#)()

```
Fl_Value_Slider::Fl_Value_Slider (
    int X,
    int Y,
    int W,
    int H,
    const char * l = 0 )
```

Creates a new [Fl_Value_Slider](#) widget using the given position, size, and label string. The default boxtype is `FL_DOWN_BOX`.

31.150.3 Member Function Documentation

31.150.3.1 draw()

```
void Fl_Value_Slider::draw ( ) [protected], [virtual]
```

Draws the widget.

Never call this function directly. FLTK will schedule redrawing whenever needed. If your widget must be redrawn as soon as possible, call [redraw\(\)](#) instead.

Override this function to draw your own widgets.

If you ever need to call another widget's draw method *from within your own draw() method*, e.g. for an embedded scrollbar, you can do it (because [draw\(\)](#) is virtual) like this:

```
Fl_Widget *s = &scroll;           // scroll is an embedded Fl_Scrollbar
s->draw();                          // calls Fl_Scrollbar::draw()
```

Reimplemented from [Fl_Slider](#).

31.150.3.2 handle()

```
int Fl_Value_Slider::handle (
    int event ) [virtual]
```

Handles the specified event.

You normally don't call this method directly, but instead let FLTK do it when the user interacts with the widget.

When implemented in a widget, this function must return 0 if the widget does not use the event or 1 otherwise.

Most of the time, you want to call the inherited [handle\(\)](#) method in your overridden method so that you don't short-circuit events that you don't handle. In this last case you should return the callee retval.

Parameters

in	<i>event</i>	the kind of event received
----	--------------	----------------------------

Return values

0	if the event was not used or understood
1	if the event was used and can be deleted

See also

[Fl_Event](#)

Reimplemented from [Fl_Slider](#).

31.150.3.3 textcolor() [1/2]

```
Fl_Color Fl_Value_Slider::textcolor ( ) const [inline]
```

Gets the color of the text in the value box.

31.150.3.4 textcolor() [2/2]

```
void Fl_Value_Slider::textcolor (
    Fl_Color s ) [inline]
```

Sets the color of the text in the value box.

31.150.3.5 `textfont()` [1/2]

```
Fl_Font Fl_Value_Slider::textfont ( ) const [inline]
```

Gets the typeface of the text in the value box.

31.150.3.6 `textfont()` [2/2]

```
void Fl_Value_Slider::textfont (
    Fl_Font s ) [inline]
```

Sets the typeface of the text in the value box.

31.150.3.7 `textsize()` [1/2]

```
Fl_Fontsize Fl_Value_Slider::textsize ( ) const [inline]
```

Gets the size of the text in the value box.

31.150.3.8 `textsize()` [2/2]

```
void Fl_Value_Slider::textsize (
    Fl_Fontsize s ) [inline]
```

Sets the size of the text in the value box.

The documentation for this class was generated from the following files:

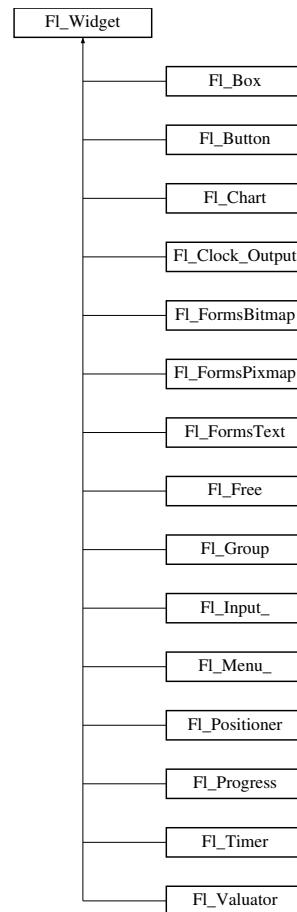
- `Fl_Value_Slider.H`
- `Fl_Value_Slider.cxx`

31.151 `Fl_Widget` Class Reference

`Fl_Widget` is the base class for all widgets in FLTK.

```
#include <Fl_Widget.H>
```

Inheritance diagram for `Fl_Widget`:



Public Member Functions

- void `_clear_fullscreen` ()
- void `_set_fullscreen` ()
- void `activate` ()
 - Activates the widget.*
- unsigned int `active` () const
 - Returns whether the widget is active.*
- int `active_r` () const
 - Returns whether the widget and all of its parents are active.*
- `FL_Align` `align` () const
 - Gets the label alignment.*
- void `align` (`FL_Align` alignment)
- long `argument` () const
 - Gets the current user data (long) argument that is passed to the callback function.*
- void `argument` (long v)
 - Sets the current user data (long) argument that is passed to the callback function.*
- virtual class `FL_Gl_Window` * `as_gl_window` ()
 - Returns an `FL_Gl_Window` pointer if this widget is an `FL_Gl_Window`.*
- virtual `FL_Group` * `as_group` ()
 - Returns an `FL_Group` pointer if this widget is an `FL_Group`.*
- virtual `FL_Window` * `as_window` ()
 - Returns an `FL_Window` pointer if this widget is an `FL_Window`.*
- `FL_Boxtype` `box` () const

- Gets the box type of the widget.*

 - void `box` (`FI_Boxtype` new_box)
- Sets the box type for the widget.*

 - `FI_Callback_p` `callback` () const
- Gets the current callback function for the widget.*

 - void `callback` (`FI_Callback` *cb)
- Sets the current callback function for the widget.*

 - void `callback` (`FI_Callback` *cb, void *p)
- Sets the current callback function for the widget.*

 - void `callback` (`FI_Callback0` *cb)
- Sets the current callback function for the widget.*

 - void `callback` (`FI_Callback1` *cb, long p=0)
- Sets the current callback function for the widget.*

 - unsigned int `changed` () const
- Checks if the widget value changed since the last callback.*

 - void `clear_active` ()
- Marks the widget as inactive without sending events or changing focus.*

 - void `clear_changed` ()
- Marks the value of the widget as unchanged.*

 - void `clear_damage` (`uchar` c=0)
- Clears or sets the damage flags.*

 - void `clear_output` ()
- Sets a widget to accept input.*

 - void `clear_visible` ()
- Hides the widget.*

 - void `clear_visible_focus` ()
- Disables keyboard focus navigation with this widget.*

 - `FI_Color` `color` () const
- Gets the background color of the widget.*

 - void `color` (`FI_Color` bg)
- Sets the background color of the widget.*

 - void `color` (`FI_Color` bg, `FI_Color` sel)
- Sets the background and selection color of the widget.*

 - `FI_Color` `color2` () const
- For back compatibility only.*

 - void `color2` (unsigned a)
- For back compatibility only.*

 - int `contains` (const `FI_Widget` *w) const
- Checks if w is a child of this widget.*

 - void `copy_label` (const char *new_label)
- Sets the current label.*

 - void `copy_tooltip` (const char *text)
- Sets the current tooltip text.*

 - `uchar` `damage` () const
- Returns non-zero if `draw()` needs to be called.*

 - void `damage` (`uchar` c)
- Sets the damage bits for the widget.*

 - void `damage` (`uchar` c, int x, int y, int w, int h)
- Sets the damage bits for an area inside the widget.*

 - int `damage_resize` (int, int, int, int)
- Internal use only.*

- void `deactivate` ()
Deactivates the widget.
- `Fl_Image * deimage` ()
Gets the image that is used as part of the widget label.
- const `Fl_Image * deimage` () const
- void `deimage` (`Fl_Image &img`)
Sets the image to use as part of the widget label.
- void `deimage` (`Fl_Image *img`)
Sets the image to use as part of the widget label.
- void `do_callback` ()
Calls the widget callback.
- void `do_callback` (`Fl_Widget *o`, long arg)
Calls the widget callback.
- void `do_callback` (`Fl_Widget *o`, void *arg=0)
Calls the widget callback.
- virtual void `draw` ()=0
Draws the widget.
- void `draw_label` (int, int, int, int, `Fl_Align`) const
Draws the label in an arbitrary bounding box with an arbitrary alignment.
- int `h` () const
Gets the widget height.
- virtual int `handle` (int event)
Handles the specified event.
- virtual void `hide` ()
Makes a widget invisible.
- `Fl_Image * image` ()
Gets the image that is used as part of the widget label.
- const `Fl_Image * image` () const
- void `image` (`Fl_Image &img`)
Sets the image to use as part of the widget label.
- void `image` (`Fl_Image *img`)
Sets the image to use as part of the widget label.
- int `inside` (const `Fl_Widget *wgt`) const
Checks if this widget is a child of wgt.
- int `is_label_copied` () const
Returns whether the current label was assigned with `copy_label()`.
- const char * `label` () const
Gets the current label text.
- void `label` (const char *text)
Sets the current label pointer.
- void `label` (`Fl_Labeltype a`, const char *b)
Shortcut to set the label text and type in one call.
- `Fl_Color labelcolor` () const
Gets the label color.
- void `labelcolor` (`Fl_Color c`)
Sets the label color.
- `Fl_Font labelfont` () const
Gets the font to use.
- void `labelfont` (`Fl_Font f`)
Sets the font to use.
- `Fl_Fontsize labelsize` () const

- Gets the font size in pixels.*

 - void `labelsize` (`FI_Fontsize` pix)
- Sets the font size in pixels.*

 - `FI_Labeltype` `labeltype` () const
- Gets the label type.*

 - void `labeltype` (`FI_Labeltype` a)
- Sets the label type.*

 - void `measure_label` (int &ww, int &hh) const
- Sets width ww and height hh accordingly with the label size.*

 - unsigned int `output` () const
- Returns if a widget is used for output only.*

 - `FI_Group` * `parent` () const
- Returns a pointer to the parent widget.*

 - void `parent` (`FI_Group` *p)
- Internal use only - "for hacks only".*

 - void `position` (int X, int Y)
- Repositions the window or widget.*

 - void `redraw` ()
- Schedules the drawing of the widget.*

 - void `redraw_label` ()
- Schedules the drawing of the label.*

 - virtual void `resize` (int x, int y, int w, int h)
- Changes the size or position of the widget.*

 - `FI_Color` `selection_color` () const
- Gets the selection color.*

 - void `selection_color` (`FI_Color` a)
- Sets the selection color.*

 - void `set_active` ()
- Marks the widget as active without sending events or changing focus.*

 - void `set_changed` ()
- Marks the value of the widget as changed.*

 - void `set_output` ()
- Sets a widget to output only.*

 - void `set_visible` ()
- Makes the widget visible.*

 - void `set_visible_focus` ()
- Enables keyboard focus navigation with this widget.*

 - virtual void `show` ()
- Makes a widget visible.*

 - void `size` (int W, int H)
- Changes the size of the widget.*

 - int `take_focus` ()
- Gives the widget the keyboard focus.*

 - unsigned int `takeevents` () const
- Returns if the widget is able to take events.*

 - int `test_shortcut` ()
- Returns true if the widget's label contains the entered '&x' shortcut.*

 - const char * `tooltip` () const
- Gets the current tooltip text.*

 - void `tooltip` (const char *text)
- Sets the current tooltip text.*

- `Fl_Widget * top_window ()` const
Returns a pointer to the top-level window for the widget.
- `Fl_Widget * top_window_offset (int &xoff, int &yoff)` const
Finds the x/y offset of the current widget relative to the top-level window.
- `uchar type ()` const
Gets the widget type.
- `void type (uchar t)`
Sets the widget type.
- `int use_accents_menu ()`
Returns non zero if MAC_USE_ACCENTS_MENU flag is set, 0 otherwise.
- `void * user_data ()` const
Gets the user data for this widget.
- `void user_data (void *v)`
Sets the user data for this widget.
- `unsigned int visible ()` const
Returns whether a widget is visible.
- `unsigned int visible_focus ()`
Checks whether this widget has a visible focus.
- `void visible_focus (int v)`
Modifies keyboard focus navigation.
- `int visible_r ()` const
Returns whether a widget and all its parents are visible.
- `int w ()` const
Gets the widget width.
- `Fl_When when ()` const
Returns the conditions under which the callback is called.
- `void when (uchar i)`
Sets the flags used to decide when a callback is called.
- `Fl_Widget * window ()` const
Returns a pointer to the nearest parent window up the widget hierarchy.
- `int x ()` const
Gets the widget position in its window.
- `int y ()` const
Gets the widget position in its window.
- `virtual ~Fl_Widget ()`
Destroys the widget.

Static Public Member Functions

- `static void default_callback (Fl_Widget *cb, void *d)`
The default callback for all widgets that don't set a callback.
- `static unsigned int label_shortcut (const char *t)`
Returns the Unicode value of the '&x' shortcut in a given text.
- `static int test_shortcut (const char *, const bool require_alt=false)`
Returns true if the given text t contains the entered '&x' shortcut.

Protected Types

- enum {
INACTIVE = 1<<0 , **INVISIBLE** = 1<<1 , **OUTPUT** = 1<<2 , **NOBORDER** = 1<<3 ,
FORCE_POSITION = 1<<4 , **NON_MODAL** = 1<<5 , **SHORTCUT_LABEL** = 1<<6 , **CHANGED** = 1<<7
 ,
OVERRIDE = 1<<8 , **VISIBLE_FOCUS** = 1<<9 , **COPIED_LABEL** = 1<<10 , **CLIP_CHILDREN** = 1<<11
 ,
MENU_WINDOW = 1<<12 , **TOOLTIP_WINDOW** = 1<<13 , **MODAL** = 1<<14 , **NO_OVERLAY** = 1<<15
 ,
GROUP_RELATIVE = 1<<16 , **COPIED_TOOLTIP** = 1<<17 , **FULLSCREEN** = 1<<18 , **MAC_USE_ACCENTS_MENU**
 = 1<<19 ,
USERFLAG3 = 1<<29 , **USERFLAG2** = 1<<30 , **USERFLAG1** = 1<<31 }
flags possible values enumeration.

Protected Member Functions

- void **clear_flag** (unsigned int c)
Clears a flag in the flags mask.
- void **draw_backdrop** () const
If FL_ALIGN_IMAGE_BACKDROP is set, the image or deimage will be drawn.
- void **draw_box** () const
Draws the widget box according its box style.
- void **draw_box** (FI_Boxtype t, FI_Color c) const
Draws a box of type t, of color c at the widget's position and size.
- void **draw_box** (FI_Boxtype t, int x, int y, int w, int h, FI_Color c) const
Draws a box of type t, of color c at the position X,Y and size W,H.
- void **draw_focus** ()
draws a focus rectangle around the widget
- void **draw_focus** (FI_Boxtype t, int x, int y, int w, int h) const
Draws a focus box for the widget at the given position and size.
- void **draw_label** () const
Draws the widget's label at the defined label position.
- void **draw_label** (int, int, int, int) const
Draws the label in an arbitrary bounding box.
- FI_Widget** (int x, int y, int w, int h, const char *label=0L)
Creates a widget at the given position and size.
- unsigned int **flags** () const
Gets the widget flags mask.
- void **h** (int v)
Internal use only.
- void **set_flag** (unsigned int c)
Sets a flag in the flags mask.
- void **w** (int v)
Internal use only.
- void **x** (int v)
Internal use only.
- void **y** (int v)
Internal use only.

Friends

- class **FI_Group**

31.151.1 Detailed Description

`Fl_Widget` is the base class for all widgets in FLTK.

You can't create one of these because the constructor is not public. However you can subclass it.

All "property" accessing methods, such as `color()`, `parent()`, or `argument()` are implemented as trivial inline functions and thus are as fast and small as accessing fields in a structure. Unless otherwise noted, the property setting methods such as `color(n)` or `label(s)` are also trivial inline functions, even if they change the widget's appearance. It is up to the user code to call `redraw()` after these.

31.151.2 Member Enumeration Documentation

31.151.2.1 anonymous enum

anonymous enum [protected]

flags possible values enumeration.

See `activate()`, `output()`, `visible()`, `changed()`, `set_visible_focus()`

Enumerator

INACTIVE	the widget can't receive focus, and is disabled but potentially visible
INVISIBLE	the widget is not drawn, but can receive a few special events
OUTPUT	for output only
NOBORDER	don't draw a decoration (Fl_Window)
FORCE_POSITION	don't let the window manager position the window (Fl_Window)
NON_MODAL	this is a hovering toolbar window (Fl_Window)
SHORTCUT_LABEL	the label contains a shortcut we need to draw
CHANGED	the widget value changed
OVERRIDE	position window on top (Fl_Window)
VISIBLE_FOCUS	accepts keyboard focus navigation if the widget can have the focus
COPIED_LABEL	the widget label is internally copied, its destruction is handled by the widget
CLIP_CHILDREN	all drawing within this widget will be clipped (Fl_Group)
MENU_WINDOW	a temporary popup window, dismissed by clicking outside (Fl_Window)
TOOLTIP_WINDOW	a temporary popup, transparent to events, and dismissed easily (Fl_Window)
MODAL	a window blocking input to all other winows (Fl_Window)
NO_OVERLAY	window not using a hardware overlay plane (Fl_Menu_Window)
GROUP_RELATIVE	Reserved, not implemented. DO NOT USE.
COPIED_TOOLTIP	the widget tooltip is internally copied, its destruction is handled by the widget
FULLSCREEN	a fullscreen window (Fl_Window)
MAC_USE_ACCENTS_MENU	On the Mac OS platform, pressing and holding a key on the keyboard opens an accented-character menu window (Fl_Input_ , Fl_Text_Editor)
USERFLAG3	reserved for 3rd party extensions
USERFLAG2	reserved for 3rd party extensions
USERFLAG1	reserved for 3rd party extensions

31.151.3 Constructor & Destructor Documentation

31.151.3.1 Fl_Widget()

```
Fl_Widget::Fl_Widget (
    int x,
    int y,
    int w,
    int h,
    const char * label = 0L ) [protected]
```

Creates a widget at the given position and size.

The [Fl_Widget](#) is a protected constructor, but all derived widgets have a matching public constructor. It takes a value for [x\(\)](#), [y\(\)](#), [w\(\)](#), [h\(\)](#), and an optional value for [label\(\)](#).

Parameters

in	<i>x,y</i>	the position of the widget relative to the enclosing window
in	<i>w,h</i>	size of the widget in pixels
in	<i>label</i>	optional text for the widget label

31.151.3.2 ~Fl_Widget()

```
Fl_Widget::~~Fl_Widget ( ) [virtual]
```

Destroys the widget.

Destroys the widget, taking care of throwing focus before if any.

Destroying single widgets is not very common. You almost always want to destroy the parent group instead, which will destroy all of the child widgets and groups in that group.

Since

FLTK 1.3, the widget's destructor removes the widget from its parent group, if it is member of a group.

Destruction removes the widget from any parent group! And groups when destroyed destroy all their children. This is convenient and fast.

31.151.4 Member Function Documentation

31.151.4.1 activate()

```
void Fl_Widget::activate ( )
```

Activates the widget.

Changing this value will send `FL_ACTIVATE` to the widget if [active_r\(\)](#) is true.

See also

[active\(\)](#), [active_r\(\)](#), [deactivate\(\)](#)

31.151.4.2 active()

```
unsigned int Fl_Widget::active ( ) const [inline]
```

Returns whether the widget is active.

Return values

0	if the widget is inactive
---	---------------------------

See also

[active_r\(\)](#), [activate\(\)](#), [deactivate\(\)](#)

31.151.4.3 active_r()

```
int Fl_Widget::active_r ( ) const
```

Returns whether the widget and all of its parents are active.

Return values

0	if this or any of the parent widgets are inactive
---	---

See also

[active\(\)](#), [activate\(\)](#), [deactivate\(\)](#)

31.151.4.4 align() [1/2]

```
Fl_Align Fl_Widget::align ( ) const [inline]
```

Gets the label alignment.

Returns

label alignment

See also

[label\(\)](#), [align\(Fl_Align\)](#), [Fl_Align](#)

31.151.4.5 align() [2/2]

```
void Fl_Widget::align (
    Fl_Align alignment ) [inline]
```

Sets the label alignment.

This controls how the label is displayed next to or inside the widget. The default value is FL_ALIGN_CENTER, which centers the label inside the widget.

Parameters

in	<i>alignment</i>	new label alignment
----	------------------	---------------------

See also

[align\(\)](#), [Fl_Align](#)

31.151.4.6 argument() [1/2]

```
long Fl_Widget::argument ( ) const [inline]
```

Gets the current user data (long) argument that is passed to the callback function.

Todo The user data value must be implemented using *intptr_t* or similar to avoid 64-bit machine incompatibilities.

31.151.4.7 argument() [2/2]

```
void Fl_Widget::argument (
    long v ) [inline]
```

Sets the current user data (long) argument that is passed to the callback function.

Todo The user data value must be implemented using *intptr_t* or similar to avoid 64-bit machine incompatibilities.

31.151.4.8 as_gl_window()

```
virtual class Fl_Gl_Window * Fl_Widget::as_gl_window ( ) [inline], [virtual]
```

Returns an [Fl_Gl_Window](#) pointer if this widget is an [Fl_Gl_Window](#).

Use this method if you have a widget (pointer) and need to know whether this widget is derived from [Fl_Gl_Window](#).

If it returns non-NULL, then the widget in question is derived from [Fl_Gl_Window](#).

Return values

NULL	if this widget is not derived from Fl_Gl_Window .
------	---

Note

This method is provided to avoid `dynamic_cast`.

See also

[Fl_Widget::as_group\(\)](#), [Fl_Widget::as_window\(\)](#)

Reimplemented in [Fl_Gl_Window](#).

31.151.4.9 as_group()

```
virtual Fl_Group * Fl_Widget::as_group ( ) [inline], [virtual]
```

Returns an [Fl_Group](#) pointer if this widget is an [Fl_Group](#).

Use this method if you have a widget (pointer) and need to know whether this widget is derived from [Fl_Group](#). If it returns non-NULL, then the widget in question is derived from [Fl_Group](#), and you can use the returned pointer to access its children or other [Fl_Group](#)-specific methods.

Example:

```
void my_callback (Fl_Widget *w, void *) {
    Fl_Group *g = w->as_group();
    if (g)
        printf ("This group has %d children\n",g->children());
    else
        printf ("This widget is not a group!\n");
}
```

Return values

NULL	if this widget is not derived from Fl_Group .
------	---

Note

This method is provided to avoid `dynamic_cast`.

See also

[Fl_Widget::as_window\(\)](#), [Fl_Widget::as_gl_window\(\)](#)

Reimplemented in [Fl_Group](#).

31.151.4.10 as_window()

```
virtual Fl\_Window * Fl_Widget::as_window ( ) [inline], [virtual]
```

Returns an [Fl_Window](#) pointer if this widget is an [Fl_Window](#).

Use this method if you have a widget (pointer) and need to know whether this widget is derived from [Fl_Window](#). If it returns non-NULL, then the widget in question is derived from [Fl_Window](#), and you can use the returned pointer to access its children or other [Fl_Window](#)-specific methods.

Return values

<i>NULL</i>	if this widget is not derived from Fl_Window .
-------------	--

Note

This method is provided to avoid `dynamic_cast`.

See also

[Fl_Widget::as_group\(\)](#), [Fl_Widget::as_gl_window\(\)](#)

Reimplemented in [Fl_Window](#).

31.151.4.11 box() [1/2]

```
Fl\_Boxtype Fl_Widget::box ( ) const [inline]
```

Gets the box type of the widget.

Returns

the current box type

See also

[box\(Fl_Boxtype\)](#), [Fl_Boxtype](#)

31.151.4.12 box() [2/2]

```
void Fl_Widget::box (
    Fl\_Boxtype new_box ) [inline]
```

Sets the box type for the widget.

This identifies a routine that draws the background of the widget. See [Fl_Boxtype](#) for the available types. The default depends on the widget, but is usually `FL_NO_BOX` or `FL_UP_BOX`.

Parameters

in	<i>new_box</i>	the new box type
----	----------------	------------------

See also

[box\(\)](#), [Fl_Boxtype](#)

31.151.4.13 callback() [1/5]

```
Fl\_Callback\_p Fl_Widget::callback ( ) const [inline]
```

Gets the current callback function for the widget.

Each widget has a single callback.

Returns

current callback

31.151.4.14 callback() [2/5]

```
void Fl_Widget::callback (
    Fl_Callback * cb ) [inline]
```

Sets the current callback function for the widget.
Each widget has a single callback.

Parameters

in	<i>cb</i>	new callback
----	-----------	--------------

31.151.4.15 callback() [3/5]

```
void Fl_Widget::callback (
    Fl_Callback * cb,
    void * p ) [inline]
```

Sets the current callback function for the widget.
Each widget has a single callback.

Parameters

in	<i>cb</i>	new callback
in	<i>p</i>	user data

31.151.4.16 callback() [4/5]

```
void Fl_Widget::callback (
    Fl_Callback0 * cb ) [inline]
```

Sets the current callback function for the widget.
Each widget has a single callback.

Parameters

in	<i>cb</i>	new callback
----	-----------	--------------

31.151.4.17 callback() [5/5]

```
void Fl_Widget::callback (
    Fl_Callback1 * cb,
    long p = 0 ) [inline]
```

Sets the current callback function for the widget.
Each widget has a single callback.

Parameters

in	<i>cb</i>	new callback
in	<i>p</i>	user data

31.151.4.18 changed()

```
unsigned int Fl_Widget::changed ( ) const [inline]
```

Checks if the widget value changed since the last callback.

"Changed" is a flag that is turned on when the user changes the value stored in the widget. This is only used by subclasses of [Fl_Widget](#) that store values, but is in the base class so it is easier to scan all the widgets in a panel and [do_callback\(\)](#) on the changed ones in response to an "OK" button.

Most widgets turn this flag off when they do the callback, and when the program sets the stored value.

Return values

<code>0</code>	if the value did not change
----------------	-----------------------------

See also

[set_changed\(\)](#), [clear_changed\(\)](#)

31.151.4.19 clear_active()

```
void Fl_Widget::clear_active ( ) [inline]
```

Marks the widget as inactive without sending events or changing focus.

This is mainly for specialized use, for normal cases you want [deactivate\(\)](#).

See also

[deactivate\(\)](#)

31.151.4.20 clear_changed()

```
void Fl_Widget::clear_changed ( ) [inline]
```

Marks the value of the widget as unchanged.

See also

[changed\(\)](#), [set_changed\(\)](#)

31.151.4.21 clear_damage()

```
void Fl_Widget::clear_damage (
    uchar c = 0 ) [inline]
```

Clears or sets the damage flags.

Damage flags are cleared when parts of the widget drawing is repaired.

The optional argument `c` specifies the bits that **are set** after the call (default: 0) and **not** the bits that are cleared!

Note

Therefore it is possible to set damage bits with this method, but this should be avoided. Use [damage\(uchar\)](#) instead.

Parameters

in	<code>c</code>	new bitmask of damage flags (default: 0)
----	----------------	--

See also

[damage\(uchar\)](#), [damage\(\)](#)

31.151.4.22 clear_output()

```
void Fl_Widget::clear_output ( ) [inline]
```

Sets a widget to accept input.

See also

[set_output\(\)](#), [output\(\)](#)

31.151.4.23 clear_visible()

```
void Fl_Widget::clear_visible ( ) [inline]
```

Hides the widget.

You must still redraw the parent to see a change in the window. Normally you want to use the [hide\(\)](#) method instead.

31.151.4.24 clear_visible_focus()

```
void Fl_Widget::clear_visible_focus ( ) [inline]
```

Disables keyboard focus navigation with this widget.

Normally, all widgets participate in keyboard focus navigation.

See also

[set_visible_focus\(\)](#), [visible_focus\(\)](#), [visible_focus\(int\)](#)

31.151.4.25 color() [1/3]

```
Fl_Color Fl_Widget::color ( ) const [inline]
```

Gets the background color of the widget.

Returns

current background color

See also

[color\(Fl_Color\)](#), [color\(Fl_Color, Fl_Color\)](#)

31.151.4.26 color() [2/3]

```
void Fl_Widget::color (
    Fl_Color bg ) [inline]
```

Sets the background color of the widget.

The color is passed to the box routine. The color is either an index into an internal table of RGB colors or an RGB color value generated using [fl_rgb_color\(\)](#).

The default for most widgets is FL_BACKGROUND_COLOR. Use [Fl::set_color\(\)](#) to redefine colors in the color map.

Parameters

in	<i>bg</i>	background color
----	-----------	------------------

See also

[color\(\)](#), [color\(Fl_Color, Fl_Color\)](#), [selection_color\(Fl_Color\)](#)

31.151.4.27 color() [3/3]

```
void Fl_Widget::color (
    Fl_Color bg,
    Fl_Color sel ) [inline]
```

Sets the background and selection color of the widget.
The two color form sets both the background and selection colors.

Parameters

in	<i>bg</i>	background color
in	<i>sel</i>	selection color

See also

[color\(unsigned\)](#), [selection_color\(unsigned\)](#)

31.151.4.28 color2() [1/2]

```
Fl_Color Fl_Widget::color2 ( ) const [inline]
```

For back compatibility only.

Deprecated Use [selection_color\(\)](#) instead.

31.151.4.29 color2() [2/2]

```
void Fl_Widget::color2 (
    unsigned a ) [inline]
```

For back compatibility only.

Deprecated Use [selection_color\(unsigned\)](#) instead.

31.151.4.30 contains()

```
int Fl_Widget::contains (
    const Fl_Widget * w ) const
```

Checks if *w* is a child of this widget.

Parameters

in	<i>w</i>	potential child widget
----	----------	------------------------

Returns

Returns 1 if *w* is a child of this widget, or is equal to this widget. Returns 0 if *w* is NULL.

31.151.4.31 copy_label()

```
void Fl_Widget::copy_label (
    const char * new_label )
```

Sets the current label.

Unlike [label\(\)](#), this method allocates a copy of the label string instead of using the original string pointer.

The internal copy will automatically be freed whenever you assign a new label or when the widget is destroyed.

Parameters

in	<i>new_label</i>	the new label text
----	------------------	--------------------

See also

[label\(\)](#)

31.151.4.32 copy_tooltip()

```
void Fl_Widget::copy_tooltip (
    const char * text )
```

Sets the current tooltip text.

Unlike [tooltip\(\)](#), this method allocates a copy of the tooltip string instead of using the original string pointer.

The internal copy will automatically be freed whenever you assign a new tooltip or when the widget is destroyed.

If no tooltip is set, the tooltip of the parent is inherited. Setting a tooltip for a group and setting no tooltip for a child will show the group's tooltip instead. To avoid this behavior, you can set the child's tooltip to an empty string ("").

Parameters

in	<i>text</i>	New tooltip text (an internal copy is made and managed)
----	-------------	---

See also

[tooltip\(const char*\)](#), [tooltip\(\)](#)

31.151.4.33 damage() [1/3]

```
uchar Fl_Widget::damage ( ) const [inline]
```

Returns non-zero if [draw\(\)](#) needs to be called.

The damage value is actually a bit field that the widget subclass can use to figure out what parts to draw.

Returns

a bitmap of flags describing the kind of damage to the widget

See also

[damage\(uchar\)](#), [clear_damage\(uchar\)](#)

31.151.4.34 damage() [2/3]

```
void Fl_Widget::damage (
    uchar c )
```

Sets the damage bits for the widget.
Setting damage bits will schedule the widget for the next redraw.

Parameters

in	c	bitmask of flags to set
----	---	-------------------------

See also

[damage\(\)](#), [clear_damage\(uchar\)](#)

31.151.4.35 damage() [3/3]

```
void Fl_Widget::damage (
    uchar c,
    int x,
    int y,
    int w,
    int h )
```

Sets the damage bits for an area inside the widget.
Setting damage bits will schedule the widget for the next redraw.

Parameters

in	c	bitmask of flags to set
in	x,y,w,h	size of damaged area

See also

[damage\(\)](#), [clear_damage\(uchar\)](#)

31.151.4.36 deactivate()

```
void Fl_Widget::deactivate ( )
```

Deactivates the widget.

Inactive widgets will be drawn "grayed out", e.g. with less contrast than the active widget. Inactive widgets will not receive any keyboard or mouse button events. Other events (including `FL_ENTER`, `FL_MOVE`, `FL_LEAVE`, `FL_SHORTCUT`, and others) will still be sent. A widget is only active if [active\(\)](#) is true on it *and all of its parents*.

Changing this value will send `FL_DEACTIVATE` to the widget if [active_r\(\)](#) is true.

Currently you cannot deactivate [Fl_Window](#) widgets.

See also

[activate\(\)](#), [active\(\)](#), [active_r\(\)](#)

31.151.4.37 default_callback()

```
void Fl_Widget::default_callback (
    Fl_Widget * cb,
    void * d ) [static]
```

The default callback for all widgets that don't set a callback.

This callback function puts a pointer to the widget on the queue returned by [Fl::readqueue\(\)](#).

Relying on the default callback and reading the callback queue with [Fl::readqueue\(\)](#) is not recommended. If you need a callback, you should set one with [Fl_Widget::callback\(Fl_Callback *cb, void *data\)](#) or one of its variants.

Parameters

in	<i>cb</i>	the widget given to the callback
in	<i>d</i>	user data associated with that callback

See also

[callback\(\)](#), [do_callback\(\)](#), [Fl::readqueue\(\)](#)

31.151.4.38 deimage() [1/3]

```
Fl_Image * Fl_Widget::deimage ( ) [inline]
```

Gets the image that is used as part of the widget label.

This image is used when drawing the widget in the inactive state.

Returns

the current image for the deactivated widget

31.151.4.39 deimage() [2/3]

```
void Fl_Widget::deimage (
    Fl_Image & img ) [inline]
```

Sets the image to use as part of the widget label.

This image is used when drawing the widget in the inactive state.

Parameters

in	<i>img</i>	the new image for the deactivated widget
----	------------	--

31.151.4.40 deimage() [3/3]

```
void Fl_Widget::deimage (
    Fl_Image * img ) [inline]
```

Sets the image to use as part of the widget label.

This image is used when drawing the widget in the inactive state.

Parameters

in	<i>img</i>	the new image for the deactivated widget
----	------------	--

31.151.4.41 do_callback() [1/3]

```
void Fl_Widget::do_callback ( ) [inline]
```

Calls the widget callback.

Causes a widget to invoke its callback function with default arguments.

See also

[callback\(\)](#)

31.151.4.42 do_callback() [2/3]

```
void Fl_Widget::do_callback (
    Fl_Widget * o,
    long arg ) [inline]
```

Calls the widget callback.

Causes a widget to invoke its callback function with arbitrary arguments.

Parameters

in	<i>o</i>	call the callback with <i>o</i> as the widget argument
in	<i>arg</i>	call the callback with <i>arg</i> as the user data argument

See also

[callback\(\)](#)

31.151.4.43 do_callback() [3/3]

```
void Fl_Widget::do_callback (
    Fl_Widget * o,
    void * arg = 0 )
```

Calls the widget callback.

Causes a widget to invoke its callback function with arbitrary arguments.

Parameters

in	<i>o</i>	call the callback with <i>o</i> as the widget argument
in	<i>arg</i>	use <i>arg</i> as the user data argument

See also

[callback\(\)](#)

31.151.4.44 draw()

```
virtual void Fl_Widget::draw ( ) [pure virtual]
```

Draws the widget.

Never call this function directly. FLTK will schedule redrawing whenever needed. If your widget must be redrawn as soon as possible, call [redraw\(\)](#) instead.

Override this function to draw your own widgets.

If you ever need to call another widget's draw method *from within your own [draw\(\)](#) method*, e.g. for an embedded scrollbar, you can do it (because [draw\(\)](#) is virtual) like this:

```
Fl_Widget *s = &scroll; // scroll is an embedded Fl_Scrollbar
```

```
s->draw(); // calls Fl_Scrollbar::draw()
```

Implemented in [Fl_Adjuster](#), [Fl_Box](#), [Fl_Browser_](#), [Fl_Button](#), [Fl_Cairo_Window](#), [Fl_Chart](#), [Fl_Choice](#), [Fl_Clock_Output](#), [Fl_Counter](#), [Fl_Dial](#), [Fl_File_Input](#), [Fl_FormsBitmap](#), [Fl_FormsPixmap](#), [Fl_Free](#), [Fl_Gl_Window](#), [Fl_Group](#), [Fl_Help_View](#), [Fl_Input](#), [Fl_Light_Button](#), [Fl_Menu_Bar](#), [Fl_Menu_Button](#), [Fl_Pack](#), [Fl_Positioner](#), [Fl_Progress](#), [Fl_Return_Button](#), [Fl_Roller](#), [Fl_Scroll](#), [Fl_Scrollbar](#), [Fl_Slider](#), [Fl_Sys_Menu_Bar](#), [Fl_Tabs](#), [Fl_Text_Display](#), [Fl_Timer](#), [Fl_Tree](#), [Fl_Value_Input](#), [Fl_Value_Output](#), [Fl_Value_Slider](#), [Fl_Window](#), [Fl_FormsText](#), [Fl_Glut_Window](#), and [Fl_Table](#).

31.151.4.45 `draw_label()` [1/3]

```
void Fl_Widget::draw_label ( ) const [protected]
```

Draws the widget's label at the defined label position.

This is the normal call for a widget's [draw\(\)](#) method.

31.151.4.46 `draw_label()` [2/3]

```
void Fl_Widget::draw_label (
    int X,
    int Y,
    int W,
    int H ) const [protected]
```

Draws the label in an arbitrary bounding box.

[draw\(\)](#) can use this instead of [draw_label\(void\)](#) to change the bounding box

31.151.4.47 `draw_label()` [3/3]

```
void Fl_Widget::draw_label (
    int X,
    int Y,
    int W,
    int H,
    Fl_Align a ) const
```

Draws the label in an arbitrary bounding box with an arbitrary alignment.

Anybody can call this to force the label to draw anywhere.

31.151.4.48 `h()` [1/2]

```
int Fl_Widget::h ( ) const [inline]
```

Gets the widget height.

Returns

the height of the widget in pixels.

31.151.4.49 `h()` [2/2]

```
void Fl_Widget::h (
    int v ) [inline], [protected]
```

Internal use only.

Use [position\(int,int\)](#), [size\(int,int\)](#) or [resize\(int,int,int,int\)](#) instead.

31.151.4.50 `handle()`

```
int Fl_Widget::handle (
    int event ) [virtual]
```

Handles the specified event.

You normally don't call this method directly, but instead let FLTK do it when the user interacts with the widget.

When implemented in a widget, this function must return 0 if the widget does not use the event or 1 otherwise. Most of the time, you want to call the inherited [handle\(\)](#) method in your overridden method so that you don't short-circuit events that you don't handle. In this last case you should return the callee retval.

Parameters

in	<i>event</i>	the kind of event received
----	--------------	----------------------------

Return values

0	if the event was not used or understood
1	if the event was used and can be deleted

See also

[Fl_Event](#)

Reimplemented in [Fl_Free](#), [Fl_Table](#), [Fl_Text_Display](#), [Fl_Text_Editor](#), [Fl_Tree](#), [Fl_Browser_](#), [Fl_File_Input](#), [Fl_Spinner](#), [Fl_Table_Row](#), [Fl_Tile](#), [Fl_Adjuster](#), [Fl_Box](#), [Fl_Button](#), [Fl_Check_Browser](#), [Fl_Choice](#), [Fl_Clock](#), [Fl_Counter](#), [Fl_Dial](#), [Fl_Gl_Window](#), [Fl_Group](#), [Fl_Help_View](#), [Fl_Input](#), [Fl_Light_Button](#), [Fl_Menu_Bar](#), [Fl_Menu_Button](#), [Fl_Positioner](#), [Fl_Repeat_Button](#), [Fl_Return_Button](#), [Fl_Roller](#), [Fl_Scroll](#), [Fl_Scrollbar](#), [Fl_Secret_Input](#), [Fl_Slider](#), [Fl_Tabs](#), [Fl_Timer](#), [Fl_Value_Input](#), [Fl_Value_Output](#), [Fl_Value_Slider](#), [Fl_Window](#), and [Fl_Glut_Window](#).

31.151.4.51 hide()

```
void Fl_Widget::hide ( ) [virtual]
Makes a widget invisible.
```

See also

[show\(\)](#), [visible\(\)](#), [visible_r\(\)](#)

Reimplemented in [Fl_Browser](#), [Fl_Double_Window](#), [Fl_Gl_Window](#), [Fl_Menu_Window](#), [Fl_Overlay_Window](#), and [Fl_Window](#).

31.151.4.52 image() [1/3]

```
Fl_Image * Fl_Widget::image ( ) [inline]
Gets the image that is used as part of the widget label.
This image is used when drawing the widget in the active state.
```

Returns

the current image

31.151.4.53 image() [2/3]

```
void Fl_Widget::image (
    Fl_Image & img ) [inline]
Sets the image to use as part of the widget label.
This image is used when drawing the widget in the active state.
```

Parameters

in	<i>img</i>	the new image for the label
----	------------	-----------------------------

31.151.4.54 image() [3/3]

```
void Fl_Widget::image (
    Fl_Image * img ) [inline]
```

Sets the image to use as part of the widget label.
This image is used when drawing the widget in the active state.

Parameters

in	<i>img</i>	the new image for the label
----	------------	-----------------------------

31.151.4.55 inside()

```
int Fl_Widget::inside (
    const Fl_Widget * wgt ) const [inline]
```

Checks if this widget is a child of *wgt*.
Returns 1 if this widget is a child of *wgt*, or is equal to *wgt*. Returns 0 if *wgt* is NULL.

Parameters

in	<i>wgt</i>	the possible parent widget.
----	------------	-----------------------------

See also

[contains\(\)](#)

31.151.4.56 is_label_copied()

```
int Fl_Widget::is_label_copied ( ) const [inline]
```

Returns whether the current label was assigned with [copy_label\(\)](#).
This can be useful for temporarily overwriting the widget's label and restoring it later.

Return values

0	current label was assigned with label() .
1	current label was assigned with copy_label() .

31.151.4.57 label() [1/3]

```
const char * Fl_Widget::label ( ) const [inline]
```

Gets the current label text.

Returns

a pointer to the current label text

See also

[label\(const char *\)](#), [copy_label\(const char *\)](#)

31.151.4.58 label() [2/3]

```
void Fl_Widget::label (
    const char * text )
```

Sets the current label pointer.

The label is shown somewhere on or next to the widget. The passed pointer is stored unchanged in the widget (the string is *not* copied), so if you need to set the label to a formatted value, make sure the buffer is static, global, or allocated. The [copy_label\(\)](#) method can be used to make a copy of the label string automatically.

Parameters

in	<i>text</i>	pointer to new label text
----	-------------	---------------------------

See also

[copy_label\(\)](#)

31.151.4.59 label() [3/3]

```
void Fl_Widget::label (
    Fl_Labeltype a,
    const char * b ) [inline]
```

Shortcut to set the label text and type in one call.

See also

[label\(const char *\)](#), [labeltype\(Fl_Labeltype\)](#)

31.151.4.60 label_shortcut()

```
unsigned int Fl_Widget::label_shortcut (
    const char * t ) [static]
```

Returns the Unicode value of the '&x' shortcut in a given text.

The given text *t* (usually a widget's label or a menu text) is searched for a '&x' shortcut label, and if found, the Unicode value (code point) of the '&x' shortcut is returned.

Parameters

<i>t</i>	text or label to search for '&x' shortcut.
----------	--

Returns

Unicode (UCS-4) value of shortcut in *t* or 0.

Note

Internal use only.

31.151.4.61 labelcolor() [1/2]

```
Fl_Color Fl_Widget::labelcolor ( ) const [inline]
```

Gets the label color.

The default color is FL_FOREGROUND_COLOR.

Returns

the current label color

31.151.4.62 labelcolor() [2/2]

```
void Fl_Widget::labelcolor (
    Fl_Color c ) [inline]
```

Sets the label color.

The default color is FL_FOREGROUND_COLOR.

Parameters

in	c	the new label color
----	---	---------------------

31.151.4.63 labelfont() [1/2]

```
Fl_Font Fl_Widget::labelfont ( ) const [inline]
```

Gets the font to use.

Fonts are identified by indexes into a table. The default value uses a Helvetica typeface (Arial for Microsoft® Windows®). The function [Fl::set_font\(\)](#) can define new typefaces.

Returns

current font used by the label

See also

[Fl_Font](#)

31.151.4.64 labelfont() [2/2]

```
void Fl_Widget::labelfont (
    Fl_Font f ) [inline]
```

Sets the font to use.

Fonts are identified by indexes into a table. The default value uses a Helvetica typeface (Arial for Microsoft® Windows®). The function [Fl::set_font\(\)](#) can define new typefaces.

Parameters

in	f	the new font for the label
----	---	----------------------------

See also

[Fl_Font](#)

31.151.4.65 labelsize() [1/2]

```
Fl_Fontsize Fl_Widget::labelsize ( ) const [inline]
```

Gets the font size in pixels.

The default size is 14 pixels.

Returns

the current font size

31.151.4.66 labelsize() [2/2]

```
void Fl_Widget::labelsize (
    Fl_Fontsize pix ) [inline]
```

Sets the font size in pixels.

Parameters

in	<i>pix</i>	the new font size
----	------------	-------------------

See also

[Fl_Fontsize labelsize\(\)](#)

31.151.4.67 labeltype() [1/2]

```
Fl_Labeltype Fl_Widget::labeltype ( ) const [inline]
```

Gets the label type.

Returns

the current label type.

See also

[Fl_Labeltype](#)

31.151.4.68 labeltype() [2/2]

```
void Fl_Widget::labeltype (
    Fl_Labeltype a ) [inline]
```

Sets the label type.

The label type identifies the function that draws the label of the widget. This is generally used for special effects such as embossing or for using the [label\(\)](#) pointer as another form of data such as an icon. The value `FL_NORMAL_LABEL` prints the label as plain text.

Parameters

in	<i>a</i>	new label type
----	----------	----------------

See also

[Fl_Labeltype](#)

31.151.4.69 measure_label()

```
void Fl_Widget::measure_label (
    int & ww,
    int & hh ) const [inline]
```

Sets width `ww` and height `hh` accordingly with the label size. Labels with images will return `w()` and `h()` of the image. This calls `fl_measure()` internally. For more information about the arguments `ww` and `hh` and word wrapping

See also

[fl_measure\(const char*, int&, int&, int\)](#)

31.151.4.70 output()

```
unsigned int Fl_Widget::output ( ) const [inline]
```

Returns if a widget is used for output only.

`output()` means the same as `!active()` except it does not change how the widget is drawn. The widget will not receive any events. This is useful for making scrollbars or buttons that work as displays rather than input devices.

Return values

0	if the widget is used for input and output
---	--

See also

[set_output\(\)](#), [clear_output\(\)](#)

31.151.4.71 parent() [1/2]

```
Fl_Group * Fl_Widget::parent ( ) const [inline]
```

Returns a pointer to the parent widget.

Usually this is a [Fl_Group](#) or [Fl_Window](#).

Return values

NULL	if the widget has no parent
------	-----------------------------

See also

[Fl_Group::add\(Fl_Widget*\)](#)

31.151.4.72 parent() [2/2]

```
void Fl_Widget::parent (
    Fl_Group * p ) [inline]
```

Internal use only - "for hacks only".

It is **STRONGLY recommended** not to use this method, because it short-circuits [Fl_Group](#)'s normal widget adding and removing methods, if the widget is already a child widget of another [Fl_Group](#).

Use [Fl_Group::add\(Fl_Widget*\)](#) and/or [Fl_Group::remove\(Fl_Widget*\)](#) instead.

31.151.4.73 position()

```
void Fl_Widget::position (
    int X,
    int Y ) [inline]
```

Repositions the window or widget.

`position(X, Y)` is a shortcut for `resize(X, Y, w(), h())`.

Parameters

in	<i>X,Y</i>	new position relative to the parent window
----	------------	--

See also

[resize\(int,int,int,int\)](#), [size\(int,int\)](#)

31.151.4.74 redraw()

```
void Fl_Widget::redraw ( )
```

Schedules the drawing of the widget.

Marks the widget as needing its [draw\(\)](#) routine called.

31.151.4.75 redraw_label()

```
void Fl_Widget::redraw_label ( )
```

Schedules the drawing of the label.

Marks the widget or the parent as needing a redraw for the label area of a widget.

31.151.4.76 resize()

```
void Fl_Widget::resize (
    int x,
    int y,
    int w,
    int h ) [virtual]
```

Changes the size or position of the widget.

This is a virtual function so that the widget may implement its own handling of resizing. The default version does *not* call the [redraw\(\)](#) method, but instead relies on the parent widget to do so because the parent may know a faster way to update the display, such as scrolling from the old position.

Some window managers under X11 call [resize\(\)](#) a lot more often than needed. Please verify that the position or size of a widget did actually change before doing any extensive calculations.

[position\(X, Y\)](#) is a shortcut for [resize\(X, Y, w\(\), h\(\)\)](#), and [size\(W, H\)](#) is a shortcut for [resize\(x\(\), y\(\), W, H\)](#).

Parameters

in	<i>x,y</i>	new position relative to the parent window
in	<i>w,h</i>	new size

See also

[position\(int,int\)](#), [size\(int,int\)](#)

Reimplemented in [Fl_Browser_](#), [Fl_Input_Choice](#), [Fl_Scroll](#), [Fl_Spinner](#), [Fl_Table](#), [Fl_Text_Display](#), [Fl_Tile](#), [Fl_Window](#), [Fl_Double_Window](#), [Fl_Gl_Window](#), [Fl_Group](#), [Fl_Help_View](#), [Fl_Input_](#), [Fl_Overlay_Window](#), [Fl_Tree](#), and [Fl_Value_Input](#).

31.151.4.77 selection_color() [1/2]

```
Fl_Color Fl_Widget::selection_color ( ) const [inline]
```

Gets the selection color.

Returns

the current selection color

See also

[selection_color\(Fl_Color\)](#), [color\(Fl_Color, Fl_Color\)](#)

31.151.4.78 selection_color() [2/2]

```
void Fl_Widget::selection_color (
    Fl_Color a ) [inline]
```

Sets the selection color.

The selection color is defined for Forms compatibility and is usually used to color the widget when it is selected, although some widgets use this color for other purposes. You can set both colors at once with [color\(Fl_Color bg, Fl_Color sel\)](#).

Parameters

in	a	the new selection color
----	---	-------------------------

See also

[selection_color\(\)](#), [color\(Fl_Color, Fl_Color\)](#)

31.151.4.79 set_active()

```
void Fl_Widget::set_active ( ) [inline]
```

Marks the widget as active without sending events or changing focus.

This is mainly for specialized use, for normal cases you want [activate\(\)](#).

See also

[activate\(\)](#)

31.151.4.80 set_changed()

```
void Fl_Widget::set_changed ( ) [inline]
```

Marks the value of the widget as changed.

See also

[changed\(\)](#), [clear_changed\(\)](#)

31.151.4.81 set_output()

```
void Fl_Widget::set_output ( ) [inline]
```

Sets a widget to output only.

See also

[output\(\)](#), [clear_output\(\)](#)

31.151.4.82 set_visible()

```
void Fl_Widget::set_visible ( ) [inline]
```

Makes the widget visible.

You must still redraw the parent widget to see a change in the window. Normally you want to use the [show\(\)](#) method instead.

31.151.4.83 set_visible_focus()

```
void Fl_Widget::set_visible_focus ( ) [inline]
```

Enables keyboard focus navigation with this widget.

Note, however, that this will not necessarily mean that the widget will accept focus, but for widgets that can accept focus, this method enables it if it has been disabled.

See also

[visible_focus\(\)](#), [clear_visible_focus\(\)](#), [visible_focus\(int\)](#)

31.151.4.84 show()

```
void Fl_Widget::show ( ) [virtual]
```

Makes a widget visible.

An invisible widget never gets redrawn and does not get keyboard or mouse events, but can receive a few other events like FL_SHOW.

The [visible\(\)](#) method returns true if the widget is set to be visible. The [visible_r\(\)](#) method returns true if the widget and all of its parents are visible. A widget is only visible if [visible\(\)](#) is true on it *and all of its parents*.

Changing it will send FL_SHOW or FL_HIDE events to the widget. *Do not change it if the parent is not visible, as this will send false FL_SHOW or FL_HIDE events to the widget.* [redraw\(\)](#) is called if necessary on this or the parent.

See also

[hide\(\)](#), [visible\(\)](#), [visible_r\(\)](#)

Reimplemented in [Fl_Browser](#), [Fl_Double_Window](#), [Fl_Gl_Window](#), [Fl_Menu_Window](#), [Fl_Overlay_Window](#), [Fl_Single_Window](#), and [Fl_Window](#).

31.151.4.85 size()

```
void Fl_Widget::size (
    int W,
    int H ) [inline]
```

Changes the size of the widget.

`size(W, H)` is a shortcut for `resize(x(), y(), W, H)`.

Parameters

in	<i>W,H</i>	new size
----	------------	----------

See also

[position\(int,int\)](#), [resize\(int,int,int,int\)](#)

31.151.4.86 take_focus()

```
int Fl_Widget::take_focus ( )
```

Gives the widget the keyboard focus.

Tries to make this widget be the [Fl::focus\(\)](#) widget, by first sending it an FL_FOCUS event, and if it returns non-zero, setting [Fl::focus\(\)](#) to this widget. You should use this method to assign the focus to a widget.

Returns

true if the widget accepted the focus.

31.151.4.87 takeevents()

```
unsigned int Fl_Widget::takeevents ( ) const [inline]
```

Returns if the widget is able to take events.

This is the same as ([active\(\)](#) && ![output\(\)](#) && [visible\(\)](#)) but is faster.

Return values

0	if the widget takes no events
---	-------------------------------

31.151.4.88 test_shortcut() [1/2]

```
int Fl_Widget::test_shortcut ( )
```

Returns true if the widget's label contains the entered '&x' shortcut.

This method must only be called in [handle\(\)](#) methods or callbacks after a keypress event (usually FL_KEYDOWN or FL_SHORTCUT). The widget's label is searched for a '&x' shortcut, and if found, this is compared with the entered key value.

[Fl::event_text\(\)](#) is used to get the entered key value.

Returns

true, if the entered text matches the widget's '&x' shortcut, false (0) otherwise.

Note

Internal use only.

31.151.4.89 test_shortcut() [2/2]

```
int Fl_Widget::test_shortcut (
    const char * t,
    const bool require_alt = false ) [static]
```

Returns true if the given text *t* contains the entered '&x' shortcut.

This method must only be called in [handle\(\)](#) methods or callbacks after a keypress event (usually FL_KEYDOWN or FL_SHORTCUT). The given text *t* (usually a widget's label or menu text) is searched for a '&x' shortcut, and if found, this is compared with the entered key value.

[Fl::event_text\(\)](#) is used to get the entered key value. [Fl::event_state\(\)](#) is used to get the Alt modifier, if *require_alt* is true.

Parameters

<i>t</i>	text or label to search for '&x' shortcut.
<i>require_alt</i>	if true: match only if Alt key is pressed.

Returns

true, if the entered text matches the '&x' shortcut in *t* false (0) otherwise.

Note

Internal use only.

31.151.4.90 tooltip() [1/2]

```
const char * Fl_Widget::tooltip ( ) const [inline]
```

Gets the current tooltip text.

Returns

a pointer to the tooltip text or NULL

See also

[tooltip\(const char*\)](#), [copy_tooltip\(const char*\)](#)

31.151.4.91 tooltip() [2/2]

```
void Fl_Widget::tooltip (
    const char * text )
```

Sets the current tooltip text.

Sets a string of text to display in a popup tooltip window when the user hovers the mouse over the widget. The string is *not* copied, so make sure any formatted string is stored in a static, global, or allocated buffer. If you want a copy made and managed for you, use the [copy_tooltip\(\)](#) method, which will manage the tooltip string automatically. If no tooltip is set, the tooltip of the parent is inherited. Setting a tooltip for a group and setting no tooltip for a child will show the group's tooltip instead. To avoid this behavior, you can set the child's tooltip to an empty string ("").

Parameters

in	<i>text</i>	New tooltip text (no copy is made)
----	-------------	------------------------------------

See also

[copy_tooltip\(const char*\)](#), [tooltip\(\)](#)

31.151.4.92 top_window()

```
Fl_Window * Fl_Widget::top_window ( ) const
```

Returns a pointer to the top-level window for the widget.

In other words, the 'window manager window' that contains this widget. This method differs from [window\(\)](#) in that it won't return sub-windows (if there are any).

Returns

the top-level window, or NULL if no top-level window is associated with this widget.

See also

[window\(\)](#)

31.151.4.93 top_window_offset()

```
Fl_Window * Fl_Widget::top_window_offset (
    int & xoff,
    int & yoff ) const
```

Finds the x/y offset of the current widget relative to the top-level window.

Parameters

out	<i>xoff,yoff</i>	Returns the x/y offset
-----	------------------	------------------------

Returns

the top-level window (or NULL for a widget that's not in any window)

31.151.4.94 type() [1/2]

```
uchar Fl_Widget::type ( ) const [inline]
```

Gets the widget type.

Returns the widget type value, which is used for Forms compatibility and to simulate RTTI.

Todo Explain "simulate RTTI" (currently only used to decide if a widget is a window, i.e. `type() >= FL_WINDOW` ?).
Is `type()` really used in a way that ensures "Forms compatibility" ?

31.151.4.95 type() [2/2]

```
void Fl_Widget::type (
    uchar t ) [inline]
```

Sets the widget type.

This is used for Forms compatibility.

31.151.4.96 user_data() [1/2]

```
void * Fl_Widget::user_data ( ) const [inline]
```

Gets the user data for this widget.

Gets the current user data (void *) argument that is passed to the callback function.

Returns

user data as a pointer

31.151.4.97 user_data() [2/2]

```
void Fl_Widget::user_data (
    void * v ) [inline]
```

Sets the user data for this widget.

Sets the new user data (void *) argument that is passed to the callback function.

Parameters

in	v	new user data
----	---	---------------

31.151.4.98 visible()

```
unsigned int Fl_Widget::visible ( ) const [inline]
```

Returns whether a widget is visible.

Return values

0	if the widget is not drawn and hence invisible.
---	---

See also

[show\(\)](#), [hide\(\)](#), [visible_r\(\)](#)

31.151.4.99 visible_focus() [1/2]

```
unsigned int Fl_Widget::visible_focus ( ) [inline]
```

Checks whether this widget has a visible focus.

Return values

0	if this widget has no visible focus.
---	--------------------------------------

See also

[visible_focus\(int\)](#), [set_visible_focus\(\)](#), [clear_visible_focus\(\)](#)

31.151.4.100 visible_focus() [2/2]

```
void Fl_Widget::visible_focus (
    int v ) [inline]
```

Modifies keyboard focus navigation.

Parameters

in	v	set or clear visible focus
----	---	----------------------------

See also

[set_visible_focus\(\)](#), [clear_visible_focus\(\)](#), [visible_focus\(\)](#)

31.151.4.101 visible_r()

```
int Fl_Widget::visible_r ( ) const
```

Returns whether a widget and all its parents are visible.

Return values

0	if the widget or any of its parents are invisible.
---	--

See also

[show\(\)](#), [hide\(\)](#), [visible\(\)](#)

31.151.4.102 w() [1/2]

```
int Fl_Widget::w ( ) const [inline]
```

Gets the widget width.

Returns

the width of the widget in pixels.

31.151.4.103 w() [2/2]

```
void Fl_Widget::w (
    int v ) [inline], [protected]
```

Internal use only.

Use [position\(int,int\)](#), [size\(int,int\)](#) or [resize\(int,int,int,int\)](#) instead.

31.151.4.104 when() [1/2]

```
Fl_When Fl_Widget::when ( ) const [inline]
```

Returns the conditions under which the callback is called.

You can set the flags with [when\(uchar\)](#), the default value is FL_WHEN_RELEASE.

Returns

set of flags

See also

[when\(uchar\)](#)

31.151.4.105 when() [2/2]

```
void Fl_Widget::when (
    uchar i ) [inline]
```

Sets the flags used to decide when a callback is called.

This controls when callbacks are done. The following values are useful, the default value is FL_WHEN_RELEASE:

- 0: The callback is not done, but [changed\(\)](#) is turned on.
- FL_WHEN_CHANGED: The callback is done each time the text is changed by the user.
- FL_WHEN_RELEASE: The callback will be done when this widget loses the focus, including when the window is unmapped. This is a useful value for text fields in a panel where doing the callback on every change is wasteful. However the callback will also happen if the mouse is moved out of the window, which means it should not do anything visible (like pop up an error message). You might do better setting this to zero, and scanning all the items for [changed\(\)](#) when the OK button on a panel is pressed.
- FL_WHEN_ENTER_KEY: If the user types the Enter key, the entire text is selected, and the callback is done if the text has changed. Normally the Enter key will navigate to the next field (or insert a newline for a [Fl_Multiline_Input](#)) - this changes the behavior.
- FL_WHEN_ENTER_KEY|FL_WHEN_NOT_CHANGED: The Enter key will do the callback even if the text has not changed. Useful for command fields. [Fl_Widget::when\(\)](#) is a set of bitflags used by subclasses of [Fl_Widget](#) to decide when to do the callback.

If the value is zero then the callback is never done. Other values are described in the individual widgets. This field is in the base class so that you can scan a panel and [do_callback\(\)](#) on all the ones that don't do their own callbacks in response to an "OK" button.

Parameters

in	<i>i</i>	set of flags
----	----------	--------------

31.151.4.106 window()

```
Fl_Window * Fl_Widget::window ( ) const
```

Returns a pointer to the nearest parent window up the widget hierarchy.

This will return sub-windows if there are any, or the parent window if there's no sub-windows. If this widget IS the top-level window, NULL is returned.

Return values

NULL	if no window is associated with this widget.
------	--

Note

for an [Fl_Window](#) widget, this returns its *parent* window (if any), not *this* window.

See also

[top_window\(\)](#)

31.151.4.107 x() [1/2]

```
int Fl_Widget::x ( ) const [inline]
```

Gets the widget position in its window.

Returns

the x position relative to the window

31.151.4.108 x() [2/2]

```
void Fl_Widget::x (
    int v ) [inline], [protected]
```

Internal use only.

Use [position\(int,int\)](#), [size\(int,int\)](#) or [resize\(int,int,int,int\)](#) instead.

31.151.4.109 y() [1/2]

```
int Fl_Widget::y ( ) const [inline]
```

Gets the widget position in its window.

Returns

the y position relative to the window

31.151.4.110 y() [2/2]

```
void Fl_Widget::y (
    int v ) [inline], [protected]
```

Internal use only.

Use [position\(int,int\)](#), [size\(int,int\)](#) or [resize\(int,int,int,int\)](#) instead.

The documentation for this class was generated from the following files:

- [Fl_Widget.H](#)
- [Fl.cxx](#)
- [fl_boxtype.cxx](#)
- [fl_labeltype.cxx](#)
- [fl_shortcut.cxx](#)
- [Fl_Tooltip.cxx](#)
- [Fl_Widget.cxx](#)
- [Fl_Window.cxx](#)

31.152 FL_Widget_Tracker Class Reference

This class should be used to control safe widget deletion.

```
#include <Fl.H>
```

Public Member Functions

- `int deleted ()`
Returns 1, if the watched widget has been deleted.
- `int exists ()`
Returns 1, if the watched widget exists (has not been deleted).
- `FL_Widget_Tracker (FL_Widget *wi)`
The constructor adds a widget to the watch list.
- `FL_Widget * widget ()`
Returns a pointer to the watched widget.
- `~FL_Widget_Tracker ()`
The destructor removes a widget from the watch list.

31.152.1 Detailed Description

This class should be used to control safe widget deletion.

You can use an `FL_Widget_Tracker` object to watch another widget, if you need to know, if this widget has been deleted during a callback.

This simplifies the use of the "safe widget deletion" methods `Fl::watch_widget_pointer()` and `Fl::release_widget_pointer()` and makes their use more reliable, because the destructor automatically releases the widget pointer from the widget watch list.

It is intended to be used as an automatic (local/stack) variable, such that the automatic destructor is called when the object's scope is left. This ensures that no stale widget pointers are left in the widget watch list (see example below).

You can also create `FL_Widget_Tracker` objects with `new`, but then it is your responsibility to delete the object (and thus remove the widget pointer from the watch list) when it is not needed any more.

Example:

```
int MyClass::handle (int event) {
    if (...) {
        FL_Widget_Tracker wp(this);           // watch myself
        do_callback();                       // call the callback
        if (wp.deleted()) return 1;          // exit, if deleted
        // Now we are sure that the widget has not been deleted.
        // It is safe to access the widget
        clear_changed();                     // access the widget
    }
}
```

31.152.2 Member Function Documentation

31.152.2.1 deleted()

```
int FL_Widget_Tracker::deleted () [inline]
```

Returns 1, if the watched widget has been deleted.

This is a convenience method. You can also use something like

```
if (wp.widget() == 0) // ...
```

where `wp` is an `FL_Widget_Tracker` object.

31.152.2.2 exists()

```
int FL_Widget_Tracker::exists () [inline]
```

Returns 1, if the watched widget exists (has not been deleted).

This is a convenience method. You can also use something like

```
if (wp.widget() != 0) // ...
```

where `wp` is an `Fl_Widget_Tracker` object.

31.152.2.3 widget()

```
Fl_Widget * Fl_Widget_Tracker::widget ( ) [inline]
```

Returns a pointer to the watched widget.

This pointer is `NULL`, if the widget has been deleted.

The documentation for this class was generated from the following files:

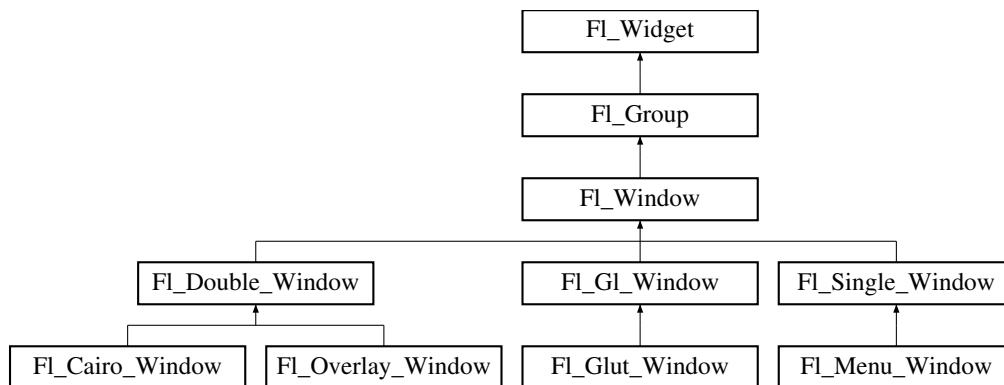
- [Fl.H](#)
- [Fl.cxx](#)

31.153 Fl_Window Class Reference

This widget produces an actual window.

```
#include <Fl_Window.H>
```

Inheritance diagram for `Fl_Window`:



Classes

- struct [shape_data_type](#)
Data supporting a non-rectangular window shape.

Public Member Functions

- virtual `Fl_Window * as_window ()`
Returns an `Fl_Window` pointer if this widget is an `Fl_Window`.
- unsigned int `border () const`
See void `Fl_Window::border(int)`
- void `border (int b)`
Sets whether or not the window manager border is around the window.
- void `clear_border ()`
Fast inline function to turn the window manager border off.
- void `clear_modal_states ()`
Clears the "modal" flags and converts a "modal" or "non-modal" window back into a "normal" window.
- void `copy_label (const char *a)`
Sets the window titlebar label to a copy of a character string.
- void `cursor (const Fl_RGB_Image *, int, int)`
Changes the cursor for this window.
- void `cursor (Fl_Cursor c, Fl_Color, Fl_Color=FL_WHITE)`
For back compatibility only.

- void `cursor` (`FL_Cursor`)
Changes the cursor for this window.
- int `decorated_h` ()
Returns the window height including any window title bar and any frame added by the window manager.
- int `decorated_w` ()
Returns the window width including any frame added by the window manager.
- void `default_cursor` (`FL_Cursor` c, `FL_Color`, `FL_Color=FL_WHITE`)
For back compatibility only.
- void `default_cursor` (`FL_Cursor`)
Sets the default window cursor.
- `FL_Window` (int w, int h, const char *title=0)
Creates a window from the given size and title.
- `FL_Window` (int x, int y, int w, int h, const char *title=0)
Creates a window from the given position, size and title.
- void `free_position` ()
Undoes the effect of a previous `resize()` or `show()` so that the next time `show()` is called the window manager is free to position the window.
- void `fullscreen` ()
Makes the window completely fill one or more screens, without any window manager border visible.
- unsigned int `fullscreen_active` () const
Returns non zero if FULLSCREEN flag is set, 0 otherwise.
- void `fullscreen_off` ()
Turns off any side effects of `fullscreen()`
- void `fullscreen_off` (int X, int Y, int W, int H)
Turns off any side effects of `fullscreen()` and does `resize(x,y,w,h)`.
- void `fullscreen_screens` (int top, int bottom, int left, int right)
Sets which screens should be used when this window is in fullscreen mode.
- virtual int `handle` (int)
Handles the specified event.
- virtual void `hide` ()
Removes the window from the screen.
- void `hotspot` (const `FL_Widget` &p, int offscreen=0)
See void `FL_Window::hotspot(int x, int y, int offscreen = 0)`
- void `hotspot` (const `FL_Widget` *, int offscreen=0)
See void `FL_Window::hotspot(int x, int y, int offscreen = 0)`
- void `hotspot` (int x, int y, int offscreen=0)
Positions the window so that the mouse is pointing at the given position, or at the center of the given widget, which may be the window itself.
- const void * `icon` () const
Gets the current icon window target dependent data.
- void `icon` (const `FL_RGB_Image` *)
Sets or resets a single window icon.
- void `icon` (const void *ic)
Sets the current icon window target dependent data.
- void `iconize` ()
Iconifies the window.
- const char * `iconlabel` () const
See void `FL_Window::iconlabel(const char)`*
- void `iconlabel` (const char *)
Sets the icon label.

- void **icons** (const [Fl_RGB_Image](#) *[], int)
 - Sets the window icons.*
- const char * **label** () const
 - See void [Fl_Window::label\(const char*\)](#)*
- void **label** (const char *)
 - Sets the window title bar label.*
- void **label** (const char *label, const char *iconlabel)
 - Sets the icon label.*
- void **make_current** ()
 - Sets things up so that the drawing functions in [<FL/fl_draw.H>](#) will go into this window.*
- unsigned int **menu_window** () const
 - Returns true if this window is a menu window.*
- unsigned int **modal** () const
 - Returns true if this window is modal.*
- unsigned int **non_modal** () const
 - Returns true if this window is modal or non-modal.*
- unsigned int **override** () const
 - Returns non zero if [FL_OVERRIDE](#) flag is set, 0 otherwise.*
- virtual void **resize** (int X, int Y, int W, int H)
 - Changes the size and position of the window.*
- void **set_menu_window** ()
 - Marks the window as a menu window.*
- void **set_modal** ()
 - A "modal" window, when [shown\(\)](#), will prevent any events from being delivered to other windows in the same program, and will also remain on top of the other windows (if the X window manager supports the "transient for" property).*
- void **set_non_modal** ()
 - A "non-modal" window (terminology borrowed from Microsoft Windows) acts like a [modal\(\)](#) one in that it remains on top, but it has no effect on event delivery.*
- void **set_override** ()
 - Activates the flags [NOBORDER|FL_OVERRIDE](#).*
- void **set_tooltip_window** ()
 - Marks the window as a tooltip window.*
- void **shape** (const [Fl_Image](#) &b)
 - Set the window's shape with an [Fl_Image](#).*
- void **shape** (const [Fl_Image](#) *img)
 - Assigns a non-rectangular shape to the window.*
- virtual void **show** ()
 - Puts the window on the screen.*
- void **show** (int argc, char **argv)
 - Puts the window on the screen and parses command-line arguments.*
- int **shown** ()
 - Returns non-zero if [show\(\)](#) has been called (but not [hide\(\)](#)).*
- void **size_range** (int minw, int minh, int maxw=0, int maxh=0, int dw=0, int dh=0, int aspect=0)
 - Sets the allowable range the user can resize this window to.*
- unsigned int **tooltip_window** () const
 - Returns true if this window is a tooltip window.*
- void **wait_for_expose** ()
 - Waits for the window to be displayed after calling [show\(\)](#).*
- int **x_root** () const
 - Gets the x position of the window on the screen.*

- const char * [xclass](#) () const
Returns the xclass for this window, or a default.
- void [xclass](#) (const char *c)
Sets the xclass for this window.
- int [y_root](#) () const
Gets the y position of the window on the screen.
- virtual [~FI_Window](#) ()
The destructor also deletes all the children.

Static Public Member Functions

- static [FI_Window](#) * [current](#) ()
Returns the last window that was made current.
- static void [default_callback](#) ([FI_Window](#) *, void *v)
Back compatibility: Sets the default callback v for win to call on close event.
- static void [default_icon](#) (const [FI_RGB_Image](#) *)
Sets a single default window icon.
- static void [default_icons](#) (const [FI_RGB_Image](#) *[], int)
Sets the default window icons.
- static const char * [default_xclass](#) ()
Returns the default xclass.
- static void [default_xclass](#) (const char *)
Sets the default window xclass.

Protected Member Functions

- virtual void [draw](#) ()
Draws the widget.
- virtual void [flush](#) ()
Forces the window to be drawn, this window is also made current and calls [draw\(\)](#).
- int [force_position](#) () const
Returns the internal state of the window's FORCE_POSITION flag.
- void [force_position](#) (int force)
Sets an internal flag that tells FLTK and the window manager to honor position requests.
- void [free_icons](#) ()
Deletes all icons previously attached to the window.

Protected Attributes

- [shape_data_type](#) * [shape_data_](#)
non-null means the window has a non-rectangular shape

Static Protected Attributes

- static [FI_Window](#) * [current_](#)
Stores the last window that was made current.

Friends

- class [FI_X](#)

Additional Inherited Members

31.153.1 Detailed Description

This widget produces an actual window.

This can either be a main window, with a border and title and all the window management controls, or a "subwindow" inside a window. This is controlled by whether or not the window has a [parent\(\)](#).

Once you create a window, you usually add children [Fl_Widget](#)'s to it by using `window->add(child)` for each new widget. See [Fl_Group](#) for more information on how to add and remove children.

There are several subclasses of [Fl_Window](#) that provide double-buffering, overlay, menu, and OpenGL support.

The window's callback is done if the user tries to close a window using the window manager and [Fl::modal\(\)](#) is zero or equal to the window. [Fl_Window](#) has a default callback that calls [Fl_Window::hide\(\)](#).

31.153.2 Constructor & Destructor Documentation

31.153.2.1 Fl_Window() [1/2]

```
Fl_Window::Fl_Window (
    int w,
    int h,
    const char * title = 0 )
```

Creates a window from the given size and title.

If [Fl_Group::current\(\)](#) is not NULL, the window is created as a subwindow of the parent window.

The (w,h) form of the constructor creates a top-level window and asks the window manager to position the window.

The (x,y,w,h) form of the constructor either creates a subwindow or a top-level window at the specified location (x,y), subject to window manager configuration. If you do not specify the position of the window, the window manager will pick a place to show the window or allow the user to pick a location. Use [position\(x,y\)](#) or [hotspot\(\)](#) before calling [show\(\)](#) to request a position on the screen. See [Fl_Window::resize\(\)](#) for some more details on positioning windows. Top-level windows initially have [visible\(\)](#) set to 0 and [parent\(\)](#) set to NULL. Subwindows initially have [visible\(\)](#) set to 1 and [parent\(\)](#) set to the parent window pointer.

[Fl_Widget::box\(\)](#) defaults to FL_FLAT_BOX. If you plan to completely fill the window with children widgets you should change this to FL_NO_BOX. If you turn the window border off you may want to change this to FL_UP_BOX.

See also

[Fl_Window\(int x, int y, int w, int h, const char* title\)](#)

31.153.2.2 Fl_Window() [2/2]

```
Fl_Window::Fl_Window (
    int x,
    int y,
    int w,
    int h,
    const char * title = 0 )
```

Creates a window from the given position, size and title.

See also

[Fl_Window\(int w, int h, const char *title\)](#)

31.153.2.3 ~Fl_Window()

```
Fl_Window::~Fl_Window ( ) [virtual]
```

The destructor *also deletes all the children*.

This allows a whole tree to be deleted at once, without having to keep a pointer to all the children in the user code. A kludge has been done so the `Fl_Window` and all of its children can be automatic (local) variables, but you must declare the `Fl_Window` *first* so that it is destroyed last.

31.153.3 Member Function Documentation

31.153.3.1 `as_window()`

```
virtual Fl_Window * Fl_Window::as_window ( ) [inline], [virtual]
```

Returns an `Fl_Window` pointer if this widget is an `Fl_Window`.

Use this method if you have a widget (pointer) and need to know whether this widget is derived from `Fl_Window`. If it returns non-NULL, then the widget in question is derived from `Fl_Window`, and you can use the returned pointer to access its children or other `Fl_Window`-specific methods.

Return values

NULL	if this widget is not derived from <code>Fl_Window</code> .
------	---

Note

This method is provided to avoid `dynamic_cast`.

See also

[Fl_Widget::as_group\(\)](#), [Fl_Widget::as_gl_window\(\)](#)

Reimplemented from [Fl_Widget](#).

31.153.3.2 `border()`

```
void Fl_Window::border (
    int b )
```

Sets whether or not the window manager border is around the window.

The default value is true. `void border(int)` can be used to turn the border on and off. *Under most X window managers this does not work after `show()` has been called, although SGI's 4DWM does work.*

31.153.3.3 `clear_border()`

```
void Fl_Window::clear_border ( ) [inline]
```

Fast inline function to turn the window manager border off.

It only works before `show()` is called.

31.153.3.4 `clear_modal_states()`

```
void Fl_Window::clear_modal_states ( ) [inline]
```

Clears the "modal" flags and converts a "modal" or "non-modal" window back into a "normal" window.

Note that there are *three* states for a window: modal, non-modal, and normal.

You can not change the "modality" of a window whilst it is shown, so it is necessary to first `hide()` the window, change its "modality" as required, then re-show the window for the new state to take effect.

This method can also be used to change a "modal" window into a "non-modal" one. On several supported platforms, the "modal" state over-rides the "non-modal" state, so the "modal" state must be cleared before the window can be set into the "non-modal" state. In general, the following sequence should work:

```
win->hide();
win->clear_modal_states();
//Set win to new state as desired, or leave "normal", e.g...
win->set_non_modal();
win->show();
```

Note

Under some window managers, the sequence of hiding the window and changing its modality will often cause it to be re-displayed at a different position when it is subsequently shown. This is an irritating feature but appears to be unavoidable at present. As a result we would advise to use this method only when absolutely necessary.

See also

void [set_modal\(\)](#), void [set_non_modal\(\)](#)

31.153.3.5 current()

```
Fl_Window * Fl_Window::current ( ) [static]
```

Returns the last window that was made current.

See also

[Fl_Window::make_current\(\)](#)

31.153.3.6 cursor() [1/3]

```
void Fl_Window::cursor (
    const Fl_RGB_Image * image,
    int hotx,
    int hoty )
```

Changes the cursor for this window.

This always calls the system, if you are changing the cursor a lot you may want to keep track of how you set it in a static variable and call this only if the new cursor is different.

The default cursor will be used if the provided image cannot be used as a cursor.

See also

[cursor\(Fl_Cursor\)](#), [default_cursor\(\)](#)

31.153.3.7 cursor() [2/3]

```
void Fl_Window::cursor (
    Fl_Cursor c,
    Fl_Color ,
    Fl_Color = FL_WHITE )
```

For back compatibility only.

Same as [Fl_Window::cursor\(Fl_Cursor\)](#)

31.153.3.8 cursor() [3/3]

```
void Fl_Window::cursor (
    Fl_Cursor c )
```

Changes the cursor for this window.

This always calls the system, if you are changing the cursor a lot you may want to keep track of how you set it in a static variable and call this only if the new cursor is different.

The type `Fl_Cursor` is an enumeration defined in [<FL/Enumerations.H>](#).

See also

[cursor\(const Fl_RGB_Image*, int, int\)](#), [default_cursor\(\)](#)

31.153.3.9 decorated_h()

```
int Fl_Window::decorated_h ( )
```

Returns the window height including any window title bar and any frame added by the window manager. Same as [h\(\)](#) if applied to a subwindow.

31.153.3.10 decorated_w()

```
int Fl_Window::decorated_w ( )
```

Returns the window width including any frame added by the window manager. Same as [w\(\)](#) if applied to a subwindow.

31.153.3.11 default_cursor() [1/2]

```
void Fl_Window::default_cursor (
    Fl_Cursor c,
    Fl_Color ,
    Fl_Color = FL_WHITE )
```

For back compatibility only.
same as [Fl_Window::default_cursor\(Fl_Cursor\)](#)

31.153.3.12 default_cursor() [2/2]

```
void Fl_Window::default_cursor (
    Fl_Cursor c )
```

Sets the default window cursor.
This is the cursor that will be used after the mouse pointer leaves a widget with a custom cursor set.

See also

[cursor\(const Fl_RGB_Image*, int, int\), default_cursor\(\)](#)

31.153.3.13 default_icon()

```
void Fl_Window::default_icon (
    const Fl_RGB_Image * icon ) [static]
```

Sets a single default window icon.
If *icon* is NULL the current default icons are removed.

Parameters

in	<i>icon</i>	default icon for all windows subsequently created or NULL
----	-------------	---

See also

[Fl_Window::default_icons\(const Fl_RGB_Image *\[\], int\)](#)

[Fl_Window::icon\(const Fl_RGB_Image *\)](#)

[Fl_Window::icons\(const Fl_RGB_Image *\[\], int\)](#)

31.153.3.14 default_icons()

```
void Fl_Window::default_icons (
    const Fl_RGB_Image * icons[],
    int count ) [static]
```

Sets the default window icons.

The default icons are used for all windows that don't have their own icons set before `show()` is called. You can change the default icons whenever you want, but this only affects windows that are created (and shown) after this call.

The given images in `icons` are copied. You can use a local variable or free the images immediately after this call.

Parameters

in	<i>icons</i>	default icons for all windows subsequently created
in	<i>count</i>	number of images in <code>icons</code> . Set to 0 to remove the current default icons

See also

[Fl_Window::default_icon\(const Fl_RGB_Image *\)](#)

[Fl_Window::icon\(const Fl_RGB_Image *\)](#)

[Fl_Window::icons\(const Fl_RGB_Image *\[\], int\)](#)

31.153.3.15 default_xclass() [1/2]

```
const char * Fl_Window::default_xclass ( ) [static]
```

Returns the default xclass.

See also

[Fl_Window::default_xclass\(const char *\)](#)

31.153.3.16 default_xclass() [2/2]

```
void Fl_Window::default_xclass (
    const char * xc ) [static]
```

Sets the default window xclass.

The default xclass is used for all windows that don't have their own xclass set before `show()` is called. You can change the default xclass whenever you want, but this only affects windows that are created (and shown) after this call.

The given string `xc` is copied. You can use a local variable or free the string immediately after this call.

If you don't call this, the default xclass for all windows will be "FLTK". You can reset the default xclass by specifying NULL for `xc`.

If you call [Fl_Window::xclass\(const char *\)](#) for any window, then this also sets the default xclass, unless it has been set before.

Parameters

in	<i>xc</i>	default xclass for all windows subsequently created
----	-----------	---

See also

[Fl_Window::xclass\(const char *\)](#)

31.153.3.17 draw()

```
void Fl_Window::draw ( ) [protected], [virtual]
```

Draws the widget.

Never call this function directly. FLTK will schedule redrawing whenever needed. If your widget must be redrawn as soon as possible, call [redraw\(\)](#) instead.

Override this function to draw your own widgets.

If you ever need to call another widget's draw method *from within your own `draw()` method*, e.g. for an embedded scrollbar, you can do it (because `draw()` is virtual) like this:

```
Fl_Widget *s = &scroll;           // scroll is an embedded Fl_Scrollbar
s->draw();                         // calls Fl_Scrollbar::draw()
```

Reimplemented from [Fl_Group](#).

Reimplemented in [Fl_Cairo_Window](#), [Fl_Gl_Window](#), and [Fl_Glut_Window](#).

31.153.3.18 flush()

```
void Fl_Window::flush ( ) [protected], [virtual]
```

Forces the window to be drawn, this window is also made current and calls `draw()`.

Reimplemented in [Fl_Double_Window](#), [Fl_Gl_Window](#), [Fl_Menu_Window](#), [Fl_Overlay_Window](#), and [Fl_Single_Window](#).

31.153.3.19 force_position() [1/2]

```
int Fl_Window::force_position ( ) const [inline], [protected]
```

Returns the internal state of the window's FORCE_POSITION flag.

Return values

1	if flag is set
0	otherwise

See also

[force_position\(int\)](#)

31.153.3.20 force_position() [2/2]

```
void Fl_Window::force_position (
    int force ) [inline], [protected]
```

Sets an internal flag that tells FLTK and the window manager to honor position requests.

This is used internally and should not be needed by user code.

Parameters

in	<i>force</i>	1 to set the FORCE_POSITION flag, 0 to clear it
----	--------------	---

31.153.3.21 free_icons()

```
void Fl_Window::free_icons ( ) [protected]
```

Deletes all icons previously attached to the window.

See also

[Fl_Window::icons\(const Fl_RGB_Image *icons\[\], int count\)](#)

31.153.3.22 free_position()

```
void Fl_Window::free_position ( ) [inline]
```

Undoes the effect of a previous `resize()` or `show()` so that the next time `show()` is called the window manager is free to position the window.

This is for Forms compatibility only.

Deprecated please use `force_position(0)` instead

31.153.3.23 fullscreen()

```
void Fl_Window::fullscreen ( )
```

Makes the window completely fill one or more screens, without any window manager border visible.

You must use `fullscreen_off()` to undo this.

Note

On some platforms, this can result in the keyboard being grabbed. The window may also be recreated, meaning `hide()` and `show()` will be called.

See also

void `Fl_Window::fullscreen_screens()`

31.153.3.24 fullscreen_screens()

```
void Fl_Window::fullscreen_screens (
    int top,
    int bottom,
    int left,
    int right )
```

Sets which screens should be used when this window is in fullscreen mode.

The window will be resized to the top of the screen with index `top`, the bottom of the screen with index `bottom`, etc.

If this method is never called, or if any argument is < 0 , then the window will be resized to fill the screen it is currently on.

See also

void `Fl_Window::fullscreen()`

31.153.3.25 handle()

```
int Fl_Window::handle (
    int event ) [virtual]
```

Handles the specified event.

You normally don't call this method directly, but instead let FLTK do it when the user interacts with the widget.

When implemented in a widget, this function must return 0 if the widget does not use the event or 1 otherwise.

Most of the time, you want to call the inherited `handle()` method in your overridden method so that you don't short-circuit events that you don't handle. In this last case you should return the callee `retval`.

Parameters

in	<i>event</i>	the kind of event received
----	--------------	----------------------------

Return values

0	if the event was not used or understood
1	if the event was used and can be deleted

See also

[Fl_Event](#)

Reimplemented from [Fl_Group](#).

Reimplemented in [Fl_Gl_Window](#), and [Fl_Glut_Window](#).

31.153.3.26 hide()

```
void Fl_Window::hide ( ) [virtual]
```

Removes the window from the screen.

If the window is already hidden or has not been shown then this does nothing and is harmless.

Reimplemented from [Fl_Widget](#).

Reimplemented in [Fl_Double_Window](#), [Fl_Gl_Window](#), [Fl_Menu_Window](#), and [Fl_Overlay_Window](#).

31.153.3.27 hotspot()

```
void Fl_Window::hotspot (
    int x,
    int y,
    int offscreen = 0 )
```

Positions the window so that the mouse is pointing at the given position, or at the center of the given widget, which may be the window itself.

If the optional offscreen parameter is non-zero, then the window is allowed to extend off the screen (this does not work with some X window managers).

See also

[position\(\)](#)

31.153.3.28 icon() [1/3]

```
const void * Fl_Window::icon ( ) const
```

Gets the current icon window target dependent data.

Deprecated in 1.3.3

31.153.3.29 icon() [2/3]

```
void Fl_Window::icon (
    const Fl_RGB_Image * icon )
```

Sets or resets a single window icon.

A window icon *can* be changed while the window is shown, but this *may* be platform and/or window manager dependent. To be sure that the window displays the correct window icon you should always set the icon before the window is shown.

If a window icon has not been set for a particular window, then the default window icon (see links below) or the system default icon will be used.

Parameters

in	<i>icon</i>	icon for this window, NULL to reset window icon.
----	-------------	--

See also

[Fl_Window::default_icon\(const Fl_RGB_Image *\)](#)
[Fl_Window::default_icons\(const Fl_RGB_Image *\[\], int\)](#)
[Fl_Window::icons\(const Fl_RGB_Image *\[\], int\)](#)

31.153.3.30 icon() [3/3]

```
void Fl_Window::icon (
    const void * ic )
```

Sets the current icon window target dependent data.

Deprecated in 1.3.3

31.153.3.31 iconize()

```
void Fl_Window::iconize ( )
```

Iconifies the window.

If you call this when [shown\(\)](#) is false it will [show\(\)](#) it as an icon. If the window is already iconified this does nothing. Call [show\(\)](#) to restore the window.

When a window is iconified/restored (either by these calls or by the user) the [handle\(\)](#) method is called with FL_HIDE and FL_SHOW events and [visible\(\)](#) is turned on and off.

There is no way to control what is drawn in the icon except with the string passed to [Fl_Window::xclass\(\)](#). You should not rely on window managers displaying the icons.

31.153.3.32 icons()

```
void Fl_Window::icons (
    const Fl_RGB_Image * icons[],
    int count )
```

Sets the window icons.

You may set multiple window icons with different sizes. Dependent on the platform and system settings the best (or the first) icon will be chosen.

The given images in `icons` are copied. You can use a local variable or free the images immediately after this call. If `count` is zero, current icons are removed. If `count` is greater than zero (must not be negative), then `icons[]` must contain at least `count` valid image pointers (not NULL). Otherwise the behavior is undefined.

Parameters

in	<i>icons</i>	icons for this window
in	<i>count</i>	number of images in <code>icons</code> . Set to 0 to remove the current icons

See also

[Fl_Window::default_icon\(const Fl_RGB_Image *\)](#)
[Fl_Window::default_icons\(const Fl_RGB_Image *\[\], int\)](#)
[Fl_Window::icon\(const Fl_RGB_Image *\)](#)

31.153.3.33 make_current()

```
void Fl_Window::make_current ( )
```

Sets things up so that the drawing functions in [<FL/fl_draw.H>](#) will go into this window.

This is useful for incremental update of windows, such as in an idle callback, which will make your program behave much better if it draws a slow graphic. **Danger: incremental update is very hard to debug and maintain!** This method only works for the [Fl_Window](#) and [Fl_Gl_Window](#) derived classes.

31.153.3.34 modal()

```
unsigned int Fl_Window::modal ( ) const [inline]
```

Returns true if this window is modal.

31.153.3.35 resize()

```
virtual void Fl_Window::resize (
    int X,
    int Y,
    int W,
    int H ) [virtual]
```

Changes the size and position of the window.

If [shown\(\)](#) is true, these changes are communicated to the window server (which may refuse that size and cause a further resize). If [shown\(\)](#) is false, the size and position are used when [show\(\)](#) is called. See [Fl_Group](#) for the effect of resizing on the child widgets.

You can also call the [Fl_Widget](#) methods [size\(x,y\)](#) and [position\(w,h\)](#), which are inline wrappers for this virtual function.

A top-level window can not force, but merely suggest a position and size to the operating system. The window manager may not be willing or able to display a window at the desired position or with the given dimensions. It is up to the application developer to verify window parameters after the resize request.

Reimplemented from [Fl_Group](#).

Reimplemented in [Fl_Double_Window](#), [Fl_Gl_Window](#), and [Fl_Overlay_Window](#).

31.153.3.36 set_menu_window()

```
void Fl_Window::set_menu_window ( ) [inline]
```

Marks the window as a menu window.

This is intended for internal use, but it can also be used if you write your own menu handling. However, this is not recommended.

This flag is used for correct "parenting" of windows in communication with the windowing system. Modern X window managers can use different flags to distinguish menu and tooltip windows from normal windows.

This must be called before the window is shown and cannot be changed later.

31.153.3.37 set_modal()

```
void Fl_Window::set_modal ( ) [inline]
```

A "modal" window, when [shown\(\)](#), will prevent any events from being delivered to other windows in the same program, and will also remain on top of the other windows (if the X window manager supports the "transient for" property).

Several modal windows may be shown at once, in which case only the last one shown gets events. You can see which window (if any) is modal by calling [Fl::modal\(\)](#).

31.153.3.38 set_non_modal()

```
void Fl_Window::set_non_modal ( ) [inline]
```

A "non-modal" window (terminology borrowed from Microsoft Windows) acts like a [modal\(\)](#) one in that it remains on top, but it has no effect on event delivery.

There are *three* states for a window: modal, non-modal, and normal.

31.153.3.39 set_tooltip_window()

```
void Fl_Window::set_tooltip_window ( ) [inline]
```

Marks the window as a tooltip window.

This is intended for internal use, but it can also be used if you write your own tooltip handling. However, this is not recommended.

This flag is used for correct "parenting" of windows in communication with the windowing system. Modern X window managers can use different flags to distinguish menu and tooltip windows from normal windows.

This must be called before the window is shown and cannot be changed later.

Note

Since `Fl_Tooltip_Window` is derived from `Fl_Menu_Window`, this also **clears** the `menu_window()` state.

31.153.3.40 shape() [1/2]

```
void Fl_Window::shape (
    const Fl_Image & b ) [inline]
```

Set the window's shape with an `Fl_Image`.

See also

```
void shape(const Fl_Image* img)
```

31.153.3.41 shape() [2/2]

```
void Fl_Window::shape (
    const Fl_Image * img )
```

Assigns a non-rectangular shape to the window.

This function gives an arbitrary shape (not just a rectangular region) to an `Fl_Window`. An `Fl_Image` of any dimension can be used as mask; it is rescaled to the window's dimension as needed.

The layout and widgets inside are unaware of the mask shape, and most will act as though the window's rectangular bounding box is available to them. It is up to you to make sure they adhere to the bounds of their masking shape.

The `img` argument can be an `Fl_Bitmap`, `Fl_Pixmap`, `Fl_RGB_Image` or `Fl_Shared_Image`:

- With `Fl_Bitmap` or `Fl_Pixmap`, the shaped window covers the image part where bitmap bits equal one, or where the pixmap is not fully transparent.
- With an `Fl_RGB_Image` with an alpha channel (depths 2 or 4), the shaped window covers the image part that is not fully transparent.
- With an `Fl_RGB_Image` of depth 1 (gray-scale) or 3 (RGB), the shaped window covers the non-black image part.
- With an `Fl_Shared_Image`, the shape is determined by rules above applied to the underlying image. The shared image should not have been scaled through `Fl_Shared_Image::scale()`.

Platform details:

- On the unix/linux platform, the SHAPE extension of the X server is required. This function does control the shape of `Fl_Gl_Window` instances.
- On the MSWindows platform, this function does nothing with class `Fl_Gl_Window`.
- On the Mac platform, OS version 10.4 or above is required. An 8-bit shape-mask is used when `img` is an `Fl_RGB_Image`: with depths 2 or 4, the image alpha channel becomes the shape mask such that areas with alpha = 0 are out of the shaped window; with depths 1 or 3, white and black are in and out of the shaped window, respectively, and other colors give intermediate masking scores. This function does nothing with class `Fl_Gl_Window`.

The window borders and caption created by the window system are turned off by default. They can be re-enabled by calling `Fl_Window::border(1)`.

A usage example is found at `example/shapedwindow.cxx`.

Version

1.3.3 (and requires compilation with `FLTK_ABI_VERSION >= 10303`)

31.153.3.42 `show()` [1/2]

```
virtual void Fl_Window::show ( ) [virtual]
```

Puts the window on the screen.

Usually (on X) this has the side effect of opening the display.

If the window is already shown then it is restored and raised to the top. This is really convenient because your program can call `show()` at any time, even if the window is already up. It also means that `show()` serves the purpose of `raise()` in other toolkits.

`Fl_Window::show(int argc, char **argv)` is used for top-level windows and allows standard arguments to be parsed from the command-line.

Note

For some obscure reasons `Fl_Window::show()` resets the current group by calling `Fl_Group::current(0)`. The comments in the code say "get rid of very common user bug: forgot end()". Although this is true it may have unwanted side effects if you `show()` an unrelated window (maybe for an error message or warning) while building a window or any other group widget.

Todo Check if we can remove resetting the current group in a later FLTK version (after 1.3.x). This may break "already broken" programs though if they rely on this "feature".

See also

[Fl_Window::show\(int argc, char **argv\)](#)

Reimplemented from [Fl_Widget](#).

Reimplemented in [Fl_Double_Window](#), [Fl_Gl_Window](#), [Fl_Menu_Window](#), [Fl_Overlay_Window](#), and [Fl_Single_Window](#).

31.153.3.43 `show()` [2/2]

```
void Fl_Window::show (
    int argc,
    char ** argv )
```

Puts the window on the screen and parses command-line arguments.

Usually (on X) this has the side effect of opening the display.

This form should be used for top-level windows, at least for the first (main) window. It allows standard arguments to be parsed from the command-line. You can use `argc` and `argv` from `main(int argc, char **argv)` for this call.

The first call also sets up some system-specific internal variables like the system colors.

Todo explain which system parameters are set up.

Parameters

<code>argc</code>	command-line argument count, usually from <code>main()</code>
<code>argv</code>	command-line argument vector, usually from <code>main()</code>

See also

virtual void [Fl_Window::show\(\)](#)

31.153.3.44 shown()

```
int Fl_Window::shown ( ) [inline]
```

Returns non-zero if [show\(\)](#) has been called (but not [hide\(\)](#)).

You can tell if a window is iconified with `(w->shown\(\) && !w->visible\(\))`.

31.153.3.45 size_range()

```
void Fl_Window::size_range (
    int minw,
    int minh,
    int maxw = 0,
    int maxh = 0,
    int dw = 0,
    int dh = 0,
    int aspect = 0 ) [inline]
```

Sets the allowable range the user can resize this window to.

This only works for top-level windows.

- `minw` and `minh` are the smallest the window can be. Either value must be greater than 0.
- `maxw` and `maxh` are the largest the window can be. If either is *equal* to the minimum then you cannot resize in that direction. If either is zero then FLTK picks a maximum size in that direction such that the window will fill the screen.
- `dw` and `dh` are size increments. The window will be constrained to widths of `minw + N * dw`, where N is any non-negative integer. If these are less or equal to 1 they are ignored (this is ignored on WIN32).
- `aspect` is a flag that indicates that the window should preserve its aspect ratio. This only works if both the maximum and minimum have the same aspect ratio (ignored on WIN32 and by many X window managers).

If this function is not called, FLTK tries to figure out the range from the setting of [resizable\(\)](#):

- If [resizable\(\)](#) is NULL (this is the default) then the window cannot be resized and the resize border and max-size control will not be displayed for the window.
- If either dimension of [resizable\(\)](#) is less than 100, then that is considered the minimum size. Otherwise the [resizable\(\)](#) has a minimum size of 100.
- If either dimension of [resizable\(\)](#) is zero, then that is also the maximum size (so the window cannot resize in that direction).

It is undefined what happens if the current size does not fit in the constraints passed to [size_range\(\)](#).

31.153.3.46 wait_for_expose()

```
void Fl_Window::wait_for_expose ( )
```

Waits for the window to be displayed after calling [show\(\)](#).

[Fl_Window::show\(\)](#) is not guaranteed to show and draw the window on all platforms immediately. Instead this is done in the background; particularly on X11 it will take a few messages (client server roundtrips) to display the window. Usually this small delay doesn't matter, but in some cases you may want to have the window instantiated and displayed synchronously.

Currently (as of FLTK 1.3.4) this method has an effect on X11 and Mac OS. On Windows, [show\(\)](#) is always synchronous. The effect of [show\(\)](#) varies with versions of Mac OS X: early versions have the window appear on the screen when [show\(\)](#) returns, later versions don't. If you want to write portable code and need this synchronous [show\(\)](#) feature, add `win->wait_for_expose\(\)` on all platforms, and FLTK will just do the right thing.

This method can be used for displaying splash screens before calling [Fl::run\(\)](#) or for having exact control over which window has the focus after calling [show\(\)](#).

If the window is not [shown\(\)](#), this method does nothing.

Note

Depending on the platform and window manager `wait_for_expose()` may not guarantee that the window is fully drawn when it is called. Under X11 it may only make sure that the window is **mapped**, i.e. the internal (OS dependent) window object was created (and maybe shown on the desktop as an empty frame or something like that). You may need to call `Fl::flush()` after `wait_for_expose()` to make sure the window and all its widgets are drawn and thus visible.

FLTK does the best it can do to make sure that all widgets get drawn if you call `wait_for_expose()` and `Fl::flush()`. However, dependent on the window manager it can not be guaranteed that this does always happen synchronously. The only guaranteed behavior that all widgets are eventually drawn is if the FLTK event loop is run continuously, for instance with `Fl::run()`.

See also

virtual void [Fl_Window::show\(\)](#)

Example code for displaying a window before calling `Fl::run()`

```
Fl_Double_Window win = new Fl_Double_Window(...);
// do more window initialization here ...
win->show();           // show window
win->wait_for_expose(); // wait, until displayed
Fl::flush();          // make sure everything gets drawn
// do more initialization work that needs some time here ...
Fl::run();            // start FLTK event loop
```

Note that the window will not be responsive until the event loop is started with `Fl::run()`.

31.153.3.47 xclass() [1/2]

```
const char * Fl_Window::xclass ( ) const
```

Returns the xclass for this window, or a default.

See also

[Fl_Window::default_xclass\(const char *\)](#)

[Fl_Window::xclass\(const char *\)](#)

31.153.3.48 xclass() [2/2]

```
void Fl_Window::xclass (
    const char * xc )
```

Sets the xclass for this window.

A string used to tell the system what type of window this is. Mostly this identifies the picture to draw in the icon. This only works if called *before* calling `show()`.

Under X, this is turned into a XA_WM_CLASS pair by truncating at the first non-alphanumeric character and capitalizing the first character, and the second one if the first is 'x'. Thus "foo" turns into "foo, Foo", and "xprog.1" turns into "xprog, XProg".

Under Microsoft Windows, this string is used as the name of the WNDCLASS structure, though it is not clear if this can have any visible effect.

Since

FLTK 1.3 the passed string is copied. You can use a local variable or free the string immediately after this call. Note that FLTK 1.1 stores the *pointer* without copying the string.

If the default xclass has not yet been set, this also sets the default xclass for all windows created subsequently.

See also

[Fl_Window::default_xclass\(const char *\)](#)

31.153.4 Member Data Documentation

31.153.4.1 current_

```
Fl_Window* Fl_Window::current_ [static], [protected]
```

Stores the last window that was made current.

See `current() const`

The documentation for this class was generated from the following files:

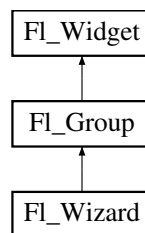
- [Fl_Window.H](#)
- [Fl.cxx](#)
- [Fl_arg.cxx](#)
- [fl_cursor.cxx](#)
- [Fl_Window.cxx](#)
- [Fl_Window_fullscreen.cxx](#)
- [Fl_Window_hotspot.cxx](#)
- [Fl_Window_iconize.cxx](#)
- [Fl_Window_shape.cxx](#)

31.154 Fl_Wizard Class Reference

This widget is based off the [Fl_Tabs](#) widget, but instead of displaying tabs it only changes "tabs" under program control.

```
#include <Fl_Wizard.H>
```

Inheritance diagram for Fl_Wizard:



Public Member Functions

- [Fl_Wizard](#) (int, int, int, int, const char *=0)
 - The constructor creates the [Fl_Wizard](#) widget at the specified position and size.*
- void [next](#) ()
 - This method shows the next child of the wizard.*
- void [prev](#) ()
 - Shows the previous child.*
- [Fl_Widget *](#) [value](#) ()
 - Gets the current visible child widget.*
- void [value](#) ([Fl_Widget *](#))
 - Sets the child widget that is visible.*

Additional Inherited Members

31.154.1 Detailed Description

This widget is based off the [Fl_Tabs](#) widget, but instead of displaying tabs it only changes "tabs" under program control.

Its primary purpose is to support "wizards" that step a user through configuration or troubleshooting tasks.

As with [Fl_Tabs](#), wizard panes are composed of child (usually [Fl_Group](#)) widgets. Navigation buttons must be added separately.

31.154.2 Constructor & Destructor Documentation

31.154.2.1 Fl_Wizard()

```
Fl_Wizard::Fl_Wizard (
    int xx,
    int yy,
    int ww,
    int hh,
    const char * l = 0 )
```

The constructor creates the [Fl_Wizard](#) widget at the specified position and size. The inherited destructor destroys the widget and its children.

31.154.3 Member Function Documentation

31.154.3.1 next()

```
void Fl_Wizard::next ( )
```

This method shows the next child of the wizard.

If the last child is already visible, this function does nothing.

The documentation for this class was generated from the following files:

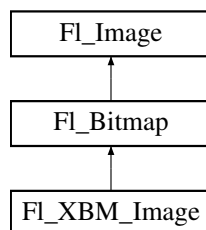
- [Fl_Wizard.H](#)
- [Fl_Wizard.cxx](#)

31.155 Fl_XBM_Image Class Reference

The [Fl_XBM_Image](#) class supports loading, caching, and drawing of X Bitmap (XBM) bitmap files.

```
#include <Fl_XBM_Image.H>
```

Inheritance diagram for [Fl_XBM_Image](#):



Public Member Functions

- [Fl_XBM_Image](#) (const char *filename)

The constructor loads the named XBM file from the given name filename.

Additional Inherited Members

31.155.1 Detailed Description

The [Fl_XBM_Image](#) class supports loading, caching, and drawing of X Bitmap (XBM) bitmap files.

31.155.2 Constructor & Destructor Documentation

31.155.2.1 Fl_XBM_Image()

```
Fl_XBM_Image::Fl_XBM_Image (
    const char * name )
```

The constructor loads the named XBM file from the given name filename.
The destructor frees all memory and server resources that are used by the image.
The documentation for this class was generated from the following files:

- Fl_XBM_Image.H
- Fl_XBM_Image.cxx

31.156 Fl_XColor Struct Reference

Public Attributes

- unsigned char **b**
- unsigned char **g**
- unsigned char **mapped**
- unsigned long **pixel**
- unsigned char **r**

The documentation for this struct was generated from the following file:

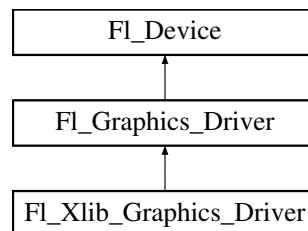
- Fl_XColor.H

31.157 Fl_Xlib_Graphics_Driver Class Reference

The Xlib-specific graphics class.

```
#include <Fl_Device.H>
```

Inheritance diagram for Fl_Xlib_Graphics_Driver:



Public Member Functions

- const char * [class_name](#) ()
Returns the name of the class of this object.
- void [color](#) (Fl_Color c)
see [fl_color\(Fl_Color c\)](#).
- void [color](#) (uchar r, uchar g, uchar b)
see [fl_color\(uchar r, uchar g, uchar b\)](#).
- void [copy_offscreen](#) (int x, int y, int w, int h, Fl_Offscreen pixmap, int srcx, int srcy)
see [fl_copy_offscreen\(\)](#)
- int [descent](#) ()
see [fl_descent\(\)](#).
- void [draw](#) (const char *str, int n, int x, int y)
*see [fl_draw\(const char *str, int n, int x, int y\)](#).*
- void [draw](#) (Fl_Bitmap *pixmap, int XP, int YP, int WP, int HP, int cx, int cy)
Draws an [Fl_Bitmap](#) object to the device.

- void `draw` (`Fl_Pixmap *pxm`, `int XP`, `int YP`, `int WP`, `int HP`, `int cx`, `int cy`)
Draws an `Fl_Pixmap` object to the device.
- void `draw` (`Fl_RGB_Image *img`, `int XP`, `int YP`, `int WP`, `int HP`, `int cx`, `int cy`)
Draws an `Fl_RGB_Image` object to the device.
- void `draw` (`int angle`, `const char *str`, `int n`, `int x`, `int y`)
*see `fl_draw(int angle, const char *str, int n, int x, int y)`.*
- void `draw_image` (`const uchar *buf`, `int X`, `int Y`, `int W`, `int H`, `int D=3`, `int L=0`)
see `fl_draw_image(const uchar buf, int X,int Y,int W,int H, int D, int L)`.*
- void `draw_image` (`Fl_Draw_Image_Cb cb`, `void *data`, `int X`, `int Y`, `int W`, `int H`, `int D=3`)
see `fl_draw_image(Fl_Draw_Image_Cb cb, void data, int X,int Y,int W,int H, int D)`.*
- void `draw_image_mono` (`const uchar *buf`, `int X`, `int Y`, `int W`, `int H`, `int D=1`, `int L=0`)
see `fl_draw_image_mono(const uchar buf, int X,int Y,int W,int H, int D, int L)`.*
- void `draw_image_mono` (`Fl_Draw_Image_Cb cb`, `void *data`, `int X`, `int Y`, `int W`, `int H`, `int D=1`)
see `fl_draw_image_mono(Fl_Draw_Image_Cb cb, void data, int X,int Y,int W,int H, int D)`.*
- void `font` (`Fl_Font face`, `Fl_Fontsize size`)
see `fl_font(Fl_Font face, Fl_Fontsize size)`.
- int `height` ()
see `fl_height()`.
- void `rtl_draw` (`const char *str`, `int n`, `int x`, `int y`)
*see `fl_rtl_draw(const char *str, int n, int x, int y)`.*
- void `text_extents` (`const char *`, `int n`, `int &dx`, `int &dy`, `int &w`, `int &h`)
see `fl_text_extents(const char, int n, int& dx, int& dy, int& w, int& h)`.*
- double `width` (`const char *str`, `int n`)
*see `fl_width(const char *str, int n)`.*
- double `width` (`unsigned int c`)
see `fl_width(unsigned int n)`.

Static Public Attributes

- static const char * `class_id` = "Fl_Xlib_Graphics_Driver"

Additional Inherited Members

31.157.1 Detailed Description

The Xlib-specific graphics class.

This class is implemented only on the Xlib platform.

31.157.2 Member Function Documentation

31.157.2.1 `class_name()`

```
const char * Fl_Xlib_Graphics_Driver::class_name ( ) [inline], [virtual]
```

Returns the name of the class of this object.

Use of the `class_name()` function is discouraged because it will be removed from future FLTK versions.

The class of an instance of an `Fl_Device` subclass can be checked with code such as:

```
if ( instance->class_name() == Fl_Printer::class_id ) { ... }
```

Reimplemented from `Fl_Graphics_Driver`.

31.157.2.2 color() [1/2]

```
void Fl_Xlib_Graphics_Driver::color (
    Fl_Color c ) [virtual]
```

see [fl_color\(Fl_Color c\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.157.2.3 color() [2/2]

```
void Fl_Xlib_Graphics_Driver::color (
    uchar r,
    uchar g,
    uchar b ) [virtual]
```

see [fl_color\(uchar r, uchar g, uchar b\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.157.2.4 copy_offscreen()

```
void Fl_Xlib_Graphics_Driver::copy_offscreen (
    int x,
    int y,
    int w,
    int h,
    Fl_Offscreen pixmap,
    int srcx,
    int srcy ) [virtual]
```

see [fl_copy_offscreen\(\)](#)

Reimplemented from [Fl_Graphics_Driver](#).

31.157.2.5 descent()

```
int Fl_Xlib_Graphics_Driver::descent ( ) [virtual]
```

see [fl_descent\(\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.157.2.6 draw() [1/5]

```
void Fl_Xlib_Graphics_Driver::draw (
    const char * str,
    int n,
    int x,
    int y ) [virtual]
```

see [fl_draw\(const char *str, int n, int x, int y\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.157.2.7 draw() [2/5]

```
void Fl_Xlib_Graphics_Driver::draw (
    Fl_Bitmap * bm,
    int XP,
    int YP,
    int WP,
    int HP,
    int cx,
    int cy ) [virtual]
```

Draws an [Fl_Bitmap](#) object to the device.

Specifies a bounding box for the image, with the origin (upper left-hand corner) of the image offset by the *cx* and *cy* arguments.

Reimplemented from [Fl_Graphics_Driver](#).

31.157.2.8 draw() [3/5]

```
void Fl_Xlib_Graphics_Driver::draw (
    Fl_Pixmap * pxm,
    int XP,
    int YP,
    int WP,
    int HP,
    int cx,
    int cy ) [virtual]
```

Draws an [Fl_Pixmap](#) object to the device.

Specifies a bounding box for the image, with the origin (upper left-hand corner) of the image offset by the *cx* and *cy* arguments.

Reimplemented from [Fl_Graphics_Driver](#).

31.157.2.9 draw() [4/5]

```
void Fl_Xlib_Graphics_Driver::draw (
    Fl_RGB_Image * rgb,
    int XP,
    int YP,
    int WP,
    int HP,
    int cx,
    int cy ) [virtual]
```

Draws an [Fl_RGB_Image](#) object to the device.

Specifies a bounding box for the image, with the origin (upper left-hand corner) of the image offset by the *cx* and *cy* arguments.

Reimplemented from [Fl_Graphics_Driver](#).

31.157.2.10 draw() [5/5]

```
void Fl_Xlib_Graphics_Driver::draw (
    int angle,
    const char * str,
    int n,
    int x,
    int y ) [virtual]
```

see [fl_draw\(int angle, const char *str, int n, int x, int y\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.157.2.11 draw_image() [1/2]

```
void Fl_Xlib_Graphics_Driver::draw_image (
    const uchar * buf,
    int X,
    int Y,
    int W,
    int H,
```

```
    int D = 3,  
    int L = 0 ) [virtual]
```

see [fl_draw_image\(const uchar* buf, int X,int Y,int W,int H, int D, int L\)](#).
Reimplemented from [Fl_Graphics_Driver](#).

31.157.2.12 draw_image() [2/2]

```
void Fl_Xlib_Graphics_Driver::draw_image (  
    Fl_Draw_Image_Cb cb,  
    void * data,  
    int X,  
    int Y,  
    int W,  
    int H,  
    int D = 3 ) [virtual]
```

see [fl_draw_image\(Fl_Draw_Image_Cb cb, void* data, int X,int Y,int W,int H, int D\)](#).
Reimplemented from [Fl_Graphics_Driver](#).

31.157.2.13 draw_image_mono() [1/2]

```
void Fl_Xlib_Graphics_Driver::draw_image_mono (  
    const uchar * buf,  
    int X,  
    int Y,  
    int W,  
    int H,  
    int D = 1,  
    int L = 0 ) [virtual]
```

see [fl_draw_image_mono\(const uchar* buf, int X,int Y,int W,int H, int D, int L\)](#).
Reimplemented from [Fl_Graphics_Driver](#).

31.157.2.14 draw_image_mono() [2/2]

```
void Fl_Xlib_Graphics_Driver::draw_image_mono (  
    Fl_Draw_Image_Cb cb,  
    void * data,  
    int X,  
    int Y,  
    int W,  
    int H,  
    int D = 1 ) [virtual]
```

see [fl_draw_image_mono\(Fl_Draw_Image_Cb cb, void* data, int X,int Y,int W,int H, int D\)](#).
Reimplemented from [Fl_Graphics_Driver](#).

31.157.2.15 font()

```
void Fl_Xlib_Graphics_Driver::font (  
    Fl_Font face,  
    Fl_Fontsize fsize ) [virtual]
```

see [fl_font\(Fl_Font face, Fl_Fontsize size\)](#).
Reimplemented from [Fl_Graphics_Driver](#).

31.157.2.16 height()

```
int Fl_Xlib_Graphics_Driver::height ( ) [virtual]
see fl\_height\(\).
```

Reimplemented from [Fl_Graphics_Driver](#).

31.157.2.17 rtl_draw()

```
void Fl_Xlib_Graphics_Driver::rtl_draw (
    const char * str,
    int n,
    int x,
    int y ) [virtual]
```

see [fl_rtl_draw\(const char *str, int n, int x, int y\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.157.2.18 text_extents()

```
void Fl_Xlib_Graphics_Driver::text_extents (
    const char * t,
    int n,
    int & dx,
    int & dy,
    int & w,
    int & h ) [virtual]
```

see [fl_text_extents\(const char*, int n, int& dx, int& dy, int& w, int& h\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.157.2.19 width() [1/2]

```
double Fl_Xlib_Graphics_Driver::width (
    const char * str,
    int n ) [virtual]
```

see [fl_width\(const char *str, int n\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

31.157.2.20 width() [2/2]

```
double Fl_Xlib_Graphics_Driver::width (
    unsigned int c ) [virtual]
```

see [fl_width\(unsigned int n\)](#).

Reimplemented from [Fl_Graphics_Driver](#).

The documentation for this class was generated from the following files:

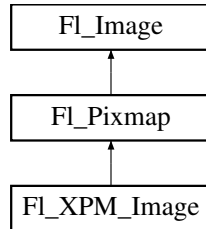
- [Fl_Device.H](#)
- [Fl_Bitmap.cxx](#)
- [fl_color.cxx](#)
- [Fl_Device.cxx](#)
- [fl_draw_image.cxx](#)
- [Fl_Image.cxx](#)
- [Fl_Pixmap.cxx](#)

31.158 Fl_XPM_Image Class Reference

The [Fl_XPM_Image](#) class supports loading, caching, and drawing of X Pixmap (XPM) images, including transparency.

```
#include <Fl_XPM_Image.H>
```

Inheritance diagram for Fl_XPM_Image:



Public Member Functions

- [Fl_XPM_Image](#) (const char *filename)
The constructor loads the XPM image from the name filename.

Additional Inherited Members

31.158.1 Detailed Description

The [Fl_XPM_Image](#) class supports loading, caching, and drawing of X Pixmap (XPM) images, including transparency.

31.158.2 Constructor & Destructor Documentation

31.158.2.1 Fl_XPM_Image()

```
Fl_XPM_Image::Fl_XPM_Image (
    const char * name )
```

The constructor loads the XPM image from the name filename.

The destructor frees all memory and server resources that are used by the image.

The documentation for this class was generated from the following files:

- Fl_XPM_Image.H
- Fl_XPM_Image.cxx

31.159 Fl_Text_Editor::Key_Binding Struct Reference

Simple linked list item associating a key/state to a function.

```
#include <Fl_Text_Editor.H>
```

Public Attributes

- [Key_Func](#) function
associated function
- int **key**
the key pressed
- [Key_Binding](#) * **next**
next key binding in the list
- int **state**
the state of key modifiers

31.159.1 Detailed Description

Simple linked list item associating a key/state to a function.

The documentation for this struct was generated from the following file:

- [FI_Text_Editor.H](#)

31.160 FI_Graphics_Driver::matrix Struct Reference

A 2D coordinate transformation matrix.

```
#include <Fl_Device.H>
```

Public Attributes

- double **a**
- double **b**
- double **c**
- double **d**
- double **x**
- double **y**

31.160.1 Detailed Description

A 2D coordinate transformation matrix.

The documentation for this struct was generated from the following file:

- [Fl_Device.H](#)

31.161 FI_Preferences::Name Class Reference

'Name' provides a simple method to create numerical or more complex procedural names for entries and groups on the fly.

```
#include <Fl_Preferences.H>
```

Public Member Functions

- [Name](#) (const char *format,...)
Creates a group name or entry name on the fly.
- [Name](#) (unsigned int n)
Creates a group name or entry name on the fly.
- **operator const char * ()**
Return the [Name](#) as a "C" string.

31.161.1 Detailed Description

'Name' provides a simple method to create numerical or more complex procedural names for entries and groups on the fly.

Example: `prefs.set(FI_Preferences::Name("File%d",i),file[i]);`

See `test/preferences.cxx` as a sample for writing arrays into preferences.

'Name' is actually implemented as a class inside [FI_Preferences](#). It casts into `const char*` and gets automatically destroyed after the enclosing call ends.

31.161.2 Constructor & Destructor Documentation

31.161.2.1 Name() [1/2]

```
Fl_Preferences::Name::Name (
    unsigned int n )
```

Creates a group name or entry name on the fly.

This version creates a simple unsigned integer as an entry name.

```
int n, i;
Fl_Preferences prev( appPrefs, "PreviousFiles" );
prev.get( "n", 0 );
for ( i=0; i<n; i++ )
    prev.get( Fl_Preferences::Name(i), prevFile[i], "" );
```

31.161.2.2 Name() [2/2]

```
Fl_Preferences::Name::Name (
    const char * format,
    ... )
```

Creates a group name or entry name on the fly.

This version creates entry names as in 'printf'.

```
int n, i;
Fl_Preferences prefs( USER, "matthiasm.com", "test" );
prev.get( "nFiles", 0 );
for ( i=0; i<n; i++ )
    prev.get( Fl_Preferences::Name( "File%d", i ), prevFile[i], "" );
```

The documentation for this class was generated from the following files:

- Fl_Preferences.H
- Fl_Preferences.cxx

31.162 Fl_Preferences::Node Class Reference**Public Member Functions**

- void **add** (const char *line)
- [Node](#) * **addChild** (const char *path)
- const char * **child** (int ix)
- [Node](#) * **childNodes** (int ix)
- void **deleteAllChildren** ()
- void **deleteAllEntries** ()
- char **deleteEntry** (const char *name)
- char **dirty** ()
- [Entry](#) & **entry** (int i)
- [Node](#) * **find** (const char *path)
- [RootNode](#) * **findRoot** ()
- const char * **get** (const char *name)
- int **getEntry** (const char *name)
- const char * **name** ()
- int **nChildren** ()
- int **nEntry** ()
- [Node](#) (const char *path)
- [Node](#) * **parent** ()
- const char * **path** ()
- char **remove** ()
- [Node](#) * **search** (const char *path, int offset=0)
- void **set** (const char *line)
- void **set** (const char *name, const char *value)
- void **setParent** ([Node](#) *parent)
- void **setRoot** ([RootNode](#) *r)
- int **write** (FILE *f)

Static Public Attributes

- static int **lastEntrySet** = -1

The documentation for this class was generated from the following files:

- FI_Preferences.H
- FI_Preferences.cxx

31.163 FI_Paged_Device::page_format Struct Reference

width, height and name of a page format

```
#include <Fl_Paged_Device.H>
```

Public Attributes

- int **height**
height in points
- const char * **name**
format name
- int **width**
width in points

31.163.1 Detailed Description

width, height and name of a page format

The documentation for this struct was generated from the following file:

- [FI_Paged_Device.H](#)

31.164 FI_Preferences::RootNode Class Reference

Public Member Functions

- char **getPath** (char *[path](#), int pathlen)
- int **read** ()
- **RootNode** ([FI_Preferences](#) *)
- **RootNode** ([FI_Preferences](#) *, const char *[path](#), const char *[vendor](#), const char *[application](#))
- **RootNode** ([FI_Preferences](#) *, [Root](#) root, const char *[vendor](#), const char *[application](#))
- int **write** ()

The documentation for this class was generated from the following files:

- FI_Preferences.H
- FI_Preferences.cxx

31.165 FI_Scroll::ScrollInfo Struct Reference

Structure to manage scrollbar and widget interior sizes.

```
#include <Fl_Scroll.H>
```

Public Attributes

- [Fl_Region_LRTB](#) **child**
child bounding box: left/right/top/bottom
- int **hneeded**
horizontal scrollbar visibility
- [Fl_Scrollbar_Data](#) **hscroll**
horizontal scrollbar region + values
- [Fl_Region_XYWH](#) **innerbox**
widget's inner box, excluding scrollbars
- [Fl_Region_XYWH](#) **innerchild**
widget's inner box, including scrollbars
- int **scrollsize**
the effective scrollbar thickness (local or global)
- int **vneeded**
vertical scrollbar visibility
- [Fl_Scrollbar_Data](#) **vscroll**
vertical scrollbar region + values

31.165.1 Detailed Description

Structure to manage scrollbar and widget interior sizes.

This is filled out by [recalc_scrollbars\(\)](#) for use in calculations that need to know the visible scroll area size, etc.

Note

Availability in FLTK_ABI_VERSION 10303 or higher.

The documentation for this struct was generated from the following file:

- [Fl_Scroll.H](#)

31.166 Fl_Window::shape_data_type Struct Reference

Data supporting a non-rectangular window shape.

```
#include <Fl_Window.H>
```

Public Attributes

- int **lh_**
height of shape image
- int **lw_**
width of shape image
- [Fl_Image](#) * **shape_**
shape image
- [Fl_Bitmap](#) * **todelete_**
auxiliary bitmap image

31.166.1 Detailed Description

Data supporting a non-rectangular window shape.

The documentation for this struct was generated from the following file:

- [Fl_Window.H](#)

31.167 FI_Text_Display::Style_Table_Entry Struct Reference

This structure associates the color, font, and font size of a string to draw with an attribute mask matching attr.
`#include <FI_Text_Display.H>`

Public Attributes

- unsigned **attr**
currently unused (this may be change in the future)
- [FI_Color](#) **color**
text color
- [FI_Font](#) **font**
text font
- [FI_Fontsize](#) **size**
text font size

31.167.1 Detailed Description

This structure associates the color, font, and font size of a string to draw with an attribute mask matching attr. There must be one entry for each style that can be used in an [FI_Text_Display](#) for displaying text. The style table is an array of struct [Style_Table_Entry](#). The style table is associated with an [FI_Text_Display](#) by using [FI_Text_Display::highlight_data\(\)](#).

See also

[FI_Text_Display::highlight_data\(\)](#)

The documentation for this struct was generated from the following file:

- [FI_Text_Display.H](#)

Chapter 32

File Documentation

32.1 abi-version.h

```
1 /* FL/abi-version.h.    Generated from abi-version.in by configure.    */
2 /*
3 ABI Configuration file for the Fast Light Tool Kit (FLTK).
4
5 =====
6 DO NOT EDIT - This file is generated by configure !
7 =====
8
9 define FL_ABI_VERSION: lxxxy for l.x.y (xx,yy with leading zero)
10 */
11
12 /* #undef FL_ABI_VERSION */
```

32.2 dirent.h

```
1 //
2 // "$Id$"
3 //
4 // Directory header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2011 by Bill Spitzak and others.
7 //
8 // This library is free software.  Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file.  If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19
20 // this file is for back-compatibility only
21 #include "filename.H"
22
23 //
24 // End of "$Id$".
25 //
```

32.3 Enumerations.H File Reference

This file contains type definitions and general enumerations.

```
#include <FL/abi-version.h>
#include "Fl_Export.H"
#include "fl_types.h"
```

Macros

Mouse and Keyboard Events

This and the following constants define the non-ASCII keys on the keyboard for `FL_KEYBOARD` and `FL_SHORTCUT` events.

```
\todo      FL_Button and FL_key... constants could be structured better
          (use an enum or some doxygen grouping ?)
```

```
\sa Fl::event_key() and Fl::get_key(int) (use ascii letters for all other keys):
```

- **#define FL_Alt_L** 0xffe9
The left alt key.
- **#define FL_Alt_R** 0xffea
The right alt key.
- **#define FL_Back** 0xEF26 /* Like back on a browser */
- **#define FL_BackSpace** 0xff08
The backspace key.
- **#define FL_Button** 0xffe8
A mouse button; use FL_Button + n for mouse button n.
- **#define FL_Caps_Lock** 0xffe5
The caps lock key.
- **#define FL_Control_L** 0xffe3
The lefthand control key.
- **#define FL_Control_R** 0xffe4
The righthand control key.
- **#define FL_Delete** 0xffff
The delete key.
- **#define FL_Down** 0xff54
The down arrow key.
- **#define FL_Eisu** 0xff2f
The Eisu key of JIS keyboards.
- **#define FL_End** 0xff57
The end key.
- **#define FL_Enter** 0xff0d
The enter key.
- **#define FL_Escape** 0xff1b
The escape key.
- **#define FL_F** 0xffbd
One of the function keys; use FL_F + n for function key n.
- **#define FL_F_Last** 0xffe0
The last function key; use to range-check function keys.
- **#define FL_Favorites** 0xEF30 /* Show favorite locations */
- **#define FL_Forward** 0xEF27 /* Like forward on a browser */
- **#define FL_Help** 0xff68
The 'help' key on Mac keyboards.
- **#define FL_Home** 0xff50
The home key.
- **#define FL_Home_Page** 0xEF18 /* Display user's home page */
- **#define FL_Insert** 0xff63
The insert key.
- **#define FL_Iso_Key** 0xff0c
The additional key of ISO keyboards.
- **#define FL_JIS_Underscore** 0xff31
The underscore key of JIS keyboards.
- **#define FL_Kana** 0xff2e
The Kana key of JIS keyboards.
- **#define FL_KP** 0xff80
One of the keypad numbers; use FL_KP + 'n' for digit n.
- **#define FL_KP_Enter** 0xff8d
The enter key on the keypad, same as FL_KP+'r'.
- **#define FL_KP_Last** 0xffbd
The last keypad key; use to range-check keypad.
- **#define FL_Left** 0xff51

- The left arrow key.*
- #define **FL_Mail** 0xEF19 /* Invoke user's mail program */
- #define **FL_Media_Next** 0xEF17 /* Next track */
- #define **FL_Media_Play** 0xEF14 /* Start playing of audio */
- #define **FL_Media_Prev** 0xEF16 /* Previous track */
- #define **FL_Media_Stop** 0xEF15 /* Stop playing audio */
- #define **FL_Menu** 0xff67
- The menu key.*
- #define **FL_Meta_L** 0xfe7
- The left meta/Windows key.*
- #define **FL_Meta_R** 0xfe8
- The right meta/Windows key.*
- #define **FL_Num_Lock** 0xff7f
- The num lock key.*
- #define **FL_Page_Down** 0xff56
- The page-down key.*
- #define **FL_Page_Up** 0xff55
- The page-up key.*
- #define **FL_Pause** 0xff13
- The pause key.*
- #define **FL_Print** 0xff61
- The print (or print-screen) key.*
- #define **FL_Refresh** 0xEF29 /* Refresh the page */
- #define **FL_Right** 0xff53
- The right arrow key.*
- #define **FL_Scroll_Lock** 0xff14
- The scroll lock key.*
- #define **FL_Search** 0xEF1B /* Search */
- #define **FL_Shift_L** 0xfe1
- The lefthand shift key.*
- #define **FL_Shift_R** 0xfe2
- The righthand shift key.*
- #define **FL_Sleep** 0xEF2F /* Put system to sleep */
- #define **FL_Stop** 0xEF28 /* Stop current operation */
- #define **FL_Tab** 0xff09
- The tab key.*
- #define **FL_Up** 0xff52
- The up arrow key.*
- #define **FL_Volume_Down** 0xEF11 /* Volume control down */
- #define **FL_Volume_Mute** 0xEF12 /* Mute sound from the system */
- #define **FL_Volume_Up** 0xEF13 /* Volume control up */
- #define **FL_Yen** 0xff30
- The Yen key of JIS keyboards.*

Mouse Buttons

These constants define the button numbers for FL_PUSH and FL_RELEASE events.

```
\sa Fl::event_button()
```

- #define **FL_LEFT_MOUSE** 1
- The left mouse button.*
- #define **FL_MIDDLE_MOUSE** 2
- The middle mouse button.*
- #define **FL_RIGHT_MOUSE** 3
- The right mouse button.*

Event States

The following constants define bits in the Fl::event_state() value.

- #define **FL_ALT** 0x00080000

- *One of the alt keys is down.*
- #define **FL_BUTTON**(n) (0x00800000<<(n))
Mouse button n (n > 0) is pushed.
- #define **FL_BUTTON1** 0x01000000
Mouse button 1 is pushed.
- #define **FL_BUTTON2** 0x02000000
Mouse button 2 is pushed.
- #define **FL_BUTTON3** 0x04000000
Mouse button 3 is pushed.
- #define **FL_BUTTONS** 0x7f000000
Any mouse button is pushed.
- #define **FL_CAPS_LOCK** 0x00020000
The caps lock is on.
- #define **FL_COMMAND** **FL_CTRL**
An alias for FL_CTRL on WIN32 and X11, or FL_META on MacOS X.
- #define **FL_CONTROL** **FL_META**
An alias for FL_META on WIN32 and X11, or FL_CTRL on MacOS X.
- #define **FL_CTRL** 0x00040000
One of the ctrl keys is down.
- #define **FL_KEY_MASK** 0x0000ffff
All keys are 16 bit for now.
- #define **FL_META** 0x00400000
One of the meta/Windows keys is down.
- #define **FL_NUM_LOCK** 0x00100000
The num lock is on.
- #define **FL_SCROLL_LOCK** 0x00800000
The scroll lock is on.
- #define **FL_SHIFT** 0x00010000
One of the shift keys is down.

Enumerations

When Conditions

- enum **Fl_When** {
FL_WHEN_NEVER = 0 , **FL_WHEN_CHANGED** = 1 , **FL_WHEN_NOT_CHANGED** = 2 , **FL_WHEN_RELEASE** = 4 ,
FL_WHEN_RELEASE_ALWAYS = 6 , **FL_WHEN_ENTER_KEY** = 8 , **FL_WHEN_ENTER_KEY_ALWAYS** = 10 , **FL_WHEN_ENTER_KEY_CHANGED** = 11 }
These constants determine when a callback is performed.

Version Numbers

FLTK defines some constants to help the programmer to find out, for which FLTK version a program is compiled. The following constants are defined:

- #define **FL_ABI_VERSION** **FL_API_VERSION**
The FLTK ABI (Application Binary Interface) version number as an int.
- #define **FL_API_VERSION** (**FL_MAJOR_VERSION***10000 + **FL_MINOR_VERSION***100 + **FL_PATCH_VERSION**)
The FLTK API version number as an int.
- enum **Fl_Event** {
FL_NO_EVENT = 0 , **FL_PUSH** = 1 , **FL_RELEASE** = 2 , **FL_ENTER** = 3 ,
FL_LEAVE = 4 , **FL_DRAG** = 5 , **FL_FOCUS** = 6 , **FL_UNFOCUS** = 7 ,
FL_KEYDOWN = 8 , **FL_KEYBOARD** = 8 , **FL_KEYUP** = 9 , **FL_CLOSE** = 10 ,
FL_MOVE = 11 , **FL_SHORTCUT** = 12 , **FL_DEACTIVATE** = 13 , **FL_ACTIVATE** = 14 ,
FL_HIDE = 15 , **FL_SHOW** = 16 , **FL_PASTE** = 17 , **FL_SELECTIONCLEAR** = 18 ,
FL_MOUSEWHEEL = 19 , **FL_DND_ENTER** = 20 , **FL_DND_DRAG** = 21 , **FL_DND_LEAVE** = 22 ,
FL_DND_RELEASE = 23 , **FL_SCREEN_CONFIGURATION_CHANGED** = 24 , **FL_FULLSCREEN** = 25 ,
FL_ZOOM_GESTURE = 26 }

Every time a user moves the mouse pointer, clicks a button, or presses a key, an event is generated and sent to your application.

- #define `FL_MAJOR_VERSION` 1
The major release version of this FLTK library.
- #define `FL_MINOR_VERSION` 3
The minor release version for this library.
- #define `FL_PATCH_VERSION` 8
The patch version for this library.
- #define `FL_VERSION`
The FLTK version number as a double.
- #define `FLTK_ABI_VERSION` `FL_ABI_VERSION`

Box Types

FLTK standard box types

This enum defines the standard box types included with FLTK.

`FL_NO_BOX` means nothing is drawn at all, so whatever is already on the screen remains. The `FL_..._FRAME` types only draw their edges, leaving the interior unchanged. The blue color in Figure 1 is the area that is not drawn by the frame types.

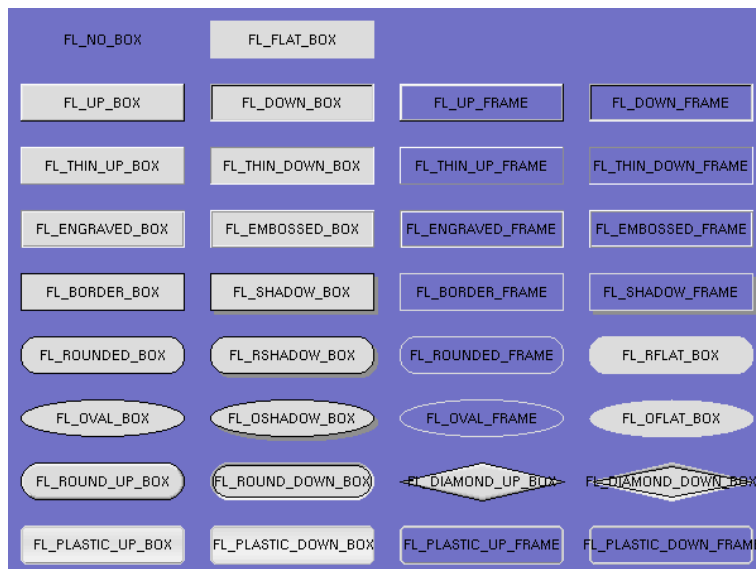


Figure 32.1 FLTK standard box types

Todo Description of boxtypes is incomplete. See below for the defined enum `Fl_Boxtype`.

See also

`src/Fl_get_system_colors.cxx`

- `Fl_Boxtype fl_box` (`Fl_Boxtype` b)
Get the filled version of a frame.
- enum `Fl_Boxtype` {
`FL_NO_BOX = 0`, `FL_FLAT_BOX`, `FL_UP_BOX`, `FL_DOWN_BOX`,
`FL_UP_FRAME`, `FL_DOWN_FRAME`, `FL_THIN_UP_BOX`, `FL_THIN_DOWN_BOX`,
`FL_THIN_UP_FRAME`, `FL_THIN_DOWN_FRAME`, `FL_ENGRAVED_BOX`, `FL_EMBOSSED_BOX`,
`FL_ENGRAVED_FRAME`, `FL_EMBOSSED_FRAME`, `FL_BORDER_BOX`, `FL_SHADOW_BOX`,
`FL_BORDER_FRAME`, `FL_SHADOW_FRAME`, `FL_ROUNDED_BOX`, `FL_RSHADOW_BOX`,
`FL_ROUNDED_FRAME`, `FL_RFLAT_BOX`, `FL_ROUND_UP_BOX`, `FL_ROUND_DOWN_BOX`,
`FL_DIAMOND_UP_BOX`, `FL_DIAMOND_DOWN_BOX`, `FL_OVAL_BOX`, `FL_OSHADOW_BOX`,
`FL_OVAL_FRAME`, `FL_OFLAT_BOX`, `FL_PLASTIC_UP_BOX`, `FL_PLASTIC_DOWN_BOX`,
`FL_PLASTIC_UP_FRAME`, `FL_PLASTIC_DOWN_FRAME`,

```

_FL_PLASTIC_UP_FRAME , _FL_PLASTIC_DOWN_FRAME , _FL_PLASTIC_THIN_UP_BOX ,
_FL_PLASTIC_THIN_DOWN_BOX ,
_FL_PLASTIC_ROUND_UP_BOX , _FL_PLASTIC_ROUND_DOWN_BOX , _FL_GTK_UP_BOX ,
_FL_GTK_DOWN_BOX ,
_FL_GTK_UP_FRAME , _FL_GTK_DOWN_FRAME , _FL_GTK_THIN_UP_BOX , _FL_GTK_THIN_DOWN_BOX
,
_FL_GTK_THIN_UP_FRAME , _FL_GTK_THIN_DOWN_FRAME , _FL_GTK_ROUND_UP_BOX ,
_FL_GTK_ROUND_DOWN_BOX ,
_FL_GLEAM_UP_BOX , _FL_GLEAM_DOWN_BOX , _FL_GLEAM_UP_FRAME , _FL_GLEAM_DOWN_FRAME
,
_FL_GLEAM_THIN_UP_BOX , _FL_GLEAM_THIN_DOWN_BOX , _FL_GLEAM_ROUND_UP_BOX ,
_FL_GLEAM_ROUND_DOWN_BOX ,
FL_FREE_BOXTYPE }
• #define FL_CIRCLE_BOX FL_ROUND_DOWN_BOX
• FL_EXPORT FI_Boxtype fl_define_FL_DIAMOND_BOX ()
• FI_Labeltype FL_EXPORT fl_define_FL_EMBOSSSED_LABEL ()
• FI_Labeltype FL_EXPORT fl_define_FL_ENGRAVED_LABEL ()
• FL_EXPORT FI_Boxtype fl_define_FL_GLEAM_UP_BOX ()
• FL_EXPORT FI_Boxtype fl_define_FL_GTK_UP_BOX ()
• FL_EXPORT FI_Boxtype fl_define_FL_OVAL_BOX ()
• FL_EXPORT FI_Boxtype fl_define_FL_PLASTIC_UP_BOX ()
• FL_EXPORT FI_Boxtype fl_define_FL_RFLAT_BOX ()
• FL_EXPORT FI_Boxtype fl_define_FL_ROUND_UP_BOX ()
• FL_EXPORT FI_Boxtype fl_define_FL_ROUNDED_BOX ()
• FL_EXPORT FI_Boxtype fl_define_FL_RSHADOW_BOX ()
• FL_EXPORT FI_Boxtype fl_define_FL_SHADOW_BOX ()
• FI_Labeltype FL_EXPORT fl_define_FL_SHADOW_LABEL ()
• #define FL_DIAMOND_BOX FL_DIAMOND_DOWN_BOX
• #define FL_DIAMOND_DOWN_BOX (FI_Boxtype)(fl_define_FL_DIAMOND_BOX()+1)
• #define FL_DIAMOND_UP_BOX fl_define_FL_DIAMOND_BOX()
• FI_Boxtype fl_down (FI_Boxtype b)
    Get the "pressed" or "down" version of a box.
• #define FL_EMBOSSSED_LABEL fl_define_FL_EMBOSSSED_LABEL()
• #define FL_ENGRAVED_LABEL fl_define_FL_ENGRAVED_LABEL()
• FI_Boxtype fl_frame (FI_Boxtype b)
    Get the unfilled, frame only version of a box.
• #define FL_FRAME FL_ENGRAVED_FRAME
• #define FL_FRAME_BOX FL_ENGRAVED_BOX
• #define FL_GLEAM_DOWN_BOX (FI_Boxtype)(fl_define_FL_GLEAM_UP_BOX()+1)
• #define FL_GLEAM_DOWN_FRAME (FI_Boxtype)(fl_define_FL_GLEAM_UP_BOX()+3)
• #define FL_GLEAM_ROUND_DOWN_BOX (FI_Boxtype)(fl_define_FL_GLEAM_UP_BOX()+7)
• #define FL_GLEAM_ROUND_UP_BOX (FI_Boxtype)(fl_define_FL_GLEAM_UP_BOX()+6)
• #define FL_GLEAM_THIN_DOWN_BOX (FI_Boxtype)(fl_define_FL_GLEAM_UP_BOX()+5)
• #define FL_GLEAM_THIN_UP_BOX (FI_Boxtype)(fl_define_FL_GLEAM_UP_BOX()+4)
• #define FL_GLEAM_UP_BOX fl_define_FL_GLEAM_UP_BOX()
• #define FL_GLEAM_UP_FRAME (FI_Boxtype)(fl_define_FL_GLEAM_UP_BOX()+2)
• #define FL_GTK_DOWN_BOX (FI_Boxtype)(fl_define_FL_GTK_UP_BOX()+1)
• #define FL_GTK_DOWN_FRAME (FI_Boxtype)(fl_define_FL_GTK_UP_BOX()+3)
• #define FL_GTK_ROUND_DOWN_BOX (FI_Boxtype)(fl_define_FL_GTK_UP_BOX()+9)
• #define FL_GTK_ROUND_UP_BOX (FI_Boxtype)(fl_define_FL_GTK_UP_BOX()+8)
• #define FL_GTK_THIN_DOWN_BOX (FI_Boxtype)(fl_define_FL_GTK_UP_BOX()+5)
• #define FL_GTK_THIN_DOWN_FRAME (FI_Boxtype)(fl_define_FL_GTK_UP_BOX()+7)
• #define FL_GTK_THIN_UP_BOX (FI_Boxtype)(fl_define_FL_GTK_UP_BOX()+4)
• #define FL_GTK_THIN_UP_FRAME (FI_Boxtype)(fl_define_FL_GTK_UP_BOX()+6)
• #define FL_GTK_UP_BOX fl_define_FL_GTK_UP_BOX()

```

- #define **FL_GTK_UP_FRAME** (*FI_Boxtype*)(fl_define_FL_GTK_UP_BOX()+2)
 - enum *FI_Labeltype* {
FL_NORMAL_LABEL = 0, **FL_NO_LABEL**, **_FL_SHADOW_LABEL**, **_FL_ENGRAVED_LABEL**,
_FL_EMBOSSSED_LABEL, **_FL_MULTI_LABEL**, **_FL_ICON_LABEL**, **_FL_IMAGE_LABEL**,
FL_FREE_LABELTYPE }
- The labeltype() method sets the type of the label.*
- #define **FL_OFLAT_BOX** (*FI_Boxtype*)(fl_define_FL_OVAL_BOX()+3)
 - #define **FL_OSHADOW_BOX** (*FI_Boxtype*)(fl_define_FL_OVAL_BOX()+1)
 - #define **FL_OVAL_BOX** fl_define_FL_OVAL_BOX()
 - #define **FL_OVAL_FRAME** (*FI_Boxtype*)(fl_define_FL_OVAL_BOX()+2)
 - #define **FL_PLASTIC_DOWN_BOX** (*FI_Boxtype*)(fl_define_FL_PLASTIC_UP_BOX()+1)
 - #define **FL_PLASTIC_DOWN_FRAME** (*FI_Boxtype*)(fl_define_FL_PLASTIC_UP_BOX()+3)
 - #define **FL_PLASTIC_ROUND_DOWN_BOX** (*FI_Boxtype*)(fl_define_FL_PLASTIC_UP_BOX()+7)
 - #define **FL_PLASTIC_ROUND_UP_BOX** (*FI_Boxtype*)(fl_define_FL_PLASTIC_UP_BOX()+6)
 - #define **FL_PLASTIC_THIN_DOWN_BOX** (*FI_Boxtype*)(fl_define_FL_PLASTIC_UP_BOX()+5)
 - #define **FL_PLASTIC_THIN_UP_BOX** (*FI_Boxtype*)(fl_define_FL_PLASTIC_UP_BOX()+4)
 - #define **FL_PLASTIC_UP_BOX** fl_define_FL_PLASTIC_UP_BOX()
 - #define **FL_PLASTIC_UP_FRAME** (*FI_Boxtype*)(fl_define_FL_PLASTIC_UP_BOX()+2)
 - #define **FL_RFLAT_BOX** fl_define_FL_RFLAT_BOX()
 - #define **FL_ROUND_DOWN_BOX** (*FI_Boxtype*)(fl_define_FL_ROUND_UP_BOX()+1)
 - #define **FL_ROUND_UP_BOX** fl_define_FL_ROUND_UP_BOX()
 - #define **FL_ROUNDED_BOX** fl_define_FL_ROUNDED_BOX()
 - #define **FL_ROUNDED_FRAME** (*FI_Boxtype*)(fl_define_FL_ROUNDED_BOX()+2)
 - #define **FL_RSHADOW_BOX** fl_define_FL_RSHADOW_BOX()
 - #define **FL_SHADOW_BOX** fl_define_FL_SHADOW_BOX()
 - #define **FL_SHADOW_FRAME** (*FI_Boxtype*)(fl_define_FL_SHADOW_BOX()+2)
 - #define **FL_SHADOW_LABEL** fl_define_FL_SHADOW_LABEL()
 - #define **FL_SYMBOL_LABEL** **FL_NORMAL_LABEL**

Sets the current label type and return its corresponding FI_Labeltype value.

Colors

The *FI_Color* type holds an FLTK color value.

Colors are either 8-bit indexes into a [virtual colormap](#) or 24-bit RGB color values. (See [Colors](#) for the default FLTK colormap)

Color indices occupy the lower 8 bits of the value, while RGB colors occupy the upper 24 bits, for a byte organization of RGBI.

```

FI_Color => 0xrrggbbii
           | | | |
           | | | +--- index between 0 and 255
           | | +----- blue color component (8 bit)
           | +----- green component (8 bit)
           +----- red component (8 bit)

```

A color can have either an index or an rgb value. Colors with rgb set and an index >0 are reserved for special use.

- const *FI_Color* **FL_BACKGROUND2_COLOR** = 7
the default background color for text, list, and valuator widgets
- const *FI_Color* **FL_BACKGROUND_COLOR** = 49
- const *FI_Color* **FL_BLACK** = 56
- const *FI_Color* **FL_BLUE** = 216
- typedef unsigned int *FI_Color*
An FLTK color value; see also Colors
- FL_EXPORT *FI_Color* fl_color_average (*FI_Color* c1, *FI_Color* c2, float weight)

- Returns the weighted average color between the two given colors.*
- #define **FL_COLOR_CUBE** (**FI_Color**)56
- **FI_Color** **fl_color_cube** (int r, int g, int b)
 - Returns a color out of the color cube.*
- FL_EXPORT **FI_Color** **fl_contrast** (**FI_Color** fg, **FI_Color** bg)
 - Returns a color that contrasts with the background color.*
- const **FI_Color** **FL_CYAN** = 223
- const **FI_Color** **FL_DARK1** = 47
- const **FI_Color** **FL_DARK2** = 45
- const **FI_Color** **FL_DARK3** = 39
- const **FI_Color** **FL_DARK_BLUE** = 136
- const **FI_Color** **FL_DARK_CYAN** = 140
- const **FI_Color** **FL_DARK_GREEN** = 60
- const **FI_Color** **FL_DARK_MAGENTA** = 152
- const **FI_Color** **FL_DARK_RED** = 72
- const **FI_Color** **FL_DARK_YELLOW** = 76
- **FI_Color** **fl_darker** (**FI_Color** c)
 - Returns a darker version of the specified color.*
- const **FI_Color** **FL_FOREGROUND_COLOR** = 0
 - the default foreground color (0) used for labels and text*
- #define **FL_FREE_COLOR** (**FI_Color**)16
- #define **FL_GRAY** FL_BACKGROUND_COLOR
- const **FI_Color** **FL_GRAY0** = 32
- #define **FL_GRAY_RAMP** (**FI_Color**)32
- **FI_Color** **fl_gray_ramp** (int i)
 - Returns a gray color value from black (i == 0) to white (i == FL_NUM_GRAY - 1).*
- const **FI_Color** **FL_GREEN** = 63
- FL_EXPORT **FI_Color** **fl_inactive** (**FI_Color** c)
 - Returns the inactive, dimmed version of the given color.*
- const **FI_Color** **FL_INACTIVE_COLOR** = 8
 - the inactive foreground color*
- const **FI_Color** **FL_LIGHT1** = 50
- const **FI_Color** **FL_LIGHT2** = 52
- const **FI_Color** **FL_LIGHT3** = 54
- **FI_Color** **fl_lighter** (**FI_Color** c)
 - Returns a lighter version of the specified color.*
- const **FI_Color** **FL_MAGENTA** = 248
- #define **FL_NUM_BLUE** 5
- #define **FL_NUM_FREE_COLOR** 16
- #define **FL_NUM_GRAY** 24
- #define **FL_NUM_GREEN** 8
- #define **FL_NUM_RED** 5
- const **FI_Color** **FL_RED** = 88
- **FI_Color** **fl_rgb_color** (uchar g)
 - Returns the 24-bit color value closest to g (grayscale).*
- **FI_Color** **fl_rgb_color** (uchar r, uchar g, uchar b)
 - Returns the 24-bit color value closest to r, g, b.*
- const **FI_Color** **FL_SELECTION_COLOR** = 15
 - the default selection/highlight color*
- const **FI_Color** **FL_WHITE** = 255
- const **FI_Color** **FL_YELLOW** = 95

Cursors

- enum { [FL_READ](#) = 1 , [FL_WRITE](#) = 4 , [FL_EXCEPT](#) = 8 }
FD "when" conditions.
- enum [Fl_Cursor](#) {
[FL_CURSOR_DEFAULT](#) = 0 , [FL_CURSOR_ARROW](#) = 35 , [FL_CURSOR_CROSS](#) = 66 , [FL_CURSOR_WAIT](#) = 76 ,
[FL_CURSOR_INSERT](#) = 77 , [FL_CURSOR_HAND](#) = 31 , [FL_CURSOR_HELP](#) = 47 , [FL_CURSOR_MOVE](#) = 27 ,
[FL_CURSOR_NS](#) = 78 , [FL_CURSOR_WE](#) = 79 , [FL_CURSOR_NWSE](#) = 80 , [FL_CURSOR_NESW](#) = 81 ,
[FL_CURSOR_N](#) = 70 , [FL_CURSOR_NE](#) = 69 , [FL_CURSOR_E](#) = 49 , [FL_CURSOR_SE](#) = 8 ,
[FL_CURSOR_S](#) = 9 , [FL_CURSOR_SW](#) = 7 , [FL_CURSOR_W](#) = 36 , [FL_CURSOR_NW](#) = 68 ,
[FL_CURSOR_NONE](#) = 255 }
The following constants define the mouse cursors that are available in FLTK.
- enum [Fl_Damage](#) {
[FL_DAMAGE_CHILD](#) = 0x01 , [FL_DAMAGE_EXPOSE](#) = 0x02 , [FL_DAMAGE_SCROLL](#) = 0x04 ,
[FL_DAMAGE_OVERLAY](#) = 0x08 ,
[FL_DAMAGE_USER1](#) = 0x10 , [FL_DAMAGE_USER2](#) = 0x20 , [FL_DAMAGE_ALL](#) = 0x80 }
Damage masks.
- #define [FL_IMAGE_WITH_ALPHA](#) 0x40000000
- enum [Fl_Mode](#) {
[FL_RGB](#) = 0 , [FL_INDEX](#) = 1 , [FL_SINGLE](#) = 0 , [FL_DOUBLE](#) = 2 ,
[FL_ACCUM](#) = 4 , [FL_ALPHA](#) = 8 , [FL_DEPTH](#) = 16 , [FL_STENCIL](#) = 32 ,
[FL_RGB8](#) = 64 , [FL_MULTISAMPLE](#) = 128 , [FL_STEREO](#) = 256 , [FL_FAKE_SINGLE](#) = 512 ,
[FL_OPENGL3](#) = 1024 }
visual types and `Fl_Gl_Window::mode()` (values match Glut)

Alignment Flags

Flags to control the label alignment.

This controls how the label is displayed next to or inside the widget. The default value is [FL_ALIGN_CENTER](#) (0) for most widgets, which centers the label inside the widget.

Flags can be or'd to achieve a combination of alignments, but there are some "magic values" (e.g. combinations of TOP and BOTTOM and of LEFT and RIGHT) that have special meanings (see below). For instance:

[FL_ALIGN_TOP_LEFT](#) == ([FL_ALIGN_TOP](#) | [FL_ALIGN_LEFT](#)) != [FL_ALIGN_LEFT_TOP](#).

Outside alignments ([FL_ALIGN_INSIDE](#) is not set):

```

      TOP_LEFT      TOP      TOP_RIGHT
+-----+
LEFT_TOP |                | RIGHT_TOP
|                |                |
LEFT |                CENTER | RIGHT
|                |                |
LEFT_BOTTOM |                | RIGHT_BOTTOM
+-----+
BOTTOM_LEFT  BOTTOM  BOTTOM_RIGHT

```

Inside alignments ([FL_ALIGN_INSIDE](#) is set):

```

+-----+
| TOP_LEFT      TOP      TOP_RIGHT |
|                |                |
| LEFT          CENTER    RIGHT    |
|                |                |
| BOTTOM_LEFT  BOTTOM  BOTTOM_RIGHT |
+-----+

```

See also

[FL_ALIGN_CENTER](#), etc.

- typedef unsigned [Fl_Align](#)
FLTK type for alignment control.
- const [Fl_Align](#) [FL_ALIGN_BOTTOM](#) = ([Fl_Align](#))2
Align the label at the bottom of the widget.
- const [Fl_Align](#) [FL_ALIGN_BOTTOM_LEFT](#) = [FL_ALIGN_BOTTOM](#) | [FL_ALIGN_LEFT](#)
- const [Fl_Align](#) [FL_ALIGN_BOTTOM_RIGHT](#) = [FL_ALIGN_BOTTOM](#) | [FL_ALIGN_RIGHT](#)

- const `Fl_Align FL_ALIGN_CENTER` = `(Fl_Align)0`
Align the label horizontally in the middle.
- const `Fl_Align FL_ALIGN_CLIP` = `(Fl_Align)64`
All parts of the label that are larger than the widget will not be drawn .
- const `Fl_Align FL_ALIGN_IMAGE_BACKDROP` = `(Fl_Align)0x0200`
If the label contains an image, draw the image or deimage in the background.
- const `Fl_Align FL_ALIGN_IMAGE_MASK` = `0x0320`
- const `Fl_Align FL_ALIGN_IMAGE_NEXT_TO_TEXT` = `(Fl_Align)0x0100`
If the label contains an image, draw the text to the right of the image.
- const `Fl_Align FL_ALIGN_IMAGE_OVER_TEXT` = `(Fl_Align)0x0000`
If the label contains an image, draw the text below the image.
- const `Fl_Align FL_ALIGN_INSIDE` = `(Fl_Align)16`
Draw the label inside of the widget.
- const `Fl_Align FL_ALIGN_LEFT` = `(Fl_Align)4`
Align the label at the left of the widget.
- const `Fl_Align FL_ALIGN_LEFT_BOTTOM` = `0x000d`
- const `Fl_Align FL_ALIGN_LEFT_TOP` = `0x0007`
- const `Fl_Align FL_ALIGN_NOWRAP` = `(Fl_Align)0`
- const `Fl_Align FL_ALIGN_POSITION_MASK` = `0x000f`
- const `Fl_Align FL_ALIGN_RIGHT` = `(Fl_Align)8`
Align the label to the right of the widget.
- const `Fl_Align FL_ALIGN_RIGHT_BOTTOM` = `0x000e`
- const `Fl_Align FL_ALIGN_RIGHT_TOP` = `0x000b`
- const `Fl_Align FL_ALIGN_TEXT_NEXT_TO_IMAGE` = `(Fl_Align)0x0120`
If the label contains an image, draw the text to the left of the image.
- const `Fl_Align FL_ALIGN_TEXT_OVER_IMAGE` = `(Fl_Align)0x0020`
If the label contains an image, draw the text on top of the image.
- const `Fl_Align FL_ALIGN_TOP` = `(Fl_Align)1`
Align the label at the top of the widget.
- const `Fl_Align FL_ALIGN_TOP_LEFT` = `FL_ALIGN_TOP | FL_ALIGN_LEFT`
- const `Fl_Align FL_ALIGN_TOP_RIGHT` = `FL_ALIGN_TOP | FL_ALIGN_RIGHT`
- const `Fl_Align FL_ALIGN_WRAP` = `(Fl_Align)128`
Wrap text that does not fit the width of the widget.

Font Numbers

The following constants define the standard FLTK fonts:

- const `Fl_Font FL_BOLD` = 1
add this to helvetica, courier, or times
- const `Fl_Font FL_BOLD_ITALIC` = 3
add this to helvetica, courier, or times
- const `Fl_Font FL_COURIER` = 4
Courier normal.
- const `Fl_Font FL_COURIER_BOLD` = 5
Courier bold.
- const `Fl_Font FL_COURIER_BOLD_ITALIC` = 7
Courier bold-italic.
- const `Fl_Font FL_COURIER_ITALIC` = 6
Courier italic.
- typedef int `Fl_Font`
A font number is an index into the internal font table.

- typedef int [Fl_Fontsize](#)
Size of a font in pixels.
- const [Fl_Font](#) **FL_FREE_FONT** = 16
first one to allocate
- const [Fl_Font](#) **FL_HELVETICA** = 0
Helvetica (or Arial) normal (0)
- const [Fl_Font](#) **FL_HELVETICA_BOLD** = 1
Helvetica (or Arial) bold.
- const [Fl_Font](#) **FL_HELVETICA_BOLD_ITALIC** = 3
Helvetica (or Arial) bold-oblique.
- const [Fl_Font](#) **FL_HELVETICA_ITALIC** = 2
Helvetica (or Arial) oblique.
- const [Fl_Font](#) **FL_ITALIC** = 2
add this to helvetica, courier, or times
- `FL_EXPORT` [Fl_Fontsize](#) **FL_NORMAL_SIZE**
normal font size
- const [Fl_Font](#) **FL_SCREEN** = 13
Default monospaced screen font.
- const [Fl_Font](#) **FL_SCREEN_BOLD** = 14
Default monospaced bold screen font.
- const [Fl_Font](#) **FL_SYMBOL** = 12
Standard symbol font.
- const [Fl_Font](#) **FL_TIMES** = 8
Times roman.
- const [Fl_Font](#) **FL_TIMES_BOLD** = 9
Times roman bold.
- const [Fl_Font](#) **FL_TIMES_BOLD_ITALIC** = 11
Times roman bold-italic.
- const [Fl_Font](#) **FL_TIMES_ITALIC** = 10
Times roman italic.
- const [Fl_Font](#) **FL_ZAPF_DINGBATS** = 15
Zapf-dingbats font.

32.3.1 Detailed Description

This file contains type definitions and general enumerations.

32.3.2 Macro Definition Documentation

32.3.2.1 FL_ABI_VERSION

```
#define FL_ABI_VERSION FL_API_VERSION
```

The FLTK ABI (Application Binary Interface) version number as an *int*.

`FL_ABI_VERSION` is an *int* that describes the major, minor, and patch ABI version numbers in the same format as `FL_API_VERSION`.

The ABI version number `FL_ABI_VERSION` is usually the same as the API version `FL_API_VERSION` with the last two digits set to '00'.

FLTK retains the ABI (Application Binary Interface) during patch releases of the same major and minor versions.

Examples:

FLTK Version	<code>FL_API_VERSION</code>	<code>FL_ABI_VERSION</code>	<code>FL_VERSION</code> (deprecated)
1.3.0	10300	10300	1.0300
1.3.4	10304	10300	1.0304

Version 1.2.3 is actually stored as 10203 to allow for more than 9 minor and patch releases.

The `FL_MAJOR_VERSION`, `FL_MINOR_VERSION`, and `FL_PATCH_VERSION` constants give the integral values for the major, minor, and patch releases respectively.

To enable new ABI-breaking features in patch releases you can configure FLTK to use a higher `FL_ABI_VERSION`.

See also

`README.abi-version.txt`

32.3.2.2 FL_API_VERSION

```
#define FL_API_VERSION (FL_MAJOR_VERSION*10000 + FL_MINOR_VERSION*100 + FL_PATCH_VERSION)
```

The FLTK API version number as an *int*.

`FL_API_VERSION` is an *int* that describes the major, minor, and patch version numbers.

Version 1.2.3 is actually stored as 10203 to allow for more than 9 minor and patch releases.

The `FL_MAJOR_VERSION`, `FL_MINOR_VERSION`, and `FL_PATCH_VERSION` constants give the integral values for the major, minor, and patch releases respectively.

Note

`FL_API_VERSION` is intended to replace the deprecated *double* `FL_VERSION`.

See also

[Fl::api_version\(\)](#)

32.3.2.3 FL_MAJOR_VERSION

```
#define FL_MAJOR_VERSION 1
```

The major release version of this FLTK library.

See also

[FL_VERSION](#)

32.3.2.4 FL_MINOR_VERSION

```
#define FL_MINOR_VERSION 3
```

The minor release version for this library.

FLTK remains mostly source-code compatible between minor version changes.

32.3.2.5 FL_PATCH_VERSION

```
#define FL_PATCH_VERSION 8
```

The patch version for this library.

FLTK remains binary compatible between patches.

32.3.2.6 FL_VERSION

```
#define FL_VERSION
```

Value:

```
( (double)FL_MAJOR_VERSION + \
(double)FL_MINOR_VERSION * 0.01 + \
(double)FL_PATCH_VERSION * 0.0001 )
```

The FLTK version number as a *double*.

`FL_VERSION` is a *double* that describes the major, minor, and patch version numbers.

Version 1.2.3 is actually stored as 1.0203 to allow for more than 9 minor and patch releases.

Deprecated This `double` version number is retained for compatibility with existing program code. New code should use `int FL_API_VERSION` instead. `FL_VERSION` is deprecated because comparisons of floating point values may fail due to rounding errors. However, there are currently no plans to remove this deprecated constant.

`FL_VERSION` is equivalent to *(double)FL_API_VERSION / 10000*.

See also

[Fl::version\(\)](#) (deprecated as well)

[FL_API_VERSION](#)

[Fl::api_version\(\)](#)

32.3.3 Typedef Documentation

32.3.3.1 Fl_Fontsize

```
typedef int Fl_Fontsize
```

Size of a font in pixels.

This is the approximate height of a font in pixels.

32.3.4 Enumeration Type Documentation

32.3.4.1 anonymous enum

```
anonymous enum
```

FD "when" conditions.

Enumerator

FL_READ	Call the callback when there is data to be read.
FL_WRITE	Call the callback when data can be written without blocking.
FL_EXCEPT	Call the callback if an exception occurs on the file.

32.3.4.2 Fl_Boxtype

```
enum Fl_Boxtype
```

Enumerator

FL_NO_BOX	nothing is drawn at all, this box is invisible
FL_FLAT_BOX	a flat box
FL_UP_BOX	see figure 1
FL_DOWN_BOX	see figure 1
FL_UP_FRAME	see figure 1
FL_DOWN_FRAME	see figure 1
FL_THIN_UP_BOX	see figure 1
FL_THIN_DOWN_BOX	see figure 1
FL_THIN_UP_FRAME	see figure 1
FL_THIN_DOWN_FRAME	see figure 1
FL_ENGRAVED_BOX	see figure 1

Enumerator

FL_EMBOSSSED_BOX	see figure 1
FL_ENGRAVED_FRAME	see figure 1
FL_EMBOSSSED_FRAME	see figure 1
FL_BORDER_BOX	see figure 1
_FL_SHADOW_BOX	see figure 1
FL_BORDER_FRAME	see figure 1
_FL_SHADOW_FRAME	see figure 1
_FL_ROUNDED_BOX	see figure 1
_FL_RSHADOW_BOX	see figure 1
_FL_ROUNDED_FRAME	see figure 1
_FL_RFLAT_BOX	see figure 1
_FL_ROUND_UP_BOX	see figure 1
_FL_ROUND_DOWN_BOX	see figure 1
_FL_DIAMOND_UP_BOX	see figure 1
_FL_DIAMOND_DOWN_BOX	see figure 1
_FL_OVAL_BOX	see figure 1
_FL_OSHADOW_BOX	see figure 1
_FL_OVAL_FRAME	see figure 1
_FL_OFLAT_BOX	see figure 1
_FL_PLASTIC_UP_BOX	plastic version of FL_UP_BOX
_FL_PLASTIC_DOWN_BOX	plastic version of FL_DOWN_BOX
_FL_PLASTIC_UP_FRAME	plastic version of FL_UP_FRAME
_FL_PLASTIC_DOWN_FRAME	plastic version of FL_DOWN_FRAME
_FL_PLASTIC_THIN_UP_BOX	plastic version of FL_THIN_UP_BOX
_FL_PLASTIC_THIN_DOWN_BOX	plastic version of FL_THIN_DOWN_BOX
_FL_PLASTIC_ROUND_UP_BOX	plastic version of FL_ROUND_UP_BOX
_FL_PLASTIC_ROUND_DOWN_BOX	plastic version of FL_ROUND_DOWN_BOX
_FL_GTK_UP_BOX	gtk+ version of FL_UP_BOX
_FL_GTK_DOWN_BOX	gtk+ version of FL_DOWN_BOX
_FL_GTK_UP_FRAME	gtk+ version of FL_UP_FRAME
_FL_GTK_DOWN_FRAME	gtk+ version of FL_DOWN_FRAME
_FL_GTK_THIN_UP_BOX	gtk+ version of FL_THIN_UP_BOX
_FL_GTK_THIN_DOWN_BOX	gtk+ version of FL_THIN_DOWN_BOX
_FL_GTK_THIN_UP_FRAME	gtk+ version of FL_THIN_UP_FRAME
_FL_GTK_THIN_DOWN_FRAME	gtk+ version of FL_THIN_DOWN_FRAME
_FL_GTK_ROUND_UP_BOX	gtk+ version of FL_ROUND_UP_BOX
_FL_GTK_ROUND_DOWN_BOX	gtk+ version of FL_ROUND_DOWN_BOX
_FL_GLEAM_UP_BOX	gleam version of FL_UP_BOX
_FL_GLEAM_DOWN_BOX	gleam version of FL_DOWN_BOX
_FL_GLEAM_UP_FRAME	gleam version of FL_UP_FRAME
_FL_GLEAM_DOWN_FRAME	gleam version of FL_DOWN_FRAME
_FL_GLEAM_THIN_UP_BOX	gleam version of FL_THIN_UP_BOX
_FL_GLEAM_THIN_DOWN_BOX	gleam version of FL_THIN_DOWN_BOX
_FL_GLEAM_ROUND_UP_BOX	gleam version of FL_ROUND_UP_BOX
_FL_GLEAM_ROUND_DOWN_BOX	gleam version of FL_ROUND_DOWN_BOX
FL_FREE_BOXTYPE	the first free box type for creation of new box types

32.3.4.3 Fl_Cursor

enum [Fl_Cursor](#)

The following constants define the mouse cursors that are available in FLTK.

Cursors are provided by the system when available, or bitmaps built into FLTK as a fallback.

Todo enum [Fl_Cursor](#) needs maybe an image.

Enumerator

FL_CURSOR_DEFAULT	the default cursor, usually an arrow.
FL_CURSOR_ARROW	an arrow pointer.
FL_CURSOR_CROSS	crosshair.
FL_CURSOR_WAIT	busy indicator (e.g. hourglass).
FL_CURSOR_INSERT	I-beam.
FL_CURSOR_HAND	pointing hand.
FL_CURSOR_HELP	question mark pointer.
FL_CURSOR_MOVE	4-pointed arrow or hand.
FL_CURSOR_NS	up/down resize.
FL_CURSOR_WE	left/right resize.
FL_CURSOR_NWSE	diagonal resize.
FL_CURSOR_NESW	diagonal resize.
FL_CURSOR_N	upwards resize.
FL_CURSOR_NE	upwards, right resize.
FL_CURSOR_E	rightwards resize.
FL_CURSOR_SE	downwards, right resize.
FL_CURSOR_S	downwards resize.
FL_CURSOR_SW	downwards, left resize.
FL_CURSOR_W	leftwards resize.
FL_CURSOR_NW	upwards, left resize.
FL_CURSOR_NONE	invisible.

32.3.4.4 Fl_Damage

enum [Fl_Damage](#)

Damage masks.

Enumerator

FL_DAMAGE_CHILD	A child needs to be redrawn.
FL_DAMAGE_EXPOSE	The window was exposed.
FL_DAMAGE_SCROLL	The Fl_Scroll widget was scrolled.
FL_DAMAGE_OVERLAY	The overlay planes need to be redrawn.
FL_DAMAGE_USER1	First user-defined damage bit.
FL_DAMAGE_USER2	Second user-defined damage bit.
FL_DAMAGE_ALL	Everything needs to be redrawn.

32.3.4.5 Fl_Event

enum `Fl_Event`

Every time a user moves the mouse pointer, clicks a button, or presses a key, an event is generated and sent to your application.

Events can also come from other programs like the window manager.

Events are identified by the integer argument passed to the `Fl_Widget::handle()` virtual method. Other information about the most recent event is stored in static locations and acquired by calling the `Fl::event_*` methods. This static information remains valid until the next event is read from the window system, so it is ok to look at it outside of the `handle()` method.

Event numbers can be converted to their actual names using the `fl_eventnames[]` array defined in `#include <FL/names.h>`

See also

`Fl::event_text()`, `Fl::event_key()`, class `Fl::`

Enumerator

<code>FL_NO_EVENT</code>	No event.
<code>FL_PUSH</code>	<p>A mouse button has gone down with the mouse pointing at this widget. You can find out what button by calling <code>Fl::event_button()</code>. You find out the mouse position by calling <code>Fl::event_x()</code> and <code>Fl::event_y()</code>.</p> <p>A widget indicates that it "wants" the mouse click by returning non-zero from its <code>Fl_Widget::handle()</code> method. It will then become the <code>Fl::pushed()</code> widget and will get <code>FL_DRAG</code> and the matching <code>FL_RELEASE</code> events.</p> <p>If <code>Fl_Widget::handle()</code> returns zero then FLTK will try sending the <code>FL_PUSH</code> to another widget.</p>
<code>FL_RELEASE</code>	<p>A mouse button has been released. You can find out what button by calling <code>Fl::event_button()</code>.</p> <p>In order to receive the <code>FL_RELEASE</code> event, the widget must return non-zero when handling <code>FL_PUSH</code>.</p>
<code>FL_ENTER</code>	<p>The mouse has been moved to point at this widget. This can be used for highlighting feedback. If a widget wants to highlight or otherwise track the mouse, it indicates this by returning non-zero from its <code>handle()</code> method. It then becomes the <code>Fl::belowmouse()</code> widget and will receive <code>FL_MOVE</code> and <code>FL_LEAVE</code> events.</p>
<code>FL_LEAVE</code>	<p>The mouse has moved out of the widget. In order to receive the <code>FL_LEAVE</code> event, the widget must return non-zero when handling <code>FL_ENTER</code>.</p>
<code>FL_DRAG</code>	<p>The mouse has moved with a button held down. The current button state is in <code>Fl::event_state()</code>. The mouse position is in <code>Fl::event_x()</code> and <code>Fl::event_y()</code>.</p> <p>In order to receive <code>FL_DRAG</code> events, the widget must return non-zero when handling <code>FL_PUSH</code>.</p>
<code>FL_FOCUS</code>	<p>This indicates an <i>attempt</i> to give a widget the keyboard focus. If a widget wants the focus, it should change itself to display the fact that it has the focus, and return non-zero from its <code>handle()</code> method. It then becomes the <code>Fl::focus()</code> widget and gets <code>FL_KEYDOWN</code>, <code>FL_KEYUP</code>, and <code>FL_UNFOCUS</code> events. The focus will change either because the window manager changed which window gets the focus, or because the user tried to navigate using tab, arrows, or other keys. You can check <code>Fl::event_key()</code> to figure out why it moved. For navigation it will be the key pressed and for interaction with the window manager it will be zero.</p>

Enumerator

FL_UNFOCUS	This event is sent to the previous Fl::focus() widget when another widget gets the focus or the window loses focus.
FL_KEYDOWN	A key was pressed (FL_KEYDOWN) or released (FL_KEYUP). FL_KEYBOARD is a synonym for FL_KEYDOWN . The key can be found in Fl::event_key() . The text that the key should insert can be found with Fl::event_text() and its length is in Fl::event_length() . If you use the key handle() should return 1. If you return zero then FLTK assumes you ignored the key and will then attempt to send it to a parent widget. If none of them want it, it will change the event into a FL_SHORTCUT event. To receive FL_KEYBOARD events you must also respond to the FL_FOCUS and FL_UNFOCUS events. If you are writing a text-editing widget you may also want to call the Fl::compose() function to translate individual keystrokes into non-ASCII characters. FL_KEYUP events are sent to the widget that currently has focus. This is not necessarily the same widget that received the corresponding FL_KEYDOWN event because focus may have changed between events.
FL_KEYBOARD	Equivalent to FL_KEYDOWN . See also FL_KEYDOWN
FL_KEYUP	Key release event. See also FL_KEYDOWN
FL_CLOSE	The user clicked the close button of a window. This event is used internally only to trigger the callback of Fl_Window derived classed. The default callback closes the window calling Fl_Window::hide() .
FL_MOVE	The mouse has moved without any mouse buttons held down. This event is sent to the Fl::belowmouse() widget. In order to receive FL_MOVE events, the widget must return non-zero when handling FL_ENTER .
FL_SHORTCUT	If the Fl::focus() widget is zero or ignores an FL_KEYBOARD event then FLTK tries sending this event to every widget it can, until one of them returns non-zero. FL_SHORTCUT is first sent to the Fl::belowmouse() widget, then its parents and siblings, and eventually to every widget in the window, trying to find an object that returns non-zero. FLTK tries really hard to not to ignore any keystrokes! You can also make "global" shortcuts by using Fl::add_handler() . A global shortcut will work no matter what windows are displayed or which one has the focus.
FL_DEACTIVATE	This widget is no longer active, due to Fl_Widget::deactivate() being called on it or one of its parents. Fl_Widget::active() may still be true after this, the widget is only active if Fl_Widget::active() is true on it and all its parents (use Fl_Widget::active_r() to check this).
FL_ACTIVATE	This widget is now active, due to Fl_Widget::activate() being called on it or one of its parents.

Enumerator

FL_HIDE	This widget is no longer visible, due to <code>Fl_Widget::hide()</code> being called on it or one of its parents, or due to a parent window being minimized. <code>Fl_Widget::visible()</code> may still be true after this, but the widget is visible only if <code>visible()</code> is true for it and all its parents (use <code>Fl_Widget::visible_r()</code> to check this).
FL_SHOW	This widget is visible again, due to <code>Fl_Widget::show()</code> being called on it or one of its parents, or due to a parent window being restored. Child <code>Fl_Windows</code> respond to this by actually creating the window if not done already, so if you subclass a window, be sure to pass <code>FL_SHOW</code> to the base class <code>Fl_Widget::handle()</code> method!
FL_PASTE	You should get this event some time after you call <code>Fl::paste()</code> . The contents of <code>Fl::event_text()</code> is the text to insert and the number of characters is in <code>Fl::event_length()</code> .
FL_SELECTIONCLEAR	The <code>Fl::selection_owner()</code> will get this event before the selection is moved to another widget. This indicates that some other widget or program has claimed the selection. Motif programs used this to clear the selection indication. Most modern programs ignore this.
FL_MOUSEWHEEL	The user has moved the mouse wheel. The <code>Fl::event_dx()</code> and <code>Fl::event_dy()</code> methods can be used to find the amount to scroll horizontally and vertically.
FL_DND_ENTER	The mouse has been moved to point at this widget. A widget that is interested in receiving drag'n'drop data must return 1 to receive <code>FL_DND_DRAG</code> , <code>FL_DND_LEAVE</code> and <code>FL_DND_RELEASE</code> events.
FL_DND_DRAG	The mouse has been moved inside a widget while dragging data. A widget that is interested in receiving drag'n'drop data should indicate the possible drop position.
FL_DND_LEAVE	The mouse has moved out of the widget.
FL_DND_RELEASE	The user has released the mouse button dropping data into the widget. If the widget returns 1, it will receive the data in the immediately following <code>FL_PASTE</code> event.
FL_SCREEN_CONFIGURATION_CHANGED	The screen configuration (number, positions) was changed. Use <code>Fl::add_handler()</code> to be notified of this event.
FL_FULLSCREEN	The fullscreen state of the window has changed.
FL_ZOOM_GESTURE	The user has made a zoom/pinch/magnification gesture. The <code>Fl::event_dy()</code> method can be used to find magnification amount, <code>Fl::event_x()</code> and <code>Fl::event_y()</code> are set as well.

32.3.4.6 Fl_Labeltype

enum `Fl_Labeltype`

The `labeltype()` method sets the type of the label.

The following standard label types are included:

Todo The doxygen comments are incomplete, and some labeltypes start with an underscore. Also, there are three external functions undocumented (yet):

- `fl_define_FL_SHADOW_LABEL()`
- `fl_define_FL_ENGRAVED_LABEL()`
- `fl_define_FL_EMBOSSED_LABEL()`

Enumerator

FL_NORMAL_LABEL	draws the text (0)
FL_NO_LABEL	does nothing
_FL_SHADOW_LABEL	draws a drop shadow under the text
_FL_ENGRAVED_LABEL	draws edges as though the text is engraved
_FL_EMBOSSSED_LABEL	draws edges as though the text is raised
_FL_MULTI_LABEL	draws a composite label See also Fl_Multi_Label
_FL_ICON_LABEL	draws the icon associated with the text
_FL_IMAGE_LABEL	the label displays an "icon" based on a Fl_Image
FL_FREE_LABELTYPE	first free labeltype to use for creating own labeltypes

32.3.4.7 Fl_When

enum [Fl_When](#)

These constants determine when a callback is performed.

See also

[Fl_Widget::when\(\)](#);

Todo doxygen comments for values are incomplete and maybe wrong or unclear

Enumerator

FL_WHEN_NEVER	Never call the callback.
FL_WHEN_CHANGED	Do the callback only when the widget value changes.
FL_WHEN_NOT_CHANGED	Do the callback whenever the user interacts with the widget.
FL_WHEN_RELEASE	Do the callback when the button or key is released and the value changes.
FL_WHEN_RELEASE_ALWAYS	Do the callback when the button or key is released, even if the value doesn't change.
FL_WHEN_ENTER_KEY	Do the callback when the user presses the ENTER key and the value changes.
FL_WHEN_ENTER_KEY_ALWAYS	Do the callback when the user presses the ENTER key, even if the value doesn't change.
FL_WHEN_ENTER_KEY_CHANGED	?

32.3.5 Function Documentation**32.3.5.1 fl_box()**

```
Fl_Boxtype fl_box (
    Fl_Boxtype b ) [inline]
```

Get the filled version of a frame.

If no filled version of a given frame exists, the behavior of this function is undefined and some random box or frame is returned.

32.3.5.2 fl_color_cube()

```
Fl_Color fl_color_cube (
    int r,
    int g,
    int b ) [inline]
```

Returns a color out of the color cube.

`r` must be in the range 0 to FL_NUM_RED (5) minus 1, `g` must be in the range 0 to FL_NUM_GREEN (8) minus 1, `b` must be in the range 0 to FL_NUM_BLUE (5) minus 1.

To get the closest color to a 8-bit set of R,G,B values use:

```
fl_color_cube(R * (FL_NUM_RED - 1) / 255,
    G * (FL_NUM_GREEN - 1) / 255,
    B * (FL_NUM_BLUE - 1) / 255);
```

32.3.5.3 fl_down()

```
Fl_Boxtype fl_down (
    Fl_Boxtype b ) [inline]
```

Get the "pressed" or "down" version of a box.

If no "down" version of a given box exists, the behavior of this function is undefined and some random box or frame is returned.

32.3.5.4 fl_frame()

```
Fl_Boxtype fl_frame (
    Fl_Boxtype b ) [inline]
```

Get the unfilled, frame only version of a box.

If no frame version of a given box exists, the behavior of this function is undefined and some random box or frame is returned.

32.3.5.5 fl_gray_ramp()

```
Fl_Color fl_gray_ramp (
    int i ) [inline]
```

Returns a gray color value from black (`i == 0`) to white (`i == FL_NUM_GRAY - 1`).

FL_NUM_GRAY is defined to be 24 in the current FLTK release. To get the closest FLTK gray value to an 8-bit grayscale color 'I' use:

```
fl_gray_ramp(I * (FL_NUM_GRAY - 1) / 255)
```

32.3.6 Variable Documentation

32.3.6.1 FL_ALIGN_LEFT

```
const Fl_Align FL_ALIGN_LEFT = (Fl_Align)4
```

Align the label at the left of the widget.

Inside labels appear left-justified starting at the left side of the widget, outside labels are right-justified and drawn to the left of the widget.

32.3.6.2 FL_ALIGN_TOP

```
const Fl_Align FL_ALIGN_TOP = (Fl_Align)1
```

Align the label at the top of the widget.

Inside labels appear below the top, outside labels are drawn on top of the widget.

32.3.6.3 FL_NORMAL_SIZE

```
FL_EXPORT Fl_Fontsize FL_NORMAL_SIZE [extern]
```

normal font size

normal font size

32.4 Enumerations.H

[Go to the documentation of this file.](#)

```

1 //
2 // Enumerations for the Fast Light Tool Kit (FLTK).
3 //
4 // Copyright 1998-2021 by Bill Spitzak and others.
5 //
6 // This library is free software. Distribution and use rights are outlined in
7 // the file "COPYING" which should have been included with this file. If this
8 // file is missing or damaged, see the license at:
9 //
10 //      https://www.fltk.org/COPYING.php
11 //
12 // Please see the following page on how to report bugs and issues:
13 //
14 //      https://www.fltk.org/bugs.php
15 //
16
21 #ifndef Fl_Enumerations_H
22 #define Fl_Enumerations_H
23
24 /*
25 *****
26 * Notes on FL_ABI_VERSION and deprecated (obsolete) FLTK_ABI_VERSION:
27 *
28 * (1) FLTK_ABI_VERSION is deprecated, but still defined below.
29 *     Do NOT define FLTK_ABI_VERSION here - it would be overwritten later.
30 *
31 * (2) FL_ABI_VERSION is now (as of FLTK 1.3.4) defined by configure
32 *     or CMake. Do NOT define it here. Its definition will be included
33 *     below by "#include <FL/abi-version.h>".
34 *
35 * (3) If you use the provided IDE files (Windows VC++ or Xcode) you should
36 *     edit the definition in the provided file abi-version.ide. The correct
37 *     file is '\path/to/fltk/abi-version.ide' .
38 *
39 *****
40 * For more informations on FL_ABI_VERSION see README.abi-version.txt.
41 *****
42 */
43
44 #include <FL/abi-version.h>
45
46 # include "Fl_Export.H"
47 # include "fl_types.h"
48
49
62 #define FL_MAJOR_VERSION      1
63
69 #define FL_MINOR_VERSION     3
70
76 #define FL_PATCH_VERSION     8
77
99 #define FL_VERSION           ( (double)FL_MAJOR_VERSION + \
100 (double)FL_MINOR_VERSION * 0.01 + \
101 (double)FL_PATCH_VERSION * 0.0001 )
102
121 #define FL_API_VERSION (FL_MAJOR_VERSION*10000 + FL_MINOR_VERSION*100 + FL_PATCH_VERSION)
122
153 #ifndef FL_ABI_VERSION
154 #define FL_ABI_VERSION (FL_MAJOR_VERSION*10000 + FL_MINOR_VERSION*100)
155 #endif
156
157 /*
158 Check if FL_ABI_VERSION is out of allowed range; redefine if necessary.
159
160 This is done to prevent users from defining an illegal ABI version.
161
162 Rule:  FL_MAJOR_VERSION * 10000 + FL_MINOR_VERSION * 100
163 <= FL_ABI_VERSION <= FL_API_VERSION.
164
165 Example (FLTK 1.3.4):
166
167 10300 <= FL_ABI_VERSION <= 10304
168
169 Note:  configure + CMake can be used to define FL_ABI_VERSION, but they
170 do not check validity. This is done here.
171 */
172
173 #if FL_ABI_VERSION < FL_MAJOR_VERSION*10000 + FL_MINOR_VERSION*100
174

```

```

175 # undef FL_ABI_VERSION
176 # define FL_ABI_VERSION (FL_MAJOR_VERSION*10000 + FL_MINOR_VERSION*100)
177
178 #elif FL_ABI_VERSION > FL_API_VERSION
179
180 # undef FL_ABI_VERSION
181 # define FL_ABI_VERSION FL_API_VERSION
182
183 #endif
184
185 /*
186 FLTK_ABI_VERSION is deprecated (replaced by FL_ABI_VERSION).
187
188 This deprecated constant will be removed in FLTK 1.4.0 and later.
189 Please use FL_ABI_VERSION when FLTK 1.4.0 has been released.
190 */
191
192 #ifdef FLTK_ABI_VERSION
193 #undef FLTK_ABI_VERSION
194 #endif
195
196 #define FLTK_ABI_VERSION FL_ABI_VERSION
197     // group: Version Numbers
198
199
200 // DEV NOTE: Keep this list in sync with FL/names.H
201 enum Fl_Event { // events
202     FL_NO_EVENT          = 0,
203
204     FL_PUSH              = 1,
205
206     FL_RELEASE          = 2,
207
208     FL_ENTER            = 3,
209
210     FL_LEAVE           = 4,
211
212     FL_DRAG             = 5,
213
214     FL_FOCUS            = 6,
215
216     FL_UNFOCUS         = 7,
217
218     FL_KEYDOWN         = 8,
219
220     FL_KEYBOARD        = 8,
221
222     FL_KEYUP           = 9,
223
224     FL_CLOSE           = 10,
225
226     FL_MOVE            = 11,
227
228     FL_SHORTCUT        = 12,
229
230     FL_DEACTIVATE      = 13,
231
232     FL_ACTIVATE        = 14,
233
234     FL_HIDE            = 15,
235
236     FL_SHOW            = 16,
237
238     FL_PASTE           = 17,
239
240     FL_SELECTIONCLEAR  = 18,
241
242     FL_MOUSEWHEEL      = 19,
243
244     FL_DND_ENTER       = 20,
245
246     FL_DND_DRAG        = 21,
247
248     FL_DND_LEAVE       = 22,
249
250     FL_DND_RELEASE     = 23,
251     FL_SCREEN_CONFIGURATION_CHANGED = 24,
252     FL_FULLSCREEN      = 25,
253     FL_ZOOM_GESTURE    = 26
254 };
255
256 enum Fl_When { // Fl_Widget::when():
257     FL_WHEN_NEVER      = 0,
258     FL_WHEN_CHANGED    = 1,
259     FL_WHEN_NOT_CHANGED = 2,
260     FL_WHEN_RELEASE    = 4,
261     FL_WHEN_RELEASE_ALWAYS = 6,
262     FL_WHEN_ENTER_KEY  = 8,

```

```

444 FL_WHEN_ENTER_KEY_ALWAYS=10,
445 FL_WHEN_ENTER_KEY_CHANGED=11
446 };
447 // group: When Conditions
448
449
450
451 // FIXME: These codes collide with valid Unicode keys
452
453 #define FL_Button 0xfee8
454 #define FL_BackSpace 0xff08
455 #define FL_Tab 0xff09
456 #define FL_Iso_Key 0xff0c
457 #define FL_Enter 0xff0d
458 #define FL_Pause 0xff13
459 #define FL_Scroll_Lock 0xff14
460 #define FL_Escape 0xff1b
461 #define FL_Kana 0xff2e
462 #define FL_Eisu 0xff2f
463 #define FL_Yen 0xff30
464 #define FL_JIS_Underscore 0xff31
465 #define FL_Home 0xff50
466 #define FL_Left 0xff51
467 #define FL_Up 0xff52
468 #define FL_Right 0xff53
469 #define FL_Down 0xff54
470 #define FL_Page_Up 0xff55
471 #define FL_Page_Down 0xff56
472 #define FL_End 0xff57
473 #define FL_Print 0xff61
474 #define FL_Insert 0xff63
475 #define FL_Menu 0xff67
476 #define FL_Help 0xff68
477 #define FL_Num_Lock 0xff7f
478 #define FL_KP 0xff80
479 #define FL_KP_Enter 0xff8d
480 #define FL_KP_Last 0xffbd
481 #define FL_F 0xffbd
482 #define FL_F_Last 0xffe0
483 #define FL_Shift_L 0xffe1
484 #define FL_Shift_R 0xffe2
485 #define FL_Control_L 0xffe3
486 #define FL_Control_R 0xffe4
487 #define FL_Caps_Lock 0xffe5
488 #define FL_Meta_L 0xffe7
489 #define FL_Meta_R 0xffe8
490 #define FL_Alt_L 0xffe9
491 #define FL_Alt_R 0xffea
492 #define FL_Delete 0xffff
493
494 // These use the Private Use Area (PUA) of the Basic Multilingual Plane
495 // of Unicode. Guaranteed not to conflict with a proper Unicode character.
496
497 // These primarily map to the XFree86 keysym range
498 #define FL_Volume_Down 0xEF11 /* Volume control down */
499 #define FL_Volume_Mute 0xEF12 /* Mute sound from the system */
500 #define FL_Volume_Up 0xEF13 /* Volume control up */
501 #define FL_Media_Play 0xEF14 /* Start playing of audio */
502 #define FL_Media_Stop 0xEF15 /* Stop playing audio */
503 #define FL_Media_Prev 0xEF16 /* Previous track */
504 #define FL_Media_Next 0xEF17 /* Next track */
505 #define FL_Home_Page 0xEF18 /* Display user's home page */
506 #define FL_Mail 0xEF19 /* Invoke user's mail program */
507 #define FL_Search 0xEF1B /* Search */
508 #define FL_Back 0xEF26 /* Like back on a browser */
509 #define FL_Forward 0xEF27 /* Like forward on a browser */
510 #define FL_Stop 0xEF28 /* Stop current operation */
511 #define FL_Refresh 0xEF29 /* Refresh the page */
512 #define FL_Sleep 0xEF2F /* Put system to sleep */
513 #define FL_Favorites 0xEF30 /* Show favorite locations */
514 // group: Mouse and Keyboard Events
515
516
517 #define FL_LEFT_MOUSE 1
518 #define FL_MIDDLE_MOUSE 2
519 #define FL_RIGHT_MOUSE 3
520 // group: Mouse Buttons
521
522 // group: Event States
523
524 // FIXME: it would be nice to have the modifiers in the upper 8 bit so that
525 // a unicode ke (24bit) can be sent as an unsigned with the modifiers.
526
527 #define FL_SHIFT 0x00010000
528 #define FL_CAPS_LOCK 0x00020000
529 #define FL_CTRL 0x00040000
530 #define FL_ALT 0x00080000
531 #define FL_NUM_LOCK 0x00100000

```

```

560                                     // most X servers do this?
561 #define FL_META          0x00400000
562                                     // correct for XFree86
563 #define FL_SCROLL_LOCK  0x00800000
564                                     // correct for XFree86
565 #define FL_BUTTON1      0x01000000
566 #define FL_BUTTON2      0x02000000
567 #define FL_BUTTON3      0x04000000
568 #define FL_BUTTONS      0x7f000000
569 #define FL_BUTTON(n)    (0x00800000«(n))
570
571 #define FL_KEY_MASK 0x0000ffff
572                                     //   FIXME: Unicode needs 24 bits!
573
574 #ifndef __APPLE__
575 #   define FL_COMMAND    FL_META
576 #   define FL_CONTROL    FL_CTRL
577 #else
578 #   define FL_COMMAND    FL_CTRL
579 #   define FL_CONTROL    FL_META
580 #endif // __APPLE__
581                                     // group:  Event States
582
583
584 enum Fl_Boxtype { // boxtypes (if you change these you must fix fl_boxtype.cxx):
585
586     FL_NO_BOX = 0,
587     FL_FLAT_BOX,
588     FL_UP_BOX,
589     FL_DOWN_BOX,
590     FL_UP_FRAME,
591     FL_DOWN_FRAME,
592     FL_THIN_UP_BOX,
593     FL_THIN_DOWN_BOX,
594     FL_THIN_UP_FRAME,
595     FL_THIN_DOWN_FRAME,
596     FL_ENGRAVED_BOX,
597     FL_ENGRAVED_FRAME,
598     FL_EMBOSSSED_BOX,
599     FL_EMBOSSSED_FRAME,
600     FL_BORDER_BOX,
601     _FL_SHADOW_BOX,
602     FL_BORDER_FRAME,
603     _FL_SHADOW_FRAME,
604     _FL_ROUNDED_BOX,
605     _FL_RSHADOW_BOX,
606     _FL_ROUNDED_FRAME,
607     _FL_RFLAT_BOX,
608     _FL_ROUND_UP_BOX,
609     _FL_ROUND_DOWN_BOX,
610     _FL_DIAMOND_UP_BOX,
611     _FL_DIAMOND_DOWN_BOX,
612     _FL_OVAL_BOX,
613     _FL_OSHADOW_BOX,
614     _FL_OVAL_FRAME,
615     _FL_OFLAT_BOX,
616     _FL_PLASTIC_UP_BOX,
617     _FL_PLASTIC_DOWN_BOX,
618     _FL_PLASTIC_UP_FRAME,
619     _FL_PLASTIC_DOWN_FRAME,
620     _FL_PLASTIC_THIN_UP_BOX,
621     _FL_PLASTIC_THIN_DOWN_BOX,
622     _FL_PLASTIC_ROUND_UP_BOX,
623     _FL_PLASTIC_ROUND_DOWN_BOX,
624     _FL_GTK_UP_BOX,
625     _FL_GTK_DOWN_BOX,
626     _FL_GTK_UP_FRAME,
627     _FL_GTK_DOWN_FRAME,
628     _FL_GTK_THIN_UP_BOX,
629     _FL_GTK_THIN_DOWN_BOX,
630     _FL_GTK_THIN_UP_FRAME,
631     _FL_GTK_THIN_DOWN_FRAME,
632     _FL_GTK_ROUND_UP_BOX,
633     _FL_GTK_ROUND_DOWN_BOX,
634     _FL_GLEAM_UP_BOX,
635     _FL_GLEAM_DOWN_BOX,
636     _FL_GLEAM_UP_FRAME,
637     _FL_GLEAM_DOWN_FRAME,
638     _FL_GLEAM_THIN_UP_BOX,
639     _FL_GLEAM_THIN_DOWN_BOX,
640     _FL_GLEAM_ROUND_UP_BOX,
641     _FL_GLEAM_ROUND_DOWN_BOX,
642     FL_FREE_BOXTYPE
643 };
644 extern FL_EXPORT Fl_Boxtype fl_define_FL_ROUND_UP_BOX();
645 #define FL_ROUND_UP_BOX fl_define_FL_ROUND_UP_BOX()
646 #define FL_ROUND_DOWN_BOX (Fl_Boxtype) (fl_define_FL_ROUND_UP_BOX()+1)
647 extern FL_EXPORT Fl_Boxtype fl_define_FL_SHADOW_BOX();

```

```

665 #define FL_SHADOW_BOX fl_define_FL_SHADOW_BOX()
666 #define FL_SHADOW_FRAME (Fl_Boxtype)(fl_define_FL_SHADOW_BOX()+2)
667 extern FL_EXPORT Fl_Boxtype fl_define_FL_ROUNDED_BOX();
668 #define FL_ROUNDED_BOX fl_define_FL_ROUNDED_BOX()
669 #define FL_ROUNDED_FRAME (Fl_Boxtype)(fl_define_FL_ROUNDED_BOX()+2)
670 extern FL_EXPORT Fl_Boxtype fl_define_FL_RFLAT_BOX();
671 #define FL_RFLAT_BOX fl_define_FL_RFLAT_BOX()
672 extern FL_EXPORT Fl_Boxtype fl_define_FL_RSHADOW_BOX();
673 #define FL_RSHADOW_BOX fl_define_FL_RSHADOW_BOX()
674 extern FL_EXPORT Fl_Boxtype fl_define_FL_DIAMOND_BOX();
675 #define FL_DIAMOND_UP_BOX fl_define_FL_DIAMOND_BOX()
676 #define FL_DIAMOND_DOWN_BOX (Fl_Boxtype)(fl_define_FL_DIAMOND_BOX()+1)
677 extern FL_EXPORT Fl_Boxtype fl_define_FL_OVAL_BOX();
678 #define FL_OVAL_BOX fl_define_FL_OVAL_BOX()
679 #define FL_OSHADOW_BOX (Fl_Boxtype)(fl_define_FL_OVAL_BOX()+1)
680 #define FL_OVAL_FRAME (Fl_Boxtype)(fl_define_FL_OVAL_BOX()+2)
681 #define FL_OFIAT_BOX (Fl_Boxtype)(fl_define_FL_OVAL_BOX()+3)
682
683 extern FL_EXPORT Fl_Boxtype fl_define_FL_PLASTIC_UP_BOX();
684 #define FL_PLASTIC_UP_BOX fl_define_FL_PLASTIC_UP_BOX()
685 #define FL_PLASTIC_DOWN_BOX (Fl_Boxtype)(fl_define_FL_PLASTIC_UP_BOX()+1)
686 #define FL_PLASTIC_UP_FRAME (Fl_Boxtype)(fl_define_FL_PLASTIC_UP_BOX()+2)
687 #define FL_PLASTIC_DOWN_FRAME (Fl_Boxtype)(fl_define_FL_PLASTIC_UP_BOX()+3)
688 #define FL_PLASTIC_THIN_UP_BOX (Fl_Boxtype)(fl_define_FL_PLASTIC_UP_BOX()+4)
689 #define FL_PLASTIC_THIN_DOWN_BOX (Fl_Boxtype)(fl_define_FL_PLASTIC_UP_BOX()+5)
690 #define FL_PLASTIC_ROUND_UP_BOX (Fl_Boxtype)(fl_define_FL_PLASTIC_UP_BOX()+6)
691 #define FL_PLASTIC_ROUND_DOWN_BOX (Fl_Boxtype)(fl_define_FL_PLASTIC_UP_BOX()+7)
692
693 extern FL_EXPORT Fl_Boxtype fl_define_FL_GTK_UP_BOX();
694 #define FL_GTK_UP_BOX fl_define_FL_GTK_UP_BOX()
695 #define FL_GTK_DOWN_BOX (Fl_Boxtype)(fl_define_FL_GTK_UP_BOX()+1)
696 #define FL_GTK_UP_FRAME (Fl_Boxtype)(fl_define_FL_GTK_UP_BOX()+2)
697 #define FL_GTK_DOWN_FRAME (Fl_Boxtype)(fl_define_FL_GTK_UP_BOX()+3)
698 #define FL_GTK_THIN_UP_BOX (Fl_Boxtype)(fl_define_FL_GTK_UP_BOX()+4)
699 #define FL_GTK_THIN_DOWN_BOX (Fl_Boxtype)(fl_define_FL_GTK_UP_BOX()+5)
700 #define FL_GTK_THIN_UP_FRAME (Fl_Boxtype)(fl_define_FL_GTK_UP_BOX()+6)
701 #define FL_GTK_THIN_DOWN_FRAME (Fl_Boxtype)(fl_define_FL_GTK_UP_BOX()+7)
702 #define FL_GTK_ROUND_UP_BOX (Fl_Boxtype)(fl_define_FL_GTK_UP_BOX()+8)
703 #define FL_GTK_ROUND_DOWN_BOX (Fl_Boxtype)(fl_define_FL_GTK_UP_BOX()+9)
704
705 extern FL_EXPORT Fl_Boxtype fl_define_FL_GLEAM_UP_BOX();
706 #define FL_GLEAM_UP_BOX fl_define_FL_GLEAM_UP_BOX()
707 #define FL_GLEAM_DOWN_BOX (Fl_Boxtype)(fl_define_FL_GLEAM_UP_BOX()+1)
708 #define FL_GLEAM_UP_FRAME (Fl_Boxtype)(fl_define_FL_GLEAM_UP_BOX()+2)
709 #define FL_GLEAM_DOWN_FRAME (Fl_Boxtype)(fl_define_FL_GLEAM_UP_BOX()+3)
710 #define FL_GLEAM_THIN_UP_BOX (Fl_Boxtype)(fl_define_FL_GLEAM_UP_BOX()+4)
711 #define FL_GLEAM_THIN_DOWN_BOX (Fl_Boxtype)(fl_define_FL_GLEAM_UP_BOX()+5)
712 #define FL_GLEAM_ROUND_UP_BOX (Fl_Boxtype)(fl_define_FL_GLEAM_UP_BOX()+6)
713 #define FL_GLEAM_ROUND_DOWN_BOX (Fl_Boxtype)(fl_define_FL_GLEAM_UP_BOX()+7)
714
715 // conversions of box types to other boxtypes:
721 inline Fl_Boxtype fl_box(Fl_Boxtype b) {
722     return (Fl_Boxtype)((b<FL_UP_BOX||b%4>1)?b:(b-2));
723 }
724
725 inline Fl_Boxtype fl_down(Fl_Boxtype b) {
726     return (Fl_Boxtype)((b<FL_UP_BOX)?b:(b|1));
727 }
728
729 inline Fl_Boxtype fl_frame(Fl_Boxtype b) {
730     return (Fl_Boxtype)((b%4<2)?b:(b+2));
731 }
732
733 // back-compatibility box types:
742 #define FL_FRAME FL_ENGRAVED_FRAME
743 #define FL_FRAME_BOX FL_ENGRAVED_BOX
744 #define FL_CIRCLE_BOX FL_ROUND_DOWN_BOX
745 #define FL_DIAMOND_BOX FL_DIAMOND_DOWN_BOX
746 // group: Box Types
747
748
749 enum Fl_Labeltype { // labeltypes:
750     FL_NORMAL_LABEL = 0,
751     FL_NO_LABEL,
752     FL_SHADOW_LABEL,
753     FL_ENGRAVED_LABEL,
754     FL_EMBOSSED_LABEL,
755     FL_MULTI_LABEL,
756     FL_ICON_LABEL,
757     FL_IMAGE_LABEL,
758     FL_FREE_LABELTYPE
759 };
760
761 #define FL_SYMBOL_LABEL FL_NORMAL_LABEL
762 extern FL_Labeltype FL_EXPORT fl_define_FL_SHADOW_LABEL();
763 #define FL_SHADOW_LABEL fl_define_FL_SHADOW_LABEL()
764 extern FL_Labeltype FL_EXPORT fl_define_FL_ENGRAVED_LABEL();
765 #define FL_ENGRAVED_LABEL fl_define_FL_ENGRAVED_LABEL()
766 extern FL_Labeltype FL_EXPORT fl_define_FL_EMBOSSED_LABEL();
767 #define FL_EMBOSSED_LABEL fl_define_FL_EMBOSSED_LABEL();

```

```

784 #define FL_EMBOSSSED_LABEL fl_define_FL_EMBOSSSED_LABEL()
826 typedef unsigned Fl_Align;
828 const Fl_Align FL_ALIGN_CENTER      = (Fl_Align)0;
831 const Fl_Align FL_ALIGN_TOP        = (Fl_Align)1;
833 const Fl_Align FL_ALIGN_BOTTOM     = (Fl_Align)2;
837 const Fl_Align FL_ALIGN_LEFT      = (Fl_Align)4;
839 const Fl_Align FL_ALIGN_RIGHT     = (Fl_Align)8;
841 const Fl_Align FL_ALIGN_INSIDE     = (Fl_Align)16;
843 const Fl_Align FL_ALIGN_TEXT_OVER_IMAGE = (Fl_Align)0x0020;
845 const Fl_Align FL_ALIGN_IMAGE_OVER_TEXT = (Fl_Align)0x0000;
847 const Fl_Align FL_ALIGN_CLIP      = (Fl_Align)64;
849 const Fl_Align FL_ALIGN_WRAP      = (Fl_Align)128;
851 const Fl_Align FL_ALIGN_IMAGE_NEXT_TO_TEXT = (Fl_Align)0x0100;
853 const Fl_Align FL_ALIGN_TEXT_NEXT_TO_IMAGE = (Fl_Align)0x0120;
855 const Fl_Align FL_ALIGN_IMAGE_BACKDROP = (Fl_Align)0x0200;
856 const Fl_Align FL_ALIGN_TOP_LEFT   = FL_ALIGN_TOP | FL_ALIGN_LEFT;
857 const Fl_Align FL_ALIGN_TOP_RIGHT  = FL_ALIGN_TOP | FL_ALIGN_RIGHT;
858 const Fl_Align FL_ALIGN_BOTTOM_LEFT = FL_ALIGN_BOTTOM | FL_ALIGN_LEFT;
859 const Fl_Align FL_ALIGN_BOTTOM_RIGHT = FL_ALIGN_BOTTOM | FL_ALIGN_RIGHT;
860 const Fl_Align FL_ALIGN_LEFT_TOP   = 0x0007; // magic value
861 const Fl_Align FL_ALIGN_RIGHT_TOP  = 0x000b; // magic value
862 const Fl_Align FL_ALIGN_LEFT_BOTTOM = 0x000d; // magic value
863 const Fl_Align FL_ALIGN_RIGHT_BOTTOM = 0x000e; // magic value
864 const Fl_Align FL_ALIGN_NOWRAP     = (Fl_Align)0; // for back compatibility
865 const Fl_Align FL_ALIGN_POSITION_MASK = 0x000f; // left, right, top, bottom
866 const Fl_Align FL_ALIGN_IMAGE_MASK = 0x0320; // l/r, t/b, backdrop
875 typedef int Fl_Font;
876
877 const Fl_Font FL_HELVETICA          = 0;
878 const Fl_Font FL_HELVETICA_BOLD    = 1;
879 const Fl_Font FL_HELVETICA_ITALIC  = 2;
880 const Fl_Font FL_HELVETICA_BOLD_ITALIC = 3;
881 const Fl_Font FL_COURIER           = 4;
882 const Fl_Font FL_COURIER_BOLD      = 5;
883 const Fl_Font FL_COURIER_ITALIC    = 6;
884 const Fl_Font FL_COURIER_BOLD_ITALIC = 7;
885 const Fl_Font FL_TIMES             = 8;
886 const Fl_Font FL_TIMES_BOLD        = 9;
887 const Fl_Font FL_TIMES_ITALIC      = 10;
888 const Fl_Font FL_TIMES_BOLD_ITALIC = 11;
889 const Fl_Font FL_SYMBOL            = 12;
890 const Fl_Font FL_SCREEN            = 13;
891 const Fl_Font FL_SCREEN_BOLD       = 14;
892 const Fl_Font FL_ZAPF_DINGBATS     = 15;
893
894 const Fl_Font FL_FREE_FONT         = 16;
895 const Fl_Font FL_BOLD              = 1;
896 const Fl_Font FL_ITALIC           = 2;
897 const Fl_Font FL_BOLD_ITALIC      = 3;
898
904 typedef int Fl_Fontsize;
905
906 extern FL_EXPORT Fl_Fontsize FL_NORMAL_SIZE;
907
932 typedef unsigned int Fl_Color;
933
934 // Standard colors.  These are used as default colors in widgets and altered as necessary
935 const Fl_Color FL_FOREGROUND_COLOR = 0;
936 const Fl_Color FL_BACKGROUND2_COLOR = 7;
937 const Fl_Color FL_INACTIVE_COLOR   = 8;
938 const Fl_Color FL_SELECTION_COLOR  = 15;
939
940 // boxtypes generally limit themselves to these colors so
941 // the whole ramp is not allocated:
942
943 const Fl_Color FL_GRAY0    = 32; // 'A'
944 const Fl_Color FL_DARK3    = 39; // 'H'
945 const Fl_Color FL_DARK2    = 45; // 'N'
946 const Fl_Color FL_DARK1    = 47; // 'P'
947 const Fl_Color FL_BACKGROUND_COLOR = 49; // 'R' default background color
948 const Fl_Color FL_LIGHT1    = 50; // 'S'
949 const Fl_Color FL_LIGHT2    = 52; // 'U'
950 const Fl_Color FL_LIGHT3    = 54; // 'W'
951
952 // FLTK provides a 5x8x5 color cube that is used with colormap visuals
953
954 const Fl_Color FL_BLACK     = 56;
955 const Fl_Color FL_RED       = 88;
956 const Fl_Color FL_GREEN     = 63;
957 const Fl_Color FL_YELLOW    = 95;
958 const Fl_Color FL_BLUE      = 216;
959 const Fl_Color FL_MAGENTA    = 248;
960 const Fl_Color FL_CYAN      = 223;
961 const Fl_Color FL_DARK_RED   = 72;
962
963 const Fl_Color FL_DARK_GREEN = 60;
964 const Fl_Color FL_DARK_YELLOW = 76;

```



```

965 const Fl_Color FL_DARK_BLUE      = 136;
966 const Fl_Color FL_DARK_MAGENTA  = 152;
967 const Fl_Color FL_DARK_CYAN     = 140;
968
969 const Fl_Color FL_WHITE          = 255;
970
971
972 #define FL_FREE_COLOR      (Fl_Color)16
973 #define FL_NUM_FREE_COLOR 16
974 #define FL_GRAY_RAMP      (Fl_Color)32
975 #define FL_NUM_GRAY      24
976 #define FL_GRAY           FL_BACKGROUND_COLOR
977 #define FL_COLOR_CUBE     (Fl_Color)56
978 #define FL_NUM_RED        5
979 #define FL_NUM_GREEN      8
980 #define FL_NUM_BLUE       5
981
982 FL_EXPORT Fl_Color fl_inactive(Fl_Color c);
983
984 FL_EXPORT Fl_Color fl_contrast(Fl_Color fg, Fl_Color bg);
985
986 FL_EXPORT Fl_Color fl_color_average(Fl_Color c1, Fl_Color c2, float weight);
987
988 inline Fl_Color fl_lighter(Fl_Color c) { return fl_color_average(c, FL_WHITE, .67f); }
989
990 inline Fl_Color fl_darker(Fl_Color c) { return fl_color_average(c, FL_BLACK, .67f); }
991
992 inline Fl_Color fl_rgb_color(uchar r, uchar g, uchar b) {
993     if (!r && !g && !b) return FL_BLACK;
994     else return (Fl_Color)((!(r < 8) | g) < 8) | b) < 8);
995 }
996
997 inline Fl_Color fl_rgb_color(uchar g) {
998     if (!g) return FL_BLACK;
999     else return (Fl_Color)((!(g < 8) | g) < 8) | g) < 8);
1000 }
1001
1002 inline Fl_Color fl_gray_ramp(int i) {return (Fl_Color)(i+FL_GRAY_RAMP);}
1003
1004 inline Fl_Color fl_color_cube(int r, int g, int b) {
1005     return (Fl_Color)((b*FL_NUM_RED + r) * FL_NUM_GREEN + g + FL_COLOR_CUBE);}
1006 // group: Colors
1007
1008 /* FIXME: We should renumber these, but that will break the ABI */
1009 enum Fl_Cursor {
1010     FL_CURSOR_DEFAULT = 0,
1011     FL_CURSOR_ARROW   = 35,
1012     FL_CURSOR_CROSS   = 66,
1013     FL_CURSOR_WAIT    = 76,
1014     FL_CURSOR_INSERT  = 77,
1015     FL_CURSOR_HAND    = 31,
1016     FL_CURSOR_HELP    = 47,
1017     FL_CURSOR_MOVE    = 27,
1018     /* Resize indicators */
1019     FL_CURSOR_NS      = 78,
1020     FL_CURSOR_WE      = 79,
1021     FL_CURSOR_NWSE    = 80,
1022     FL_CURSOR_NESW    = 81,
1023     FL_CURSOR_N       = 70,
1024     FL_CURSOR_NE      = 69,
1025     FL_CURSOR_E       = 49,
1026     FL_CURSOR_SE      = 8,
1027     FL_CURSOR_S       = 9,
1028     FL_CURSOR_SW      = 7,
1029     FL_CURSOR_W       = 36,
1030     FL_CURSOR_NW      = 68,
1031     FL_CURSOR_NONE    = 255
1032 }; // group: Cursors
1033
1034 enum { // values for "when" passed to Fl::add_fd()
1035     FL_READ   = 1,
1036     FL_WRITE  = 4,
1037     FL_EXCEPT = 8
1038 };
1039
1040 enum Fl_Mode {
1041     FL_RGB      = 0,
1042     FL_INDEX    = 1,
1043     FL_SINGLE   = 0,
1044     FL_DOUBLE   = 2,
1045     FL_ACCUM    = 4,
1046     FL_ALPHA    = 8,
1047     FL_DEPTH    = 16,
1048     FL_STENCIL  = 32,
1049     FL_RGB8     = 64,
1050     FL_MULTISAMPLE= 128,

```

```

1093  FL_STEREO      = 256,
1094  FL_FAKE_SINGLE = 512, // Fake single buffered windows using double-buffer
1095  FL_OPENGL3     = 1024
1096 };
1097
1098 // image alpha blending
1099
1100 #define FL_IMAGE_WITH_ALPHA 0x40000000
1101
1102 enum Fl_Damage {
1103     FL_DAMAGE_CHILD      = 0x01,
1104     FL_DAMAGE_EXPOSE     = 0x02,
1105     FL_DAMAGE_SCROLL     = 0x04,
1106     FL_DAMAGE_OVERLAY   = 0x08,
1107     FL_DAMAGE_USER1     = 0x10,
1108     FL_DAMAGE_USER2     = 0x20,
1109     FL_DAMAGE_ALL       = 0x80
1110 };
1111 };
1112
1113 // FLTK 1.0.x compatibility definitions...
1114 # ifdef FLTK_1_0_COMPAT
1115 #     define contrast      fl_contrast
1116 #     define down         fl_down
1117 #     define frame        fl_frame
1118 #     define inactive     fl_inactive
1119 # endif // FLTK_1_0_COMPAT
1120
1121 #endif
1122
1123 //
1124 // End of "$Id$".
1125 //

```

32.5 filename.H File Reference

File names and URI utility functions.

```

#include "Fl_Export.H"
#include <sys/types.h>
#include <dirent.h>

```

Macros

- #define **fl_dirent_h_cyclic_include**
- #define **FL_FILENAME_H**
- #define **FL_PATH_MAX** 2048
all path buffers should use this length

Typedefs

- typedef int() **Fl_File_Sort_F**(struct dirent **, struct dirent **)
File sorting function.

Functions

- FL_EXPORT void **fl_decode_uri** (char *uri)
Decodes a URL-encoded string.
- FL_EXPORT int **fl_filename_absolute** (char *to, int tolen, const char *from)
Makes a filename absolute from a relative filename.
- FL_EXPORT int **fl_filename_expand** (char *to, int tolen, const char *from)
Expands a filename containing shell variables and tilde (~).
- FL_EXPORT const char * **fl_filename_ext** (const char *buf)
Gets the extensions of a filename.
- FL_EXPORT void **fl_filename_free_list** (struct dirent ***l, int n)
Free the list of filenames that is generated by fl_filename_list().
- FL_EXPORT int **fl_filename_isdir** (const char *name)

- Determines if a file exists and is a directory from its filename.*

 - FL_EXPORT int `fl_filename_list` (const char *d, struct dirent ***, [FL_File_Sort_F](#) *s=fl_numericSort)

Portable and const-correct wrapper for the scandir() function.
- FL_EXPORT int `fl_filename_match` (const char *name, const char *pattern)

Checks if a string s matches a pattern p.
- FL_EXPORT const char * `fl_filename_name` (const char *filename)

Gets the file name from a path.
- FL_EXPORT int `fl_filename_relative` (char *to, int tolen, const char *from)

Makes a filename relative to the current working directory.
- FL_EXPORT char * `fl_filename_setext` (char *to, int tolen, const char *ext)

Replaces the extension in buf of max.
- FL_EXPORT int `fl_open_uri` (const char *uri, char *msg, int msglen)

Opens the specified Uniform Resource Identifier (URI).

32.5.1 Detailed Description

File names and URI utility functions.

32.6 filename.H

[Go to the documentation of this file.](#)

```

1 /*
2  * "$Id$"
3  *
4  * Filename header file for the Fast Light Tool Kit (FLTK).
5  *
6  * Copyright 1998-2010 by Bill Spitzak and others.
7  *
8  * This library is free software.  Distribution and use rights are outlined in
9  * the file "COPYING" which should have been included with this file.  If this
10 * file is missing or damaged, see the license at:
11 *
12 *     http://www.fltk.org/COPYING.php
13 *
14 * Please report all bugs and problems on the following page:
15 *
16 *     http://www.fltk.org/str.php
17 */
18
19 /* Xcode on OS X includes files by recursing down into directories.
20 * This code catches the cycle and directly includes the required file.
21 */
22 #ifndef fl_dirent_h_cyclic_include
23 # include "/usr/include/dirent.h"
24 #endif
25
26 #ifndef FL_FILENAME_H
27 # define FL_FILENAME_H
28
29 # include "Fl_Export.H"
30
31 # define FL_PATH_MAX 2048
32 FL_EXPORT const char *fl_filename_name(const char * filename);
33 FL_EXPORT const char *fl_filename_ext(const char *buf);
34 FL_EXPORT char *fl_filename_setext(char *to, int tolen, const char *ext);
35 FL_EXPORT int fl_filename_expand(char *to, int tolen, const char *from);
36 FL_EXPORT int fl_filename_absolute(char *to, int tolen, const char *from);
37 FL_EXPORT int fl_filename_relative(char *to, int tolen, const char *from);
38 FL_EXPORT int fl_filename_match(const char *name, const char *pattern);
39 FL_EXPORT int fl_filename_isdir(const char *name);
40
41 # if defined(__cplusplus) && !defined(FL_DOXYGEN)
42 /*
43 * Under WIN32, we include filename.H from numericSort.c; this should probably change...
44 */
45
46 inline char *fl_filename_setext(char *to, const char *ext) { return fl_filename_setext(to, FL_PATH_MAX,
47 ext); }
48 inline int fl_filename_expand(char *to, const char *from) { return fl_filename_expand(to, FL_PATH_MAX,
49 from); }
50 inline int fl_filename_absolute(char *to, const char *from) { return fl_filename_absolute(to,
51 FL_PATH_MAX, from); }
52 FL_EXPORT int fl_filename_relative(char *to, int tolen, const char *from, const char *cwd);

```

```

72 inline int fl_filename_relative(char *to, const char *from) { return fl_filename_relative(to,
    FL_PATH_MAX, from); }
73 # endif /* __cplusplus */
74
75
76 # if defined(WIN32) && !defined(__MINGW32__) && !defined(__CYGWIN__) && !defined(__WATCOMC__)
77
78 struct dirent {char d_name[1];};
79
80 # elif defined(__WATCOMC__)
81 #     include <sys/types.h>
82 #     include <direct.h>
83
84 # else
85 /*
86 * WARNING: on some systems (very few nowadays?) <dirent.h> may not exist.
87 * The correct information is in one of these files:
88 *
89 *     #include <sys/ndir.h>
90 *     #include <sys/dir.h>
91 *     #include <ndir.h>
92 *
93 * plus you must do the following #define:
94 *
95 *     #define dirent direct
96 *
97 * It would be best to create a <dirent.h> file that does this...
98 */
99 #     include <sys/types.h>
100 #     define fl_dirent_h_cyclic_include
101 #     include <dirent.h>
102 #     undef fl_dirent_h_cyclic_include
103 # endif
104
105 # if defined (__cplusplus)
106 extern "C" {
107 # endif /* __cplusplus */
108
109 # if !defined(FL_DOXYGEN)
110 FL_EXPORT int fl_alphasort(struct dirent **, struct dirent **);
111 FL_EXPORT int fl_casealphasort(struct dirent **, struct dirent **);
112 FL_EXPORT int fl_casenumERICsort(struct dirent **, struct dirent **);
113 FL_EXPORT int fl_numericSORT(struct dirent **, struct dirent **);
114 # endif
115
116 typedef int (Fl_File_Sort_F)(struct dirent **, struct dirent **);
117 # if defined(__cplusplus)
118 }
119
120
121 /*
122 * Portable "scandir" function.    Ugly but necessary...
123 */
124
125 FL_EXPORT int fl_filename_list(const char *d, struct dirent ***l,
126                               Fl_File_Sort_F *s = fl_numericSORT);
127 FL_EXPORT void fl_filename_free_list(struct dirent ***l, int n);
128
129 /*
130 * Generic function to open a Uniform Resource Identifier (URI) using a
131 * system-defined program (added in FLTK 1.1.8)
132 */
133
134 FL_EXPORT int fl_open_uri(const char *uri, char *msg = (char *)0,
135                          int msglen = 0);
136
137 FL_EXPORT void fl_decode_uri(char *uri);
138
139 # ifndef FL_DOXYGEN
140 /*
141 * _fl_filename_isdir_quick() is a private function that checks for a
142 * trailing slash and assumes that the passed name is a directory if
143 * it finds one.    This function is used by Fl_File_Browser and
144 * Fl_File_Chooser to avoid extra stat() calls, but is not supported
145 * outside of FLTK...
146 */
147 int _fl_filename_isdir_quick(const char *name);
148 # endif
149
150 # endif /* __cplusplus */
151
152 /*
153 * FLTK 1.0.x compatibility definitions...
154 */
155
156 # ifdef FLTK_1_0_COMPAT
157 #     define filename_absolute    fl_filename_absolute
158 #     define filename_expand      fl_filename_expand

```

```

159 #   define filename_ext      fl_filename_ext
160 #   define filename_isdir    fl_filename_isdir
161 #   define filename_list     fl_filename_list
162 #   define filename_match    fl_filename_match
163 #   define filename_name     fl_filename_name
164 #   define filename_relative fl_filename_relative
165 #   define filename_setext   fl_filename_setext
166 #   define numeric_sort     fl_numeric_sort
167 #   endif /* FLTK_1_0_COMPAT */
168
169
170 #endif /* FL_FILENAME_H */
171
172 /*
173 * End of "$Id$".
174 */

```

32.7 Fl.H File Reference

Fl static class.

```

#include <FL/Fl_Export.H>
#include <FL/Fl_Cairo.H>
#include "fl_utf8.h"
#include "Enumerations.H"

```

Classes

- class [Fl](#)
The Fl is the FLTK global (static) class containing state information and global methods for the current application.
- class [Fl_Widget_Tracker](#)
This class should be used to control safe widget deletion.

Macros

- #define [Fl_Object Fl_Widget](#)
for back compatibility - use Fl_Widget!
- #define [FL_SOCKET](#) int

Typedefs

- typedef void(* [Fl_Abort_Handler](#)) (const char *format,...)
Signature of set_abort functions passed as parameters.
- typedef int(* [Fl_Args_Handler](#)) (int argc, char **argv, int &i)
Signature of args functions passed as parameters.
- typedef void(* [Fl_Atclose_Handler](#)) ([Fl_Window](#) *window, void *data)
Signature of set_atclose functions passed as parameters.
- typedef void(* [Fl_Awake_Handler](#)) (void *data)
Signature of some wakeup callback functions passed as parameters.
- typedef void() [Fl_Box_Draw_F](#)(int x, int y, int w, int h, [Fl_Color](#) color)
Signature of some box drawing functions passed as parameters.
- typedef void(* [Fl_Clipboard_Notify_Handler](#)) (int source, void *data)
Signature of add_clipboard_notify functions passed as parameters.
- typedef int(* [Fl_Event_Dispatch](#)) (int event, [Fl_Window](#) *w)
Signature of event_dispatch functions passed as parameters.
- typedef int(* [Fl_Event_Handler](#)) (int event)
Signature of add_handler functions passed as parameters.
- typedef void(* [Fl_FD_Handler](#)) ([FL_SOCKET](#) fd, void *data)
Signature of add_fd functions passed as parameters.

- typedef void(* **Fl_Idle_Handler**) (void *data)
Signature of add_idle callback functions passed as parameters.
- typedef void() **Fl_Label_Draw_F**(const [Fl_Label](#) *label, int x, int y, int w, int h, [Fl_Align](#) align)
Signature of some label drawing functions passed as parameters.
- typedef void() **Fl_Label_Measure_F**(const [Fl_Label](#) *label, int &width, int &height)
Signature of some label measurement functions passed as parameters.
- typedef void(* **Fl_Old_Idle_Handler**) ()
Signature of set_idle callback functions passed as parameters.
- typedef int(* **Fl_System_Handler**) (void *event, void *data)
Signature of add_system_handler functions passed as parameters.
- typedef void(* **Fl_Timeout_Handler**) (void *data)
Signature of some timeout callback functions passed as parameters.

Variables

- FL_EXPORT const char * **fl_local_alt**
string pointer used in shortcuts, you can change it to another language
- FL_EXPORT const char * **fl_local_ctrl**
string pointer used in shortcuts, you can change it to another language
- FL_EXPORT const char * **fl_local_meta**
string pointer used in shortcuts, you can change it to another language
- FL_EXPORT const char * **fl_local_shift**
string pointer used in shortcuts, you can change it to another language

32.7.1 Detailed Description

[Fl](#) static class.

32.8 Fl.H

[Go to the documentation of this file.](#)

```

1 //
2 // "$Id$"
3 //
4 // Main header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2016 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file.  If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
23 #ifndef Fl_H
24 # define Fl_H
25
26 #include <FL/Fl_Export.H>
27
28 #ifdef FLTK_HAVE_CAIRO
29 # include <FL/Fl_Cairo.H>
30 #endif
31
32 # include "fl_utf8.h"
33 # include "Enumerations.H"
34 # ifndef Fl_Object
35 #   define Fl_Object Fl_Widget
36 # endif
37
38 # ifdef check
39 #   undef check
40 # endif

```

```

41
42
43 class Fl_Widget;
44 class Fl_Window;
45 class Fl_Image;
46 struct Fl_Label;
47
48 // Keep avoiding having the socket deps at that level but mke sure it will work in both 32 & 64 bit
   builds
49 #if defined(WIN32) && !defined(__CYGWIN__)
50 # if defined(_WIN64)
51 # define FL_SOCKET unsigned __int64
52 # else
53 # define FL_SOCKET int
54 # endif
55 #else
56 # define FL_SOCKET int
57 #endif
58
59
60 // Pointers you can use to change FLTK to a foreign language.
61 // Note: Similar pointers are defined in FL/fl_ask.H and src/fl_ask.cxx
62 extern FL_EXPORT const char* fl_local_ctrl;
63 extern FL_EXPORT const char* fl_local_meta;
64 extern FL_EXPORT const char* fl_local_alt;
65 extern FL_EXPORT const char* fl_local_shift;
66
67
68
69
70
71
72
73 typedef void (Fl_Label_Draw_F)(const Fl_Label *label, int x, int y, int w, int h, Fl_Align align);
74
75
76 typedef void (Fl_Label_Measure_F)(const Fl_Label *label, int &width, int &height);
77
78
79 typedef void (Fl_Box_Draw_F)(int x, int y, int w, int h, Fl_Color color);
80
81
82
83
84
85
86
87
88
89
90
91
92 typedef void (*Fl_Timeout_Handler)(void *data);
93
94
95 typedef void (*Fl_Awake_Handler)(void *data);
96
97
98 typedef void (*Fl_Idle_Handler)(void *data);
99
100
101 typedef void (*Fl_Old_Idle_Handler)();
102
103
104 typedef void (*Fl_FD_Handler)(FL_SOCKET fd, void *data);
105
106
107 typedef int (*Fl_Event_Handler)(int event);
108
109
110 typedef int (*Fl_System_Handler)(void *event, void *data);
111
112
113 typedef void (*Fl_Abort_Handler)(const char *format,...);
114
115
116 typedef void (*Fl_Atclose_Handler)(Fl_Window *window, void *data);
117
118
119 typedef int (*Fl_Args_Handler)(int argc, char **argv, int &i);
120
121
122 typedef int (*Fl_Event_Dispatch)(int event, Fl_Window *w);
123
124
125 typedef void (*Fl_Clipboard_Notify_Handler)(int source, void *data);
126 /* group callback_functions */
127
128
129
130
131
132
133
134
135 class FL_EXPORT Fl {
136   Fl() {}; // no constructor!
137
138 private:
139   static int use_high_res_GL_;
140
141 public: // should be private!
142 #ifndef FL_DOXYGEN
143   static int e_number;
144   static int e_x;
145   static int e_y;
146   static int e_x_root;
147   static int e_y_root;
148   static int e_dx;
149   static int e_dy;
150   static int e_state;
151   static int e_clicks;
152   static int e_is_click;
153   static int e_keysym;
154   static char* e_text;
155   static int e_length;
156   static void *e_clipboard_data;
157   static const char *e_clipboard_type;
158   static Fl_Event_Dispatch e_dispatch;
159   static Fl_Widget* belowmouse_;
160   static Fl_Widget* pushed_;
161   static Fl_Widget* focus_;
162   static int damage_;

```

```

163 static Fl_Widget* selection_owner_;
164 static Fl_Window* modal_;
165 static Fl_Window* grab_;
166 static int compose_state; // used for dead keys (WIN32) or marked text (MacOS)
167 static void call_screen_init(); // recompute screen number and dimensions
168 #ifdef __APPLE__
169 static void reset_marked_text(); // resets marked text
170 static void insertion_point_location(int x, int y, int height); // sets window coordinates & height of
    insertion point
171 #endif
172 #endif // FL_DOXYGEN
173
174
175 static void damage(int d) {damage_ = d;}
176
177 public:
178 typedef enum {
179     OPTION_ARROW_FOCUS = 0,
180     // When switched on, FLTK will use the file chooser dialog that comes
181     // with your operating system whenever possible. When switched off, FLTK
182     // will present its own file chooser.
183     // \todo implement me
184     // OPTION_NATIVE_FILECHOOSER,
185     // When Filechooser Preview is enabled, the FLTK or native file chooser
186     // will show a preview of a selected file (if possible) before the user
187     // decides to choose the file.
188     // \todo implement me
189     //OPTION_FILECHOOSER_PREVIEW,
190     OPTION_VISIBLE_FOCUS,
191     OPTION_DND_TEXT,
192     OPTION_SHOW_TOOLTIPS,
193     OPTION_FNFC_USES_GTK,
194     // don't change this, leave it always as the last element
195     OPTION_LAST
196 } Fl_Option;
197
198 private:
199 static unsigned char options_[OPTION_LAST];
200 static unsigned char options_read_;
201
202 public:
203 /*
204 Return a global setting for all FLTK applications, possibly overridden
205 by a setting specifically for this application.
206 */
207 static bool option(Fl_Option opt);
208
209 /*
210 Override an option while the application is running.
211 */
212 static void option(Fl_Option opt, bool val);
213
214 static void (*idle)();
215
216 #ifndef FL_DOXYGEN
217 static Fl_Awake_Handler *awake_ring_;
218 static void **awake_data_;
219 static int awake_ring_size_;
220 static int awake_ring_head_;
221 static int awake_ring_tail_;
222 static const char* scheme_;
223 static Fl_Image* scheme_bg_;
224
225 static int e_original_keysym; // late addition
226 static int scrollbar_size_;
227 #endif
228
229 static int add_aware_handler_(Fl_Awake_Handler, void*);
230 static int get_aware_handler_(Fl_Awake_Handler&, void*&);
231
232 public:
233 // API version number
234 static double version();
235 static int api_version();
236
237 // ABI version number
238 static int abi_version();
239
240 static inline int abi_check(const int val = FL_ABI_VERSION) {
241     return val == abi_version();
242 }
243
244 // argument parsers:
245 static int arg(int argc, char **argv, int& i);
246 static int args(int argc, char **argv, int& i, Fl_Args_Handler cb = 0);

```



```

310 static void args(int argc, char **argv);
311 static const char* const help;
312
313 // things called by initialization:
314 static void display(const char*);
315 static int visual(int);
316 static int gl_visual(int, int *alist=0); // platform dependent
317 static void own_colormap();
318 static void get_system_colors();
319 static void foreground(uchar, uchar, uchar);
320 static void background(uchar, uchar, uchar);
321 static void background2(uchar, uchar, uchar);
322
323 // schemes:
324 static int scheme(const char *name);
325 static const char* scheme() {return scheme_;}
326
327 static int is_scheme(const char *name) {
328     return (scheme_ && name && !strcmp(name,scheme_));
329 }
330 static int reload_scheme(); // platform dependent
331 static int scrollbar_size();
332 static void scrollbar_size(int W);
333
334 // execution:
335 static int wait();
336 static double wait(double time);
337 static int check();
338 static int ready();
339 static int run();
340 static Fl_Widget* readqueue();
341 static void add_timeout(double t, Fl_Timeout_Handler,void* = 0); // platform dependent
342 static void repeat_timeout(double t, Fl_Timeout_Handler, void* = 0); // platform dependent
343 static int has_timeout(Fl_Timeout_Handler, void* = 0);
344 static void remove_timeout(Fl_Timeout_Handler, void* = 0);
345 static void add_check(Fl_Timeout_Handler, void* = 0);
346 static int has_check(Fl_Timeout_Handler, void* = 0);
347 static void remove_check(Fl_Timeout_Handler, void* = 0);
348 static void add_fd(int fd, int when, Fl_FD_Handler cb, void* = 0); // platform dependent
349 static void add_fd(int fd, Fl_FD_Handler cb, void* = 0); // platform dependent
350 static void remove_fd(int, int when); // platform dependent
351 static void remove_fd(int); // platform dependent
352
353 static void add_idle(Fl_Idle_Handler cb, void* data = 0);
354 static int has_idle(Fl_Idle_Handler cb, void* data = 0);
355 static void remove_idle(Fl_Idle_Handler cb, void* data = 0);
356 static int damage() {return damage_;}
357 static void redraw();
358 static void flush();
359 static void (*warning)(const char*, ...);
360 static void (*error)(const char*, ...);
361 static void (*fatal)(const char*, ...);
362 static Fl_Window* first_window();
363 static void first_window(Fl_Window*);
364 static Fl_Window* next_window(const Fl_Window*);
365
366 static Fl_Window* modal() {return modal_;}
367 static Fl_Window* grab() {return grab_;}
368 static void grab(Fl_Window*); // platform dependent
369 // event information:
370 static int event() {return e_number;}
371 static int event_x() {return e_x;}
372 static int event_y() {return e_y;}
373 static int event_x_root() {return e_x_root;}
374 static int event_y_root() {return e_y_root;}
375 static int event_dx() {return e_dx;}
376 static int event_dy() {return e_dy;}
377 static void get_mouse(int &,int &); // platform dependent
378 static int event_clicks() {return e_clicks;}
379 static void event_clicks(int i) {e_clicks = i;}
380 static int event_is_click() {return e_is_click;}
381 static void event_is_click(int i) {e_is_click = i;}
382 static int event_button() {return e_keysym-FL_Button;}
383 static int event_state() {return e_state;}
384
385 static int event_state(int mask) {return e_state&mask;}
386 static int event_key() {return e_keysym;}
387 static int event_original_key() {return e_original_keysym;}
388 static int event_key(int key);
389 static int get_key(int key); // platform dependent
390 static const char* event_text() {return e_text;}
391 static int event_length() {return e_length;}
392
393 static void *event_clipboard() { return e_clipboard_data; }
394 static const char *event_clipboard_type() {return e_clipboard_type; }
395
396
397
398
399
400

```

```

811 static int compose(int &del);
812 static void compose_reset();
813 static int event_inside(int,int,int,int);
814 static int event_inside(const Fl_Widget*);
815 static int test_shortcut(Fl_Shortcut);
816
821 static void enable_im();
826 static void disable_im();
827
828 // event destinations:
829 static int handle(int, Fl_Window*);
830 static int handle_(int, Fl_Window*);
833 static Fl_Widget* belowmouse() {return belowmouse_;}
834 static void belowmouse(Fl_Widget*);
837 static Fl_Widget* pushed() {return pushed_;}
838 static void pushed(Fl_Widget*);
840 static Fl_Widget* focus() {return focus_;}
841 static void focus(Fl_Widget*);
842 static void add_handler(Fl_Event_Handler h);
843 static void remove_handler(Fl_Event_Handler h);
844 static void add_system_handler(Fl_System_Handler h, void *data);
845 static void remove_system_handler(Fl_System_Handler h);
846 static void event_dispatch(Fl_Event_Dispatch d);
847 static Fl_Event_Dispatch event_dispatch();
853 // cut/paste:
869 #if FLTK_ABI_VERSION >= 10303 || defined(FL_DOXYGEN)
870 static void copy(const char* stuff, int len, int destination = 0, const char *type =
    Fl::clipboard_plain_text); // platform dependent
871 #else
872 static void copy(const char* stuff, int len, int destination, const char *type);
873 static void copy(const char* stuff, int len, int destination = 0);
874 #endif
875
876 #if !(defined(__APPLE__) || defined(WIN32) || defined(FL_DOXYGEN))
877 static void copy_image(const unsigned char* data, int W, int H, int destination = 0); // platform
    dependent
878 #endif
917 #if FLTK_ABI_VERSION >= 10303 || defined(FL_DOXYGEN)
918 static void paste(Fl_Widget &receiver, int source, const char *type = Fl::clipboard_plain_text); //
    platform dependent
919 #else
920 static void paste(Fl_Widget &receiver, int source, const char *type);
921 static void paste(Fl_Widget &receiver, int source /*=0*/);
922 #endif
944 static void add_clipboard_notify(Fl_Clipboard_Notify_Handler h, void *data = 0);
949 static void remove_clipboard_notify(Fl_Clipboard_Notify_Handler h);
953 static int clipboard_contains(const char *type);
956 char const * const clipboard_plain_text;
959 char const * const clipboard_image;
960
970 static int dnd(); // platform dependent
971
972 // These are for back-compatibility only:
975 static Fl_Widget* selection_owner() {return selection_owner_;}
976 static void selection_owner(Fl_Widget*);
977 static void selection(Fl_Widget &owner, const char*, int len);
978 static void paste(Fl_Widget &receiver);
983 // screen size:
985 static int x(); // platform dependent
987 static int y(); // platform dependent
989 static int w(); // platform dependent
991 static int h(); // platform dependent
992
993 // multi-head support:
994 static int screen_count();
1000 static void screen_xywh(int &X, int &Y, int &W, int &H) {
1001     int x, y;
1002     Fl::get_mouse(x, y);
1003     screen_xywh(X, Y, W, H, x, y);
1004 }
1005 static void screen_xywh(int &X, int &Y, int &W, int &H, int mx, int my);
1006 static void screen_xywh(int &X, int &Y, int &W, int &H, int n);
1007 static void screen_xywh(int &X, int &Y, int &W, int &H, int mx, int my, int mw, int mh);
1008 static int screen_num(int x, int y);
1009 static int screen_num(int x, int y, int w, int h);
1010 static void screen_dpi(float &h, float &v, int n=0);
1011 static void screen_work_area(int &X, int &Y, int &W, int &H, int mx, int my);
1012 static void screen_work_area(int &X, int &Y, int &W, int &H, int n);
1018 static void screen_work_area(int &X, int &Y, int &W, int &H) {
1019     int x, y;
1020     Fl::get_mouse(x, y);
1021     screen_work_area(X, Y, W, H, x, y);
1022 }
1023
1031 // color map:
1032 static void set_color(Fl_Color, uchar, uchar, uchar);
1037 static void set_color(Fl_Color i, unsigned c); // platform dependent

```

```

1038 static unsigned get_color(Fl_Color i);
1039 static void get_color(Fl_Color i, uchar &red, uchar &green, uchar &blue);
1045 static void free_color(Fl_Color i, int overlay = 0); // platform dependent
1046
1047 // fonts:
1048 static const char* get_font(Fl_Font);
1061 static const char* get_font_name(Fl_Font, int* attributes = 0);
1073 static int get_font_sizes(Fl_Font, int*& sizep);
1074 static void set_font(Fl_Font, const char*);
1075 static void set_font(Fl_Font, Fl_Font);
1094 static Fl_Font set_fonts(const char* = 0); // platform dependent
1095
1102 // <Hack to re-order the 'Drawing functions' group>
1105 // labeltypes:
1106 static void set_labeltype(Fl_Labeltype, Fl_Label_Draw_F*, Fl_Label_Measure_F*);
1108 static void set_labeltype(Fl_Labeltype, Fl_Labeltype from); // is it defined?
1109
1110 // boxtypes:
1111 static Fl_Box_Draw_F *get_boxtype(Fl_Boxtype);
1112 static void set_boxtype(Fl_Boxtype, Fl_Box_Draw_F*, uchar, uchar, uchar);
1113 static void set_boxtype(Fl_Boxtype, Fl_Boxtype from);
1114 static int box_dx(Fl_Boxtype);
1115 static int box_dy(Fl_Boxtype);
1116 static int box_dw(Fl_Boxtype);
1117 static int box_dh(Fl_Boxtype);
1118
1119 static int draw_box_active();
1120 static Fl_Color box_color(Fl_Color);
1121 static void set_box_color(Fl_Color);
1122
1123 // back compatibility:
1127 static void set_abort(Fl_Abort_Handler f) {fatal = f;}
1128 static void (*atclose)(Fl_Window*, void*);
1129 static void default_atclose(Fl_Window*, void*);
1133 static void set_atclose(Fl_Atclose_Handler f) {atclose = f;}
1139 static int event_shift() {return e_state&FL_SHIFT;}
1141 static int event_ctrl() {return e_state&FL_CTRL;}
1143 static int event_command() {return e_state&FL_COMMAND;}
1145 static int event_alt() {return e_state&FL_ALT;}
1154 static int event_buttons() {return e_state&0x7f000000;}
1159 static int event_button1() {return e_state&FL_BUTTON1;}
1164 static int event_button2() {return e_state&FL_BUTTON2;}
1169 static int event_button3() {return e_state&FL_BUTTON3;}
1177 static void set_idle(Fl_Old_Idle_Handler cb) {idle = cb;}
1179 static void grab(Fl_Window& win) {grab(&win);}
1183 static void release() {grab(0);}
1184
1185 // Visible focus methods...
1191 static void visible_focus(int v) { option(OPTION_VISIBLE_FOCUS, (v!=0)); }
1197 static int visible_focus() { return option(OPTION_VISIBLE_FOCUS); }
1198
1199 // Drag-n-drop text operation methods...
1206 static void dnd_text_ops(int v) { option(OPTION_DND_TEXT, (v!=0)); }
1213 static int dnd_text_ops() { return option(OPTION_DND_TEXT); }
1218 // Multithreading support:
1219 static int lock();
1220 static void unlock();
1221 static void awake(void* message = 0);
1223 static int awake(Fl_Awake_Handler cb, void* message = 0);
1230 static void* thread_message(); // platform dependent
1262 // Widget deletion:
1263 static void delete_widget(Fl_Widget *w);
1264 static void do_widget_deletion();
1265 static void watch_widget_pointer(Fl_Widget *&w);
1266 static void release_widget_pointer(Fl_Widget *&w);
1267 static void clear_widget_pointer(Fl_Widget const *w);
1274 static void use_high_res_GL(int val) { use_high_res_GL_ = val; }
1280 static int use_high_res_GL() { return use_high_res_GL_; }
1281
1282 #ifdef FLTK_HAVE_CAIRO
1286 public:
1287 // Cairo support API
1288 static cairo_t * cairo_make_current(Fl_Window* w);
1303 static void cairo_autolink_context(bool alink) {cairo_state_.autolink(alink);}
1311 static bool cairo_autolink_context() {return cairo_state_.autolink();}
1313 static cairo_t * cairo_cc() { return cairo_state_.cc(); }
1318 static void cairo_cc(cairo_t * c, bool own=false){ cairo_state_.cc(c, own); }
1319
1320 private:
1321 static cairo_t * cairo_make_current(void* gc);
1322 static cairo_t * cairo_make_current(void* gc, int W, int H);
1323 static Fl_Cairo_State cairo_state_;
1324 public:
1327 #endif // FLTK_HAVE_CAIRO
1328
1329 };
1330

```

```

1371 class FL_EXPORT Fl_Widget_Tracker {
1372
1373     Fl_Widget* wp_;
1374
1375 public:
1376
1377     Fl_Widget_Tracker(Fl_Widget *wi);
1378     ~Fl_Widget_Tracker();
1379
1385     Fl_Widget *widget() {return wp_;}
1386
1396     int deleted() {return wp_ == 0;}
1397
1407     int exists() {return wp_ != 0;}
1408
1409 };
1410
1416 #endif // !Fl_H
1417
1418 //
1419 // End of "$Id$".
1420 //

```

32.9 Fl_Adjuster.H

```

1 //
2 // "$Id$"
3 //
4 // Adjuster widget header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Adjuster widget . */
21
22 // 3-button "slider", made for Nuke
23
24 #ifndef Fl_Adjuster_H
25 #define Fl_Adjuster_H
26
27 #ifndef Fl_Valuator_H
28 #include "Fl_Valuator.H"
29 #endif
30
43 class FL_EXPORT Fl_Adjuster : public Fl_Valuator {
44     int drag;
45     int ix;
46     int soft_;
47 protected:
48     void draw();
49     int handle(int);
50     void value_damage();
51 public:
52     Fl_Adjuster(int X,int Y,int W,int H,const char *l=0);
59     void soft(int s) {soft_ = s;}
66     int soft()const {return soft_;}
67 };
68
69 #endif
70
71 //
72 // End of "$Id$".
73 //

```

32.10 fl_ask.H File Reference

API for common dialogs.

```
#include "Enumerations.H"
```

Macros

- #define `__fl_attr(x)`

Enumerations

- enum `FL_Beep` {
`FL_BEEP_DEFAULT = 0`, `FL_BEEP_MESSAGE`, `FL_BEEP_ERROR`, `FL_BEEP_QUESTION`,
`FL_BEEP_PASSWORD`, `FL_BEEP_NOTIFICATION` }

Different system beeps available.

Functions

- FL_EXPORT void FL_EXPORT void **fl_alert** (const char *,...) `__fl_attr((__format__ (__printf__`
- FL_EXPORT void FL_EXPORT void FL_EXPORT int **fl_ask** (const char *,...) `__fl_attr((__format__ (__printf__`
- FL_EXPORT void **fl_beep** (int type=`FL_BEEP_DEFAULT`)
Emits a system beep message.
- FL_EXPORT int **fl_choice** (const char *q, const char *b0, const char *b1, const char *b2,...) `__fl_attr((__format__ (__printf__`
- FL_EXPORT int FL_EXPORT const char FL_EXPORT const char FL_EXPORT int **fl_choice_n** (const char *q, const char *b0, const char *b1, const char *b2,...) `__fl_attr((__format__ (__printf__`
- FL_EXPORT int FL_EXPORT const char * **fl_input** (const char *label, const char *deflt=0,...) `__fl_attr((__format__ (__printf__`
- FL_EXPORT void **fl_message** (const char *,...) `__fl_attr((__format__ (__printf__`
- void **fl_message_font** (`FL_Font` f, `FL_Fontsize` s)
- FL_EXPORT void **fl_message_hotspot** (int enable)
Sets whether or not to move the common message box used in many common dialogs like [fl_message\(\)](#), [fl_alert\(\)](#), [fl_ask\(\)](#), [fl_choice\(\)](#), [fl_input\(\)](#), [fl_password\(\)](#) to follow the mouse pointer.
- FL_EXPORT int **fl_message_hotspot** (void)
Gets whether or not to move the common message box used in many common dialogs like [fl_message\(\)](#), [fl_alert\(\)](#), [fl_ask\(\)](#), [fl_choice\(\)](#), [fl_input\(\)](#), [fl_password\(\)](#) to follow the mouse pointer.
- FL_EXPORT int FL_EXPORT const char FL_EXPORT const char FL_EXPORT int FL_EXPORT `FL_Widget` * **fl_message_icon** ()
Gets the [FL_Box](#) icon container of the current default dialog used in many common dialogs like [fl_message\(\)](#), [fl_alert\(\)](#), [fl_ask\(\)](#), [fl_choice\(\)](#), [fl_input\(\)](#), [fl_password\(\)](#)
- FL_EXPORT void **fl_message_title** (const char *title)
Sets the title of the dialog window used in many common dialogs.
- FL_EXPORT void **fl_message_title_default** (const char *title)
Sets the default title of the dialog window used in many common dialogs.
- FL_EXPORT int FL_EXPORT const char FL_EXPORT const char * **fl_password** (const char *label, const char *deflt=0,...) `__fl_attr((__format__ (__printf__`

Variables

- FL_EXPORT void FL_EXPORT void FL_EXPORT int `__deprecated__`
- FL_EXPORT const char * **fl_cancel**
string pointer used in common dialogs, you can change it to another language
- FL_EXPORT const char * **fl_close**
string pointer used in common dialogs, you can change it to another language
- FL_EXPORT `FL_Font` **fl_message_font_**
- FL_EXPORT `FL_Fontsize` **fl_message_size_**
- FL_EXPORT const char * **fl_no**
string pointer used in common dialogs, you can change it to another language
- FL_EXPORT const char * **fl_ok**

string pointer used in common dialogs, you can change it to another language

- FL_EXPORT const char * **fl_yes**

string pointer used in common dialogs, you can change it to another language

32.10.1 Detailed Description

API for common dialogs.

32.10.2 Enumeration Type Documentation

32.10.2.1 Fl_Beep

enum [Fl_Beep](#)

Different system beeps available.

See also

[fl_beep\(int\)](#)

Enumerator

FL_BEEP_DEFAULT	Default beep.
FL_BEEP_MESSAGE	Message beep.
FL_BEEP_ERROR	Error beep.
FL_BEEP_QUESTION	Question beep.
FL_BEEP_PASSWORD	Password beep.
FL_BEEP_NOTIFICATION	Notification beep.

32.11 fl_ask.H

[Go to the documentation of this file.](#)

```

1 //
2 // "$Id$"
3 //
4 // Standard dialog header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2011 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file.  If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 #ifndef fl_ask_H
20 # define fl_ask_H
21
22 # include "Enumerations.H"
23
24 class Fl_Widget;
25
26 enum Fl_Beep {
27     FL_BEEP_DEFAULT = 0,
28     FL_BEEP_MESSAGE,
29     FL_BEEP_ERROR,
30     FL_BEEP_QUESTION,
31     FL_BEEP_PASSWORD,
32     FL_BEEP_NOTIFICATION
33 };

```

```

40
41 # ifdef __GNUC__
42 /* the GNUC-specific attribute appearing below in prototypes with a variable list of arguments
43 helps detection of mismatches between format string and argument list at compilation time */
44 #   define __fl_attr(x) __attribute__ (x)
45 #   else
46 #   define __fl_attr(x)
47 #   endif // __GNUC__
48
49 FL_EXPORT void fl_beep(int type = FL_BEEP_DEFAULT);
50 FL_EXPORT void fl_message(const char *,...) __fl_attr((__format__ (__printf__, 1, 2)));
51 FL_EXPORT void fl_alert(const char *,...) __fl_attr((__format__ (__printf__, 1, 2)));
52 // fl_ask() is deprecated since it uses "Yes" and "No" for the buttons,
53 // which does not conform to the current FLTK Human Interface Guidelines.
54 // Use fl_choice() instead with the appropriate verbs instead.
55 FL_EXPORT int fl_ask(const char *,...) __fl_attr((__format__ (__printf__, 1, 2), __deprecated__));
56 FL_EXPORT int fl_choice(const char *q,const char *b0,const char *b1,const char *b2,...)
57   __fl_attr((__format__ (__printf__, 1, 5)));
57 FL_EXPORT const char *fl_input(const char *label, const char *deflt = 0, ...) __fl_attr((__format__
58   (__printf__, 1, 3)));
58 FL_EXPORT const char *fl_password(const char *label, const char *deflt = 0, ...) __fl_attr((__format__
59   (__printf__, 1, 3)));
60 // since FLTK 1.3.8:
61 FL_EXPORT int fl_choice_n(const char *q,const char *b0,const char *b1,const char *b2,...)
62   __fl_attr((__format__ (__printf__, 1, 5)));
62
63 FL_EXPORT Fl_Widget *fl_message_icon();
64 extern FL_EXPORT Fl_Font fl_message_font_;
65 extern FL_EXPORT Fl_Fontsize fl_message_size_;
66 inline void fl_message_font(Fl_Font f, Fl_Fontsize s) {
67   fl_message_font_ = f; fl_message_size_ = s;}
68
69 FL_EXPORT void fl_message_hotspot(int enable);
70 FL_EXPORT int fl_message_hotspot(void);
71
72 FL_EXPORT void fl_message_title(const char *title);
73 FL_EXPORT void fl_message_title_default(const char *title);
74
75 // pointers you can use to change FLTK to a foreign language:
76 extern FL_EXPORT const char* fl_no;
77 extern FL_EXPORT const char* fl_yes;
78 extern FL_EXPORT const char* fl_ok;
79 extern FL_EXPORT const char* fl_cancel;
80 extern FL_EXPORT const char* fl_close;
81 #endif // !fl_ask_H
82
83 //
84 // End of "$Id$".
85 //

```

32.12 Fl_Bitmap.H

```

1 //
2 // "$Id$"
3 //
4 // Bitmap header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 //   http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //   http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Bitmap widget . */
21
22 #ifndef Fl_Bitmap_H
23 #define Fl_Bitmap_H
24 # include "Fl_Image.H"
25
26 class Fl_Widget;
27 struct Fl_Menu_Item;
28
29 class FL_EXPORT Fl_Bitmap : public Fl_Image {
30   friend class Fl_Quartz_Graphics_Driver;
31   friend class Fl_GDI_Graphics_Driver;
32   friend class Fl_GDI_Printer_Graphics_Driver;

```

```

37 friend class Fl_Xlib_Graphics_Driver;
38 public:
39
40 const uchar *array;
41 int alloc_array;
42
43 private:
44 int start(int XP, int YP, int WP, int HP, int &cx, int &cy,
45           int &X, int &Y, int &W, int &H);
46 #if defined(__APPLE__) || defined(WIN32)
47 void *id_;
48 #else
49 unsigned id_;
50 #endif // __APPLE__ || WIN32
51
52 public:
53 Fl_Bitmap(const uchar *bits, int W, int H) :
54   Fl_Image(W,H,0), array(bits), alloc_array(0), id_(0) {data((const char **)&array, 1);}
55 Fl_Bitmap(const char *bits, int W, int H) :
56   Fl_Image(W,H,0), array((const uchar *)bits), alloc_array(0), id_(0) {data((const char **)&array, 1);}
57 virtual ~Fl_Bitmap();
58 virtual Fl_Image *copy(int W, int H);
59 Fl_Image *copy() { return copy(w(), h()); }
60 virtual void draw(int X, int Y, int W, int H, int cx=0, int cy=0);
61 void draw(int X, int Y) {draw(X, Y, w(), h(), 0, 0);}
62 virtual void label(Fl_Widget*w);
63 virtual void label(Fl_Menu_Item*m);
64 virtual void uncache();
65 };
66
67 #endif
68 //
69 // End of "$Id$".
70 //

```

32.13 Fl_BMP_Image.H

```

1 //
2 // "$Id$"
3 //
4 // BMP image header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 // http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 // http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_BMP_Image widget . */
21
22 #ifndef Fl_BMP_Image_H
23 #define Fl_BMP_Image_H
24 # include "Fl_Image.H"
25
26 class FL_EXPORT Fl_BMP_Image : public Fl_RGB_Image {
27 public:
28   Fl_BMP_Image(const char* filename);
29 };
30 #endif
31 //
32 // End of "$Id$".
33 //

```

32.14 Fl_Box.H

```

1 //
2 // "$Id$"
3 //

```



```

4 // Box header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software.  Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file.  If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Box widget . */
21
22 #ifndef Fl_Box_H
23 #define Fl_Box_H
24
25 #ifndef Fl_Widget_H
26 #include "Fl_Widget.H"
27 #endif
28
29 class FL_EXPORT Fl_Box : public Fl_Widget {
30 protected:
31     void draw();
32 public:
33     Fl_Box(int X, int Y, int W, int H, const char *l=0);
34     Fl_Box(Fl_Boxtype b, int X, int Y, int W, int H, const char *l);
35     virtual int handle(int);
36 };
37
38 #endif
39 //
40 // End of "$Id$".
41 //

```

32.15 Fl_Browser.H

```

1 //
2 // "$Id$"
3 //
4 // Browser header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2016 by Bill Spitzak and others.
7 //
8 // This library is free software.  Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file.  If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Browser widget . */
21
22 // Forms-compatible browser.  Probably useful for other
23 // lists of textual data.  Notice that the line numbers
24 // start from 1, and 0 means "no line".
25
26 #ifndef Fl_Browser_H
27 #define Fl_Browser_H
28
29 #include "Fl_Browser.H"
30 #include "Fl_Image.H"
31
32 struct FL_BLINE;
33
34 class FL_EXPORT Fl_Browser : public Fl_Browser_ {
35     FL_BLINE *first;           // the array of lines
36     FL_BLINE *last;
37     FL_BLINE *cache;
38     int cacheline;           // line number of cache
39     int lines;               // Number of lines
40     int full_height_;

```

```

88  const int* column_widths_;
89  char format_char_;           // alternative to @-sign
90  char column_char_;         // alternative to tab
91
92  protected:
93
94  // required routines for Fl_Browser_ subclass:
95  void* item_first() const ;
96  void* item_next(void* item) const ;
97  void* item_prev(void* item) const ;
98  void* item_last()const ;
99  int item_selected(void* item) const ;
100 void item_select(void* item, int val);
101 int item_height(void* item) const ;
102 int item_width(void* item) const ;
103 void item_draw(void* item, int X, int Y, int W, int H) const ;
104 int full_height() const ;
105 int incr_height() const ;
106 const char *item_text(void *item) const;
112 void item_swap(void *a, void *b) { swap((FL_BLINE*)a, (FL_BLINE*)b); }
118 void *item_at(int line)const { return (void*)find_line(line); }
119
120 FL_BLINE* find_line(int line) const ;
121 FL_BLINE* _remove(int line) ;
122 void insert(int line, FL_BLINE* item);
123 int lineno(void *item) const ;
124 void swap(FL_BLINE *a, FL_BLINE *b);
125
126 public:
127
128 void remove(int line);
129 void add(const char* newtext, void* d = 0);
130 void insert(int line, const char* newtext, void* d = 0);
131 void move(int to, int from);
132 int load(const char* filename);
133 void swap(int a, int b);
134 void clear();
135
141 int size()const { return lines; }
142 void size(int W, int H) { Fl_Widget::size(W, H); }
143
147 Fl_Fontsize textsize()const { return Fl_Browser_::textsize(); }
148
149 /*
150 Sets the default text size for the lines in the browser to newSize.
151 Defined and documented in Fl_Browser.cxx
152 */
153 void textsize(Fl_Fontsize newSize);
154
155 int topline() const ;
156 enum Fl_Line_Position { TOP, BOTTOM, MIDDLE };
157 void lineposition(int line, Fl_Line_Position pos);
158 void topline(int line) { lineposition(line, TOP); }
159 void bottomline(int line) { lineposition(line, BOTTOM); }
160 void middleline(int line) { lineposition(line, MIDDLE); }
161
181 int select(int line, int val=1);
182 int selected(int line) const ;
183 void show(int line);
184 void show() { Fl_Widget::show(); }
185 void hide(int line);
186 void hide() { Fl_Widget::hide(); }
187 int visible(int line) const ;
188
191 int value() const ;
192 void value(int line) { select(line); }
193 const char* text(int line) const ;
194 void text(int line, const char* newtext);
195 void* data(int line) const ;
196 void data(int line, void* d);
197
203 Fl_Browser(int X, int Y, int W, int H, const char *L = 0);
204 ~Fl_Browser() { clear(); }
205
238 char format_char()const { return format_char_; }
239 void format_char(char c) { format_char_ = c; }
240 char column_char()const { return column_char_; }
241 void column_char(char c) { column_char_ = c; }
242 const int* column_widths()const { return column_widths_; }
243 void column_widths(const int* arr) { column_widths_ = arr; }
244
297 int displayed(int line)const { return Fl_Browser_::displayed(find_line(line)); }
298
306 void make_visible(int line) {
307     if (line < 1) Fl_Browser_::display(find_line(1));
308     else if (line > lines) Fl_Browser_::display(find_line(lines));
309     else Fl_Browser_::display(find_line(line));

```

```

310 }
311
312 // icon support
313 void icon(int line, Fl_Image* icon);
314 Fl_Image* icon(int line) const;
315 void remove_icon(int line);
316
317 void replace(int a, const char* b) { text(a, b); }
318 void display(int line, int val=1);
319 };
320 };
321
322 #endif
323
324 //
325 // End of "$Id$".
326 //

```

32.16 Fl_Browser_.H

```

1 //
2 // "$Id$"
3 //
4 // Common browser header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2016 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Browser_ widget . */
21
22 // Yes, I know this should be a template...
23
24 #ifndef Fl_Browser__H
25 #define Fl_Browser__H
26
27 #ifndef Fl_Group_H
28 #include "Fl_Group.H"
29 #endif
30 #include "Fl_Scrollbar.H"
31 #include <FL/Fl.H> // Fl::scrollbar_size()
32
33 #define FL_NORMAL_BROWSER 0
34 #define FL_SELECT_BROWSER 1
35 #define FL_HOLD_BROWSER 2
36 #define FL_MULTI_BROWSER 3
37 #define FL_SORT_ASCENDING 0
38 #define FL_SORT_DESCENDING 1
39
40 class FL_EXPORT Fl_Browser_ : public Fl_Group {
41     int position_; // where user wants it scrolled to
42     int real_position_; // the current vertical scrolling position
43     int hposition_; // where user wants it panned to
44     int real_hposition_; // the current horizontal scrolling position
45     int offset_; // how far down top_ item the real_position_ is
46     int max_width_; // widest object seen so far
47     uchar has_scrollbar_; // which scrollbars are enabled
48     Fl_Font textfont_;
49     Fl_Fonsize textsize_;
50     Fl_Color textcolor_;
51     void* top_; // which item scrolling position is in
52     void* selection_; // which is selected (except for FL_MULTI_BROWSER)
53     void *redraw1,*redraw2; // minimal update pointers
54     void* max_width_item_; // which item has max_width_
55     int scrollbar_size_; // size of scrollbar trough
56
57     void update_top();
58
59 protected:
60
61 // All of the following must be supplied by the subclass:
62     virtual void *item_first() const = 0;
63     virtual void *item_next(void *item) const = 0;
64     virtual void *item_prev(void *item) const = 0;
65     virtual void *item_last()const { return 0L; }
66     virtual int item_height(void *item) const = 0;
67     virtual int item_width(void *item) const = 0;

```

```

140 virtual int item_quick_height(void *item) const ;
141 virtual void item_draw(void *item,int X,int Y,int W,int H) const = 0;
142 virtual const char *item_text(void *item)const { (void)item; return 0L; }
143 virtual void item_swap(void *a,void *b) { (void)a; (void)b; }
144 virtual void *item_at(int index)const { (void)index; return 0L; }
145 // you don't have to provide these but it may help speed it up:
146 virtual int full_width() const ; // current width of all items
147 virtual int full_height() const ; // current height of all items
148 virtual int incr_height() const ; // average height of an item
149 // These only need to be done by subclass if you want a multi-browser:
150 virtual void item_select(void *item,int val=1);
151 virtual int item_selected(void *item) const ;
152
153 // things the subclass may want to call:
154 void *top()const { return top_; }
155 void *selection()const { return selection_; }
156 void new_list(); // completely clobber all data, as though list replaced
157 void deleting(void *item); // get rid of any pointers to item
158 void replacing(void *a,void *b); // change a pointers to b
159 void swapping(void *a,void *b); // exchange pointers a and b
160 void inserting(void *a,void *b); // insert b near a
161 int displayed(void *item) const ; // true if this item is visible
162 void redraw_line(void *item); // minimal update, no change in size
163 void redraw_lines() { damage(FL_DAMAGE_SCROLL); } // redraw all of them
164 void bbox(int &X,int &Y,int &W,int &H) const;
165 int leftedge() const; // x position after scrollbar & border
166 void *find_item(int ypos); // item under mouse
167
168 void draw();
169 Fl_Browser_(int X,int Y,int W,int H,const char *L=0);
170
171 public:
172
173 Fl_Scrollbar scrollbar;
174 Fl_Scrollbar hscrollbar;
175
176 int handle(int event);
177 void resize(int X,int Y,int W,int H);
178
179 int select(void *item,int val=1,int docallbacks=0);
180 int select_only(void *item,int docallbacks=0);
181 int deselect(int docallbacks=0);
182 int position()const { return position_; }
183 void position(int pos); // scroll to here
184 int hposition()const { return hposition_; }
185 void hposition(int); // pan to here
186 void display(void *item); // scroll so this item is shown
187
188 enum { // values for has_scrollbar()
189     HORIZONTAL = 1,
190     VERTICAL = 2,
191     BOTH = 3,
192     ALWAYS_ON = 4,
193     HORIZONTAL_ALWAYS = 5,
194     VERTICAL_ALWAYS = 6,
195     BOTH_ALWAYS = 7
196 };
197 uchar has_scrollbar()const { return has_scrollbar_; }
198 void has_scrollbar(uchar mode) { has_scrollbar_ = mode; }
199
200 Fl_Font textfont()const { return textfont_; }
201 void textfont(Fl_Font font) { textfont_ = font; }
202
203 Fl_Fontsize textsize()const { return textsize_; }
204 void textsize(Fl_Fontsize newSize) { textsize_ = newSize; }
205
206 Fl_Color textcolor()const { return textcolor_; }
207 void textcolor(Fl_Color col) { textcolor_ = col; }
208
209 int scrollbar_size()const {
210     return(scrollbar_size_);
211 }
212 void scrollbar_size(int newSize) {
213     scrollbar_size_ = newSize;
214 }
215 int scrollbar_width()const {
216     return(Fl::scrollbar_size());
217 }
218 void scrollbar_width(int width) {
219     Fl::scrollbar_size(width);
220     scrollbar_size_ = 0;
221 }
222 void scrollbar_right() { scrollbar.align(FL_ALIGN_RIGHT); }
223 void scrollbar_left() { scrollbar.align(FL_ALIGN_LEFT); }
224 void sort(int flags=0);
225 };
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383

```

```

384 #endif
385
386 //
387 // End of "$Id$".
388 //

```

32.17 Fl_Button.H

```

1 //
2 // "$Id$"
3 //
4 // Button header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2014 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Button widget . */
21
22 #ifndef Fl_Button_H
23 #define Fl_Button_H
24
25 #ifndef Fl_Widget_H
26 #include "Fl_Widget.H"
27 #endif
28
29 // values for type()
30 #define FL_NORMAL_BUTTON      0
31 #define FL_TOGGLE_BUTTON     1
32 #define FL_RADIO_BUTTON      (FL_RESERVED_TYPE+2)
33 #define FL_HIDDEN_BUTTON     3
34
35
36
37
38 extern FL_EXPORT Fl_Shortcut fl_old_shortcut(const char*);
39
40 class Fl_Widget_Tracker;
41
42 class FL_EXPORT Fl_Button : public Fl_Widget {
43
44     int shortcut_;
45     char value_;
46     char oldval;
47     uchar down_box_;
48
49 protected:
50
51     static Fl_Widget_Tracker *key_release_tracker;
52     static void key_release_timeout(void*);
53     void simulate_key_action();
54
55     virtual void draw();
56
57 public:
58
59     virtual int handle(int);
60
61     Fl_Button(int X, int Y, int W, int H, const char *L = 0);
62
63     int value(int v);
64
65     char value()const {return value_;}
66
67     int set() {return value(1);}
68
69     int clear() {return value(0);}
70
71     void setonly(); // this should only be called on FL_RADIO_BUTTONs
72
73     int shortcut()const {return shortcut_;}
74
75     void shortcut(int s) {shortcut_ = s;}
76
77     Fl_Boxtype down_box()const {return (Fl_Boxtype)down_box_;}
78
79     void down_box(Fl_Boxtype b) {down_box_ = b;}
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161

```

```

163 void shortcut(const char *s) {shortcut(fl_old_shortcut(s));}
164
166 Fl_Color down_color()const {return selection_color();}
167
169 void down_color(unsigned c) {selection_color(c);}
170 };
171
172 #endif
173
174 //
175 // End of "$Id$".
176 //

```

32.18 Fl_Cairo.H

```

1 //
2 // "$Id$"
3 //
4 // Main header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 // http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 // http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Handling transparently platform dependent cairo include files
21 */
22
23 #ifndef FL_CAIRO_H
24 # define FL_CAIRO_H
25 # ifdef FLTK_HAVE_CAIRO
26
27 // Cairo is currently supported for the following platforms:
28 // Win32, Apple Quartz, X11
29
30 # include <FL/Fl_Export.H>
31
32 # include <cairo.h>
33
34 class FL_EXPORT Fl_Cairo_State {
35 public:
36 Fl_Cairo_State() : cc_(0), own_cc_(false), autolink_(false), window_(0), gc_(0) {}
37
38 // access attributes
39 cairo_t* cc()const {return cc_;}
40 bool autolink()const {return autolink_;}
41 void cc(cairo_t* c, bool own=true) {
42     if (cc_ && own_cc_) cairo_destroy(cc_);
43     cc_=c;
44     if (!cc_) window_=0;
45     own_cc_=own;
46 }
47 void autolink(bool b);
48 void window(void* w) {window_=w;}
49 void* window()const {return window_;}
50 void gc(void* c) {gc_=c;}
51 void* gc()const {return gc_;}
52
53 private:
54     cairo_t * cc_; // contains the unique autoupdated cairo context
55     bool own_cc_; // indicates whether we must delete the cc, useful for internal cleanup
56     bool autolink_; // false by default, prevents the automatic cairo mapping on fltk windows
57 // for custom cairo implementations.
58     void* window_, *gc_; // for keeping track internally of last win+gc treated
59 };
60
61 # endif // FLTK_HAVE_CAIRO
62 #endif // FL_CAIRO_H
63
64 //
65 // End of "$Id$" .
66 //

```

32.19 Fl_Cairo_Window.H

```

1 //
2 // "$Id$"
3 //
4 // Main header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software.  Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file.  If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Cairo_Window Handling transparently a fltk window incorporate a cairo draw callback.
21 */
22
23 #ifndef FL_CAIRO_WINDOW_H
24 # define FL_CAIRO_WINDOW_H
25 # ifdef FLTK_HAVE_CAIRO
26
27 // Cairo is currently supported for the following platforms:
28 // Win32, Apple Quartz, X11
29 # include <FL/Fl.H>
30 # include <FL/Fl_Double_Window.H>
31
48 class FL_EXPORT Fl_Cairo_Window : public Fl_Double_Window {
49
50 public:
51     Fl_Cairo_Window(int w, int h) : Fl_Double_Window(w,h),draw_cb_(0) {}
52
53 protected:
54     void draw() {
55         Fl_Double_Window::draw();
56         // manual method ? if yes explicitly get a cairo_context here
57         if (!Fl::cairo_autolink_context())
58             Fl::cairo_make_current(this);
59         if (draw_cb_) draw_cb_(this, Fl::cairo_cc());
60     }
61
62
63 public:
64     typedef void (*cairo_draw_cb) (Fl_Cairo_Window* self, cairo_t* def);
65     void set_draw_cb(cairo_draw_cb cb){draw_cb_=cb;}
66 private:
67     cairo_draw_cb draw_cb_;
68 };
69
70 # endif // FLTK_HAVE_CAIRO
71 #endif // FL_CAIRO_WINDOW_H
72
73 //
74 // End of "$Id$" .
75 //

```

32.20 Fl_Chart.H

```

1 //
2 // "$Id$"
3 //
4 // Forms chart header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software.  Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file.  If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Chart widget . */

```

```

21
22 #ifndef Fl_Chart_H
23 #define Fl_Chart_H
24
25 #ifndef Fl_Widget_H
26 #include "Fl_Widget.H"
27 #endif
28
29 // values for type()
30 #define FL_BAR_CHART          0
31 #define FL_HORBAR_CHART     1
32 #define FL_LINE_CHART       2
33 #define FL_FILL_CHART       3
34 #define FL_SPIKE_CHART      4
35 #define FL_PIE_CHART        5
36 #define FL_SPECIALPIE_CHART 6
37 #define FL_FILLED_CHART     FL_FILL_CHART
38 #define FL_CHART_MAX        128
39 #define FL_CHART_LABEL_MAX   18
40
41 struct FL_CHART_ENTRY {
42     float val;
43     unsigned col;
44     char str[FL_CHART_LABEL_MAX+1];
45 };
46
47 class FL_EXPORT Fl_Chart : public Fl_Widget {
48     int numb;
49     int maxnumb;
50     int sizenumb;
51     FL_CHART_ENTRY *entries;
52     double min,max;
53     uchar autosize_;
54     Fl_Font textfont_;
55     Fl_Fontsize textsize_;
56     Fl_Color textcolor_;
57 protected:
58     void draw();
59 public:
60     Fl_Chart(int X, int Y, int W, int H, const char *L = 0);
61     ~Fl_Chart();
62     void clear();
63     void add(double val, const char *str = 0, unsigned col = 0);
64     void insert(int ind, double val, const char *str = 0, unsigned col = 0);
65     void replace(int ind, double val, const char *str = 0, unsigned col = 0);
66     void bounds(double *a,double *b)const { *a = min; *b = max;}
67     void bounds(double a,double b);
68     int size()const {return numb;}
69     void size(int W, int H) { Fl_Widget::size(W, H); }
70     int maxsize()const {return maxnumb;}
71     void maxsize(int m);
72     Fl_Font textfont()const {return textfont_;}
73     void textfont(Fl_Font s) {textfont_ = s;}
74     Fl_Fontsize textsize()const {return textsize_;}
75     void textsize(Fl_Fontsize s) {textsize_ = s;}
76     Fl_Color textcolor()const {return textcolor_;}
77     void textcolor(Fl_Color n) {textcolor_ = n;}
78     uchar autosize()const {return autosize_;}
79     void autosize(uchar n) {autosize_ = n;}
80 };
81 #endif
82 //
83 // End of "$Id$".
84 //

```

32.21 Fl_Check_Browser.H

```
1 //
```



```

2 // "$Id$"
3 //
4 // Fl_Check_Browser header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Check_Browser widget . */
21
22 #ifndef Fl_Check_Browser_H
23 #define Fl_Check_Browser_H
24
25 #include "Fl.H"
26 #include "Fl_Browser_.H"
27
32 class FL_EXPORT Fl_Check_Browser : public Fl_Browser_ {
33     /* required routines for Fl_Browser_ subclass: */
34
35     void *item_first() const;
36     void *item_next(void *) const;
37     void *item_prev(void *) const;
38     int item_height(void *) const;
39     int item_width(void *) const;
40     void item_draw(void *, int, int, int, int) const;
41     void item_select(void *, int);
42     int item_selected(void *) const;
43
44     /* private data */
45
46     public: // IRIX 5.3 C++ compiler doesn't support private structures...
47
48 #ifndef FL_DOXYGEN
49     struct cb_item {
50         cb_item *next;
51         cb_item *prev;
52         char checked;
53         char selected;
54         char *text;
55     };
56 #endif // !FL_DOXYGEN
57
58     private:
59
60     cb_item *first;
61     cb_item *last;
62     cb_item *cache;
63     int cached_item;
64     int nitems_;
65     int nchecked_;
66     cb_item *find_item(int) const;
67     int lineno(cb_item *) const;
68
69     public:
70
71     Fl_Check_Browser(int x, int y, int w, int h, const char *l = 0);
72     ~Fl_Check_Browser() { clear(); }
73     int add(char *s); // add an (unchecked) item
74     int add(char *s, int b); // add an item and set checked
75     // both return the new nitems()
76     int remove(int item); // delete an item. Returns nitems()
77
78     // inline const char * methods to avoid breaking binary compatibility...
79     int add(const char *s) { return add((char *)s); }
80     int add(const char *s, int b) { return add((char *)s, b); }
81
82     void clear(); // delete all items
83     int nitems()const { return nitems_; }
84     int nchecked()const { return nchecked_; }
85     int checked(int item) const;
86     void checked(int item, int b);
87     void set_checked(int item) { checked(item, 1); }
88     void check_all();
89     void check_none();
90     int value() const; // currently selected item
91     char *text(int item) const; // returns pointer to internal buffer
92
93 };

```

```

103     protected:
104
105     int handle(int);
106 };
107
108 #endif // Fl_Check_Browser_H
109
110 //
111 // End of "$Id$".
112 //
113

```

32.22 Fl_Check_Button.H

```

1 //
2 // "$Id$"
3 //
4 // Check button header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2014 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 #ifndef Fl_Check_Button_H
20 #define Fl_Check_Button_H
21
22 #include "Fl_Light_Button.H"
23
24 /*
25 class: Fl_Check_Button.
26
27 A button with a "checkmark" to show its status.
28 */
29
30 class FL_EXPORT Fl_Check_Button : public Fl_Light_Button {
31 public:
32     Fl_Check_Button(int X, int Y, int W, int H, const char *L = 0);
33 };
34
35 #endif
36
37 //
38 // End of "$Id$".
39 //

```

32.23 Fl_Choice.H

```

1 //
2 // "$Id$"
3 //
4 // Choice header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Choice widget . */
21
22 #ifndef Fl_Choice_H
23 #define Fl_Choice_H
24
25 #include "Fl_Menu_.H"
26

```

```

83 class FL_EXPORT Fl_Choice : public Fl_Menu_ {
84 protected:
85   void draw();
86 public:
87   int handle(int);
88
89   Fl_Choice(int X, int Y, int W, int H, const char *L = 0);
90
91   int value()const {return Fl_Menu_::value();}
92
93   int value(int v);
94
95   int value(const Fl_Menu_Item* v);
96 };
97
98 #endif
99
100 //
101 // End of "$Id$".
102 //

```

32.24 Fl_Clock.H

```

1 //
2 // "$Id$"
3 //
4 // Clock header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 //   http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //   http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Clock, Fl_Clock_Output widgets . */
21
22 #ifndef Fl_Clock_H
23 #define Fl_Clock_H
24
25 #ifndef Fl_Widget_H
26 #include "Fl_Widget.H"
27 #endif
28
29 // values for type:
30 #define FL_SQUARE_CLOCK      0
31 #define FL_ROUND_CLOCK      1
32 #define FL_ANALOG_CLOCK     FL_SQUARE_CLOCK
33 #define FL_DIGITAL_CLOCK    FL_SQUARE_CLOCK
34
35 // fabien: Please keep the horizontal formatting of both images in class desc,
36 // don't lose vert. space for nothing!
37
38 class FL_EXPORT Fl_Clock_Output : public Fl_Widget {
39   int hour_, minute_, second_;
40   ulong value_;
41   void drawhands(Fl_Color, Fl_Color); // part of draw
42 protected:
43   void draw();
44   void draw(int X, int Y, int W, int H);
45 public:
46
47   Fl_Clock_Output(int X, int Y, int W, int H, const char *L = 0);
48
49   void value(ulong v); // set to this Unix time
50
51   void value(int H, int m, int s);
52
53   ulong value()const {return value_;}
54
55   int hour()const {return hour_;}
56
57   int minute()const {return minute_;}
58
59   int second()const {return second_;}
60 };
61
62 // a Fl_Clock displays the current time always by using a timeout:
63

```

```

113 class FL_EXPORT Fl_Clock : public Fl_Clock_Output {
114 public:
115     int handle(int);
116
117     Fl_Clock(int X, int Y, int W, int H, const char *L = 0);
118
119     Fl_Clock(uchar t, int X, int Y, int W, int H, const char *L);
120
121     ~Fl_Clock();
122 };
123
124 #endif
125
126 //
127 // End of "$Id$".
128 //

```

32.25 Fl_Color_Chooser.H File Reference

[Fl_Color_Chooser](#) widget .

```

#include <FL/Fl_Group.H>
#include <FL/Fl_Box.H>
#include <FL/Fl_Return_Button.H>
#include <FL/Fl_Choice.H>
#include <FL/Fl_Value_Input.H>

```

Classes

- class [Fl_Color_Chooser](#)

The [Fl_Color_Chooser](#) widget provides a standard RGB color chooser.

32.25.1 Detailed Description

[Fl_Color_Chooser](#) widget .

32.26 Fl_Color_Chooser.H

[Go to the documentation of this file.](#)

```

1 //
2 // "$Id$"
3 //
4 // Color chooser header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
22 // The color chooser object and the color chooser popup. The popup
23 // is just a window containing a single color chooser and some boxes
24 // to indicate the current and cancelled color.
25
26 #ifndef Fl_Color_Chooser_H
27 #define Fl_Color_Chooser_H
28
29 #include <FL/Fl_Group.H>
30 #include <FL/Fl_Box.H>
31 #include <FL/Fl_Return_Button.H>
32 #include <FL/Fl_Choice.H>
33 #include <FL/Fl_Value_Input.H>
34
35 #ifndef FL_DOXYGEN
36
38 class FL_EXPORT Flcc_HueBox : public Fl_Widget {

```

```

39  int px, py;
40 protected:
41  void draw();
42  int handle_key(int);
43 public:
44  int handle(int);
45  Flcc_HueBox(int X, int Y, int W, int H) : Fl_Widget(X,Y,W,H) {
46  px = py = 0;}
47 };
48
49
50 class FL_EXPORT Flcc_ValueBox : public Fl_Widget {
51  int py;
52 protected:
53  void draw();
54  int handle_key(int);
55 public:
56  int handle(int);
57  Flcc_ValueBox(int X, int Y, int W, int H) : Fl_Widget(X,Y,W,H) {
58  py = 0;}
59 };
60
61
62 class FL_EXPORT Flcc_Value_Input : public Fl_Value_Input {
63 public:
64  int format(char*);
65  Flcc_Value_Input(int X, int Y, int W, int H) : Fl_Value_Input(X,Y,W,H) {}
66 };
67
68 #endif // !FL_DOXYGEN
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107 class FL_EXPORT Fl_Color_Chooser : public Fl_Group {
108  Flcc_HueBox huebox;
109  Flcc_ValueBox valuebox;
110  Fl_Choice choice;
111  Flcc_Value_Input rvalue;
112  Flcc_Value_Input gvalue;
113  Flcc_Value_Input bvalue;
114  Fl_Box resize_box;
115  double hue_, saturation_, value_;
116  double r_, g_, b_;
117  void set_valuators();
118  static void rgb_cb(Fl_Widget*, void*);
119  static void mode_cb(Fl_Widget*, void*);
120 public:
121
122
123
124
125  int mode() {return choice.value();}
126
127
128  void mode(int newMode);
129
130
131
132  double hue()const {return hue_;}
133
134
135
136  double saturation()const {return saturation_;}
137
138
139
140  double value()const {return value_;}
141
142
143
144  double r()const {return r_;}
145
146
147  double g()const {return g_;}
148
149
150  double b()const {return b_;}
151
152
153
154  int hsv(double H, double S, double V);
155
156
157  int rgb(double R, double G, double B);
158
159
160  static void hsv2rgb(double H, double S, double V, double& R, double& G, double& B);
161
162
163  static void rgb2hsv(double R, double G, double B, double& H, double& S, double& V);
164
165
166  Fl_Color_Chooser(int X, int Y, int W, int H, const char *L = 0);
167 };
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183 FL_EXPORT int fl_color_chooser(const char* name, double& r, double& g, double& b, int m=-1);
184 FL_EXPORT int fl_color_chooser(const char* name, uchar& r, uchar& g, uchar& b, int m=-1);
185
186 #endif
187
188 //
189 // End of "$Id$".
190 //

```

32.27 Fl_Copy_Surface.H

```

1 //
2 // "$Id$"
3 //

```

```

4 // Copy-to-clipboard code for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2014 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 #ifndef Fl_Copy_Surface_H
20 #define Fl_Copy_Surface_H
21
22 #include <FL/Fl_Paged_Device.H>
23 #include <FL/Fl_Printer.H>
24 #include <FL/x.H>
25
26 class FL_EXPORT Fl_Copy_Surface : public Fl_Surface_Device {
27 private:
28     int width;
29     int height;
30     Fl_Paged_Device *helper;
31 #ifdef __APPLE__
32     CFMutableDataRef pdfdata;
33     CGContextRef oldgc;
34     CGContextRef gc;
35     void prepare_copy_pdf_and_tiff(int w, int h);
36     void complete_copy_pdf_and_tiff();
37     void init_PDF_context(int w, int h);
38     static size_t MyPutBytes(void* info, const void* buffer, size_t count);
39 #elif defined(WIN32)
40     HDC oldgc;
41     HDC gc;
42 #else // Xlib
43     Fl_Offscreen xid;
44     Window oldwindow;
45     Fl_Surface_Device *_ss;
46 #endif
47 public:
48     static const char *class_id;
49     const char *class_name() {return class_id;};
50     Fl_Copy_Surface(int w, int h);
51     ~Fl_Copy_Surface();
52     void set_current();
53     void draw(Fl_Widget* widget, int delta_x = 0, int delta_y = 0);
54     void draw_decorated_window(Fl_Window* win, int delta_x = 0, int delta_y = 0);
55     int w() { return width; }
56     int h() { return height; }
57 };
58
59 #if defined(__APPLE__)
60 /* Mac class to reimplement Fl_Paged_Device::printable_rect() */
61 class FL_EXPORT Fl_Quartz_Surface_ : public Fl_System_Printer {
62 protected:
63     int width;
64     int height;
65 public:
66     static const char *class_id;
67     const char *class_name() {return class_id;};
68     Fl_Quartz_Surface_(int w, int h);
69     virtual int printable_rect(int *w, int *h);
70     virtual ~Fl_Quartz_Surface_() {};
71 };
72 #elif defined(WIN32)
73 /* Win class to implement translate()/untranslate() */
74 class FL_EXPORT Fl_GDI_Surface_ : public Fl_Paged_Device {
75     int width;
76     int height;
77     unsigned depth;
78     POINT origins[10];
79 public:
80     static const char *class_id;
81     const char *class_name() {return class_id;};
82     Fl_GDI_Surface_();
83     virtual void translate(int x, int y);
84     virtual void untranslate();
85     virtual ~Fl_GDI_Surface_();
86 };
87
88 #endif

```

```

118 #elif !defined(FL_DOXYGEN)
119
120 /* Xlib class to implement translate()/untranslate() */
121 class FL_EXPORT Fl_Xlib_Surface_ : public Fl_Paged_Device {
122 public:
123     static const char *class_id;
124     const char *class_name() {return class_id;};
125     Fl_Xlib_Surface_();
126     virtual void translate(int x, int y);
127     virtual void untranslate();
128     virtual ~Fl_Xlib_Surface_();
129 };
130
131 #endif
132
133 #endif // Fl_Copy_Surface_H
134
135 //
136 // End of "$Id$".
137 //

```

32.28 Fl_Counter.H

```

1 //
2 // "$Id$"
3 //
4 // Counter header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Counter widget . */
21
22 // A numerical value with up/down step buttons. From Forms.
23
24 #ifndef Fl_Counter_H
25 #define Fl_Counter_H
26
27 #ifndef Fl_Valuator_H
28 #include "Fl_Valuator.H"
29 #endif
30
31 // values for type():
32 #define FL_NORMAL_COUNTER      0
33 #define FL_SIMPLE_COUNTER      1
34
35 class FL_EXPORT Fl_Counter : public Fl_Valuator {
36 public:
37     Fl_Font textfont_;
38     Fl_Fontsize textsize_;
39     Fl_Color textcolor_;
40     double lstep_;
41     uchar mouseobj;
42     static void repeat_callback(void *);
43     int calc_mouseobj();
44     void increment_cb();
45
46 protected:
47     void draw();
48
49 public:
50     int handle(int);
51
52     Fl_Counter(int X, int Y, int W, int H, const char* L = 0);
53     ~Fl_Counter();
54
55     void lstep(double a) {lstep_ = a;}
56
57     void step(double a, double b) {Fl_Valuator::step(a); lstep_ = b;}
58
59     void step(double a) {Fl_Valuator::step(a);}
60
61     double step()const {return Fl_Valuator::step();}

```

```

93
95 Fl_Font textfont()const {return textfont_;}
97 void textfont(Fl_Font s) {textfont_ = s;}
98
100 Fl_Fontsize textsize()const {return textsize_;}
102 void textsize(Fl_Fontsize s) {textsize_ = s;}
103
105 Fl_Color textcolor()const {return textcolor_;}
107 void textcolor(Fl_Color s) {textcolor_ = s;}
108
109 };
110
111 #endif
112
113 //
114 // End of "$Id$".
115 //

```

32.29 FI_Device.H File Reference

declaration of classes [Fl_Device](#), [Fl_Graphics_Driver](#), [Fl_Surface_Device](#), [Fl_Display_Device](#), [Fl_Device_Plugin](#).

```

#include <FL/x.H>
#include <FL/Fl_Plugin.H>
#include <FL/Fl_Image.H>
#include <FL/Fl_Bitmap.H>
#include <FL/Fl_Pixmap.H>
#include <FL/Fl_RGB_Image.H>
#include <stdlib.h>

```

Classes

- class [Fl_Device](#)
All graphical output devices and all graphics systems.
- class [Fl_Device_Plugin](#)
This plugin socket allows the integration of new device drivers for special window or screen types.
- class [Fl_Display_Device](#)
A display to which the computer can draw.
- class [Fl_GDI_Graphics_Driver](#)
The MSWindows-specific graphics class.
- class [Fl_GDI_Printer_Graphics_Driver](#)
The graphics driver used when printing on MSWindows.
- class [Fl_Graphics_Driver](#)
A virtual class subclassed for each graphics driver FLTK uses.
- class [Fl_Quartz_Graphics_Driver](#)
The Mac OS X-specific graphics class.
- class [Fl_Surface_Device](#)
A drawing surface that's susceptible to receive graphical output.
- class [Fl_Xlib_Graphics_Driver](#)
The Xlib-specific graphics class.
- struct [Fl_Graphics_Driver::matrix](#)
A 2D coordinate transformation matrix.

Macros

- #define **FL_MATRIX_STACK_SIZE** 32
- #define **FL_REGION_STACK_SIZE** 10
- #define **XPOINT** XPoint

Typedefs

- typedef short **COORD_T**
- typedef void(* **Fl_Draw_Image_Cb**) (void *data, int x, int y, int w, **uchar** *buf)
signature of image generation callback function.

Variables

- **FL_EXPORT Fl_Graphics_Driver * fl_graphics_driver**
Points to the driver that currently receives all graphics requests.

32.29.1 Detailed Description

declaration of classes [Fl_Device](#), [Fl_Graphics_Driver](#), [Fl_Surface_Device](#), [Fl_Display_Device](#), [Fl_Device_Plugin](#).

32.29.2 Typedef Documentation

32.29.2.1 Fl_Draw_Image_Cb

typedef void(* **Fl_Draw_Image_Cb**) (void *data, int x, int y, int w, **uchar** *buf)
signature of image generation callback function.

Parameters

in	<i>data</i>	user data passed to function
in	<i>x,y,w</i>	position and width of scan line in image
out	<i>buf</i>	buffer for generated image data. You must copy <i>w</i> pixels from scanline <i>y</i> , starting at pixel <i>x</i> to this buffer.

32.30 Fl_Device.H

[Go to the documentation of this file.](#)

```

1 //
2 // "$Id$"
3 //
4 // Definition of classes Fl_Device, Fl_Graphics_Driver, Fl_Surface_Device, Fl_Display_Device
5 // for the Fast Light Tool Kit (FLTK).
6 //
7 // Copyright 2010-2014 by Bill Spitzak and others.
8 //
9 // This library is free software. Distribution and use rights are outlined in
10 // the file "COPYING" which should have been included with this file. If this
11 // file is missing or damaged, see the license at:
12 //
13 // http://www.fltk.org/COPYING.php
14 //
15 // Please report all bugs and problems on the following page:
16 //
17 // http://www.fltk.org/str.php
18 //
19
25 #ifndef Fl_Device_H
26 #define Fl_Device_H
27
28 #include <FL/x.H>
29 #include <FL/Fl_Plugin.H>
30 #include <FL/Fl_Image.H>
31 #include <FL/Fl_Bitmap.H>
32 #include <FL/Fl_Pixmap.H>
33 #include <FL/Fl_RGB_Image.H>
34 #include <stdlib.h>
35
36 class Fl_Graphics_Driver;
37 class Fl_Font_Descriptor;
38
39 FL_EXPORT extern Fl_Graphics_Driver *fl_graphics_driver;

```

```

40
41 typedef void (*Fl_Draw_Image_Cb)(void* data,int x,int y,int w,uchar* buf);
42
43 // typedef what the x,y fields in a point are:
44 #ifdef WIN32
45 typedef int COORD_T;
46 # define XPOINT XPoint
47 #elif defined(__APPLE__)
48 typedef float COORD_T;
49 typedef struct { float x; float y; } QPoint;
50 # define XPOINT QPoint
51 extern float fl_quartz_line_width_;
52 #else
53 typedef short COORD_T;
54 # define XPOINT XPoint
55 #endif
56
57 class FL_EXPORT Fl_Device {
58 public:
59     static const char *class_id;
60     virtual const char *class_name() {return class_id;};
61     virtual ~Fl_Device() {};
62 };
63
64 #define FL_REGION_STACK_SIZE 10
65 #define FL_MATRIX_STACK_SIZE 32
66 class FL_EXPORT Fl_Graphics_Driver : public Fl_Device {
67 public:
68     struct matrix {double a, b, c, d, x, y;};
69 private:
70     static const matrix m0;
71     Fl_Font font_; // current font
72     Fl_Fontsize size_; // current font size
73     Fl_Color color_; // current color
74     int sptr;
75     static const int matrix_stack_size = FL_MATRIX_STACK_SIZE;
76     matrix stack[FL_MATRIX_STACK_SIZE];
77     matrix m;
78     int n, p_size, gap_;
79     XPOINT *p;
80     int what;
81     int fl_clip_state_number;
82     int rstackptr;
83     static const int region_stack_max = FL_REGION_STACK_SIZE - 1;
84     Fl_Region rstack[FL_REGION_STACK_SIZE];
85 #ifdef WIN32
86     int numcount;
87     int counts[20];
88 #endif
89     Fl_Font_Descriptor *font_descriptor_;
90     void transformed_vertex0(COORD_T x, COORD_T y);
91     void fixloop();
92
93 protected:
94 #ifndef FL_DOXYGEN
95     enum {LINE, LOOP, POLYGON, POINT_};
96     inline int vertex_no() { return n; }
97     inline XPOINT *vertices() {return p;};
98     inline int vertex_kind() {return what;};
99 #endif
100
101 /* ** \brief red color for background and/or mixing if device does not support masking or alpha *
102 uchar bg_r_;
103 ** \brief green color for background and/or mixing if device does not support masking or alpha *
104 uchar bg_g_;
105 ** \brief blue color for background and/or mixing if device does not support masking or alpha *
106 uchar bg_b_; */
107 friend class Fl_Pixmap;
108 friend class Fl_Bitmap;
109 friend class Fl_RGB_Image;
110 friend void fl_rect(int x, int y, int w, int h);
111 friend void fl_rectf(int x, int y, int w, int h);
112 friend void fl_line_style(int style, int width, char* dashes);
113 friend void fl_xyline(int x, int y, int x1);
114 friend void fl_xyline(int x, int y, int x1, int y2);
115 friend void fl_xyline(int x, int y, int x1, int y2, int x3);
116 friend void fl_yxline(int x, int y, int y1);
117 friend void fl_yxline(int x, int y, int y1, int x2);
118 friend void fl_yxline(int x, int y, int y1, int x2, int y3);
119 friend void fl_line(int x, int y, int x1, int y1);
120 friend void fl_line(int x, int y, int x1, int y1, int x2, int y2);
121 friend void fl_draw(const char *str, int n, int x, int y);
122 #ifdef __APPLE__
123 friend void fl_draw(const char *str, int n, float x, float y);
124 #endif
125 friend void fl_draw(int angle, const char *str, int n, int x, int y);
126 friend void fl_rtl_draw(const char *str, int n, int x, int y);
127 friend void fl_font(Fl_Font face, Fl_Fontsize size);

```

```

173 friend void fl_color(Fl_Color c);
174 friend void fl_color(uchar r, uchar g, uchar b);
175 friend void fl_point(int x, int y);
176 friend void fl_loop(int x0, int y0, int x1, int y1, int x2, int y2);
177 friend void fl_loop(int x0, int y0, int x1, int y1, int x2, int y2, int x3, int y3);
178 friend void fl_polygon(int x0, int y0, int x1, int y1, int x2, int y2);
179 friend void fl_polygon(int x0, int y0, int x1, int y1, int x2, int y2, int x3, int y3);
180 friend void fl_begin_points();
181 friend void fl_begin_line();
182 friend void fl_begin_loop();
183 friend void fl_begin_polygon();
184 friend void fl_vertex(double x, double y);
185 friend void fl_curve(double X0, double Y0, double X1, double Y1, double X2, double Y2, double X3,
double Y3);
186 friend void fl_circle(double x, double y, double r);
187 friend void fl_arc(double x, double y, double r, double start, double end);
188 friend void fl_arc(int x, int y, int w, int h, double a1, double a2);
189 friend void fl_pie(int x, int y, int w, int h, double a1, double a2);
190 friend void fl_end_points();
191 friend void fl_end_line();
192 friend void fl_end_loop();
193 friend void fl_end_polygon();
194 friend void fl_transformed_vertex(double xf, double yf);
195 friend void fl_push_clip(int x, int y, int w, int h);
196 friend int fl_clip_box(int x, int y, int w, int h, int &X, int &Y, int &W, int &H);
197 friend int fl_not_clipped(int x, int y, int w, int h);
198 friend void fl_push_no_clip();
199 friend void fl_pop_clip();
200 friend void fl_begin_complex_polygon();
201 friend void fl_gap();
202 friend void fl_end_complex_polygon();
203 friend void fl_push_matrix();
204 friend void fl_pop_matrix();
205 friend void fl_mult_matrix(double a, double b, double c, double d, double x, double y);
206 friend void fl_scale(double x, double y);
207 friend void fl_scale(double x);
208 friend void fl_translate(double x, double y);
209 friend void fl_rotate(double d);
210 friend double fl_transform_x(double x, double y);
211 friend double fl_transform_y(double x, double y);
212 friend double fl_transform_dx(double x, double y);
213 friend double fl_transform_dy(double x, double y);
214 friend Fl_Region fl_clip_region();
215 friend void fl_clip_region(Fl_Region r);
216 friend void fl_restore_clip();
217
218 friend void fl_draw_image(const uchar* buf, int X,int Y,int W,int H, int D, int L);
219 friend void fl_draw_image_mono(const uchar* buf, int X,int Y,int W,int H, int D, int L);
220 friend void fl_draw_image(Fl_Draw_Image_Cb cb, void* data, int X,int Y,int W,int H, int D);
221 friend FL_EXPORT void fl_draw_image_mono(Fl_Draw_Image_Cb cb, void* data, int X,int Y,int W,int H, int
D);
222 friend FL_EXPORT void gl_start();
223 friend FL_EXPORT void fl_copy_offscreen(int x, int y, int w, int h, Fl_Offscreen pixmap, int srcx, int
srcy);
224 matrix *fl_matrix;
225 Fl_Graphics_Driver();
226 virtual void rect(int x, int y, int w, int h);
227 virtual void rectf(int x, int y, int w, int h);
228 virtual void line_style(int style, int width=0, char* dashes=0);
229 virtual void xyline(int x, int y, int x1);
230 virtual void xyline(int x, int y, int x1, int y2);
231 virtual void xyline(int x, int y, int x1, int y2, int x3);
232 virtual void yxline(int x, int y, int y1);
233 virtual void yxline(int x, int y, int y1, int x2);
234 virtual void yxline(int x, int y, int y1, int x2, int y3);
235 virtual void line(int x, int y, int x1, int y1);
236 virtual void line(int x, int y, int x1, int y1, int x2, int y2);
237 virtual void draw(const char *str, int n, int x, int y) {}
238 #ifdef __APPLE__
239 virtual void draw(const char *str, int n, float x, float y) { draw(str, n, (int)(x+0.5),
(int)(y+0.5));}
240 #endif
241 virtual void draw(int angle, const char *str, int n, int x, int y) {}
242 virtual void rtl_draw(const char *str, int n, int x, int y) {};
243 virtual void color(Fl_Color c) {color_ = c;}
244 virtual void color(uchar r, uchar g, uchar b) {}
245 virtual void point(int x, int y);
246 virtual void loop(int x0, int y0, int x1, int y1, int x2, int y2);
247 virtual void loop(int x0, int y0, int x1, int y1, int x2, int y2, int x3, int y3);
248 virtual void polygon(int x0, int y0, int x1, int y1, int x2, int y2);
249 virtual void polygon(int x0, int y0, int x1, int y1, int x2, int y2, int x3, int y3);
250 virtual void begin_points();
251 virtual void begin_line();
252 virtual void begin_loop();
253 virtual void begin_polygon();
254 virtual void vertex(double x, double y);
255 virtual void curve(double X0, double Y0, double X1, double Y1, double X2, double Y2, double X3, double

```

```

Y3);
286 virtual void circle(double x, double y, double r);
288 virtual void arc(double x, double y, double r, double start, double end);
290 virtual void arc(int x, int y, int w, int h, double a1, double a2);
292 virtual void pie(int x, int y, int w, int h, double a1, double a2);
294 virtual void end_points();
296 virtual void end_line();
298 virtual void end_loop();
300 virtual void end_polygon();
302 virtual void begin_complex_polygon();
304 virtual void gap();
306 virtual void end_complex_polygon();
308 virtual void transformed_vertex(double xf, double yf);
310 virtual void push_clip(int x, int y, int w, int h);
312 virtual int clip_box(int x, int y, int w, int h, int &X, int &Y, int &W, int &H);
314 virtual int not_clipped(int x, int y, int w, int h);
316 virtual void push_no_clip();
318 virtual void pop_clip();
319
321 void push_matrix();
323 void pop_matrix();
325 void mult_matrix(double a, double b, double c, double d, double x, double y);
327 inline void scale(double x, double y) { mult_matrix(x,0,0,y,0,0); }
329 inline void scale(double x) { mult_matrix(x,0,0,x,0,0); }
331 inline void translate(double x,double y) { mult_matrix(1,0,0,1,x,y); }
333 void rotate(double d);
335 double transform_x(double x, double y);
337 double transform_y(double x, double y);
339 double transform_dx(double x, double y);
341 double transform_dy(double x, double y);
343 Fl_Region clip_region();
345 void clip_region(Fl_Region r);
347 void restore_clip();
348
349 // Images
351 virtual void draw_image(const uchar* buf, int X,int Y,int W,int H, int D=3, int L=0) {}
353 virtual void draw_image_mono(const uchar* buf, int X,int Y,int W,int H, int D=1, int L=0) {}
355 virtual void draw_image(Fl_Draw_Image_Cb cb, void* data, int X,int Y,int W,int H, int D=3) {}
357 virtual void draw_image_mono(Fl_Draw_Image_Cb cb, void* data, int X,int Y,int W,int H, int D=1) {}
358 // Image classes
364 virtual void draw(Fl_RGB_Image * rgb,int XP, int YP, int WP, int HP, int cx, int cy) {}
370 virtual void draw(Fl_Pixmap * pxm,int XP, int YP, int WP, int HP, int cx, int cy) {}
376 virtual void draw(Fl_Bitmap *bm, int XP, int YP, int WP, int HP, int cx, int cy) {}
377 #if FLTK_ABI_VERSION >= 10301
378 virtual
379 #endif
380 void copy_offscreen(int x, int y, int w, int h, Fl_Offscreen pixmap, int srcx, int srcy);
381
382 public:
383 static const char *class_id;
384 virtual const char *class_name() {return class_id;};
386 virtual void font(Fl_Font face, Fl_Fonsize fsize) {font_ = face; size_ = fsize;};
388 Fl_Font font() {return font_; }
390 Fl_Fonsize size() {return size_; }
392 virtual double width(const char *str, int n) {return 0;};
394 virtual inline double width(unsigned int c) { char ch = (char)c; return width(&ch, 1); }
396 virtual void text_extents(const char*, int n, int& dx, int& dy, int& w, int& h);
398 virtual int height() {return size();};
400 virtual int descent() {return 0;};
402 Fl_Color color() {return color_;};
404 inline Fl_Font_Descriptor *font_descriptor() { return font_descriptor_;}
406 inline void font_descriptor(Fl_Font_Descriptor *d) { font_descriptor_ = d;};
407 #if FLTK_ABI_VERSION >= 10304 || defined(FL_DOXYGEN)
408 virtual
409 #endif
410 int draw_scaled(Fl_Image *img, int X, int Y, int W, int H);
412 virtual ~Fl_Graphics_Driver() { if (p) free(p); }
413 };
414
415 #if defined(__APPLE__) || defined(FL_DOXYGEN)
421 class FL_EXPORT Fl_Quartz_Graphics_Driver : public Fl_Graphics_Driver {
422 public:
423 static const char *class_id;
424 const char *class_name() {return class_id;};
425 void color(Fl_Color c);
426 void color(uchar r, uchar g, uchar b);
427 void draw(const char* str, int n, int x, int y);
428 #ifndef __APPLE__
429 void draw(const char *str, int n, float x, float y);
430 #endif
431 void draw(int angle, const char *str, int n, int x, int y);
432 void rtl_draw(const char* str, int n, int x, int y);
433 void font(Fl_Font face, Fl_Fonsize size);
434 void draw(Fl_Pixmap *pxm, int XP, int YP, int WP, int HP, int cx, int cy);
435 void draw(Fl_Bitmap *pxm, int XP, int YP, int WP, int HP, int cx, int cy);
436 void draw(Fl_RGB_Image *img, int XP, int YP, int WP, int HP, int cx, int cy);
437 int draw_scaled(Fl_Image *img, int XP, int YP, int WP, int HP);

```

```

438 void draw_image(const uchar* buf, int X,int Y,int W,int H, int D=3, int L=0);
439 void draw_image(Fl_Draw_Image_Cb cb, void* data, int X,int Y,int W,int H, int D=3);
440 void draw_image_mono(const uchar* buf, int X,int Y,int W,int H, int D=1, int L=0);
441 void draw_image_mono(Fl_Draw_Image_Cb cb, void* data, int X,int Y,int W,int H, int D=1);
442 double width(const char *str, int n);
443 double width(unsigned int c);
444 void text_extents(const char*, int n, int& dx, int& dy, int& w, int& h);
445 int height();
446 int descent();
447 #if ! defined(FL_DOXYGEN)
448     static Fl_Offscreen create_offscreen_with_alpha(int w, int h);
449 #endif
450 void copy_offscreen(int x, int y, int w, int h, Fl_Offscreen pixmap, int srcx, int srcy);
451 };
452 #endif
453 #if defined(WIN32) || defined(FL_DOXYGEN)
454 class FL_EXPORT Fl_GDI_Graphics_Driver : public Fl_Graphics_Driver {
455 public:
456     static const char *class_id;
457     const char *class_name() {return class_id;};
458     void color(Fl_Color c);
459     void color(uchar r, uchar g, uchar b);
460     void draw(const char* str, int n, int x, int y);
461     void draw(int angle, const char *str, int n, int x, int y);
462     void rtl_draw(const char* str, int n, int x, int y);
463     void font(Fl_Font face, Fl_Fontsize size);
464     void draw(Fl_Pixmap *pixmap, int XP, int YP, int WP, int HP, int cx, int cy);
465     void draw(Fl_Bitmap *pixmap, int XP, int YP, int WP, int HP, int cx, int cy);
466     void draw(Fl_RGB_Image *img, int XP, int YP, int WP, int HP, int cx, int cy);
467     void draw_image(const uchar* buf, int X,int Y,int W,int H, int D=3, int L=0);
468     void draw_image(Fl_Draw_Image_Cb cb, void* data, int X,int Y,int W,int H, int D=3);
469     void draw_image_mono(const uchar* buf, int X,int Y,int W,int H, int D=1, int L=0);
470     void draw_image_mono(Fl_Draw_Image_Cb cb, void* data, int X,int Y,int W,int H, int D=1);
471     double width(const char *str, int n);
472     double width(unsigned int c);
473     void text_extents(const char*, int n, int& dx, int& dy, int& w, int& h);
474     int height();
475     int descent();
476 #if ! defined(FL_DOXYGEN)
477     void copy_offscreen_with_alpha(int x,int y,int w,int h,HBITMAP bitmap,int srcx,int srcy);
478 #endif
479     void copy_offscreen(int x, int y, int w, int h, Fl_Offscreen pixmap, int srcx, int srcy);
480 };
481 class FL_EXPORT Fl_GDI_Printer_Graphics_Driver : public Fl_GDI_Graphics_Driver {
482 public:
483     static const char *class_id;
484     const char *class_name() {return class_id;};
485     void draw(Fl_Pixmap *pixmap, int XP, int YP, int WP, int HP, int cx, int cy);
486     void draw(Fl_Bitmap *bm, int XP, int YP, int WP, int HP, int cx, int cy);
487     int draw_scaled(Fl_Image *img, int XP, int YP, int WP, int HP);
488 };
489 #endif
490 #if !(defined(__APPLE__) || defined(WIN32))
491 class FL_EXPORT Fl_Xlib_Graphics_Driver : public Fl_Graphics_Driver {
492 public:
493     static const char *class_id;
494     const char *class_name() {return class_id;};
495     void color(Fl_Color c);
496     void color(uchar r, uchar g, uchar b);
497     void draw(const char* str, int n, int x, int y);
498     void draw(int angle, const char *str, int n, int x, int y);
499     void rtl_draw(const char* str, int n, int x, int y);
500     void font(Fl_Font face, Fl_Fontsize size);
501     void draw(Fl_Pixmap *pixmap, int XP, int YP, int WP, int HP, int cx, int cy);
502     void draw(Fl_Bitmap *pixmap, int XP, int YP, int WP, int HP, int cx, int cy);
503     void draw(Fl_RGB_Image *img, int XP, int YP, int WP, int HP, int cx, int cy);
504     void draw_image(const uchar* buf, int X,int Y,int W,int H, int D=3, int L=0);
505     void draw_image(Fl_Draw_Image_Cb cb, void* data, int X,int Y,int W,int H, int D=3);
506     void draw_image_mono(const uchar* buf, int X,int Y,int W,int H, int D=1, int L=0);
507     void draw_image_mono(Fl_Draw_Image_Cb cb, void* data, int X,int Y,int W,int H, int D=1);
508     double width(const char *str, int n);
509     double width(unsigned int c);
510     void text_extents(const char*, int n, int& dx, int& dy, int& w, int& h);
511     int height();
512     int descent();
513     void copy_offscreen(int x, int y, int w, int h, Fl_Offscreen pixmap, int srcx, int srcy);
514 #if ! defined(FL_DOXYGEN)
515     void copy_offscreen_with_alpha(int x, int y, int w, int h, Fl_Offscreen pixmap, int srcx, int srcy);
516 #endif
517 };
518 #endif
519 class FL_EXPORT Fl_Surface_Device : public Fl_Device {
520 public:
521     Fl_Graphics_Driver *_driver;
522     static Fl_Surface_Device *_surface; // the surface that currently receives graphics output
523     static Fl_Surface_Device *default_surface(); // create surface if none exists yet

```

```

561 protected:
562     Fl_Surface_Device(Fl_Graphics_Driver *graphics_driver) {_driver = graphics_driver; };
563 public:
564     static const char *class_id;
565     const char *class_name() {return class_id;};
566     virtual void set_current(void);
567     inline void driver(Fl_Graphics_Driver *graphics_driver) {_driver = graphics_driver;};
568     inline Fl_Graphics_Driver *driver() {return _driver; };
569     static inline Fl_Surface_Device *surface() {
570         return _surface ? _surface : default_surface();
571     };
572     virtual ~Fl_Surface_Device() {}
573 };
574
575 class FL_EXPORT Fl_Display_Device : public Fl_Surface_Device {
576     static Fl_Display_Device *_display; // the platform display device
577 #ifdef __APPLE__
578     friend class Fl_X;
579     friend class Fl_Graphics_Driver;
580     static bool high_res_window_; //< true when drawing to a window of a retina display (Mac OS X only)
581     static bool high_resolution() {return high_res_window_;}
582 #endif
583 public:
584     static const char *class_id;
585     const char *class_name() {return class_id;};
586     Fl_Display_Device(Fl_Graphics_Driver *graphics_driver);
587     static Fl_Display_Device *display_device();
588 };
589
590 class FL_EXPORT Fl_Device_Plugin : public Fl_Plugin {
591 public:
592     Fl_Device_Plugin(const char *pluginName)
593     : Fl_Plugin(klass(), pluginName) {}
594     virtual const char *klass() { return "fltk:device"; }
595     virtual const char *name() = 0;
596     virtual int print(Fl_Widget* w, int x, int y, int height) = 0;
597 #ifdef FL_LIBRARY
598     virtual
599 #endif
600     Fl_RGB_Image* rectangle_capture(Fl_Widget *widget, int x, int y, int w, int h) {return NULL;}
601 };
602 #endif // Fl_Device_H
603
604 //
605 // End of "$Id$".
606 //

```

32.31 Fl_Dial.H

```

1 //
2 // "$Id$"
3 //
4 // Dial header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Dial widget . */
21
22 #ifndef Fl_Dial_H
23 #define Fl_Dial_H
24
25 #ifndef Fl_Valuator_H
26 #include "Fl_Valuator.H"
27 #endif
28
29 // values for type():
30 #define FL_NORMAL_DIAL 0
31 #define FL_LINE_DIAL 1
32 #define FL_FILL_DIAL 2
33
34 class FL_EXPORT Fl_Dial : public Fl_Valuator {
35
36     short a1,a2;

```

```

50
51 protected:
52
53 // these allow subclasses to put the dial in a smaller area:
54 void draw(int X, int Y, int W, int H);
55 int handle(int event, int X, int Y, int W, int H);
56 void draw();
57
58 public:
59
60 int handle(int);
61 Fl_Dial(int x,int y,int w,int h, const char *l = 0);
62 short angle1()const {return a1;}
63 void angle1(short a) {a1 = a;}
64 short angle2()const {return a2;}
65 void angle2(short a) {a2 = a;}
66 void angles(short a, short b) {a1 = a; a2 = b;}
67
68 };
69
70 #endif
71
72 //
73 // End of "$Id$".
74 //

```

32.32 Fl_Double_Window.H

```

1 //
2 // "$Id$"
3 //
4 // Double-buffered window header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 // http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 // http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Double_Window widget . */
21
22 #ifndef Fl_Double_Window_H
23 #define Fl_Double_Window_H
24
25 #include "Fl_Window.H"
26
27 class FL_EXPORT Fl_Double_Window : public Fl_Window {
28 protected:
29 void flush(int eraseoverlay);
30 char force_doublebuffering_;
31 public:
32 void show();
33 void show(int a, char **b) {Fl_Window::show(a,b);}
34 void flush();
35 void resize(int,int,int,int);
36 void hide();
37 ~Fl_Double_Window();
38
39 Fl_Double_Window(int W, int H, const char *l = 0);
40
41 Fl_Double_Window(int X, int Y, int W, int H, const char *l = 0);
42
43 };
44
45 #endif
46
47 //
48 // End of "$Id$".
49 //

```

32.33 fl_draw.H File Reference

utility header to pull drawing functions together

```
#include <FL/x.H>
#include <FL/Enumerations.H>
#include <FL/Fl_Window.H>
#include <FL/Fl_Device.H>
```

Macros

- `#define fl_clip fl_push_clip`
Intersects the current clip region with a rectangle and pushes this new region onto the stack (deprecated).

Enumerations

- enum {
`FL_SOLID = 0` , `FL_DASH = 1` , `FL_DOT = 2` , `FL_DASHDOT = 3` ,
`FL_DASHDOTDOT = 4` , `FL_CAP_FLAT = 0x100` , `FL_CAP_ROUND = 0x200` , `FL_CAP_SQUARE = 0x300` ,
`FL_JOIN_MITER = 0x1000` , `FL_JOIN_ROUND = 0x2000` , `FL_JOIN_BEVEL = 0x3000` }

Functions

- `FL_EXPORT int fl_add_symbol` (const char *name, void(*drawit)(`Fl_Color`), int scalable)
Adds a symbol to the system.
- void `fl_arc` (double x, double y, double r, double start, double end)
Adds a series of points to the current path on the arc of a circle.
- void `fl_arc` (int x, int y, int w, int h, double a1, double a2)
Draw ellipse sections using integer coordinates.
- void `fl_begin_complex_polygon` ()
Starts drawing a complex filled polygon.
- void `fl_begin_line` ()
Starts drawing a list of lines.
- void `fl_begin_loop` ()
Starts drawing a closed sequence of lines.
- void `fl_begin_points` ()
Starts drawing a list of points.
- void `fl_begin_polygon` ()
Starts drawing a convex filled polygon.
- `FL_EXPORT char fl_can_do_alpha_blending` ()
Checks whether platform supports true alpha blending for RGBA images.
- `FL_EXPORT void fl_chord` (int x, int y, int w, int h, double a1, double a2)
fl_chord declaration is a place holder - the function does not yet exist
- void `fl_circle` (double x, double y, double r)
fl_circle() is equivalent to fl_arc(x,y,r,0,360), but may be faster.
- int `fl_clip_box` (int x, int y, int w, int h, int &X, int &Y, int &W, int &H)
Intersects the rectangle with the current clip region and returns the bounding box of the result.
- `Fl_Region fl_clip_region` ()
Returns the current clipping region.
- void `fl_clip_region` (`Fl_Region` r)
Replaces the top of the clipping stack with a clipping region of any shape.
- `Fl_Color fl_color` ()
Returns the last fl_color() that was set.
- void `fl_color` (`Fl_Color` c)
Sets the color for all subsequent drawing operations.
- void `fl_color` (int c)

- for back compatibility - use `fl_color(FI_Color c)` instead*
- void `fl_color` (`uchar r`, `uchar g`, `uchar b`)
Sets the color for all subsequent drawing operations.
 - FL_EXPORT void `fl_cursor` (`FI_Cursor`)
Sets the cursor for the current window to the specified shape and colors.
 - FL_EXPORT void `fl_cursor` (`FI_Cursor`, `FI_Color fg`, `FI_Color bg=FL_WHITE`)
 - void `fl_curve` (`double X0`, `double Y0`, `double X1`, `double Y1`, `double X2`, `double Y2`, `double X3`, `double Y3`)
Adds a series of points on a Bezier curve to the path.
 - int `fl_descent` ()
Returns the recommended distance above the bottom of a `fl_height()` tall box to draw the text at so it looks centered vertically in that box.
 - void `fl_draw` (`const char *str`, `int n`, `int x`, `int y`)
Draws starting at the given `x`, `y` location a UTF-8 string of length `n` bytes.
 - FL_EXPORT void `fl_draw` (`const char *str`, `int x`, `int y`)
Draws a nul-terminated UTF-8 string starting at the given `x`, `y` location.
 - FL_EXPORT void `fl_draw` (`const char *str`, `int x`, `int y`, `int w`, `int h`, `FI_Align align`, `FI_Image *img=0`, `int draw↵_symbols=1`)
Fancy string drawing function which is used to draw all the labels.
 - FL_EXPORT void `fl_draw` (`const char *str`, `int x`, `int y`, `int w`, `int h`, `FI_Align align`, `void(*callthis)(const char *, int, int, int)`, `FI_Image *img=0`, `int draw_symbols=1`)
The same as `fl_draw(const char,int,int,int,int,FI_Align,FI_Image*,int)` with the addition of the `callthis` parameter, which is a pointer to a text drawing function such as `fl_draw(const char*, int, int, int)` to do the real work.*
 - void `fl_draw` (`int angle`, `const char *str`, `int n`, `int x`, `int y`)
Draws at the given `x`, `y` location a UTF-8 string of length `n` bytes rotating `angle` degrees counter-clockwise.
 - FL_EXPORT void `fl_draw` (`int angle`, `const char *str`, `int x`, `int y`)
Draws a nul-terminated UTF-8 string starting at the given `x`, `y` location and rotating `angle` degrees counter-clockwise.
 - FL_EXPORT void `fl_draw_box` (`FI_Boxtype`, `int x`, `int y`, `int w`, `int h`, `FI_Color`)
Draws a box using given type, position, size and color.
 - void `fl_draw_image` (`const uchar *buf`, `int X`, `int Y`, `int W`, `int H`, `int D=3`, `int L=0`)
Draws an 8-bit per color RGB or luminance image.
 - void `fl_draw_image` (`FI_Draw_Image_Cb cb`, `void *data`, `int X`, `int Y`, `int W`, `int H`, `int D=3`)
Draws an image using a callback function to generate image data.
 - void `fl_draw_image_mono` (`const uchar *buf`, `int X`, `int Y`, `int W`, `int H`, `int D=1`, `int L=0`)
Draws a gray-scale (1 channel) image.
 - void `fl_draw_image_mono` (`FI_Draw_Image_Cb cb`, `void *data`, `int X`, `int Y`, `int W`, `int H`, `int D=1`)
Draws a gray-scale image using a callback function to generate image data.
 - FL_EXPORT int `fl_draw_pixmap` (`char *const *data`, `int x`, `int y`, `FI_Color=FL_GRAY`)
Draw XPM image data, with the top-left corner at the given position.
 - FL_EXPORT int `fl_draw_pixmap` (`const char *const *cdata`, `int x`, `int y`, `FI_Color=FL_GRAY`)
Draw XPM image data, with the top-left corner at the given position.
 - FL_EXPORT int `fl_draw_symbol` (`const char *label`, `int x`, `int y`, `int w`, `int h`, `FI_Color`)
Draw the named symbol in the given rectangle using the given color.
 - void `fl_end_complex_polygon` ()
Ends complex filled polygon, and draws.
 - void `fl_end_line` ()
Ends list of lines, and draws.
 - void `fl_end_loop` ()
Ends closed sequence of lines, and draws.
 - void `fl_end_points` ()
Ends list of points, and draws.
 - void `fl_end_polygon` ()

- Ends convex filled polygon, and draws.*

 - FL_EXPORT const char * [fl_expand_text](#) (const char *from, char *buf, int maxbuf, double maxw, int &n, double &width, int wrap, int draw_symbols=0)
 - Copy from to buf, replacing control characters with ^X.*
- [Fl_Font fl_font](#) ()
 - Returns the *face* set by the most recent call to [fl_font\(\)](#).
- void [fl_font](#) (Fl_Font face, Fl_Fontsize fsize)
 - Sets the current font, which is then used in various drawing routines.
- FL_EXPORT void [fl_frame](#) (const char *s, int x, int y, int w, int h)
 - Draws a series of line segments around the given box.
- FL_EXPORT void [fl_frame2](#) (const char *s, int x, int y, int w, int h)
 - Draws a series of line segments around the given box.
- void [fl_gap](#) ()
 - Call [fl_gap\(\)](#) to separate loops of the path.
- int [fl_height](#) ()
 - Returns the recommended minimum line spacing for the current font.
- FL_EXPORT int [fl_height](#) (int font, int size)
 - This function returns the actual height of the specified font and size.
- FL_EXPORT const char * [fl_latin1_to_local](#) (const char *t, int n=-1)
 - Converts text from Windows/X11 latin1 character set to local encoding.
- void [fl_line](#) (int x, int y, int x1, int y1)
 - Draws a line from (x,y) to (x1,y1)
- void [fl_line](#) (int x, int y, int x1, int y1, int x2, int y2)
 - Draws a line from (x,y) to (x1,y1) and another from (x1,y1) to (x2,y2)
- void [fl_line_style](#) (int style, int width=0, char *dashes=0)
 - Sets how to draw lines (the "pen").
- FL_EXPORT const char * [fl_local_to_latin1](#) (const char *t, int n=-1)
 - Converts text from local encoding to Windows/X11 latin1 character set.
- FL_EXPORT const char * [fl_local_to_mac_roman](#) (const char *t, int n=-1)
 - Converts text from local encoding to Mac Roman character set.
- void [fl_loop](#) (int x, int y, int x1, int y1, int x2, int y2)
 - Outlines a 3-sided polygon with lines.
- void [fl_loop](#) (int x, int y, int x1, int y1, int x2, int y2, int x3, int y3)
 - Outlines a 4-sided polygon with lines.
- FL_EXPORT const char * [fl_mac_roman_to_local](#) (const char *t, int n=-1)
 - Converts text from Mac Roman character set to local encoding.
- FL_EXPORT void [fl_measure](#) (const char *str, int &x, int &y, int draw_symbols=1)
 - Measure how wide and tall the string will be when printed by the [fl_draw\(\)](#) function with *align* parameter.
- FL_EXPORT int [fl_measure_pixmap](#) (char *const *data, int &w, int &h)
 - Get the dimensions of a pixmap.
- FL_EXPORT int [fl_measure_pixmap](#) (const char *const *cdata, int &w, int &h)
 - Get the dimensions of a pixmap.
- void [fl_mult_matrix](#) (double a, double b, double c, double d, double x, double y)
 - Concatenates another transformation onto the current one.
- int [fl_not_clipped](#) (int x, int y, int w, int h)
 - Does the rectangle intersect the current clip region?
- FL_EXPORT unsigned int [fl_old_shortcut](#) (const char *s)
 - Emulation of XForms named shortcuts.
- FL_EXPORT void [fl_overlay_clear](#) ()
 - Erase a selection rectangle without drawing a new one.
- FL_EXPORT void [fl_overlay_rect](#) (int x, int y, int w, int h)

- Draws a selection rectangle, erasing a previous one by XOR'ing it first.*

 - void **fl_pie** (int x, int y, int w, int h, double a1, double a2)

Draw filled ellipse sections using integer coordinates.
- void **fl_point** (int x, int y)

Draws a single pixel at the given coordinates.
- void **fl_polygon** (int x, int y, int x1, int y1, int x2, int y2)

Fills a 3-sided polygon.
- void **fl_polygon** (int x, int y, int x1, int y1, int x2, int y2, int x3, int y3)

Fills a 4-sided polygon.
- void **fl_pop_clip** ()

Restores the previous clip region.
- void **fl_pop_matrix** ()

Restores the current transformation matrix from the stack.
- void **fl_push_clip** (int x, int y, int w, int h)

Intersects the current clip region with a rectangle and pushes this new region onto the stack.
- void **fl_push_matrix** ()

Saves the current transformation matrix on the stack.
- void **fl_push_no_clip** ()

Pushes an empty clip region onto the stack so nothing will be clipped.
- FL_EXPORT **uchar** * **fl_read_image** (**uchar** *p, int X, int Y, int W, int H, int alpha=0)

Reads an RGB(A) image from the current window or off-screen buffer.
- void **fl_rect** (int x, int y, int w, int h)

Draws a 1-pixel border inside the given bounding box.
- void **fl_rect** (int x, int y, int w, int h, **FL_Color** c)

Draws with passed color a 1-pixel border inside the given bounding box.
- void **fl_rectf** (int x, int y, int w, int h)

Colors with current color a rectangle that exactly fills the given bounding box.
- void **fl_rectf** (int x, int y, int w, int h, **FL_Color** c)

Colors with passed color a rectangle that exactly fills the given bounding box.
- FL_EXPORT void **fl_rectf** (int x, int y, int w, int h, **uchar** r, **uchar** g, **uchar** b)

Colors a rectangle with "exactly" the passed r, g, b color.
- FL_EXPORT void **fl_reset_spot** (void)
- void **fl_restore_clip** ()

Undoes any clobbering of clip done by your program.
- void **fl_rotate** (double d)

Concatenates rotation transformation onto the current one.
- void **fl_rtl_draw** (const char *str, int n, int x, int y)

Draws a UTF-8 string of length n bytes right to left starting at the given x, y location.
- void **fl_scale** (double x)

Concatenates scaling transformation onto the current one.
- void **fl_scale** (double x, double y)

Concatenates scaling transformation onto the current one.
- FL_EXPORT void **fl_scroll** (int X, int Y, int W, int H, int dx, int dy, void(*draw_area)(void *, int, int, int, int), void *data)

Scroll a rectangle and draw the newly exposed portions.
- FL_EXPORT void **fl_set_spot** (int font, int size, int X, int Y, int W, int H, **FL_Window** *win=0)
- FL_EXPORT void **fl_set_status** (int X, int Y, int W, int H)
- FL_EXPORT const char * **fl_shortcut_label** (unsigned int shortcut)

Get a human-readable string from a shortcut value.
- FL_EXPORT const char * **fl_shortcut_label** (unsigned int shortcut, const char **eom)

Get a human-readable string from a shortcut value.

- `FL_Fontsize fl_size ()`
Returns the `size` set by the most recent call to `fl_font()`.
- `FL_EXPORT void fl_text_extents (const char *, int &dx, int &dy, int &w, int &h)`
Determines the minimum pixel dimensions of a nul-terminated string.
- `void fl_text_extents (const char *t, int n, int &dx, int &dy, int &w, int &h)`
Determines the minimum pixel dimensions of a sequence of `n` characters.
- `double fl_transform_dx (double x, double y)`
Transforms distance using current transformation matrix.
- `double fl_transform_dy (double x, double y)`
Transforms distance using current transformation matrix.
- `double fl_transform_x (double x, double y)`
Transforms coordinate using the current transformation matrix.
- `double fl_transform_y (double x, double y)`
Transforms coordinate using the current transformation matrix.
- `void fl_transformed_vertex (double xf, double yf)`
Adds coordinate pair to the vertex list without further transformations.
- `void fl_translate (double x, double y)`
Concatenates translation transformation onto the current one.
- `void fl_vertex (double x, double y)`
Adds a single vertex to the current path.
- `FL_EXPORT double fl_width (const char *txt)`
Returns the typographical width of a nul-terminated string using the current font face and size.
- `double fl_width (const char *txt, int n)`
Returns the typographical width of a sequence of `n` characters using the current font face and size.
- `double fl_width (unsigned int c)`
Returns the typographical width of a single character using the current font face and size.
- `void fl_xyline (int x, int y, int x1)`
Draws a horizontal line from (x,y) to $(x1,y)$
- `void fl_xyline (int x, int y, int x1, int y2)`
Draws a horizontal line from (x,y) to $(x1,y)$, then vertical from $(x1,y)$ to $(x1,y2)$
- `void fl_xyline (int x, int y, int x1, int y2, int x3)`
Draws a horizontal line from (x,y) to $(x1,y)$, then a vertical from $(x1,y)$ to $(x1,y2)$ and then another horizontal from $(x1,y2)$ to $(x3,y2)$
- `void fl_yxline (int x, int y, int y1)`
Draws a vertical line from (x,y) to $(x,y1)$
- `void fl_yxline (int x, int y, int y1, int x2)`
Draws a vertical line from (x,y) to $(x,y1)$, then a horizontal from $(x,y1)$ to $(x2,y1)$
- `void fl_yxline (int x, int y, int y1, int x2, int y3)`
Draws a vertical line from (x,y) to $(x,y1)$ then a horizontal from $(x,y1)$ to $(x2,y1)$, then another vertical from $(x2,y1)$ to $(x2,y3)$

Variables

- `FL_EXPORT char fl_draw_shortcut`

32.33.1 Detailed Description

utility header to pull drawing functions together

32.34 fl_draw.H

[Go to the documentation of this file.](#)

```

1 //
2 // "$Id$"
3 //
4 // Portable drawing function header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2016 by Bill Spitzak and others.
7 //
8 // This library is free software.  Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file.  If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
24 #ifndef fl_draw_H
25 #define fl_draw_H
26
27 #include <FL/x.H>           // for Fl_Region
28 #include <FL/Enumerations.H> // for the color names
29 #include <FL/Fl_Window.H>   // for fl_set_spot()
30 #include <FL/Fl_Device.H>   // for fl_graphics_driver
31
32 // Image class...
33 class Fl_Image;
34
35 // Label flags...
36 FL_EXPORT extern char fl_draw_shortcut;
37
42 // Colors:
52 inline void fl_color(Fl_Color c) {fl_graphics_driver->color(c); } // select indexed color
54 inline void fl_color(int c) {fl_color((Fl_Color)c); }
65 inline void fl_color(uchar r, uchar g, uchar b) {fl_graphics_driver->color(r,g,b); } // select actual
   color
70 inline Fl_Color fl_color() {return fl_graphics_driver->color(); }
76 // clip:
82 inline void fl_push_clip(int x, int y, int w, int h) {fl_graphics_driver->push_clip(x,y,w,h); }
91 #define fl_clip fl_push_clip
95 inline void fl_push_no_clip() {fl_graphics_driver->push_no_clip(); }
103 inline void fl_pop_clip() {fl_graphics_driver->pop_clip(); }
114 inline int fl_not_clipped(int x, int y, int w, int h) {return fl_graphics_driver->not_clipped(x,y,w,h);
   }
126 inline int fl_clip_box(int x, int y, int w, int h, int& X, int& Y, int& W, int& H)
127   {return fl_graphics_driver->clip_box(x,y,w,h,X,Y,W,H); }
129 inline void fl_restore_clip() { fl_graphics_driver->restore_clip(); }
136 inline void fl_clip_region(Fl_Region r) { fl_graphics_driver->clip_region(r); }
140 inline Fl_Region fl_clip_region() { return fl_graphics_driver->clip_region(); }
141
142
143 // points:
147 inline void fl_point(int x, int y) { fl_graphics_driver->point(x,y); }
148
149 // line type:
177 inline void fl_line_style(int style, int width=0, char* dashes=0)
   {fl_graphics_driver->line_style(style,width,dashes); }
178 enum {
179     FL_SOLID      = 0,
180     FL_DASH      = 1,
181     FL_DOT       = 2,
182     FL_DASHDOT   = 3,
183     FL_DASHDOTDOT = 4,
184
185     FL_CAP_FLAT   = 0x100,
186     FL_CAP_ROUND = 0x200,
187     FL_CAP_SQUARE = 0x300,
188
189     FL_JOIN_MITER = 0x1000,
190     FL_JOIN_ROUND = 0x2000,
191     FL_JOIN_BEVEL = 0x3000
192 };
193
194 // rectangles tweaked to exactly fill the pixel rectangle:
195
201 inline void fl_rect(int x, int y, int w, int h) { fl_graphics_driver->rect(x,y,w,h); }
202
204 inline void fl_rect(int x, int y, int w, int h, Fl_Color c) {fl_color(c); fl_rect(x,y,w,h); }
206 inline void fl_rectf(int x, int y, int w, int h) { fl_graphics_driver->rectf(x,y,w,h); }
208 inline void fl_rectf(int x, int y, int w, int h, Fl_Color c) {fl_color(c); fl_rectf(x,y,w,h); }
209
216 /* note: doxygen comment here to avoid triplcation in os-specific files */

```

```

217 FL_EXPORT void fl_rectf(int x, int y, int w, int h, uchar r, uchar g, uchar b);
218
219 // line segments:
223 inline void fl_line(int x, int y, int x1, int y1) {fl_graphics_driver->line(x,y,x1,y1); }
227 inline void fl_line(int x, int y, int x1, int y1, int x2, int y2)
    {fl_graphics_driver->line(x,y,x1,y1,x2,y2); }
228
229 // closed line segments:
233 inline void fl_loop(int x, int y, int x1, int y1, int x2, int y2)
    {fl_graphics_driver->loop(x,y,x1,y1,x2,y2); }
237 inline void fl_loop(int x, int y, int x1, int y1, int x2, int y2, int x3, int y3)
    {fl_graphics_driver->loop(x,y,x1,y1,x2,y2,x3,y3); }
238
239
240 // filled polygons
244 inline void fl_polygon(int x, int y, int x1, int y1, int x2, int y2)
    {fl_graphics_driver->polygon(x,y,x1,y1,x2,y2); }
248 inline void fl_polygon(int x, int y, int x1, int y1, int x2, int y2, int x3, int y3)
    { fl_graphics_driver->polygon(x,y,x1,y1,x2,y2,x3,y3); }
249
250
251 // draw rectilinear lines, horizontal segment first:
255 inline void fl_xyline(int x, int y, int x1) {fl_graphics_driver->xyline(x,y,x1);}
259 inline void fl_xyline(int x, int y, int x1, int y2) {fl_graphics_driver->xyline(x,y,x1,y2);}
264 inline void fl_xyline(int x, int y, int x1, int y2, int x3) {fl_graphics_driver->xyline(x,y,x1,y2,x3);}
265
266 // draw rectilinear lines, vertical segment first:
270 inline void fl_yxline(int x, int y, int y1) {fl_graphics_driver->yxline(x,y,y1);}
274 inline void fl_yxline(int x, int y, int y1, int x2) {fl_graphics_driver->yxline(x,y,y1,x2);}
279 inline void fl_yxline(int x, int y, int y1, int x2, int y3) {fl_graphics_driver->yxline(x,y,y1,x2,y3);}
280
281 // circular lines and pie slices (code in fl_arci.C):
304 inline void fl_arc(int x, int y, int w, int h, double a1, double a2)
    {fl_graphics_driver->arc(x,y,w,h,a1,a2); }
317 inline void fl_pie(int x, int y, int w, int h, double a1, double a2)
    {fl_graphics_driver->pie(x,y,w,h,a1,a2); }
319 FL_EXPORT void fl_chord(int x, int y, int w, int h, double a1, double a2); // nyi
320
321 // scalable drawing code (code in fl_vertex.C and fl_arc.C):
326 inline void fl_push_matrix() { fl_graphics_driver->push_matrix(); }
330 inline void fl_pop_matrix() { fl_graphics_driver->pop_matrix(); }
335 inline void fl_scale(double x, double y) { fl_graphics_driver->scale(x, y); }
340 inline void fl_scale(double x) { fl_graphics_driver->scale(x, x); }
345 inline void fl_translate(double x, double y) { fl_graphics_driver->translate(x, y); }
350 inline void fl_rotate(double d) { fl_graphics_driver->rotate(d); }
357 inline void fl_mult_matrix(double a, double b, double c, double d, double x,double y)
    { fl_graphics_driver->mult_matrix(a, b, c, d, x, y); }
358
362 inline void fl_begin_points() {fl_graphics_driver->begin_points(); }
366 inline void fl_begin_line() {fl_graphics_driver->begin_line(); }
370 inline void fl_begin_loop() {fl_graphics_driver->begin_loop(); }
374 inline void fl_begin_polygon() {fl_graphics_driver->begin_polygon(); }
379 inline void fl_vertex(double x, double y) {fl_graphics_driver->vertex(x,y); }
388 inline void fl_curve(double X0, double Y0, double X1, double Y1, double X2, double X3, double
    Y3)
389 {fl_graphics_driver->curve(X0,Y0,X1,Y1,X2,Y2,X3,Y3); }
416 inline void fl_arc(double x, double y, double r, double start, double end)
    {fl_graphics_driver->arc(x,y,r,start,end); }
424 inline void fl_circle(double x, double y, double r) {fl_graphics_driver->circle(x,y,r); }
428 inline void fl_end_points() {fl_graphics_driver->end_points(); }
432 inline void fl_end_line() {fl_graphics_driver->end_line(); }
436 inline void fl_end_loop() {fl_graphics_driver->end_loop(); }
440 inline void fl_end_polygon() {fl_graphics_driver->end_polygon(); }
455 inline void fl_begin_complex_polygon() {fl_graphics_driver->begin_complex_polygon(); }
462 inline void fl_gap() {fl_graphics_driver->gap(); }
466 inline void fl_end_complex_polygon() {fl_graphics_driver->end_complex_polygon(); }
467 // get and use transformed positions:
472 inline double fl_transform_x(double x, double y) {return fl_graphics_driver->transform_x(x, y); }
477 inline double fl_transform_y(double x, double y) {return fl_graphics_driver->transform_y(x, y); }
482 inline double fl_transform_dx(double x, double y) {return fl_graphics_driver->transform_dx(x, y); }
487 inline double fl_transform_dy(double x, double y) {return fl_graphics_driver->transform_dy(x, y); }
492 inline void fl_transformed_vertex(double xf, double yf) {fl_graphics_driver->transformed_vertex(xf,yf);
    }
497 /* NOTE: doxygen comments here to avoid triplication in os-specific sources */
498
499 // Fonts:
509 inline void fl_font(Fl_Font face, Fl_Fontsize fsize) { fl_graphics_driver->font(face,fsize); }
510
515 inline Fl_Font fl_font() {return fl_graphics_driver->font();}
520 inline Fl_Fontsize fl_size() {return fl_graphics_driver->size();}
521
522 // information you can get about the current font:
527 inline int fl_height() {return fl_graphics_driver->height();}
528 FL_EXPORT int fl_height(int font, int size);
533 inline int fl_descent() {return fl_graphics_driver->descent();}
536 FL_EXPORT double fl_width(const char* txt);
539 inline double fl_width(const char* txt, int n) {return fl_graphics_driver->width(txt, n);}
544 inline double fl_width(unsigned int c) {return fl_graphics_driver->width(c);}
555 FL_EXPORT void fl_text_extents(const char*, int& dx, int& dy, int& w, int& h); // NO fltk symbol
    expansion will be performed

```

```

559 inline void fl_text_extents(const char *t, int n, int& dx, int& dy, int& w, int& h)
560   {fl_graphics_driver->text_extents(t, n, dx, dy, w, h);}
561
562 // font encoding:
563 // Note: doxygen comments here to avoid duplication for os-sepecific cases
570 FL_EXPORT const char *fl_latin1_to_local(const char *t, int n=-1);
577 FL_EXPORT const char *fl_local_to_latin1(const char *t, int n=-1);
584 FL_EXPORT const char *fl_mac_roman_to_local(const char *t, int n=-1);
591 FL_EXPORT const char *fl_local_to_mac_roman(const char *t, int n=-1);
606 FL_EXPORT void fl_draw(const char* str, int x, int y);
614 FL_EXPORT void fl_draw(int angle, const char* str, int x, int y);
618 inline void fl_draw(const char* str, int n, int x, int y) {fl_graphics_driver->draw(str,n,x,y); }
628 inline void fl_draw(int angle, const char* str, int n, int x, int y)
   {fl_graphics_driver->draw(angle,str,n,x,y); }
632 inline void fl_rtl_draw(const char* str, int n, int x, int y) {fl_graphics_driver->rtl_draw(str,n,x,y);
   }
633 FL_EXPORT void fl_measure(const char* str, int& x, int& y,
634   int draw_symbols = 1);
635 FL_EXPORT void fl_draw(const char* str, int x, int y, int w, int h,
636   Fl_Align align,
637   Fl_Image* img=0, int draw_symbols = 1);
638 FL_EXPORT void fl_draw(const char* str, int x, int y, int w, int h,
639   Fl_Align align,
640   void (*callthis)(const char *,int,int,int),
641   Fl_Image* img=0, int draw_symbols = 1);
642
643 // boxtypes:
644 FL_EXPORT void fl_frame(const char* s, int x, int y, int w, int h);
645 FL_EXPORT void fl_frame2(const char* s, int x, int y, int w, int h);
646 FL_EXPORT void fl_draw_box(Fl_Boxtype, int x, int y, int w, int h, Fl_Color);
647
648 // images:
649
685 inline void fl_draw_image(const uchar* buf, int X,int Y,int W,int H, int D=3, int L=0)
686   { fl_graphics_driver->draw_image(buf, X, Y, W, H, D, L); }
687
692 inline void fl_draw_image_mono(const uchar* buf, int X,int Y,int W,int H, int D=1, int L=0)
693   { fl_graphics_driver->draw_image_mono(buf, X, Y, W, H, D, L); }
694
728 inline void fl_draw_image(Fl_Draw_Image_Cb cb, void* data, int X,int Y,int W,int H, int D=3)
729   { fl_graphics_driver->draw_image(cb, data, X, Y, W, H, D); }
730
735 inline void fl_draw_image_mono(Fl_Draw_Image_Cb cb, void* data, int X,int Y,int W,int H, int D=1)
736   { fl_graphics_driver->draw_image_mono(cb, data, X, Y, W, H, D); }
737
743 /* note: doxygen comment here to avoid triplification in os-speciic files */
744 FL_EXPORT char fl_can_do_alpha_blending();
745
763 /* note: doxygen comment here to avoid triplification in os-speciic files */
764 FL_EXPORT uchar *fl_read_image(uchar *p,int X,int Y,int W,int H,int alpha=0);
765
766 // pixmaps:
767 FL_EXPORT int fl_draw_pixmap(/*const*/ char* const* data, int x,int y,Fl_Color=FL_GRAY);
768 FL_EXPORT int fl_draw_pixmap(const char* const* cdata, int x,int y,Fl_Color=FL_GRAY);
769 FL_EXPORT int fl_measure_pixmap(/*const*/ char* const* data, int &w, int &h);
770 FL_EXPORT int fl_measure_pixmap(const char* const* cdata, int &w, int &h);
771
772 // other:
773 FL_EXPORT void fl_scroll(int X, int Y, int W, int H, int dx, int dy,
774   void (*draw_area)(void*, int,int,int,int), void* data);
775 FL_EXPORT const char* fl_shortcut_label(unsigned int shortcut);
776 FL_EXPORT const char* fl_shortcut_label(unsigned int shortcut, const char **eom);
777 FL_EXPORT unsigned int fl_old_shortcut(const char* s);
778 FL_EXPORT void fl_overlay_rect(int x,int y,int w,int h);
779 FL_EXPORT void fl_overlay_clear();
780 FL_EXPORT void fl_cursor(Fl_Cursor);
781 FL_EXPORT void fl_cursor(Fl_Cursor, Fl_Color fg, Fl_Color bg=FL_WHITE);
782 FL_EXPORT const char* fl_expand_text(const char* from, char* buf, int maxbuf,
783   double maxw, int& n, double &width,
784   int wrap, int draw_symbols = 0);
785
786 // XIM:
788 FL_EXPORT void fl_set_status(int X, int Y, int W, int H);
790 FL_EXPORT void fl_set_spot(int font, int size, int X, int Y, int W, int H, Fl_Window *win=0);
792 FL_EXPORT void fl_reset_spot(void);
793
794
795
796 // XForms symbols:
797 FL_EXPORT int fl_draw_symbol(const char* label,int x,int y,int w,int h, Fl_Color);
798 FL_EXPORT int fl_add_symbol(const char* name, void (*drawit)(Fl_Color), int scalable);
801 #endif
802
803 //
804 // End of "$Id$".
805 //

```

32.35 Fl_Export.H

```

1 /*
2 * "$Id$"
3 *
4 * WIN32 DLL export .
5 *
6 * Copyright 1998-2010 by Bill Spitzak and others.
7 *
8 * This library is free software.  Distribution and use rights are outlined in
9 * the file "COPYING" which should have been included with this file.  If this
10 * file is missing or damaged, see the license at:
11 *
12 *     http://www.fltk.org/COPYING.php
13 *
14 * Please report all bugs and problems on the following page:
15 *
16 *     http://www.fltk.org/str.php
17 */
18
19 #ifndef Fl_Export_H
20 # define Fl_Export_H
21
22 /*
23 * The following is only used when building DLLs under WIN32...
24 */
25
26 # if defined(FL_DLL)
27 #   ifdef FL_LIBRARY
28 #     define FL_EXPORT __declspec(dllexport)
29 #   else
30 #     define FL_EXPORT __declspec(dllimport)
31 #   endif /* FL_LIBRARY */
32 #   elif __GNUC__ >= 4
33 #     define FL_EXPORT __attribute__((visibility ("default")))
34 #   else
35 #     define FL_EXPORT
36 #   endif /* FL_DLL */
37
38 #endif /* !Fl_Export_H */
39
40 /*
41 * End of "$Id$".
42 */

```

32.36 Fl_File_Browser.H

```

1 //
2 // "$Id$"
3 //
4 // FileBrowser definitions.
5 //
6 // Copyright 1999-2010 by Michael Sweet.
7 //
8 // This library is free software.  Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file.  If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_File_Browser widget . */
21
22 //
23 // Include necessary header files...
24 //
25
26 #ifndef _Fl_File_Browser_H_
27 # define _Fl_File_Browser_H_
28
29 # include "Fl_Browser.H"
30 # include "Fl_File_Icon.H"
31 # include "filename.H"
32
33
34 //
35 // Fl_File_Browser class...
36 //
37

```



```

39 class FL_EXPORT Fl_File_Browser : public Fl_Browser {
40
41     int             filetype_;
42     const char     *directory_;
43     uchar          iconsize_;
44     const char     *pattern_;
45
46     int             full_height() const;
47     int             item_height(void *) const;
48     int             item_width(void *) const;
49     void            item_draw(void *, int, int, int, int) const;
50     int             incr_height()const { return (item_height(0)); }
51
52 public:
53     enum { FILES, DIRECTORIES };
54
55     Fl_File_Browser(int, int, int, int, const char * = 0);
56
57     uchar           iconsize()const { return (iconsize_); };
58     void            iconsize(uchar s) { iconsize_ = s; redraw(); };
59
60     void            filter(const char *pattern);
61     const char     *filter()const { return (pattern_); };
62
63     int             load(const char *directory, Fl_File_Sort_F *sort = fl_numericsort);
64
65     Fl_Fontsize     textsize()const { return Fl_Browser::textsize(); };
66     void            textsize(Fl_Fontsize s) { Fl_Browser::textsize(s); iconsize_ = (uchar)(3 * s / 2); };
67
68     int             filetype()const { return (filetype_); };
69     void            filetype(int t) { filetype_ = t; };
70 };
71
72 #endif // !_Fl_File_Browser_H_
73
74 //
75 // End of "$Id$".
76 //

```

32.37 Fl_File_Chooser.H

```

1 //
2 // "$Id$"
3 //
4 // Fl_File_Chooser dialog for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2015 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18 // =====
19 // DO NOT EDIT FL/Fl_File_Chooser.H and src/Fl_File_Chooser.cxx !!!
20 // =====
21 // Please use fluid to change src/Fl_File_Chooser.fl interactively
22 // and then use fluid to "write code" or edit and use fluid -c .
23 // =====
24 //
25
26 // generated by Fast Light User Interface Designer (fluid) version 1.0308
27
28 #ifndef Fl_File_Chooser_H
29 #define Fl_File_Chooser_H
30 #include <FL/Fl.H>
31 #include <FL/Fl_Double_Window.H>
32 #include <stdio.h>
33 #include <stdlib.h>
34 #include <string.h>
35 #include <FL/Fl_Group.H>
36 #include <FL/Fl_Choice.H>
37 #include <FL/Fl_Menu_Button.H>
38 #include <FL/Fl_Button.H>
39 #include <FL/Fl_Preferences.H>
40 #include <FL/Fl_Tile.H>
41 #include <FL/Fl_File_Browser.H>
42 #include <FL/Fl_Box.H>
43 #include <FL/Fl_Check_Button.H>
44 #include <FL/Fl_File_Input.H>

```

```

45 #include <FL/Fl_Return_Button.H>
46 #include <FL/fl_ask.H>
47
48 class FL_EXPORT Fl_File_Chooser {
49 public:
50     enum { SINGLE = 0, MULTI = 1, CREATE = 2, DIRECTORY = 4 };
51 private:
52     static Fl_Preferences *prefs_;
53     void (*callback_)(Fl_File_Chooser*, void *);
54     void *data_;
55     char directory_[FL_PATH_MAX];
56     char pattern_[FL_PATH_MAX];
57     char preview_text_[2048];
58     int type_;
59     void favoritesButtonCB();
60     void favoritesCB(Fl_Widget *w);
61     void fileListCB();
62     void fileNameCB();
63     void newdir();
64     static void previewCB(Fl_File_Chooser *fc);
65     void showChoiceCB();
66     void update_favorites();
67     void update_preview();
68 public:
69     Fl_File_Chooser(const char *d, const char *p, int t, const char *title);
70 private:
71     Fl_Double_Window *window;
72     inline void cb_window_i(Fl_Double_Window*, void*);
73     static void cb_window(Fl_Double_Window*, void*);
74     Fl_Choice *showChoice;
75     inline void cb_showChoice_i(Fl_Choice*, void*);
76     static void cb_showChoice(Fl_Choice*, void*);
77     Fl_Menu_Button *favoritesButton;
78     inline void cb_favoritesButton_i(Fl_Menu_Button*, void*);
79     static void cb_favoritesButton(Fl_Menu_Button*, void*);
80 public:
81     Fl_Button *newButton;
82 private:
83     inline void cb_newButton_i(Fl_Button*, void*);
84     static void cb_newButton(Fl_Button*, void*);
85     inline void cb__i(Fl_Tile*, void*);
86     static void cb_(Fl_Tile*, void*);
87     Fl_File_Browser *fileList;
88     inline void cb_fileList_i(Fl_File_Browser*, void*);
89     static void cb_fileList(Fl_File_Browser*, void*);
90     Fl_Box *previewBox;
91 public:
92     Fl_Check_Button *previewButton;
93 private:
94     inline void cb_previewButton_i(Fl_Check_Button*, void*);
95     static void cb_previewButton(Fl_Check_Button*, void*);
96 public:
97     Fl_Check_Button *showHiddenButton;
98 private:
99     inline void cb_showHiddenButton_i(Fl_Check_Button*, void*);
100     static void cb_showHiddenButton(Fl_Check_Button*, void*);
101     Fl_File_Input *fileName;
102     inline void cb_fileName_i(Fl_File_Input*, void*);
103     static void cb_fileName(Fl_File_Input*, void*);
104     Fl_Return_Button *okButton;
105     inline void cb_okButton_i(Fl_Return_Button*, void*);
106     static void cb_okButton(Fl_Return_Button*, void*);
107     Fl_Button *cancelButton;
108     inline void cb_cancelButton_i(Fl_Button*, void*);
109     static void cb_cancelButton(Fl_Button*, void*);
110     Fl_Double_Window *favWindow;
111     Fl_File_Browser *favList;
112     inline void cb_favList_i(Fl_File_Browser*, void*);
113     static void cb_favList(Fl_File_Browser*, void*);
114     Fl_Button *favUpButton;
115     inline void cb_favUpButton_i(Fl_Button*, void*);
116     static void cb_favUpButton(Fl_Button*, void*);
117     Fl_Button *favDeleteButton;
118     inline void cb_favDeleteButton_i(Fl_Button*, void*);
119     static void cb_favDeleteButton(Fl_Button*, void*);
120     Fl_Button *favDownButton;
121     inline void cb_favDownButton_i(Fl_Button*, void*);
122     static void cb_favDownButton(Fl_Button*, void*);
123     Fl_Button *favCancelButton;
124     inline void cb_favCancelButton_i(Fl_Button*, void*);
125     static void cb_favCancelButton(Fl_Button*, void*);
126     Fl_Return_Button *favOkButton;
127     inline void cb_favOkButton_i(Fl_Return_Button*, void*);
128     static void cb_favOkButton(Fl_Return_Button*, void*);
129 public:
130     ~Fl_File_Chooser();
131     void callback(void (*cb)(Fl_File_Chooser *, void *), void *d = 0);

```

```

132 void color(Fl_Color c);
133 Fl_Color color();
134 int count();
135 void directory(const char *d);
136 char * directory();
137 void filter(const char *p);
138 const char * filter();
139 int filter_value();
140 void filter_value(int f);
141 void hide();
142 void iconsize(uchar s);
143 uchar iconsize();
144 void label(const char *l);
145 const char * label();
146 void ok_label(const char *l);
147 const char * ok_label();
148 void preview(int e);
149 int preview()const { return previewButton->value(); };
150 private:
151 void showHidden(int e);
152 void remove_hidden_files();
153 public:
154 void rescan();
155 void rescan_keep_filename();
156 void show();
157 int shown();
158 void textcolor(Fl_Color c);
159 Fl_Color textcolor();
160 void textfont(Fl_Font f);
161 Fl_Font textfont();
162 void textsize(Fl_Fontsize s);
163 Fl_Fontsize textsize();
164 void type(int t);
165 int type();
166 void * user_data() const;
167 void user_data(void *d);
168 const char *value(int f = 1);
169 void value(const char *filename);
170 int visible();
174 static const char *add_favorites_label;
178 static const char *all_files_label;
182 static const char *custom_filter_label;
186 static const char *existing_file_label;
190 static const char *favorites_label;
194 static const char *filename_label;
198 static const char *filesystems_label;
202 static const char *manage_favorites_label;
206 static const char *new_directory_label;
210 static const char *new_directory_tooltip;
214 static const char *preview_label;
218 static const char *save_label;
222 static const char *show_label;
226 static const char *hidden_label;
231 static Fl_File_Sort_F *sort;
232 private:
233 Fl_Widget* ext_group;
234 public:
235 Fl_Widget* add_extra(Fl_Widget* gr);
236 };
237 FL_EXPORT char *fl_dir_chooser(const char *message,const char *fname,int relative=0);
238 FL_EXPORT char *fl_file_chooser(const char *message,const char *pat,const char *fname,int relative=0);
239 FL_EXPORT void fl_file_chooser_callback(void (*cb)(const char*));
240 FL_EXPORT void fl_file_chooser_ok_label(const char*l);
241 #endif
242
243 //
244 // End of "$Id$".
245 //

```

32.38 Fl_File_Icon.H

```

1 //
2 // "$Id$"
3 //
4 // Fl_File_Icon definitions.
5 //
6 // Copyright 1999-2010 by Michael Sweet.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 // http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:

```

```

15 //
16 //      http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_File_Icon widget . */
21
22 //
23 // Include necessary header files...
24 //
25
26 #ifndef _Fl_Fl_File_Icon_H_
27 # define _Fl_Fl_File_Icon_H_
28
29 # include "Fl.H"
30
31
32 //
33 // Special color value for the icon color.
34 //
35
36 # define FL_ICON_COLOR (Fl_Color)0xffffffff
37 //
38 // Fl_File_Icon class...
39 //
40 //
41 //
42
43 class FL_EXPORT Fl_File_Icon {
44
45     static Fl_File_Icon *first_;      // Pointer to first icon/filetype
46     Fl_File_Icon      *next_;        // Pointer to next icon/filetype
47     const char        *pattern_;      // Pattern string
48     int                type_;         // Match only if directory or file?
49     int                num_data_;     // Number of data elements
50     int                alloc_data_;   // Number of allocated elements
51     short              *data_;        // Icon data
52
53 public:
54
55     enum                // File types
56     {
57         ANY,            // Any kind of file
58         PLAIN,          // Only plain files
59         FIFO,           // Only named pipes
60         DEVICE,         // Only character and block devices
61         LINK,           // Only symbolic links
62         DIRECTORY      // Only directories
63     };
64
65     enum                // Data opcodes
66     {
67         END,            // End of primitive/icon
68         COLOR,          // Followed by color value (2 shorts)
69         LINE,           // Start of line
70         CLOSEDLINE,    // Start of closed line
71         POLYGON,       // Start of polygon
72         OUTLINEPOLYGON, // Followed by outline color (2 shorts)
73         VERTEX          // Followed by scaled X,Y
74     };
75
76     Fl_File_Icon(const char *p, int t, int nd = 0, short *d = 0);
77     ~Fl_File_Icon();
78
79     short          *add(short d);
80
81     short          *add_color(Fl_Color c)
82     { short *d = add((short)COLOR); add((short)(c >> 16)); add((short)c); return (d); }
83
84     short          *add_vertex(int x, int y)
85     { short *d = add((short)VERTEX); add((short)x); add((short)y); return (d); }
86
87     short          *add_vertex(float x, float y)
88     { short *d = add((short)VERTEX); add((short)(x * 10000.0));
89       add((short)(y * 10000.0)); return (d); }
90
91     void          clear() { num_data_ = 0; }
92
93     void          draw(int x, int y, int w, int h, Fl_Color ic, int active = 1);
94
95     void          label(Fl_Widget *w);
96
97     static void   labeltype(const Fl_Label *o, int x, int y, int w, int h, Fl_Align a);
98     void          load(const char *f);
99     int           load_fti(const char *fti);
100    int           load_image(const char *i);
101
102    Fl_File_Icon  *next() { return (next_); }
103
104
105

```

```

127  const char    *pattern() { return (pattern_); }
128
130  int           size() { return (num_data_); }
131
143  int           type() { return (type_); }
144
146  short        *value() { return (data_); }
147
148  static Fl_File_Icon *find(const char *filename, int filetype = ANY);
149
151  static Fl_File_Icon *first() { return (first_); }
152  static void    load_system_icons(void);
153 };
154
155 #endif // !_Fl_File_Icon_H_
156
157 //
158 // End of "$Id$".
159 //

```

32.39 Fl_File_Input.H

```

1 //
2 // "$Id$"
3 //
4 // File_Input header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 // Original version Copyright 1998 by Curtis Edwards.
8 //
9 // This library is free software. Distribution and use rights are outlined in
10 // the file "COPYING" which should have been included with this file. If this
11 // file is missing or damaged, see the license at:
12 //
13 //     http://www.fltk.org/COPYING.php
14 //
15 // Please report all bugs and problems on the following page:
16 //
17 //     http://www.fltk.org/str.php
18 //
19
20 /* \file
21 Fl_File_Input widget . */
22
23 #ifndef Fl_File_Input_H
24 # define Fl_File_Input_H
25
26 # include <FL/Fl_Input.H>
27
28 class FL_EXPORT Fl_File_Input : public Fl_Input {
29
30     Fl_Color    errorcolor_;
31     char        ok_entry_;
32     uchar       down_box_;
33     short       buttons_[200];
34     short       pressed_;
35
36     void        draw_buttons();
37     int         handle_button(int event);
38     void        update_buttons();
39
40 public:
41
42     Fl_File_Input(int X, int Y, int W, int H, const char *L=0);
43
44     virtual int handle(int event);
45
46 protected:
47     virtual void draw();
48
49 public:
50     Fl_Boxtype down_box()const { return (Fl_Boxtype)down_box_; }
51     void        down_box(Fl_Boxtype b) { down_box_ = b; }
52
53     Fl_Color    errorcolor()const { return errorcolor_; }
54     void        errorcolor(Fl_Color c) { errorcolor_ = c; }
55
56     int         value(const char *str);
57     int         value(const char *str, int len);
58
59     const char *value() { return Fl_Input::value(); }
60 };
61
62 #endif // !Fl_File_Input_H
63

```

```
94
95 //
96 // End of "$Id$".
97 //
```

32.40 Fl_Fill_Dial.H

```
1 //
2 // "$Id$"
3 //
4 // Filled dial header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 // http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 // http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Fill_Dial widget . */
21
22 #ifndef Fl_Fill_Dial_H
23 #define Fl_Fill_Dial_H
24
25 #include "Fl_Dial.H"
26
27 class FL_EXPORT Fl_Fill_Dial : public Fl_Dial {
28 public:
29     Fl_Fill_Dial(int X,int Y,int W,int H, const char *L);
30 };
31
32 #endif
33
34 //
35 // End of "$Id$".
36 //
```

32.41 Fl_Fill_Slider.H

```
1 //
2 // "$Id$"
3 //
4 // Filled slider header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 // http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 // http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Fill_Slider widget . */
21
22 #ifndef Fl_Fill_Slider_H
23 #define Fl_Fill_Slider_H
24
25 #include "Fl_Slider.H"
26
27 class FL_EXPORT Fl_Fill_Slider : public Fl_Slider {
28 public:
29     Fl_Fill_Slider(int X,int Y,int W,int H,const char *L=0);
30 };
31
32 #endif
33
34 //
35 // End of "$Id$".
36 //
```

32.42 Fl_Float_Input.H

```

1 //
2 // "$Id$"
3 //
4 // Floating point input header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2011 by Bill Spitzak and others.
7 //
8 // This library is free software.  Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file.  If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Float_Input widget . */
21
22 #ifndef Fl_Float_Input_H
23 #define Fl_Float_Input_H
24
25 #include "Fl_Input.H"
26
27
28
29
30
31
32 class FL_EXPORT Fl_Float_Input : public Fl_Input {
33 public:
34     Fl_Float_Input(int X,int Y,int W,int H,const char *l = 0);
35 };
36
37 #endif
38
39 //
40 // End of "$Id$".
41 //

```

32.43 Fl_FormsBitmap.H

```

1 //
2 // "$Id$"
3 //
4 // Forms bitmap header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software.  Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file.  If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_FormsBitmap widget . */
21
22 #ifndef Fl_FormsBitmap_H
23 #define Fl_FormsBitmap_H
24
25 #include "Fl_Bitmap.H"
26
27
28
29
30
31
32 class FL_EXPORT Fl_FormsBitmap : public Fl_Widget {
33 public:
34     Fl_FormsBitmap(Fl_Boxtype, int, int, int, int, const char * = 0);
35     void set(int W, int H, const uchar *bits);
36     void bitmap(Fl_Bitmap *B) {b = B;}
37     Fl_Bitmap *bitmap()const {return b;}
38 };
39
40 #endif
41
42 //
43 // End of "$Id$".
44 //

```

32.44 Fl_FormsPixmap.H

```

1 //
2 // "$Id$"
3 //
4 // Forms pixmap header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software.  Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file.  If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_FormsPixmap widget . */
21
22 #ifndef Fl_FormsPixmap_H
23 #define Fl_FormsPixmap_H
24
25 #include "Fl_Pixmap.H"
26
27 class FL_EXPORT Fl_FormsPixmap : public Fl_Widget {
28     Fl_Pixmap *b;
29 protected:
30     void draw();
31 public:
32     Fl_FormsPixmap(Fl_Boxtype t, int X, int Y, int W, int H, const char *L= 0);
33     void set(const char * bits);
34     void Pixmap(Fl_Pixmap *B) {b = B;}
35     Fl_Pixmap *Pixmap()const {return b;}
36 };
37
38 #endif
39
40 // End of "$Id$".
41 //

```

32.45 Fl_Free.H

```

1 //
2 // "$Id$"
3 //
4 // Forms free header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software.  Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file.  If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Free widget . */
21
22 #ifndef Fl_Free_H
23 #define Fl_Free_H
24
25 #ifndef Fl_Widget_H
26 #include "Fl_Widget.H"
27 #endif
28
29 #define FL_NORMAL_FREE          1
30 #define FL_SLEEPING_FREE       2
31 #define FL_INPUT_FREE          3
32 #define FL_CONTINUOUS_FREE     4
33 #define FL_ALL_FREE            5
34
35 typedef int (*FL_HANDLEPTR)(Fl_Widget *, int, float, float, char);

```



```

37
57 class FL_EXPORT Fl_Free : public Fl_Widget {
58     FL_HANDLEPTR hfunc;
59     static void step(void *);
60 protected:
61     void draw();
62 public:
63     int handle(int e);
64     Fl_Free(uchar t,int X,int Y,int W,int H,const char *L,FL_HANDLEPTR hdl);
65     ~Fl_Free();
66 };
67
68 // old event names for compatibility:
69 #define FL_MOUSE      FL_DRAG
70 #define FL_DRAW      100
71 #define FL_STEP      101
72 #define FL_FREEMEM   102
73 #define FL_FREEZE    103
74 #define FL_THAW      104
75 #endif
76
77
78 //
79 // End of "$Id$".
80 //

```

32.46 Fl_GIF_Image.H

```

1 //
2 // "$Id$"
3 //
4 // GIF image header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software.  Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file.  If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_GIF_Image widget . */
21
22 #ifndef Fl_GIF_Image_H
23 #define Fl_GIF_Image_H
24 # include "Fl_Pixmap.H"
25
31 class FL_EXPORT Fl_GIF_Image : public Fl_Pixmap {
32
33     public:
34
35     Fl_GIF_Image(const char* filename);
36 };
37
38 #endif
39
40 //
41 // End of "$Id$".
42 //

```

32.47 Fl_GL_Window.H

```

1 //
2 // "$Id$"
3 //
4 // OpenGL header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2015 by Bill Spitzak and others.
7 //
8 // This library is free software.  Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file.  If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //

```

```

16 //      http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Gl_Window widget . */
21
22 #ifndef Fl_Gl_Window_H
23 #define Fl_Gl_Window_H
24
25 #include "Fl_Window.H"
26
27 #ifndef GLContext
31 typedef void* GLContext; // actually a GLXContext or HGLDC
32 #endif
33
34 class Fl_Gl_Choice; // structure to hold result of glXChooseVisual
35
36 class FL_EXPORT Fl_Gl_Window : public Fl_Window {
37
38     int mode_;
39     const int *alist;
40     Fl_Gl_Choice *g;
41     GLContext context_;
42     char valid_f_;
43     char damage1_; // damage() of back buffer
44     virtual void draw_overlay();
45     void init();
46
47     void *overlay;
48     void make_overlay();
49     friend class _Fl_Gl_Overlay;
50
51     static int can_do(int, const int *);
52     int mode(int, const int *);
53     static int gl_plugin_linkage();
54
55 public:
56
57     void show();
58     void show(int a, char **b) {Fl_Window::show(a,b);}
59     void flush();
60     void hide();
61     void resize(int,int,int,int);
62     int handle(int);
63
64     char valid()const {return valid_f_ & 1;}
65     void valid(char v) {if (v) valid_f_ |= 1; else valid_f_ &= 0xfe;}
66     void invalidate();
67
68     char context_valid()const {return valid_f_ & 2;}
69     void context_valid(char v) {if (v) valid_f_ |= 2; else valid_f_ &= 0xfd;}
70
71     static int can_do(int m) {return can_do(m,0);}
72     static int can_do(const int *m) {return can_do(0, m);}
73     int can_do() {return can_do(mode_,alist);}
74     Fl_Mode mode()const {return (Fl_Mode)mode_;}
75     int mode(int a) {return mode(a,0);}
76     int mode(const int *a) {return mode(0, a);}
77     void* context()const {return context_;}
78     void context(void*, int destroy_flag = 0);
79     void make_current();
80     void swap_buffers();
81     void ortho();
82
83     int can_do_overlay();
84     void redraw_overlay();
85     void hide_overlay();
86     void make_overlay_current();
87
88     // Note: Doxygen docs in Fl_Widget.H to avoid redundancy.
89     virtual Fl_Gl_Window* as_gl_window() {return this;}
90
91 #ifdef __APPLE__
92     float pixels_per_unit();
93 #else
94     float pixels_per_unit() { return 1; }
95 #endif
96     int pixel_w() { return int(pixels_per_unit() * w() + 0.5); }
97     int pixel_h() { return int(pixels_per_unit() * h() + 0.5); }
98
99     ~Fl_Gl_Window();
100     Fl_Gl_Window(int W, int H, const char *l=0) : Fl_Window(W,H,1) {init();}
101     Fl_Gl_Window(int X, int Y, int W, int H, const char *l=0)
102         : Fl_Window(X,Y,W,H,1) {init();}
103
104 protected:
105     virtual void draw();

```

```

278 };
279
280 #endif
281
282 //
283 // End of "$Id$".
284 //

```

32.48 Fl_Group.H

```

1 //
2 // "$Id$"
3 //
4 // Group header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Group, Fl_End classes . */
21
22 #ifndef Fl_Group_H
23 #define Fl_Group_H
24
25 #ifndef Fl_Widget_H
26 #include "Fl_Widget.H"
27 #endif
28
29 class FL_EXPORT Fl_Group : public Fl_Widget {
30
31     Fl_Widget** array_;
32     Fl_Widget* savedfocus_;
33     Fl_Widget* resizable_;
34     int children_;
35     int *sizes_; // remembered initial sizes of children
36
37     int navigation(int);
38     static Fl_Group *current_;
39
40     // unimplemented copy ctor and assignment operator
41     Fl_Group(const Fl_Group&);
42     Fl_Group& operator=(const Fl_Group&);
43
44 protected:
45     void draw();
46     void draw_child(Fl_Widget& widget) const;
47     void draw_children();
48     void draw_outside_label(const Fl_Widget& widget) const;
49     void update_child(Fl_Widget& widget) const;
50     int *sizes();
51
52 public:
53     int handle(int);
54     void begin();
55     void end();
56     static Fl_Group *current();
57     static void current(Fl_Group *g);
58
59     int children()const {return children_;}
60     Fl_Widget* child(int n)const {return array()[n];}
61     int find(const Fl_Widget*) const;
62     int find(const Fl_Widget& o)const {return find(&o);}
63     Fl_Widget* const* array() const;
64
65     void resize(int,int,int,int);
66     Fl_Group(int,int,int,int, const char * = 0);
67     virtual ~Fl_Group();
68     void add(Fl_Widget&);
69     void add(Fl_Widget* o) {add(*o);}
70     void insert(Fl_Widget&, int i);
71     void insert(Fl_Widget& o, Fl_Widget* before) {insert(o,find(before));}
72     void remove(int index);
73     void remove(Fl_Widget&);
74     void remove(Fl_Widget* o) {remove(*o);}

```

```

112 void clear();
113
114 void resizable(Fl_Widget& o) {resizable_ = &o;}
148 void resizable(Fl_Widget* o) {resizable_ = o;}
152 Fl_Widget* resizable()const {return resizable_;}
156 void add_resizable(Fl_Widget& o) {resizable_ = &o; add(o);}
157 void init_sizes();
158
168 void clip_children(int c) { if (c) set_flag(CLIP_CHILDREN); else clear_flag(CLIP_CHILDREN); }
176 unsigned int clip_children() { return (flags() & CLIP_CHILDREN) != 0; }
177
178 // Note: Doxygen docs in Fl_Widget.H to avoid redundancy.
179 virtual Fl_Group* as_group() { return this; }
180
181 // back compatibility functions:
182
188 void focus(Fl_Widget* W) {W->take_focus();}
189
191 Fl_Widget* & _ddfdesign_kludge() {return resizable_;}
192
194 void forms_end();
195 };
196
197 // dummy class used to end child groups in constructors for complex
198 // subclasses of Fl_Group:
218 class FL_EXPORT Fl_End {
219 public:
221 Fl_End() {Fl_Group::current()->end();}
222 };
223
224 #endif
225
226 //
227 // End of "$Id$".
228 //

```

32.49 Fl_Help_Dialog.H

```

1 //
2 // "$Id$"
3 //
4 // Fl_Help_Dialog dialog for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2015 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 // http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 // http://www.fltk.org/str.php
17 //
18 // =====
19 // DO NOT EDIT FL/Fl_Help_Dialog.H and src/Fl_Help_Dialog.cxx !!!
20 // =====
21 // Please use fluid to change src/Fl_Help_Dialog.fl interactively
22 // and then use fluid to "write code" or edit and use fluid -c .
23 // =====
24 //
25
26 // generated by Fast Light User Interface Designer (fluid) version 1.0308
27
28 #ifndef Fl_Help_Dialog_H
29 #define Fl_Help_Dialog_H
30 #include <FL/Fl.H>
31 #include <FL/Fl_Double_Window.H>
32 #include <FL/Fl_Group.H>
33 #include <FL/Fl_Button.H>
34 #include <FL/Fl_Input.H>
35 #include <FL/Fl_Box.H>
36 #include <FL/Fl_Help_View.H>
37
38 class FL_EXPORT Fl_Help_Dialog {
39 int index_;
40 int max_;
41 int line_[100]; // FIXME: we must remove those static numbers
42 char file_[100][FL_PATH_MAX]; // FIXME: we must remove those static numbers
43 int find_pos_;
44 public:
45 Fl_Help_Dialog();
46 private:
47 Fl_Double_Window *window_;

```

```

48 Fl_Button *back_;
49 inline void cb_back__i(Fl_Button*, void*);
50 static void cb_back_(Fl_Button*, void*);
51 Fl_Button *forward_;
52 inline void cb_forward__i(Fl_Button*, void*);
53 static void cb_forward_(Fl_Button*, void*);
54 Fl_Button *smaller_;
55 inline void cb_smaller__i(Fl_Button*, void*);
56 static void cb_smaller_(Fl_Button*, void*);
57 Fl_Button *larger_;
58 inline void cb_larger__i(Fl_Button*, void*);
59 static void cb_larger_(Fl_Button*, void*);
60 Fl_Input *find_;
61 inline void cb_find__i(Fl_Input*, void*);
62 static void cb_find_(Fl_Input*, void*);
63 Fl_Help_View *view_;
64 inline void cb_view__i(Fl_Help_View*, void*);
65 static void cb_view_(Fl_Help_View*, void*);
66 public:
67 ~Fl_Help_Dialog();
68 int h();
69 void hide();
70 void load(const char *f);
71 void position(int xx, int yy);
72 void resize(int xx, int yy, int ww, int hh);
73 void show();
74 void show(int argc, char **argv);
75 void textsize(Fl_Fontsize s);
76 Fl_Fontsize textsize();
77 void topline(const char *n);
78 void topline(int n);
79 void value(const char *f);
80 const char * value() const;
81 int visible();
82 int w();
83 int x();
84 int y();
85 };
86 #endif
87
88 //
89 // End of "$Id$".
90 //

```

32.50 Fl_Help_View.H

```

1 //
2 // "$Id$"
3 //
4 // Help Viewer widget definitions.
5 //
6 // Copyright 1997-2010 by Easy Software Products.
7 // Image support by Matthias Melcher, Copyright 2000-2009.
8 //
9 // This library is free software. Distribution and use rights are outlined in
10 // the file "COPYING" which should have been included with this file. If this
11 // file is missing or damaged, see the license at:
12 //
13 // http://www.fltk.org/COPYING.php
14 //
15 // Please report all bugs and problems on the following page:
16 //
17 // http://www.fltk.org/str.php
18 //
19
20 /* \file
21 Fl_Help_View widget . */
22
23 #ifndef Fl_Help_View_H
24 # define Fl_Help_View_H
25
26 //
27 // Include necessary header files...
28 //
29
30 # include <stdio.h>
31 # include "Fl.H"
32 # include "Fl_Group.H"
33 # include "Fl_Scrollbar.H"
34 # include "fl_draw.H"
35 # include "Fl_Shared_Image.H"
36 # include "filename.H"
37
38
39 //

```

```

40 // Fl_Help_Func type - link callback function for files...
41 //
42
43
44 typedef const char *(Fl_Help_Func)(Fl_Widget *, const char *);
45
46
47 //
48 // Fl_Help_Block structure...
49 //
50
51 struct Fl_Help_Block {
52     const char    *start,           // Start of text
53     const char    *end;             // End of text
54     uchar         border;           // Draw border?
55     Fl_Color      bgcolor;          // Background color
56     int           x,                 // Indentation/starting X coordinate
57                 y,                 // Starting Y coordinate
58                 w,                 // Width
59                 h;                 // Height
60     int           line[32];         // Left starting position for each line
61 };
62
63 //
64 // Fl_Help_Link structure...
65 //
66
67 struct Fl_Help_Link {
68     char          filename[192],
69     char          name[32];
70     int           x,
71                 y,
72                 w,
73                 h;
74 };
75
76 /*
77 * Fl_Help_View font stack opaque implementation
78 */
79
80
81 struct FL_EXPORT Fl_Help_Font_Style {
82     Fl_Font       f;
83     Fl_Fontsize  s;
84     Fl_Color      c;
85     void get(Fl_Font &afont, Fl_Fontsize &asize, Fl_Color &acolor) {afont=f; asize=s; acolor=c;}
86     void set(Fl_Font afont, Fl_Fontsize asize, Fl_Color acolor) {f=afont; s=asize; c=acolor;}
87     Fl_Help_Font_Style(Fl_Font afont, Fl_Fontsize asize, Fl_Color acolor) {set(afont, asize, acolor);}
88     Fl_Help_Font_Style(){} // For in table use
89 };
90
91
92 const size_t MAX_FL_HELP_FS_ELTS = 100;
93
94 struct FL_EXPORT Fl_Help_Font_Stack {
95     Fl_Help_Font_Style() {
96         nfonts_ = 0;
97     }
98 }
99
100 void init(Fl_Font f, Fl_Fontsize s, Fl_Color c) {
101     nfonts_ = 0;
102     elts_[nfonts_].set(f, s, c);
103     fl_font(f, s);
104     fl_color(c);
105 }
106
107 void top(Fl_Font &f, Fl_Fontsize &s, Fl_Color &c) { elts_[nfonts_].get(f, s, c); }
108
109 void push(Fl_Font f, Fl_Fontsize s, Fl_Color c) {
110     if (nfonts_ < MAX_FL_HELP_FS_ELTS-1) nfonts_ ++;
111     elts_[nfonts_].set(f, s, c);
112     fl_font(f, s); fl_color(c);
113 }
114
115 void pop(Fl_Font &f, Fl_Fontsize &s, Fl_Color &c) {
116     if (nfonts_ > 0) nfonts_ --;
117     top(f, s, c);
118     fl_font(f, s); fl_color(c);
119 }
120
121 size_t count()const {return nfonts_;} // Gets the current number of fonts in the stack
122
123 protected:
124     size_t nfonts_;
125     Fl_Help_Font_Style elts_[100];
126 };
127
128
129 struct Fl_Help_Target {
130     char          name[32];
131     int           y;
132 };
133 };
134
135
136 class FL_EXPORT Fl_Help_View : public Fl_Group { // Help viewer widget
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201

```

```

202  enum { RIGHT = -1, CENTER, LEFT };
203
204  char          title_[1024];
205  Fl_Color      defcolor_,
206              bgcolor_,
207              textcolor_,
208              linkcolor_;
209  Fl_Font       textfont_;
210  Fl_Fontsize   textsize_;
211  const char    *value_;
212  Fl_Help_Font_Stack fstack_;
213  int           nblocks_,
214              ablocks_;
215  Fl_Help_Block *blocks_;
216
217  Fl_Help_Func  *link_;
218
219  int           nlinks_,
220              alinks_;
221  Fl_Help_Link  *links_;
222
223  int           ntargets_,
224              atargets_;
225  Fl_Help_Target *targets_;
226
227  char          directory_[FL_PATH_MAX];
228  char          filename_[FL_PATH_MAX];
229  int           topline_,
230              leftline_,
231              size_,
232              hsize_,
233              scrollbar_size_;
234  Fl_Scrollbar  scrollbar_,
235              hscrollbar_;
236
237  static int    selection_first;
238  static int    selection_last;
239  static int    selection_push_first;
240  static int    selection_push_last;
241  static int    selection_drag_first;
242  static int    selection_drag_last;
243  static int    selected;
244  static int    draw_mode;
245  static int    mouse_x;
246  static int    mouse_y;
247  static int    current_pos;
248  static Fl_Help_View *current_view;
249  static Fl_Color hv_selection_color;
250  static Fl_Color hv_selection_text_color;
251
252
253  void initfont(Fl_Font &f, Fl_Fontsize &s, Fl_Color &c) { f = textfont_; s = textsize_; c = textcolor_;
  fstack_.init(f, s, c); }
254  void pushfont(Fl_Font f, Fl_Fontsize s) {fstack_.push(f, s, textcolor_);}
255  void pushfont(Fl_Font f, Fl_Fontsize s, Fl_Color c) {fstack_.push(f, s, c);}
256  void popfont(Fl_Font &f, Fl_Fontsize &s, Fl_Color &c) {fstack_.pop(f, s, c);}
257
258  Fl_Help_Block *add_block(const char *s, int xx, int yy, int ww, int hh, uchar border = 0);
259  void          add_link(const char *n, int xx, int yy, int ww, int hh);
260  void          add_target(const char *n, int yy);
261  static int    compare_targets(const Fl_Help_Target *t0, const Fl_Help_Target *t1);
262  int          do_align(Fl_Help_Block *block, int line, int xx, int a, int &l);
263  #if FLTK_ABI_VERSION >= 10303
264  protected:
265  #endif
266  void          draw();
267  #if FLTK_ABI_VERSION >= 10303
268  private:
269  #endif
270  void          format();
271  void          format_table(int *table_width, int *columns, const char *table);
272  void          free_data();
273  int          get_align(const char *p, int a);
274  const char   *get_attr(const char *p, const char *n, char *buf, int bufsize);
275  Fl_Color     get_color(const char *n, Fl_Color c);
276  Fl_Shared_Image *get_image(const char *name, int W, int H);
277  int          get_length(const char *l);
278  #if FLTK_ABI_VERSION >= 10303
279  public:
280  #endif
281  int          handle(int);
282  #if FLTK_ABI_VERSION >= 10303
283  private:
284  #endif
285
286  void          hv_draw(const char *t, int x, int y, int entity_extra_length = 0);
287  char          begin_selection();

```

```

288 char          extend_selection();
289 void          end_selection(int c=0);
290 void          clear_global_selection();
291 Fl_Help_Link  *find_link(int, int);
292 void          follow_link(Fl_Help_Link*);
293
294 public:
295
296 Fl_Help_View(int xx, int yy, int ww, int hh, const char *l = 0);
297 ~Fl_Help_View();
298 const char    *directory()const { if (directory_[0]) return (directory_);
299                                     else return ((const char *)0); }
300 const char    *filename()const { if (filename_[0]) return (filename_);
301                                     else return ((const char *)0); }
302 int          find(const char *s, int p = 0);
303 void          link(Fl_Help_Func *fn) { link_ = fn; }
304 int          load(const char *f);
305 void          resize(int,int,int,int);
306 int          size()const { return (size_); }
307 void          size(int W, int H) { Fl_Widget::size(W, H); }
308 void          textcolor(Fl_Color c) { if (textcolor_ == defcolor_) textcolor_ = c; defcolor_ = c; }
309 Fl_Color      textcolor()const { return (defcolor_); }
310 void          textfont(Fl_Font f) { textfont_ = f; format(); }
311 Fl_Font       textfont()const { return (textfont_); }
312 void          textsize(Fl_Fontsize s) { textsize_ = s; format(); }
313 Fl_Fontsize   textsize()const { return (textsize_); }
314 const char    *title() { return (title_); }
315 void          topline(const char *n);
316 void          topline(int);
317 int          topline()const { return (topline_); }
318 void          leftline(int);
319 int          leftline()const { return (leftline_); }
320 void          value(const char *val);
321 const char    *value()const { return (value_); }
322 void          clear_selection();
323 void          select_all();
324 int scrollbar_size()const {
325     return(scrollbar_size_);
326 }
327 void scrollbar_size(int newSize) {
328     scrollbar_size_ = newSize;
329 }
330 };
331 #endif // !Fl_Help_View_H
332
333 //
334 // End of "$Id$".
335 //

```

32.51 Fl_Hold_Browser.H

```

1 //
2 // "$Id$"
3 //
4 // Hold browser header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software.  Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file.  If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Hold_Browser widget . */
21
22 #ifndef Fl_Hold_Browser_H
23 #define Fl_Hold_Browser_H
24
25 #include "Fl_Browser.H"
26
27 class FL_EXPORT Fl_Hold_Browser : public Fl_Browser {
28 public:
29     Fl_Hold_Browser(int X,int Y,int W,int H,const char *L=0);
30 };
31 #endif
32

```



```
49 //
50 // End of "$Id$".
51 //
```

32.52 Fl_Hor_Fill_Slider.H

```
1 //
2 // "$Id$"
3 //
4 // Horizontal fill slider header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Hor_Fill_Slider widget . */
21
22 #ifndef Fl_Hor_Fill_Slider_H
23 #define Fl_Hor_Fill_Slider_H
24
25 #include "Fl_Slider.H"
26
27 class FL_EXPORT Fl_Hor_Fill_Slider : public Fl_Slider {
28 public:
29     Fl_Hor_Fill_Slider(int X,int Y,int W,int H,const char *L=0);
30 };
31
32 #endif
33
34 //
35 // End of "$Id$".
36 //
```

32.53 Fl_Hor_Nice_Slider.H

```
1 //
2 // "$Id$"
3 //
4 // Horizontal "nice" slider header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Hor_Nice_Slider widget . */
21
22 #ifndef Fl_Hor_Nice_Slider_H
23 #define Fl_Hor_Nice_Slider_H
24
25 #include "Fl_Slider.H"
26
27 class FL_EXPORT Fl_Hor_Nice_Slider : public Fl_Slider {
28 public:
29     Fl_Hor_Nice_Slider(int X,int Y,int W,int H,const char *L=0);
30 };
31
32 #endif
33
34 //
35 // End of "$Id$".
36 //
```

32.54 Fl_Hor_Slider.H

```

1 //
2 // "$Id$"
3 //
4 // Horizontal slider header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2011 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Hor_Slider widget . */
21
22 #ifndef Fl_Hor_Slider_H
23 #define Fl_Hor_Slider_H
24
25 #include "Fl_Slider.H"
26
27 class FL_EXPORT Fl_Hor_Slider : public Fl_Slider {
28 public:
29     Fl_Hor_Slider(int X,int Y,int W,int H,const char *l=0);
30 };
31
32 #endif
33
34 //
35 // End of "$Id$".
36 //

```

32.55 Fl_Hor_Value_Slider.H

```

1 //
2 // "$Id$"
3 //
4 // Horizontal value slider header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Hor_Value_Slider widget . */
21
22 #ifndef Fl_Hor_Value_Slider_H
23 #define Fl_Hor_Value_Slider_H
24
25 #include "Fl_Value_Slider.H"
26
27 class FL_EXPORT Fl_Hor_Value_Slider : public Fl_Value_Slider {
28 public:
29     Fl_Hor_Value_Slider(int X,int Y,int W,int H,const char *l=0);
30 };
31
32 #endif
33
34 //
35 // End of "$Id$".
36 //

```

32.56 Fl_Image.H File Reference

[Fl_Image](#), [Fl_RGB_Image](#) classes.

```
#include "Enumerations.H"
#include <stdlib.h>
```

Classes

- class [Fl_Image](#)
Base class for image caching and drawing.
- class [Fl_RGB_Image](#)
The [Fl_RGB_Image](#) class supports caching and drawing of full-color images with 1 to 4 channels of color information.

Enumerations

- enum [Fl_RGB_Scaling](#) { [FL_RGB_SCALING_NEAREST](#) = 0 , [FL_RGB_SCALING_BILINEAR](#) }
The scaling algorithm to use for RGB images.

32.56.1 Detailed Description

[Fl_Image](#), [Fl_RGB_Image](#) classes.

32.56.2 Enumeration Type Documentation

32.56.2.1 Fl_RGB_Scaling

enum [Fl_RGB_Scaling](#)

The scaling algorithm to use for RGB images.

Enumerator

FL_RGB_SCALING_NEAREST	default RGB image scaling algorithm
FL_RGB_SCALING_BILINEAR	more accurate, but slower RGB image scaling algorithm

32.57 Fl_Image.H

[Go to the documentation of this file.](#)

```
1 //
2 // "$Id$"
3 //
4 // Image header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2016 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 // http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 // http://www.fltk.org/str.php
17 //
18
22 #ifndef Fl_Image_H
23 # define Fl_Image_H
24
25 # include "Enumerations.H"
26 #include <stdlib.h>
```

```

27
28 class Fl_Widget;
29 class Fl_Pixmap;
30 struct Fl_Menu_Item;
31 struct Fl_Label;
32
33
34
35
36
37 enum Fl_RGB_Scaling {
38     FL_RGB_SCALING_NEAREST = 0,
39     FL_RGB_SCALING_BILINEAR
40 };
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55 class FL_EXPORT Fl_Image {
56
57 public:
58     static const int ERR_NO_IMAGE = -1;
59     static const int ERR_FILE_ACCESS = -2;
60     static const int ERR_FORMAT = -3;
61
62 private:
63     int w_, h_, d_, ld_, count_;
64     const char * const *data_;
65     static Fl_RGB_Scaling RGB_scaling_;
66
67     // Forbid use of copy constructor and assign operator
68     Fl_Image & operator=(const Fl_Image &);
69     Fl_Image(const Fl_Image &);
70
71 protected:
72
73     void w(int W) {w_ = W;}
74     void h(int H) {h_ = H;}
75     void d(int D) {d_ = D;}
76     void ld(int LD) {ld_ = LD;}
77     void data(const char * const *p, int c) {data_ = p; count_ = c;}
78     void draw_empty(int X, int Y);
79
80     static void labeltype(const Fl_Label *lo, int lx, int ly, int lw, int lh, Fl_Align la);
81     static void measure(const Fl_Label *lo, int &lw, int &lh);
82
83 public:
84
85     int w()const {return w_;}
86     int h()const {return h_;}
87     int d()const {return d_;}
88     int ld()const {return ld_;}
89     int count()const {return count_;}
90     const char * const *data()const {return data_;}
91     int fail();
92     Fl_Image(int W, int H, int D);
93     virtual ~Fl_Image();
94     virtual Fl_Image *copy(int W, int H);
95     Fl_Image *copy() { return copy(w(), h()); }
96     virtual void color_average(Fl_Color c, float i);
97     void inactive() { color_average(FL_GRAY, .33f); }
98     virtual void desaturate();
99     virtual void label(Fl_Widget*w);
100    virtual void label(Fl_Menu_Item*m);
101    virtual void draw(int X, int Y, int W, int H, int cx=0, int cy=0); // platform dependent
102    void draw(int X, int Y) {draw(X, Y, w(), h(), 0, 0);} // platform dependent
103    virtual void uncache();
104
105    // set RGB image scaling method
106    static void RGB_scaling(Fl_RGB_Scaling);
107
108    // get RGB image scaling method
109    static Fl_RGB_Scaling RGB_scaling();
110 };
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202 class FL_EXPORT Fl_RGB_Image : public Fl_Image {
203     friend class Fl_Quartz_Graphics_Driver;
204     friend class Fl_GDI_Graphics_Driver;
205     friend class Fl_GDI_Printer_Graphics_Driver;
206     friend class Fl_Xlib_Graphics_Driver;
207     static size_t max_size_;
208 public:
209
210     const uchar *array;
211     int alloc_array;
212
213 private:
214
215     #if defined(__APPLE__) || defined(WIN32)
216     void *id_; // for internal use
217     void *mask_; // for internal use (mask bitmap)
218
219
220
221

```

```

222 #else
223     unsigned id_; // for internal use
224     unsigned mask_; // for internal use (mask bitmap)
225 #endif // __APPLE__ || WIN32
226
227 public:
228
229     Fl_RGB_Image(const uchar *bits, int W, int H, int D=3, int LD=0);
230     Fl_RGB_Image(const Fl_Pixmap *pxm, Fl_Color bg=FL_GRAY);
231     virtual ~Fl_RGB_Image();
232     virtual Fl_Image *copy(int W, int H);
233     Fl_Image *copy() { return copy(w(), h()); }
234     virtual void color_average(Fl_Color c, float i);
235     virtual void desaturate();
236     virtual void draw(int X, int Y, int W, int H, int cx=0, int cy=0);
237     void draw(int X, int Y) {draw(X, Y, w(), h(), 0, 0);}
238     virtual void label(Fl_Widget*w);
239     virtual void label(Fl_Menu_Item*m);
240     virtual void uncache();
241     static void max_size(size_t size) { max_size_ = size;}
242     static size_t max_size() {return max_size_;}
243 };
244
245 #endif // !Fl_Image_H
246
247 //
248 // End of "$Id$".
249 //

```

32.58 Fl_Image_Surface.H

```

1 //
2 // "$Id$"
3 //
4 // Draw-to-image code for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2014 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 #ifndef Fl_Image_Surface_H
20 #define Fl_Image_Surface_H
21
22 #include <FL/Fl_Copy_Surface.H>
23 #include <FL/Fl_Image.H>
24 #include <FL/Fl_Shared_Image.H>
25
26
27 class FL_EXPORT Fl_Image_Surface : public Fl_Surface_Device {
28 private:
29     void prepare_(int w, int h, int highres);
30     Fl_Offscreen offscreen;
31     int width;
32     int height;
33     Fl_Paged_Device *helper;
34 #ifdef __APPLE__
35 #elif defined(WIN32)
36     HDC _sgc;
37     Window _sw;
38     Fl_Surface_Device *_ss;
39     int _savedc;
40 #else
41     Fl_Surface_Device *previous;
42     Window pre_window;
43     GC gc;
44 #endif
45 public:
46     static const char *class_id;
47     const char *class_name() {return class_id;};
48     #if FLTK_ABI_VERSION >= 10304 || defined(FL_DOXYGEN)
49     Fl_Image_Surface(int w, int h, int highres = 0);
50 #else
51     Fl_Image_Surface(int w, int h, int highres);
52     Fl_Image_Surface(int w, int h);
53 #endif
54     ~Fl_Image_Surface();

```

```

75 void set_current();
76 void draw(Fl_Widget*, int delta_x = 0, int delta_y = 0);
77 void draw_decorated_window(Fl_Window* win, int delta_x = 0, int delta_y = 0);
78 Fl_RGB_Image *image();
79 Fl_Shared_Image *highres_image();
80 };
81
82 #ifdef __APPLE__
83 /* Mac class to implement translate()/untranslate() for a flipped bitmap graphics context */
84 class FL_EXPORT Fl_Quartz_Flipped_Surface_ : public Fl_Quartz_Surface_ {
85 public:
86     static const char *class_id;
87     const char *class_name() {return class_id;};
88     Fl_Quartz_Flipped_Surface_(int w, int h);
89     void translate(int x, int y);
90     void untranslate();
91     virtual ~Fl_Quartz_Flipped_Surface_() {};
92 };
93 #endif
94
95 #endif // Fl_Image_Surface_H
96
97 //
98 // End of "$Id$".
99 //

```

32.59 Fl_Input.H

```

1 //
2 // "$Id$"
3 //
4 // Input header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Input widget . */
21
22 #ifndef Fl_Input_H
23 #define Fl_Input_H
24
25 #include "Fl_Input_.H"
26
27 class FL_EXPORT Fl_Input : public Fl_Input_ {
28     int handle_key();
29     int shift_position(int p);
30     int shift_up_down_position(int p);
31     void handle_mouse(int keepmark=0);
32
33     // Private keyboard functions
34     int kf_lines_up(int repeat_num);
35     int kf_lines_down(int repeat_num);
36     int kf_page_up();
37     int kf_page_down();
38     int kf_insert_toggle();
39     int kf_delete_word_right();
40     int kf_delete_word_left();
41     int kf_delete_sol();
42     int kf_delete_eol();
43     int kf_delete_char_right();
44     int kf_delete_char_left();
45     int kf_move_sol();
46     int kf_move_eol();
47     int kf_clear_eol();
48     int kf_move_char_left();
49     int kf_move_char_right();
50     int kf_move_word_left();
51     int kf_move_word_right();
52     int kf_move_up_and_sol();
53     int kf_move_down_and_eol();
54     int kf_top();
55     int kf_bottom();
56     int kf_select_all();
57     int kf_undo();

```

```

253 int kf_redo();
254 int kf_copy();
255 int kf_paste();
256 int kf_copy_cut();
257
258 protected:
259 void draw();
260 public:
261 int handle(int);
262 Fl_Input(int,int,int,int,const char * = 0);
263 };
264
265 #endif
266
267 //
268 // End of "$Id$".
269 //

```

32.60 Fl_Input_H

```

1 //
2 // "$Id$"
3 //
4 // Input base class header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2015 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 // http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 // http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Input_ widget . */
21
22 #ifndef Fl_Input__H
23 #define Fl_Input__H
24
25 #ifndef Fl_Widget_H
26 #include "Fl_Widget.H"
27 #endif
28
29 #define FL_NORMAL_INPUT 0
30 #define FL_FLOAT_INPUT 1
31 #define FL_INT_INPUT 2
32 #define FL_HIDDEN_INPUT 3
33 #define FL_MULTILINE_INPUT 4
34 #define FL_SECRET_INPUT 5
35 #define FL_INPUT_TYPE 7
36 #define FL_INPUT_READONLY 8
37 #define FL_NORMAL_OUTPUT (FL_NORMAL_INPUT | FL_INPUT_READONLY)
38 #define FL_MULTILINE_OUTPUT (FL_MULTILINE_INPUT | FL_INPUT_READONLY)
39 #define FL_INPUT_WRAP 16
40 #define FL_MULTILINE_INPUT_WRAP (FL_MULTILINE_INPUT | FL_INPUT_WRAP)
41 #define FL_MULTILINE_OUTPUT_WRAP (FL_MULTILINE_INPUT | FL_INPUT_READONLY | FL_INPUT_WRAP)
42
94 class FL_EXPORT Fl_Input_ : public Fl_Widget {
95
97 const char* value_;
98
100 char* buffer;
101
103 int size_;
104
106 int bufsize;
107
109 int position_;
110
113 int mark_;
114
118 int tab_nav_;
119
121 int xscroll_, yscroll_;
122
125 int mu_p;
126
128 int maximum_size_;
129
131 int shortcut_;

```

```

132
133
134     uchar erase_cursor_only;
135
136
137     Fl_Font textfont_;
138
139
140     Fl_Fontsize textsize_;
141
142
143     Fl_Color textcolor_;
144
145
146     Fl_Color cursor_color_;
147
148
149     static double up_down_pos;
150
151
152     static int was_up_down;
153
154     /* Convert a given text segment into the text that will be rendered on screen. */
155     const char* expand(const char*, char*) const;
156
157     /* Calculates the width in pixels of part of a text buffer. */
158     double expandpos(const char*, const char*, const char*, int*) const;
159
160     /* Mark a range of characters for update. */
161     void minimal_update(int, int);
162
163     /* Mark a range of characters for update. */
164     void minimal_update(int p);
165
166     /* Copy the value from a possibly static entry into the internal buffer. */
167     void put_in_buffer(int newsize);
168
169     /* Set the current font and font size. */
170     void setfont() const;
171
172 protected:
173
174     /* Find the start of a word. */
175     int word_start(int i) const;
176
177     /* Find the end of a word. */
178     int word_end(int i) const;
179
180     /* Find the start of a line. */
181     int line_start(int i) const;
182
183     /* Find the end of a line. */
184     int line_end(int i) const;
185
186     /* Draw the text in the passed bounding box. */
187     void drawtext(int, int, int, int);
188
189     /* Move the cursor to the column given by up_down_pos. */
190     int up_down_position(int, int keepmark=0);
191
192     /* Handle mouse clicks and mouse moves. */
193     void handle_mouse(int, int, int, int, int keepmark=0);
194
195     /* Handle all kinds of text field related events. */
196     int handletext(int e, int, int, int, int);
197
198     /* Check the when() field and do a callback if indicated. */
199     void maybe_do_callback();
200
201
202     int xscroll()const {return xscroll_;}
203
204
205     int yscroll()const {return yscroll_;}
206     void yscroll(int yOffset) { yscroll_ = yOffset; damage(FL_DAMAGE_EXPOSE);}
207
208     /* Return the number of lines displayed on a single page. */
209     int linesPerPage();
210
211 public:
212
213     /* Change the size of the widget. */
214     void resize(int, int, int, int);
215
216     /* Constructor */
217     Fl_Input_(int, int, int, int, const char* = 0);
218
219     /* Destructor */
220     ~Fl_Input_();
221
222     /* Changes the widget text. */
223     int value(const char*);
224
225     /* Changes the widget text. */
226     int value(const char*, int);
227

```



```

228  /* Changes the widget text. */
229  int static_value(const char*);
230
231  /* Changes the widget text. */
232  int static_value(const char*, int);
233
234  const char* value()const {return value_;}
235
236  /* Returns the character at index \p i. */
237  Fl_Char index(int i) const;
238
239  int size()const {return size_;}
240
241  void size(int W, int H) { Fl_Widget::size(W, H); }
242
243  int maximum_size()const {return maximum_size_;}
244
245  void maximum_size(int m) {maximum_size_ = m;}
246
247  int position()const {return position_;}
248
249  int mark()const {return mark_;}
250
251  /* Sets the index for the cursor and mark. */
252  int position(int p, int m);
253
254  int position(int p) {return position(p, p);}
255
256  int mark(int m) {return position(position(), m);}
257
258  /* Deletes text from \p b to \p e and inserts the new string \p text. */
259  int replace(int b, int e, const char *text, int ilen=0);
260
261  int cut() {return replace(position(), mark(), 0);}
262
263  int cut(int n) {return replace(position(), position()+n, 0);}
264
265  int cut(int a, int b) {return replace(a, b, 0);}
266
267  int insert(const char* t, int l=0){return replace(position_, mark_, t, l);}
268
269  /* Put the current selection into the clipboard. */
270  int copy(int clipboard);
271
272  /* Undo previous changes to the text buffer. */
273  int undo();
274
275  /* Copy the yank buffer to the clipboard. */
276  int copy_cuts();
277
278  int shortcut()const {return shortcut_;}
279
280  void shortcut(int s) {shortcut_ = s;}
281
282  Fl_Font textfont()const {return textfont_;}
283
284  void textfont(Fl_Font s) {textfont_ = s;}
285
286  Fl_Fontsize textsize()const {return textsize_;}
287
288  void textsize(Fl_Fontsize s) {textsize_ = s;}
289
290  Fl_Color textcolor()const {return textcolor_;}
291
292  void textcolor(Fl_Color n) {textcolor_ = n;}
293
294  Fl_Color cursor_color()const {return cursor_color_;}
295
296  void cursor_color(Fl_Color n) {cursor_color_ = n;}
297
298  int input_type()const {return type() & FL_INPUT_TYPE; }
299
300  void input_type(int t) { type((uchar)(t | readonly())); }
301
302  int readonly()const { return type() & FL_INPUT_READONLY; }
303
304  void readonly(int b) { if (b) type((uchar)(type() | FL_INPUT_READONLY));
305                        else type((uchar)(type() & ~FL_INPUT_READONLY)); }
306
307  int wrap()const { return type() & FL_INPUT_WRAP; }
308
309  void wrap(int b) { if (b) type((uchar)(type() | FL_INPUT_WRAP));
310                  else type((uchar)(type() & ~FL_INPUT_WRAP)); }
311
312  void tab_nav(int val) {
313      tab_nav_ = val;
314  }

```

```

479
490 int tab_nav()const {
491     return tab_nav_;
492 }
493 };
494
495 #endif
496
497 //
498 // End of "$Id$".
499 //

```

32.61 Fl_Input_Choice.H

```

1 //
2 // "$Id$"
3 //
4 // An input/chooser widget.
5 //
6 //      ┌──────────┬──┴──┐
7 //      │ input area │  √  │
8 //      └──────────┴──┬──┘
9 //
10 // Copyright 1998-2010 by Bill Spitzak and others.
11 // Copyright 2004 by Greg Ercolano.
12 //
13 // This library is free software. Distribution and use rights are outlined in
14 // the file "COPYING" which should have been included with this file. If this
15 // file is missing or damaged, see the license at:
16 //
17 //     http://www.fltk.org/COPYING.php
18 //
19 // Please report all bugs and problems on the following page:
20 //
21 //     http://www.fltk.org/str.php
22 //
23
24 /* \file
25 Fl_Input_Choice widget . */
26
27 #ifndef Fl_Input_Choice_H
28 #define Fl_Input_Choice_H
29
30 #include <FL/Fl.H>
31 #include <FL/Fl_Group.H>
32 #include <FL/Fl_Input.H>
33 #include <FL/Fl_Menu_Button.H>
34 #include <FL/fl_draw.H>
35 #include <string.h>
36
95 class FL_EXPORT Fl_Input_Choice : public Fl_Group {
96     // Private class to handle slightly 'special' behavior of menu button
97     class InputMenuButton : public Fl_Menu_Button {
98     void draw() {
99         draw_box(FL_UP_BOX, color());
100         fl_color(active_r() ? labelcolor() : fl_inactive(labelcolor()));
101         int xc = x()+w()/2, yc=y()+h()/2;
102         fl_polygon(xc-5,yc-3,xc+5,yc-3,xc,yc+3);
103         if (Fl::focus() == this) draw_focus();
104     }
105     public:
106     InputMenuButton(int X,int Y,int W,int H,const char*L=0) :
107         Fl_Menu_Button(X, Y, W, H, L) { box(FL_UP_BOX); }
108     };
109
110     Fl_Input *inp_;
111     InputMenuButton *menu_;
112
113     // note: this is used by the Fl_Input_Choice ctor defined in Fl_Group.
114     static void menu_cb(Fl_Widget*, void *data) {
115         Fl_Input_Choice *o=(Fl_Input_Choice *)data;
116         Fl_Widget_Tracker wp(o);
117         const Fl_Menu_Item *item = o->menubutton()->mvalue();
118         if (item && item->flags & (FL_SUBMENU|FL_SUBMENU_POINTER)) return; // ignore submenus
119         if (!strcmp(o->inp_->value(), o->menu_->text()))
120         {
121             o->Fl_Widget::clear_changed();
122             if (o->when() & FL_WHEN_NOT_CHANGED)
123                 o->do_callback();
124         }
125         else
126         {
127             o->inp_->value(o->menu_->text());
128             o->inp_->set_changed();
129             o->Fl_Widget::set_changed();

```

```

130     if (o->when() & (FL_WHEN_CHANGED|FL_WHEN_RELEASE))
131         o->do_callback();
132     }
133
134     if (wp.deleted()) return;
135
136     if (o->callback() != default_callback)
137     {
138         o->Fl_Widget::clear_changed();
139         o->inp_->clear_changed();
140     }
141 }
142
143 // note: this is used by the Fl_Input_Choice ctor defined in Fl_Group.
144 static void inp_cb(Fl_Widget*, void *data) {
145     Fl_Input_Choice *o=(Fl_Input_Choice *)data;
146     Fl_Widget_Tracker wp(o);
147     if (o->inp_->changed()) {
148         o->Fl_Widget::set_changed();
149         if (o->when() & (FL_WHEN_CHANGED|FL_WHEN_RELEASE))
150             o->do_callback();
151     } else {
152         o->Fl_Widget::clear_changed();
153         if (o->when() & FL_WHEN_NOT_CHANGED)
154             o->do_callback();
155     }
156
157     if (wp.deleted()) return;
158
159     if (o->callback() != default_callback)
160         o->Fl_Widget::clear_changed();
161 }
162
163 // Custom resize behavior -- input stretches, menu button doesn't
164 inline int inp_x() { return(x() + Fl::box_dx(box())); }
165 inline int inp_y() { return(y() + Fl::box_dy(box())); }
166 inline int inp_w() { return(w() - Fl::box_dw(box()) - 20); }
167 inline int inp_h() { return(h() - Fl::box_dh(box())); }
168
169 inline int menu_x() { return(x() + w() - 20 - Fl::box_dx(box())); }
170 inline int menu_y() { return(y() + Fl::box_dy(box())); }
171 inline int menu_w() { return(20); }
172 inline int menu_h() { return(h() - Fl::box_dh(box())); }
173
174 public:
175 Fl_Input_Choice(int X,int Y,int W,int H,const char*L=0);
176
177 void add(const char *s) { menu_->add(s); }
178 int changed()const { return inp_->changed() | Fl_Widget::changed(); }
179 void clear_changed() {
180     inp_->clear_changed();
181     Fl_Widget::clear_changed();
182 }
183 void set_changed() {
184     inp_->set_changed();
185     // no need to call Fl_Widget::set_changed()
186 }
187 void clear() { menu_->clear(); }
188 Fl_Boxtype down_box()const { return (menu_->down_box()); }
189 void down_box(Fl_Boxtype b) { menu_->down_box(b); }
190 const Fl_Menu_Item *menu() { return (menu_->menu()); }
191 void menu(const Fl_Menu_Item *m) { menu_->menu(m); }
192 void resize(int X, int Y, int W, int H) {
193     Fl_Group::resize(X,Y,W,H);
194     inp_->resize(inp_x(), inp_y(), inp_w(), inp_h());
195     menu_->resize(menu_x(), menu_y(), menu_w(), menu_h());
196 }
197 Fl_Color textcolor()const { return (inp_->textcolor()); }
198 void textcolor(Fl_Color c) { inp_->textcolor(c); }
199 Fl_Font textfont()const { return (inp_->textfont()); }
200 void textfont(Fl_Font f) { inp_->textfont(f); }
201 Fl_Fontsize textsize()const { return (inp_->textsize()); }
202 void textsize(Fl_Fontsize s) { inp_->textsize(s); }
203 const char* value()const { return (inp_->value()); }
204 void value(const char *val) { inp_->value(val); }
205 void value(int val) {
206     menu_->value(val);
207     inp_->value(menu_->text(val));
208 }
209 Fl_Menu_Button *menubutton() { return menu_; }
210 Fl_Input *input() { return inp_; }
211 };
212
213 #endif // !Fl_Input_Choice_H
214
215 //
216 // End of "$Id$".

```

268 //

32.62 Fl_Int_Input.H

```
1 //
2 // "$Id$"
3 //
4 // Integer input header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Int_Input widget . */
21
22 #ifndef Fl_Int_Input_H
23 #define Fl_Int_Input_H
24
25 #include "Fl_Input.H"
26
27 class FL_EXPORT Fl_Int_Input : public Fl_Input {
28 public:
29     Fl_Int_Input(int X,int Y,int W,int H,const char *l = 0);
30 };
31
32 #endif
33
34 //
35 // End of "$Id$".
36 //
```

32.63 Fl_JPEG_Image.H

```
1 //
2 // "$Id$"
3 //
4 // JPEG image header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_JPEG_Image class . */
21
22 #ifndef Fl_JPEG_Image_H
23 #define Fl_JPEG_Image_H
24 #include "Fl_Image.H"
25
26 class FL_EXPORT Fl_JPEG_Image : public Fl_RGB_Image {
27 public:
28     Fl_JPEG_Image(const char *filename);
29     Fl_JPEG_Image(const char *name, const unsigned char *data);
30 };
31
32 #endif
33
34 //
35 // End of "$Id$".
36 //
```

32.64 Fl_Light_Button.H

```
1 //
2 // "$Id$"
3 //
4 // Lighted button header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software.  Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file.  If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Light_Button widget . */
21
22 #ifndef Fl_Light_Button_H
23 #define Fl_Light_Button_H
24
25 #include "Fl_Button.H"
26
27 class FL_EXPORT Fl_Light_Button : public Fl_Button {
28 protected:
29     virtual void draw();
30 public:
31     virtual int handle(int);
32     Fl_Light_Button(int x,int y,int w,int h,const char *l = 0);
33 };
34
35 #endif
36
37 // End of "$Id$".
38 //
```

32.65 Fl_Line_Dial.H

```
1 //
2 // "$Id$"
3 //
4 // Line dial header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software.  Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file.  If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Line_Dial widget . */
21
22 #ifndef Fl_Line_Dial_H
23 #define Fl_Line_Dial_H
24
25 #include "Fl_Dial.H"
26
27 class FL_EXPORT Fl_Line_Dial : public Fl_Dial {
28 public:
29     Fl_Line_Dial(int X,int Y,int W,int H, const char *L = 0);
30 };
31
32 #endif
33
34 // End of "$Id$".
35 //
```

32.66 Fl_Menu.H

```

1 //
2 // "$Id$"
3 //
4 // Old menu header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software.  Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file.  If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 // this include file is for back compatibility only
20 #include "Fl_Menu_Item.H"
21
22 //
23 // End of "$Id$".
24 //

```

32.67 Fl_Menu_.H

```

1 //
2 // "$Id$"
3 //
4 // Menu base class header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2016 by Bill Spitzak and others.
7 //
8 // This library is free software.  Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file.  If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Menu_ widget . */
21
22 #ifndef Fl_Menu__H
23 #define Fl_Menu__H
24
25 #ifndef Fl_Widget_H
26 #include "Fl_Widget.H"
27 #endif
28 #include "Fl_Menu_Item.H"
29
30 class FL_EXPORT Fl_Menu_ : public Fl_Widget {
31
32     Fl_Menu_Item *menu_;
33     const Fl_Menu_Item *value_;
34
35 protected:
36
37     uchar alloc; // flag indicates if menu_ is a dynamic copy (=1) or not (=0)
38     uchar down_box_;
39     Fl_Font textfont_;
40     Fl_Fontsize textsize_;
41     Fl_Color textcolor_;
42
43     int item_pathname(char *name, int namelen, const Fl_Menu_Item *finditem,
44                     const Fl_Menu_Item *menu=0) const;
45
46 public:
47     Fl_Menu_(int,int,int,int,const char * =0);
48     ~Fl_Menu_();
49
50     int item_pathname(char *name, int namelen, const Fl_Menu_Item *finditem=0) const;
51     const Fl_Menu_Item* picked(const Fl_Menu_Item*);
52     const Fl_Menu_Item* find_item(const char *name);
53     const Fl_Menu_Item* find_item(Fl_Callback*);
54     int find_index(const char *name) const;
55     int find_index(const Fl_Menu_Item *item) const;
56     int find_index(Fl_Callback *cb) const;
57
58 };

```

```

77
89  const Fl_Menu_Item* test_shortcut() {return picked(menu()->test_shortcut());}
90  void global();
91
110  const Fl_Menu_Item *menu()const {return menu_;}
111  void menu(const Fl_Menu_Item *m);
112  void copy(const Fl_Menu_Item *m, void* user_data = 0);
113  int insert(int index, const char*, int shortcut, Fl_Callback*, void* = 0, int = 0);
114  int add(const char*, int shortcut, Fl_Callback*, void* = 0, int = 0); // see src/Fl_Menu_add.cxx
116  int add(const char* a, const char* b, Fl_Callback* c, void* d = 0, int e = 0) {
117      return add(a, fl_old_shortcut(b),c,d,e);
118  }
120  int insert(int index, const char* a, const char* b, Fl_Callback* c, void* d = 0, int e = 0) {
121      return insert(index,a,fl_old_shortcut(b),c,d,e);
122  }
123  int add(const char *);
124  int size() const ;
125  void size(int W, int H) { Fl_Widget::size(W, H); }
126  void clear();
127  int clear_submenu(int index);
128  void replace(int,const char *);
129  void remove(int);
131  void shortcut(int i, int s) {menu_[i].shortcut(s);}
133  void mode(int i,int fl) {menu_[i].flags = fl;}
135  int mode(int i)const {return menu_[i].flags;}
136
138  const Fl_Menu_Item *mvalue()const {return value_;}
140  int value()const {return value_ ? (int)(value_-menu_) : -1;}
141  int value(const Fl_Menu_Item*);
148  int value(int i) {return value(menu_+i);}
150  const char *text()const {return value_ ? value_->text : 0;}
152  const char *text(int i)const {return menu_[i].text;}
153
155  Fl_Font textfont()const {return textfont_;}
157  void textfont(Fl_Font c) {textfont_=c;}
159  Fl_Fontsize textsize()const {return textsize_;}
161  void textsize(Fl_Fontsize c) {textsize_=c;}
163  Fl_Color textcolor()const {return textcolor_;}
165  void textcolor(Fl_Color c) {textcolor_=c;}
166
173  Fl_Boxtype down_box()const {return (Fl_Boxtype)down_box_;}
175  void down_box(Fl_Boxtype b) {down_box_ = b;}
176
178  Fl_Color down_color()const {return selection_color();}
180  void down_color(unsigned c) {selection_color(c);}
181  void setonly(Fl_Menu_Item* item);
182 };
183
184 #endif
185
186 //
187 // End of "$Id$".
188 //

```

32.68 Fl_Menu_Bar.H

```

1 //
2 // "$Id$"
3 //
4 // Menu bar header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2016 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Menu_Bar widget . */
21
22 #ifndef Fl_Menu_Bar_H
23 #define Fl_Menu_Bar_H
24
25 #include "Fl_Menu_.H"
26
27 class FL_EXPORT Fl_Menu_Bar : public Fl_Menu_ {
28 protected:
29     void draw();

```

```

70 public:
71     int handle(int);
90     Fl_Menu_Bar(int X, int Y, int W, int H, const char *l=0);
91 };
92
93 #endif
94
95 //
96 // End of "$Id$".
97 //

```

32.69 Fl_Menu_Button.H

```

1 //
2 // "$Id$"
3 //
4 // Menu button header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Menu_Button widget . */
21
22 #ifndef Fl_Menu_Button_H
23 #define Fl_Menu_Button_H
24
25 #include "Fl_Menu_.H"
26
27 class FL_EXPORT Fl_Menu_Button : public Fl_Menu_ {
28 protected:
29     void draw();
30 public:
31     enum popup_buttons {POPUP1 = 1,
32         POPUP2,
33         POPUP12,
34         POPUP3,
35         POPUP13,
36         POPUP23,
37         POPUP123
38     };
39     int handle(int);
40     const Fl_Menu_Item* popup();
41     Fl_Menu_Button(int,int,int,int,const char * =0);
42 };
43 #endif
44
45 //
46 // End of "$Id$".
47 //

```

32.70 Fl_Menu_Item.H File Reference

```

#include "Fl_Widget.H"
#include "Fl_Image.H"

```

Classes

- struct [Fl_Menu_Item](#)
The *Fl_Menu_Item* structure defines a single menu item that is used by the *Fl_Menu_* class.

Typedefs

- typedef [Fl_Menu_Item](#) **Fl_Menu**

Enumerations

- enum {
[FL_MENU_INACTIVE](#) = 1 , [FL_MENU_TOGGLE](#) = 2 , [FL_MENU_VALUE](#) = 4 , [FL_MENU_RADIO](#) = 8 ,
[FL_MENU_INVISIBLE](#) = 0x10 , [FL_SUBMENU_POINTER](#) = 0x20 , [FL_SUBMENU](#) = 0x40 , [FL_MENU_DIVIDER](#)
= 0x80 ,
[FL_MENU_HORIZONTAL](#) = 0x100 }
- enum {
FL_PUP_NONE = 0 , **FL_PUP_GREY** = [FL_MENU_INACTIVE](#) , **FL_PUP_GRAY** = [FL_MENU_INACTIVE](#) ,
FL_MENU_BOX = [FL_MENU_TOGGLE](#) ,
FL_PUP_BOX = [FL_MENU_TOGGLE](#) , **FL_MENU_CHECK** = [FL_MENU_VALUE](#) , **FL_PUP_CHECK** = [FL_MENU_VALUE](#) ,
FL_PUP_RADIO = [FL_MENU_RADIO](#) ,
FL_PUP_INVISIBLE = [FL_MENU_INVISIBLE](#) , **FL_PUP_SUBMENU** = [FL_SUBMENU_POINTER](#) }

Functions

- FL_EXPORT [Fl_Shortcut fl_old_shortcut](#) (const char *)
Emulation of XForms named shortcuts.

32.70.1 Enumeration Type Documentation

32.70.1.1 anonymous enum

anonymous enum

Enumerator

FL_MENU_INACTIVE	Deactivate menu item (gray out)
FL_MENU_TOGGLE	Item is a checkbox toggle (shows checkbox for on/off state)
FL_MENU_VALUE	The on/off state for checkbox/radio buttons (if set, state is 'on')
FL_MENU_RADIO	Item is a radio button (one checkbox of many can be on)
FL_MENU_INVISIBLE	Item will not show up (shortcut will work)
FL_SUBMENU_POINTER	Indicates user_data() is a pointer to another menu array.
FL_SUBMENU	This item is a submenu to other items.
FL_MENU_DIVIDER	Creates divider line below this item. Also ends a group of radio buttons.
FL_MENU_HORIZONTAL	??? – reserved

32.71 Fl_Menu_Item.H

[Go to the documentation of this file.](#)

```

1 //
2 // "$Id$"
3 //
4 // Menu item header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 #ifndef Fl_Menu_Item_H
20 #define Fl_Menu_Item_H

```

```

21
22 # include "Fl_Widget.H"
23 # include "Fl_Image.H"
24
25 # if defined(__APPLE__) && defined(check)
26 #   undef check
27 # endif
28
29 // doxygen needs the following line to enable e.g. ::FL_MENU_TOGGLE to link to the enums
30
31
32 enum { // values for flags:
33     FL_MENU_INACTIVE = 1,
34     FL_MENU_TOGGLE = 2,
35     FL_MENU_VALUE = 4,
36     FL_MENU_RADIO = 8,
37     FL_MENU_INVISIBLE = 0x10,
38     FL_SUBMENU_POINTER = 0x20,
39     FL_SUBMENU = 0x40,
40     FL_MENU_DIVIDER = 0x80,
41     FL_MENU_HORIZONTAL = 0x100
42 };
43
44 extern FL_EXPORT Fl_Shortcut fl_old_shortcut(const char*);
45
46 class Fl_Menu_;
47
112 struct FL_EXPORT Fl_Menu_Item {
113     const char *text;
114     int shortcut_;
115     Fl_Callback *callback_;
116     void *user_data_;
117     int flags;
118     uchar labeltype_;
119     Fl_Font labelfont_;
120     Fl_Fontsize labelsize_;
121     Fl_Color labelcolor_;
122
123     // advance N items, skipping submenus:
124     const Fl_Menu_Item *next(int=1) const;
125
126     Fl_Menu_Item *next(int i=1) {
127         return (Fl_Menu_Item*)((const Fl_Menu_Item*)this)->next(i);}
128
129     const Fl_Menu_Item *first()const { return next(0); }
130
131     Fl_Menu_Item *first() { return next(0); }
132
133     // methods on menu items:
134     const char* label()const {return text;}
135
136     void label(const char* a) {text=a;}
137
138     void label(Fl_Labeltype a,const char* b) {labeltype_ = a; text = b;}
139
140     Fl_Labeltype labeltype()const {return (Fl_Labeltype)labeltype_;}
141
142     void labeltype(Fl_Labeltype a) {labeltype_ = a;}
143
144     Fl_Color labelcolor()const {return labelcolor_;}
145
146     void labelcolor(Fl_Color a) {labelcolor_ = a;}
147     Fl_Font labelfont()const {return labelfont_;}
148
149     void labelfont(Fl_Font a) {labelfont_ = a;}
150
151     Fl_Fontsize labelsize()const {return labelsize_;}
152
153     void labelsize(Fl_Fontsize a) {labelsize_ = a;}
154
155     Fl_Callback_p callback()const {return callback_;}
156
157     void callback(Fl_Callback* c, void* p) {callback_=c; user_data_=p;}
158
159     void callback(Fl_Callback* c) {callback_=c;}
160
161     void callback(Fl_Callback0*c) {callback_=(Fl_Callback*)c;}
162
163     void callback(Fl_Callback1*c, long p=0) {callback_=(Fl_Callback*)c; user_data_=(void*)(fl_intptr_t)p;}
164
165     void* user_data()const {return user_data_;}
166     void user_data(void* v) {user_data_ = v;}
167     long argument()const {return (long)(fl_intptr_t)user_data_;}
168     void argument(long v) {user_data_ = (void*)(fl_intptr_t)v;}
169
170     int shortcut()const {return shortcut_;}
171
172     void shortcut(int s) {shortcut_ = s;}

```

```

298 int submenu()const {return flags&(FL_SUBMENU|FL_SUBMENU_POINTER);}
303 int checkbox()const {return flags&FL_MENU_TOGGLE;}
310 int radio()const {return flags&FL_MENU_RADIO;}
318 int value()const {return flags&FL_MENU_VALUE;}
323 void set() {flags |= FL_MENU_VALUE;}
324
326 void clear() {flags &= ~FL_MENU_VALUE;}
327
328 void setonly();
329
331 int visible()const {return !(flags&FL_MENU_INVISIBLE);}
332
334 void show() {flags &= ~FL_MENU_INVISIBLE;}
335
337 void hide() {flags |= FL_MENU_INVISIBLE;}
338
340 int active()const {return !(flags&FL_MENU_INACTIVE);}
341
343 void activate() {flags &= ~FL_MENU_INACTIVE;}
348 void deactivate() {flags |= FL_MENU_INACTIVE;}
350 int activevisible()const {return !(flags & (FL_MENU_INACTIVE|FL_MENU_INVISIBLE));}
351
352 // compatibility for FLUID so it can set the image of a menu item...
353
355 void image(Fl_Image* a) {a->label(this);}
356
358 void image(Fl_Image& a) {a.label(this);}
359
360 // used by menubar:
361 int measure(int* h, const Fl_Menu_*) const;
362 void draw(int x, int y, int w, int h, const Fl_Menu_*, int t=0) const;
363
364 // popup menus without using an Fl_Menu_ widget:
365 const Fl_Menu_Item* popup(
366     int X, int Y,
367     const char *title = 0,
368     const Fl_Menu_Item* picked=0,
369     const Fl_Menu_* = 0) const;
370 const Fl_Menu_Item* pulldown(
371     int X, int Y, int W, int H,
372     const Fl_Menu_Item* picked = 0,
373     const Fl_Menu_* = 0,
374     const Fl_Menu_Item* title = 0,
375     int menubar=0) const;
376 const Fl_Menu_Item* test_shortcut() const;
377 const Fl_Menu_Item* find_shortcut(int *ip=0, const bool require_alt = false) const;
378
384 void do_callback(Fl_Widget* o)const {callback_(o, user_data_);}
385
391 void do_callback(Fl_Widget* o,void* arg)const {callback_(o, arg);}
392
400 void do_callback(Fl_Widget* o,long arg)const {callback_(o, (void*)(fl_intptr_t)arg);}
401
402 // back-compatibility, do not use:
403
405 int checked()const {return flags&FL_MENU_VALUE;}
406
408 void check() {flags |= FL_MENU_VALUE;}
409
411 void uncheck() {flags &= ~FL_MENU_VALUE;}
412
413 int insert(int,const char*,int,Fl_Callback*,void* =0, int =0);
414 int add(const char*, int shortcut, Fl_Callback*, void* =0, int = 0);
415
417 int add(const char*a, const char* b, Fl_Callback* c,
418         void* d = 0, int e = 0) {
419     return add(a,fl_old_shortcut(b),c,d,e);}
420
421 int size() const ;
422 };
423
424 typedef Fl_Menu_Item Fl_Menu; // back compatibility
425
426 enum { // back-compatibility enum:
427     FL_PUP_NONE = 0,
428     FL_PUP_GREY = FL_MENU_INACTIVE,
429     FL_PUP_GRAY = FL_MENU_INACTIVE,
430     FL_MENU_BOX = FL_MENU_TOGGLE,
431     FL_PUP_BOX = FL_MENU_TOGGLE,
432     FL_MENU_CHECK = FL_MENU_VALUE,
433     FL_PUP_CHECK = FL_MENU_VALUE,
434     FL_PUP_RADIO = FL_MENU_RADIO,
435     FL_PUP_INVISIBLE = FL_MENU_INVISIBLE,
436     FL_PUP_SUBMENU = FL_SUBMENU_POINTER
437 };
438
439 #endif

```

```
440
441 //
442 // End of "$Id$".
443 //
```

32.72 Fl_Menu_Window.H

```
1 //
2 // "$Id$"
3 //
4 // Menu window header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software.  Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file.  If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Menu_Window widget . */
21
22 #ifndef Fl_Menu_Window_H
23 #define Fl_Menu_Window_H
24
25 #include "Fl_Single_Window.H"
26
27 class FL_EXPORT Fl_Menu_Window : public Fl_Single_Window {
28 public:
29     void show();
30     void erase();
31     void flush();
32     void hide();
33     unsigned int overlay() {return !(flags() & NO_OVERLAY);}
34     void set_overlay() {clear_flag(NO_OVERLAY);}
35     void clear_overlay() {set_flag(NO_OVERLAY);}
36     ~Fl_Menu_Window();
37     Fl_Menu_Window(int W, int H, const char *l = 0);
38     Fl_Menu_Window(int X, int Y, int W, int H, const char *l = 0);
39 };
40
41 #endif
42
43 //
44 // End of "$Id$".
45 //
```

32.73 fl_message.H

```
1 //
2 // "$Id$"
3 //
4 // Standard message header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software.  Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file.  If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 #include "fl_ask.H"
20
21 //
22 // End of "$Id$".
23 //
```

32.74 Fl_Multi_Browser.H

```

1 //
2 // "$Id$"
3 //
4 // Multi browser header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software.  Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file.  If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Multi_Browser widget . */
21
22 #ifndef Fl_Multi_Browser_H
23 #define Fl_Multi_Browser_H
24
25 #include "Fl_Browser.H"
26
27
28
29 class FL_EXPORT Fl_Multi_Browser : public Fl_Browser {
30 public:
31     Fl_Multi_Browser(int X,int Y,int W,int H,const char *L=0);
32 };
33
34 #endif
35
36 //
37 // End of "$Id$".
38 //

```

32.75 Fl_Multi_Label.H

```

1 //
2 // "$Id$"
3 //
4 // Multi-label header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2015 by Bill Spitzak and others.
7 //
8 // This library is free software.  Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file.  If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 #ifndef Fl_Multi_Label_H
20 #define Fl_Multi_Label_H
21
22 class Fl_Widget;
23 struct Fl_Menu_Item;
24
25
26
27 struct FL_EXPORT Fl_Multi_Label {
28     const char* labela;
29     const char* labelb;
30     uchar typea;
31     uchar typeb;
32
33     void label(Fl_Widget*);
34     void label(Fl_Menu_Item*);
35 };
36
37 #endif
38
39 //
40 // End of "$Id$".
41 //

```

32.76 Fl_Multiline_Input.H

```
1 //
2 // "$Id$"
3 //
4 // Multiline input header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2011 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 // http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 // http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Multiline_Input widget . */
21
22 #ifndef Fl_Multiline_Input_H
23 #define Fl_Multiline_Input_H
24
25 #include "Fl_Input.H"
26
27 class FL_EXPORT Fl_Multiline_Input : public Fl_Input {
28 public:
29     Fl_Multiline_Input(int X,int Y,int W,int H,const char *l = 0);
30 };
31
32 #endif
33 //
34 // End of "$Id$".
35 //
```

32.77 Fl_Multiline_Output.H

```
1 //
2 // "$Id$"
3 //
4 // Multi line output header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2011 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 // http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 // http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Multiline_Output widget . */
21
22 #ifndef Fl_Multiline_Output_H
23 #define Fl_Multiline_Output_H
24
25 #include "Fl_Output.H"
26
27 class FL_EXPORT Fl_Multiline_Output : public Fl_Output {
28 public:
29     Fl_Multiline_Output(int X,int Y,int W,int H,const char *l = 0);
30 };
31
32 #endif
33 //
34 // End of "$Id$".
35 //
```

32.78 Fl_Native_File_Chooser.H File Reference

[Fl_Native_File_Chooser](#) widget.

```
#include <FL/Fl_File_Chooser.H>
```

Classes

- class [Fl_FLTK_File_Chooser](#)
- class [Fl_GTK_File_Chooser](#)
- class [Fl_Native_File_Chooser](#)

This class lets an FLTK application easily and consistently access the operating system's native file chooser.

32.78.1 Detailed Description

[Fl_Native_File_Chooser](#) widget.

32.79 Fl_Native_File_Chooser.H

[Go to the documentation of this file.](#)

```
1 //
2 // "$Id$"
3 //
4 // FLTK native OS file chooser widget
5 //
6 // Copyright 1998-2014 by Bill Spitzak and others.
7 // Copyright 2004 Greg Ercolano.
8 //
9 // This library is free software. Distribution and use rights are outlined in
10 // the file "COPYING" which should have been included with this file. If this
11 // file is missing or damaged, see the license at:
12 //
13 // http://www.fltk.org/COPYING.php
14 //
15 // Please report all bugs and problems on the following page:
16 //
17 // http://www.fltk.org/str.php
18 //
19
23 #ifndef FL_NATIVE_FILE_CHOOSER_H
24 #define FL_NATIVE_FILE_CHOOSER_H
25
26 // Use Windows' chooser
27 #ifdef WIN32
28 // #define _WIN32_WINNT 0x0501 // needed for OPENFILENAME's 'FlagsEx'
29 #if defined(FL_LIBRARY) || FLTK_ABI_VERSION < 10304
30 # include <windows.h>
31 # include <commdlg.h> // OPENFILENAMEW, GetOpenFileName()
32 # include <shlobj.h> // BROWSEINFOW, SHBrowseForFolder()
33 typedef OPENFILENAMEW fl_OPENFILENAMEW;
34 typedef BROWSEINFOW fl_BROWSEINFOW;
35 #else
36 typedef void fl_OPENFILENAMEW;
37 typedef void fl_BROWSEINFOW;
38 #endif
39 #endif
40
41 // Use Apple's chooser
42 #ifdef __APPLE__
43 # define MAXFILTERS 80
44 #endif
45
46 // All else falls back to FLTK's own chooser
47 #if ! defined(__APPLE__) && !defined(WIN32)
48 # include <FL/Fl_File_Chooser.H>
49 #else
50 # include <FL/filename.H> // FL_EXPORT
51 #endif
52
53 class Fl_FLTK_File_Chooser;
54 class Fl_GTK_File_Chooser;
55
56 #if FL_EXPORT
57 class FL_EXPORT Fl_Native_File_Chooser {
58 public:
59 enum Type {
60 BROWSE_FILE = 0,
```

```

115     BROWSE_DIRECTORY,
116     BROWSE_MULTI_FILE,
117     BROWSE_MULTI_DIRECTORY,
118     BROWSE_SAVE_FILE,
119     BROWSE_SAVE_DIRECTORY
120 };
121 enum Option {
122     NO_OPTIONS      = 0x0000,
123     SAVEAS_CONFIRM  = 0x0001,
124     NEW_FOLDER      = 0x0002,
125     PREVIEW         = 0x0004,
126     USE_FILTER_EXT  = 0x0008
127 };
128 static const char *file_exists_message;
129
130 public:
131 Fl_Native_File_Chooser(int val=BROWSE_FILE);
132 ~Fl_Native_File_Chooser();
133
134 // Public methods
135 void type(int t);
136 int type() const;
137 void options(int o);
138 int options() const;
139 int count() const;
140 const char *filename() const;
141 const char *filename(int i) const;
142 void directory(const char *val);
143 const char *directory() const;
144 void title(const char *t);
145 const char* title() const;
146 const char *filter() const;
147 void filter(const char *f);
148 int filters() const;
149 void filter_value(int i);
150 int filter_value() const;
151 void preset_file(const char*f);
152 const char* preset_file() const;
153 const char *errmsg() const;
154 int show();
155
156 #ifdef WIN32
157 private:
158     int _btype;                // kind-of browser to show()
159     int _options;              // general options
160 #if FLTK_ABI_VERSION >= 10304
161     fl_OPENFILENAMEW *_ofn_ptr; // GetOpenFileName() & GetSaveFileName() struct
162     fl_BROWSEINFOW *_binf_ptr;  // SHBrowseForFolder() struct
163     WCHAR *_wpattern;          // pattern buffer for filter
164 #else
165     fl_OPENFILENAMEW _ofn;
166     fl_BROWSEINFOW _binf;
167 #endif
168     char **_pathnames;        // array of pathnames
169     int _tpathnames;          // total pathnames
170     char *_directory;         // default pathname to use
171     char *_title;             // title for window
172     char *_filter;            // user-side search filter
173     char *_parsedfilt;        // filter parsed for Windows dialog
174     int _nfilters;            // number of filters parse_filter counted
175     char *_preset_file;       // the file to preselect
176     char *_errmsg;            // error message
177
178 // Private methods
179 void errmsg(const char *msg);
180
181 void clear_pathnames();
182 void set_single_pathname(const char *s);
183 void add_pathname(const char *s);
184
185 void ClearOFN();
186 void ClearBINF();
187 void Win2Unix(char *s);
188 void Unix2Win(char *s);
189 int showfile();
190 int showdir();
191
192 void parse_filter(const char *);
193 void clear_filters();
194 void add_filter(const char *, const char *);
195 #endif
196 #ifdef __APPLE__
197 private:
198     int _btype;                // kind-of browser to show()
199     int _options;              // general options
200     void *_panel;

```



```

203 char      **_pathnames;          // array of pathnames
204 int        _tpathnames;         // total pathnames
205 char      *_directory;          // default pathname to use
206 char      *_title;              // title for window
207 char      *_preset_file;        // the 'save as' filename
208
209 char      *_filter;              // user-side search filter, eg:
210 // C Files\t*.[ch]\nText Files\t*.txt"
211
212 char      *_filt_names;         // filter names (tab delimited)
213 // eg. "C Files\tText Files"
214
215 char      *_filt_patt[MAXFILTERS];
216 // array of filter patterns, eg:
217 //   _filt_patt[0]="*.{cxx,h}"
218 //   _filt_patt[1]="*.txt"
219
220 int        _filt_total;         // parse_filter() # of filters loaded
221 int        _filt_value;         // index of the selected filter
222 char      *_errmsg;             // error message
223
224 // Private methods
225 void errmsg(const char *msg);
226 void clear_pathnames();
227 void set_single_pathname(const char *s);
228 int  get_saveas_basename(void);
229 void clear_filters();
230 void add_filter(const char *, const char *);
231 void parse_filter(const char *from);
232 int  post();
233 int  runmodal();
234 #endif
235
236 #if ! defined(__APPLE__) && !defined(WIN32)
237 private:
238 #if FLTK_ABI_VERSION <= 10302
239 int  _btype;                    // kind-of browser to show()
240 int  _options;                  // general options
241 int  _nfilters;
242 char *_filter;                  // user supplied filter
243 char *_parsedfilt;              // parsed filter
244 int  _filtvalue;                // selected filter
245 char *_preset_file;
246 char *_prevvalue;              // Returned filename
247 char *_directory;
248 char *_errmsg;                  // error message
249 #endif
250 static int have_looked_for_GTK_libs;
251 union {
252     Fl_FLTK_File_Chooser *_x11_file_chooser;
253     Fl_GTK_File_Chooser *_gtk_file_chooser;
254 };
255 #endif
256 };
257
258 #if !defined(__APPLE__) && !defined(WIN32)
259 class FL_EXPORT Fl_FLTK_File_Chooser {
260     friend class Fl_Native_File_Chooser;
261 protected:
262 int  _btype;                    // kind-of browser to show()
263 int  _options;                  // general options
264 int  _nfilters;
265 char *_filter;                  // user supplied filter
266 char *_parsedfilt;              // parsed filter
267 int  _filtvalue;                // selected filter
268 char *_preset_file;
269 char *_prevvalue;              // Returned filename
270 char *_directory;
271 char *_errmsg;                  // error message
272 Fl_FLTK_File_Chooser(int val);
273 virtual ~Fl_FLTK_File_Chooser();
274 void errmsg(const char *msg);
275 int  type_fl_file(int);
276 void parse_filter();
277 int  exist_dialog();
278 Fl_File_Chooser *_file_chooser;
279 virtual void type(int);
280 int  type() const;
281 void options(int);
282 int  options() const;
283 virtual int count() const;
284 virtual const char *filename() const;
285 virtual const char *filename(int i) const;
286 void directory(const char *val);
287 const char *directory() const;
288 virtual void title(const char *);
289 virtual const char* title() const;

```

```

290  const char *filter() const;
291  void filter(const char *);
292  int filters() const;
293  void filter_value(int i);
294  int filter_value() const;
295  void preset_file(const char*);
296  const char* preset_file() const;
297  const char *errmsg() const;
298  virtual int show();
299 };
300
301
302 class FL_EXPORT Fl_GTK_File_Chooser : public Fl_FLTK_File_Chooser {
303     friend class Fl_Native_File_Chooser;
304 private:
305     typedef struct _GtkWidget GtkWidget;
306     typedef struct _GtkFileFilterInfo GtkFileFilterInfo;
307     struct pair {
308         Fl_GTK_File_Chooser* running; // the running Fl_GTK_File_Chooser
309         const char *filter; // a filter string of the chooser
310         pair(Fl_GTK_File_Chooser* c, const char *f) {
311             running = c;
312             filter = strdup(f);
313         };
314         ~pair() {
315             free((char*)filter);
316         };
317     };
318     GtkWidget *gtkw_ptr; // used to hold a GtkWidget* without pulling GTK into everything...
319     void *gtkw_slist; // used to hold a GList GSList...
320     unsigned gtkw_count; // number of files read back - if any
321     mutable char *gtkw_filename; // last name we read back
322     char *gtkw_title; // the title to be applied to the dialog
323     const char *previous_filter;
324
325     int fl_gtk_chooser_wrapper(); // method that wraps the GTK widget
326     Fl_GTK_File_Chooser(int val);
327     virtual ~Fl_GTK_File_Chooser();
328     static int did_find_GTK_libs;
329     static void probe_for_GTK_libs(void);
330     virtual void type(int);
331     virtual int count() const;
332     virtual const char *filename() const;
333     virtual const char *filename(int i) const;
334     virtual void title(const char *);
335     virtual const char* title() const;
336     virtual int show();
337     void changed_output_type(const char *filter);
338
339     static int custom_gtk_filter_function(const GtkFileFilterInfo*, Fl_GTK_File_Chooser::pair*);
340     static void free_pair(pair *p);
341 };
342 #endif // !defined(__APPLE__) && !defined(WIN32)
343
344 #endif /*FL_NATIVE_FILE_CHOOSER_H*/
345
346 //
347 // End of "$Id$".
348 //

```

32.80 Fl_Nice_Slider.H

```

1 //
2 // "$Id$"
3 //
4 // "Nice" slider header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Nice_Slider widget . */
21
22 #ifndef Fl_Nice_Slider_H
23 #define Fl_Nice_Slider_H

```

```

24
25 #include "Fl_Slider.H"
26
27 class FL_EXPORT Fl_Nice_Slider : public Fl_Slider {
28 public:
29   Fl_Nice_Slider(int X,int Y,int W,int H,const char *L=0);
30 };
31
32 #endif
33
34 //
35 // End of "$Id$".
36 //

```

32.81 Fl_Object.H

```

1 //
2 // "$Id$"
3 //
4 // Old Fl_Object header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software.  Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file.  If this
10 // file is missing or damaged, see the license at:
11 //
12 //   http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //   http://www.fltk.org/str.php
17 //
18
19 // This file is provided for back compatibility only.  Please use Fl_Widget
20 #ifndef Fl_Object
21 #define Fl_Object Fl_Widget
22 #endif
23 #include "Fl_Widget.H"
24
25 //
26 // End of "$Id$".
27 //

```

32.82 Fl_Output.H

```

1 //
2 // "$Id$"
3 //
4 // Output header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2011 by Bill Spitzak and others.
7 //
8 // This library is free software.  Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file.  If this
10 // file is missing or damaged, see the license at:
11 //
12 //   http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //   http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Output widget . */
21
22 #ifndef Fl_Output_H
23 #define Fl_Output_H
24
25 #include "Fl_Input.H"
26 class FL_EXPORT Fl_Output : public Fl_Input {
27 public:
28   Fl_Output(int X,int Y,int W,int H, const char *l = 0);
29 };
30
31 #endif
32
33 //
34 // End of "$Id$".
35 //

```

32.83 Fl_Overlay_Window.H

```

1 //
2 // "$Id$"
3 //
4 // Overlay window header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software.  Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file.  If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Overlay_Window class . */
21
22 #ifndef Fl_Overlay_Window_H
23 #define Fl_Overlay_Window_H
24
25 #include "Fl_Double_Window.H"
26
27
28 class FL_EXPORT Fl_Overlay_Window : public Fl_Double_Window {
29 #ifndef FL_DOXYGEN
30     friend class _Fl_Overlay;
31 #endif
32 protected:
33     virtual void draw_overlay() = 0;
34 private:
35     Fl_Window *overlay_;
36 public:
37     void show();
38     void flush();
39     void hide();
40     void resize(int, int, int, int);
41     ~Fl_Overlay_Window();
42     int can_do_overlay();
43     void redraw_overlay();
44 protected:
45     Fl_Overlay_Window(int W, int H, const char *l=0);
46     Fl_Overlay_Window(int X, int Y, int W, int H, const char *l=0);
47 public:
48     void show(int a, char **b) {Fl_Double_Window::show(a,b);}
49 };
50
51 #endif
52 //
53 // End of "$Id$".
54 //

```

32.84 Fl_Pack.H

```

1 //
2 // "$Id$"
3 //
4 // Pack header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software.  Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file.  If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Pack widget . */
21
22 #ifndef Fl_Pack_H
23 #define Fl_Pack_H
24
25 #include <FL/Fl_Group.H>

```

```

26
43 class FL_EXPORT Fl_Pack : public Fl_Group {
44     int spacing_;
45
46 public:
47     enum { // values for type(int)
48         VERTICAL = 0,
49         HORIZONTAL = 1
50     };
51
52 protected:
53     void draw();
54
55 public:
56     Fl_Pack(int x,int y,int w ,int h,const char *l = 0);
61     int spacing()const {return spacing_;}
66     void spacing(int i) {spacing_ = i;}
68     uchar horizontal()const {return type();}
69 };
70
71 #endif
72
73 //
74 // End of "$Id$".
75 //

```

32.85 Fl_Paged_Device.H File Reference

declaration of class [Fl_Paged_Device](#).

```
#include <FL/Fl_Device.H>
```

```
#include <FL/Fl_Window.H>
```

Classes

- class [Fl_Paged_Device](#)
Represents page-structured drawing surfaces.
- struct [Fl_Paged_Device::page_format](#)
width, height and name of a page format

Macros

- #define [NO_PAGE_FORMATS](#) 30 /* MSVC6 compilation fix */
Number of elements in enum Page_Format.

32.85.1 Detailed Description

declaration of class [Fl_Paged_Device](#).

32.86 Fl_Paged_Device.H

[Go to the documentation of this file.](#)

```

1 //
2 // "$Id$"
3 //
4 // Printing support for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 2010-2016 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18

```

```

23 #ifndef Fl_Paged_Device_H
24 #define Fl_Paged_Device_H
25
26 #include <FL/Fl_Device.H>
27 #include <FL/Fl_Window.H>
28
29 #define NO_PAGE_FORMATS 30 /* MSVC6 compilation fix */
30
31 class FL_EXPORT Fl_Paged_Device : public Fl_Surface_Device {
32 #ifndef __APPLE__
33     friend class Fl_Copy_Surface;
34     friend class Fl_Image_Surface;
35     void draw_decorated_window(Fl_Window *win, int x_offset, int y_offset, Fl_Surface_Device *toset);
36 #endif
37 public:
38     enum Page_Format {
39         A0 = 0,
40         A1,
41         A2,
42         A3,
43         A4,
44         A5,
45         A6,
46         A7,
47         A8,
48         A9,
49         B0,
50         B1,
51         B2,
52         B3,
53         B4,
54         B5,
55         B6,
56         B7,
57         B8,
58         B9,
59         B10,
60         C5E,
61         DLE,
62         EXECUTIVE,
63         FOLIO,
64         LEDGER,
65         LEGAL,
66         LETTER,
67         TABLOID,
68         ENVELOPE,
69         MEDIA = 0x1000
70     };
71     enum Page_Layout {
72         PORTRAIT = 0,
73         LANDSCAPE = 0x100,
74         REVERSED = 0x200,
75         ORIENTATION = 0x300
76     };
77     typedef struct {
78         int width;
79         int height;
80         const char *name;
81     } page_format;
82     static const page_format page_formats[NO_PAGE_FORMATS];
83 private:
84     void traverse(Fl_Widget *widget); // finds subwindows of widget and prints them
85 protected:
86     int x_offset;
87     int y_offset;
88     Fl_Paged_Device() : Fl_Surface_Device(NULL), x_offset(0), y_offset(0) {};
89 #if FLTK_ABI_VERSION >= 10301
90 public:
91     virtual ~Fl_Paged_Device() {};
92 #else
93     virtual ~Fl_Paged_Device() {};
94 public:
95 #endif // FLTK_ABI_VERSION
96     static const char *class_id;
97     const char *class_name() {return class_id;};
98     virtual int start_job(int pagecount, int *frompage = NULL, int *topage = NULL);
99     virtual int start_page(void);
100    virtual int printable_rect(int *w, int *h);
101    virtual void margins(int *left, int *top, int *right, int *bottom);
102    virtual void origin(int x, int y);
103    virtual void origin(int *x, int *y);
104    virtual void scale(float scale_x, float scale_y = 0.);
105    virtual void rotate(float angle);
106    virtual void translate(int x, int y);
107    virtual void untranslate(void);
108    virtual void print_widget(Fl_Widget* widget, int delta_x = 0, int delta_y = 0);

```

```

144 void print_window(Fl_Window *win, int x_offset = 0, int y_offset = 0);
145 virtual void print_window_part(Fl_Window *win, int x, int y, int w, int h, int delta_x = 0, int
delta_y = 0);
146 virtual int end_page (void);
147 virtual void end_job (void);
148 };
149
150 #endif // Fl_Paged_Device_H
151
152 //
153 // End of "$Id$"
154 //
155

```

32.87 Fl_Pixmap.H

```

1 //
2 // "$Id$"
3 //
4 // Pixmap header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2012 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 // http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 // http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Pixmap widget . */
21
22 #ifndef Fl_Pixmap_H
23 #define Fl_Pixmap_H
24 # include "Fl_Image.H"
25 #if defined(WIN32)
26 # include "x.H"
27 #endif
28
29 class Fl_Widget;
30 struct Fl_Menu_Item;
31
32 // Older C++ compilers don't support the explicit keyword... :(
33 # if defined(__sgi) && !defined(_COMPILER_VERSION)
34 #   define explicit
35 # endif // __sgi && !_COMPILER_VERSION
36
37 class FL_EXPORT Fl_Pixmap : public Fl_Image {
38     friend class Fl_Quartz_Graphics_Driver;
39     friend class Fl_GDI_Graphics_Driver;
40     friend class Fl_GDI_Printer_Graphics_Driver;
41     friend class Fl_Xlib_Graphics_Driver;
42     void copy_data();
43     void delete_data();
44     void set_data(const char * const *p);
45     int prepare(int XP, int YP, int WP, int HP, int &cx, int &cy,
46               int &X, int &Y, int &W, int &H);
47
48     protected:
49
50     void measure();
51
52     public:
53
54     int alloc_data; // Non-zero if data was allocated
55
56     private:
57
58     #if defined(WIN32)
59     #if FLTK_ABI_VERSION < 10301
60     static // a static member is needed for ABI compatibility
61     #endif
62     UINT pixmap_bg_color; // RGB color used for pixmap background
63     #endif // WIN32
64     #if defined(__APPLE__) || defined(WIN32)
65     void *id_; // for internal use
66     void *mask_; // for internal use (mask bitmap)
67     #else
68     unsigned id_; // for internal use
69     unsigned mask_; // for internal use (mask bitmap)
70

```

```

74 #endif // __APPLE__ || WIN32
75
76 public:
77
79 explicit Fl_Pixmap(char * const * D) : Fl_Image(-1,0,1), alloc_data(0), id_(0), mask_(0)
    {set_data((const char*const*)D); measure();}
81 explicit Fl_Pixmap(uchar* const * D) : Fl_Image(-1,0,1), alloc_data(0), id_(0), mask_(0)
    {set_data((const char*const*)D); measure();}
83 explicit Fl_Pixmap(const char * const * D) : Fl_Image(-1,0,1), alloc_data(0), id_(0), mask_(0)
    {set_data((const char*const*)D); measure();}
85 explicit Fl_Pixmap(const uchar* const * D) : Fl_Image(-1,0,1), alloc_data(0), id_(0), mask_(0)
    {set_data((const char*const*)D); measure();}
86 virtual ~Fl_Pixmap();
87 virtual Fl_Image *copy(int W, int H);
88 Fl_Image *copy() { return copy(w(), h()); }
89 virtual void color_average(Fl_Color c, float i);
90 virtual void desaturate();
91 virtual void draw(int X, int Y, int W, int H, int cx=0, int cy=0);
92 void draw(int X, int Y) {draw(X, Y, w(), h(), 0, 0);}
93 virtual void label(Fl_Widget*w);
94 virtual void label(Fl_Menu_Item*m);
95 virtual void uncache();
96 };
97
98 #endif
99
100 //
101 // End of "$Id$".
102 //

```

32.88 Fl_Plugin.H

```

1 //
2 // "$Id: Fl_Plugin.H 6995 2010-01-12 08:48:55Z matt $"
3 //
4 // A Plugin system for FLTK, implemented in Fl_Preferences.cxx.
5 //
6 // Copyright 2002-2010 by Matthias Melcher.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Plugin class . */
21
22 #ifndef Fl_Plugin_H
23 # define Fl_Plugin_H
24
25 # include "Fl_Preferences.H"
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92

```



```
93 //
94 // End of "$Id: Fl_Preferences.H 6995 2010-01-12 08:48:55Z matt $".
95 //
```

32.89 FI_PNG_Image.H

```
1 //
2 // "$Id$"
3 //
4 // PNG image header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_PNG_Image class . */
21
22 #ifndef Fl_PNG_Image_H
23 #define Fl_PNG_Image_H
24 # include "Fl_Image.H"
25
26 class FL_EXPORT Fl_PNG_Image : public Fl_RGB_Image {
27 public:
28     Fl_PNG_Image(const char* filename);
29     Fl_PNG_Image (const char *name_png, const unsigned char *buffer, int datasize);
30 private:
31     void load_png_(const char *name_png, const unsigned char *buffer_png, int datasize);
32 };
33
34 #endif
35 //
36 // End of "$Id$".
37 //
```

32.90 FI_PNM_Image.H

```
1 //
2 // "$Id$"
3 //
4 // PNM image header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_PNM_Image class . */
21
22 #ifndef Fl_PNM_Image_H
23 #define Fl_PNM_Image_H
24 # include "Fl_Image.H"
25
26 class FL_EXPORT Fl_PNM_Image : public Fl_RGB_Image {
27 public:
28     Fl_PNM_Image(const char* filename);
29 };
30
31 #endif
```

```

40
41 //
42 // End of "$Id$".
43 //

```

32.91 Fl_Positioner.H

```

1 //
2 // "$Id$"
3 //
4 // Positioner header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 // http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 // http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Positioner widget . */
21
22 #ifndef Fl_Positioner_H
23 #define Fl_Positioner_H
24
25 #ifndef Fl_Widget_H
26 #include "Fl_Widget.H"
27 #endif
28
29 class FL_EXPORT Fl_Positioner : public Fl_Widget {
30
31     double xmin, ymin;
32     double xmax, ymax;
33     double xvalue_, yvalue_;
34     double xstep_, ystep_;
35
36 protected:
37
38     // these allow subclasses to put the dial in a smaller area:
39     void draw(int, int, int, int);
40     int handle(int, int, int, int, int);
41     void draw();
42
43 public:
44
45     int handle(int);
46     Fl_Positioner(int x,int y,int w,int h, const char *l=0);
47     double xvalue()const {return xvalue_;}
48     double yvalue()const {return yvalue_;}
49     int xvalue(double);
50     int yvalue(double);
51     int value(double,double);
52     void xbounds(double, double);
53     double xminimum()const {return xmin;}
54     void xminimum(double a) {xbounds(a,xmax);}
55     double xmaximum()const {return xmax;}
56     void xmaximum(double a) {xbounds(xmin,a);}
57     void ybounds(double, double);
58     double yminimum()const {return ymin;}
59     void yminimum(double a) {ybounds(a,ymax);}
60     double ymaximum()const {return ymax;}
61     void ymaximum(double a) {ybounds(ymin, a);}
62     void xstep(double a) {xstep_ = a;}
63     void ystep(double a) {ystep_ = a;}
64 };
65
66 #endif
67 //
68 // End of "$Id$".
69 //

```

32.92 Fl_PostScript.H File Reference

declaration of classes [Fl_PostScript_Graphics_Driver](#), [Fl_PostScript_File_Device](#).

```
#include <FL/Fl_Paged_Device.H>
#include <FL/fl_draw.H>
#include <stdarg.h>
```

Classes

- class [Fl_PostScript_File_Device](#)
To send graphical output to a PostScript file.
- class [Fl_PostScript_Graphics_Driver](#)
PostScript graphical backend.

Typedefs

- typedef int() [Fl_PostScript_Close_Command](#)(FILE *)

32.92.1 Detailed Description

declaration of classes [Fl_PostScript_Graphics_Driver](#), [Fl_PostScript_File_Device](#).

32.93 Fl_PostScript.H

[Go to the documentation of this file.](#)

```
1 //
2 // "$Id$"
3 //
4 // Support for graphics output to PostScript file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 2010-2011 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 // http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 // http://www.fltk.org/str.php
17 //
18
23 #ifndef Fl_PostScript_H
24 #define Fl_PostScript_H
25
26 #include <FL/Fl_Paged_Device.H>
27 #include <FL/fl_draw.H>
28 #include <stdarg.h>
29
30 /* Signature of Fl_PostScript::close_command() functions passed as parameters. */
31 extern "C" {
32     typedef int (Fl_PostScript_Close_Command)(FILE *);
33 }
34
61 class FL_EXPORT Fl_PostScript_Graphics_Driver : public Fl_Graphics_Driver {
62 private:
63     void transformed_draw_extra(const char* str, int n, double x, double y, int w, bool rtl);
64     void *prepare_rle85();
65     void write_rle85(uchar b, void *data);
66     void close_rle85(void *data);
67     void *prepare85();
68     void write85(void *data, const uchar *p, int len);
69     void close85(void *data);
70 public:
71     static const char *class_id;
72     const char *class_name() {return class_id;};
73     Fl_PostScript_Graphics_Driver();
74 #ifndef FL_DOXYGEN
75     enum SHAPE{NONE=0, LINE, LOOP, POLYGON, POINTS};
76
77     class Clip {
78     public:
79         int x, y, w, h;
80         Clip *prev;
81     };
```

```

82  Clip * clip_;
83
84  int lang_level_;
85  int gap_;
86  int pages_;
87
88  double width_;
89  double height_;
90
91  int shape_;
92  int linewidth_; // need for clipping, lang level 1-2
93  int linestyle_; //
94  int interpolate_; //interpolation of images
95  unsigned char cr_, cg_, cb_;
96  char linedash_[256]; //should be enough
97  void concat(); // transform ror scalable dradings...
98  void reconcat(); //invert
99  void recover(); //recovers the state after grestore (such as line styles...)
100 void reset();
101
102  uchar * mask;
103  int mx; // width of mask;
104  int my; // mask lines
105  //Fl_Color bg_;
106  Fl_PostScript_Close_Command* close_cmd_;
107  int page_policy_;
108  int nPages;
109  int orientation_;
110
111  float scale_x;
112  float scale_y;
113  float angle;
114  int left_margin;
115  int top_margin;
116
117  FILE *output;
118  double pw_, ph_;
119
120  uchar bg_r, bg_g, bg_b;
121  int start_postscript (int pagecount, enum Fl_Paged_Device::Page_Format format, enum
  Fl_Paged_Device::Page_Layout layout);
122  /* int alpha_mask(const uchar * data, int w, int h, int D, int LD=0);
123 */
124  void transformed_draw(const char* s, int n, double x, double y); //precise text placing
125  void transformed_draw(const char* s, double x, double y);
126  int alpha_mask(const uchar * data, int w, int h, int D, int LD=0);
127
128  enum Fl_Paged_Device::Page_Format page_format_;
129  char *ps_filename_;
130
131  void page_policy(int p);
132  int page_policy(){return page_policy_;};
133  void close_command(Fl_PostScript_Close_Command* cmd){close_cmd_=cmd;};
134  FILE * file() {return output;};
135  //void orientation (int o);
136  //Fl_PostScript_Graphics_Driver(FILE *o, int lang_level, int pages = 0); // ps (also multi-page)
  constructor
137  //Fl_PostScript_Graphics_Driver(FILE *o, int lang_level, int x, int y, int w, int h); //eps
  constructor
138  void interpolate(int i){interpolate_=i;};
139  int interpolate(){return interpolate_;}
140
141  void page(double pw, double ph, int media = 0);
142  void page(int format);
143 #endif // FL_DOXYGEN
144
145  // implementation of drawing methods
146  void color(Fl_Color c);
147  void color(uchar r, uchar g, uchar b);
148
149  void push_clip(int x, int y, int w, int h);
150  int clip_box(int x, int y, int w, int h, int &X, int &Y, int &W, int &H);
151  int not_clipped(int x, int y, int w, int h);
152  void push_no_clip();
153  void pop_clip();
154
155  void line_style(int style, int width=0, char* dashes=0);
156
157  void rect(int x, int y, int w, int h);
158  void rectf(int x, int y, int w, int h);
159
160  void xyline(int x, int y, int x1);
161  void xyline(int x, int y, int x1, int y2);
162  void xyline(int x, int y, int x1, int y2, int x3);
163
164  void yxline(int x, int y, int y1);
165  void yxline(int x, int y, int y1, int x2);

```

```

166 void yxline(int x, int y, int y1, int x2, int y3);
167
168 void line(int x1, int y1, int x2, int y2);
169 void line(int x1, int y1, int x2, int y2, int x3, int y3);
170
171 void loop(int x0, int y0, int x1, int y1, int x2, int y2);
172 void loop(int x0, int y0, int x1, int y1, int x2, int y2, int x3, int y3);
173 void polygon(int x0, int y0, int x1, int y1, int x2, int y2);
174 void polygon(int x0, int y0, int x1, int y1, int x2, int y2, int x3, int y3);
175 void point(int x, int y);
176
177 void begin_points();
178 void begin_line();
179 void begin_loop();
180 void begin_polygon();
181 void vertex(double x, double y);
182 void curve(double x, double y, double x1, double y1, double x2, double y2, double x3, double y3);
183 void circle(double x, double y, double r);
184 void arc(double x, double y, double r, double start, double a);
185 void arc(int x, int y, int w, int h, double a1, double a2);
186 void pie(int x, int y, int w, int h, double a1, double a2);
187 void end_points();
188 void end_line();
189 void end_loop();
190 void end_polygon();
191 void begin_complex_polygon(){begin_polygon();};
192 void gap(){gap_=1;};
193 void end_complex_polygon(){end_polygon();};
194 void transformed_vertex(double x, double y);
195
196 void draw_image(const uchar* d, int x,int y,int w,int h, int delta=3, int ldelta=0);
197 void draw_image_mono(const uchar* d, int x,int y,int w,int h, int delta=1, int ld=0);
198 void draw_image(Fl_Draw_Image_Cb call, void* data, int x,int y, int w, int h, int delta=3);
199 void draw_image_mono(Fl_Draw_Image_Cb call, void* data, int x,int y, int w, int h, int delta=1);
200
201 void draw(const char* s, int nBytes, int x, int y) {transformed_draw(s,nBytes,x,y); };
202 #ifdef __APPLE__
203 void draw(const char* s, int nBytes, float x, float y) {transformed_draw(s,nBytes,x,y); };
204 #endif
205 void draw(int angle, const char *str, int n, int x, int y);
206 void rtl_draw(const char* s, int n, int x, int y);
207 void font(int face, int size);
208 double width(const char *, int);
209 double width(unsigned int u);
210 void text_extents(const char *c, int n, int &dx, int &dy, int &w, int &h);
211 int height();
212 int descent();
213 void draw(Fl_Pixmap * pxm,int XP, int YP, int WP, int HP, int cx, int cy);
214 void draw(Fl_Bitmap * bitmap,int XP, int YP, int WP, int HP, int cx, int cy);
215 void draw(Fl_RGB_Image * rgb,int XP, int YP, int WP, int HP, int cx, int cy);
216 int draw_scaled(Fl_Image *img, int XP, int YP, int WP, int HP);
217 int clocale_printf(const char *format, ...);
218 ~Fl_PostScript_Graphics_Driver();
219 };
220
221 class FL_EXPORT Fl_PostScript_File_Device : public Fl_Paged_Device {
222 #ifdef __APPLE__
223     CGContextRef gc;
224 #endif
225 protected:
226     Fl_PostScript_Graphics_Driver *driver();
227 public:
228     static const char *class_id;
229     const char *class_name() {return class_id;};
230     Fl_PostScript_File_Device();
231     ~Fl_PostScript_File_Device();
232     int start_job(int pagecount, int* from, int* to);
233     int start_job(int pagecount, enum Fl_Paged_Device::Page_Format format = Fl_Paged_Device::A4,
234                 enum Fl_Paged_Device::Page_Layout layout = Fl_Paged_Device::PORTRAIT);
235     int start_job(FILE *ps_output, int pagecount, enum Fl_Paged_Device::Page_Format format =
236                 Fl_Paged_Device::A4,
237                 enum Fl_Paged_Device::Page_Layout layout = Fl_Paged_Device::PORTRAIT);
238     int start_page (void);
239     int printable_rect(int *w, int *h);
240     void margins(int *left, int *top, int *right, int *bottom);
241     void origin(int *x, int *y);
242     void origin(int x, int y);
243     void scale (float scale_x, float scale_y = 0.);
244     void rotate(float angle);
245     void translate(int x, int y);
246     void untranslate(void);
247     int end_page (void);
248     void end_job(void);
249 #ifdef __APPLE__
250     void set_current() { fl_gc = gc; Fl_Paged_Device::set_current(); }
251 #endif
252 };

```

```

257 static const char *file_chooser_title;
258 };
259
260 #endif // Fl_PostScript_H
261
262 //
263 // End of "$Id$"
264 //

```

32.94 Fl_Preferences.H

```

1 //
2 // "$Id$"
3 //
4 // Preferences .
5 //
6 // Copyright 2002-2010 by Matthias Melcher.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 // http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 // http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Preferences class . */
21
22 #ifndef Fl_Preferences_H
23 # define Fl_Preferences_H
24
25 # include <stdio.h>
26 # include "Fl_Export.H"
27
28
29 class FL_EXPORT Fl_Preferences {
30 public:
31     enum Root {
32         SYSTEM=0,
33         USER
34     };
35
36     typedef void *ID;
37
38     static const char *newUUID();
39
40     Fl_Preferences( Root root, const char *vendor, const char *application );
41     Fl_Preferences( const char *path, const char *vendor, const char *application );
42     Fl_Preferences( Fl_Preferences &parent, const char *group );
43     Fl_Preferences( Fl_Preferences *parent, const char *group );
44     Fl_Preferences( Fl_Preferences &parent, int groupIndex );
45     Fl_Preferences( Fl_Preferences *parent, int groupIndex );
46     Fl_Preferences(const Fl_Preferences&);
47     Fl_Preferences( ID id );
48     virtual ~Fl_Preferences();
49
50     ID id() { return (ID)node; }
51
52     static char remove(ID id_) { return ((Node*)id_)->remove(); }
53
54     const char *name() { return node->name(); }
55
56     const char *path() { return node->path(); }
57
58     int groups();
59     const char *group( int num_group );
60     char groupExists( const char *key );
61     char deleteGroup( const char *group );
62     char deleteAllGroups();
63
64     int entries();
65     const char *entry( int index );
66     char entryExists( const char *key );
67     char deleteEntry( const char *entry );
68     char deleteAllEntries();
69
70     char clear();
71
72     char set( const char *entry, int value );
73     char set( const char *entry, float value );
74     char set( const char *entry, float value, int precision );

```

```

125 char set( const char *entry, double value );
126 char set( const char *entry, double value, int precision );
127 char set( const char *entry, const char *value );
128 char set( const char *entry, const void *value, int size );
129
130 char get( const char *entry, int &value, int defaultValue );
131 char get( const char *entry, float &value, float defaultValue );
132 char get( const char *entry, double &value, double defaultValue );
133 char get( const char *entry, char *&value, const char *defaultValue );
134 char get( const char *entry, char *value, const char *defaultValue, int maxSize );
135 char get( const char *entry, void *&value, const void *defaultValue, int defaultSize );
136 char get( const char *entry, void *value, const void *defaultValue, int defaultSize, int maxSize );
137
138 int size( const char *entry );
139
140 char getUserdataPath( char *path, int pathlen );
141
142 void flush();
143
144 // char export( const char *filename, Type fileFormat );
145 // char import( const char *filename );
146
147 class FL_EXPORT Name {
148
149     char *data_;
150
151 public:
152     Name( unsigned int n );
153     Name( const char *format, ... );
154
155     operator const char *() { return data_; }
156     ~Name();
157 };
158
159 struct Entry {
160     char *name, *value;
161 };
162
163 private:
164     Fl_Preferences() : node(0), rootNode(0) { }
165     Fl_Preferences &operator=(const Fl_Preferences&);
166
167     static char nameBuffer[128];
168     static char uuidBuffer[40];
169     static Fl_Preferences *runtimePrefs;
170
171 public: // older Sun compilers need this (public definition of the following classes)
172     class RootNode;
173
174     class FL_EXPORT Node { // a node contains a list to all its entries
175                             // and all means to manage the tree structure
176     public:
177         Node *child_, *next_;
178         union { // these two are mutually exclusive
179             Node *parent_; // top_bit clear
180             RootNode *root_; // top_bit set
181         };
182         char *path_;
183         Entry *entry_;
184         int nEntry_, NEntry_;
185         unsigned char dirty_:1;
186         unsigned char top_:1;
187         unsigned char indexed_:1;
188         // indexing routines
189         Node **index_;
190         int nIndex_, NIndex_;
191         void createIndex();
192         void updateIndex();
193         void deleteIndex();
194     public:
195         static int lastEntrySet;
196     public:
197         Node( const char *path );
198         ~Node();
199         // node methods
200         int write( FILE *f );
201         const char *name();
202         const char *path() { return path_; }
203         Node *find( const char *path );
204         Node *search( const char *path, int offset=0 );
205         Node *childNode( int ix );
206         Node *addChild( const char *path );
207         void setParent( Node *parent );
208         Node *parent() { return top_?0L:parent_; }
209         void setRoot( RootNode *r ) { root_ = r; top_ = 1; }
210         RootNode *findRoot();
211         char remove();
212         char dirty();

```

```

229 void deleteAllChildren();
230 // entry methods
231 int nChildren();
232 const char *child( int ix );
233 void set( const char *name, const char *value );
234 void set( const char *line );
235 void add( const char *line );
236 const char *get( const char *name );
237 int getEntry( const char *name );
238 char deleteEntry( const char *name );
239 void deleteAllEntries();
240 int nEntry() { return nEntry_; }
241 Entry &entry(int i) { return entry_[i]; }
242 };
243 friend class Node;
244
245 class FL_EXPORT RootNode { // the root node manages file paths and basic reading and
writing
246     Fl_Preferences *prefs_;
247     char *filename_;
248     char *vendor_, *application_;
249 public:
250     RootNode( Fl_Preferences *, Root root, const char *vendor, const char *application );
251     RootNode( Fl_Preferences *, const char *path, const char *vendor, const char *application );
252     RootNode( Fl_Preferences * );
253     ~RootNode();
254     int read();
255     int write();
256     char getPath( char *path, int pathlen );
257 };
258 friend class RootNode;
259
260 protected:
261     Node *node;
262     RootNode *rootNode;
263 };
264
265 #endif // !Fl_Preferences_H
266
267 //
268 // End of "$Id$".
269 //

```

32.95 FI_Printer.H File Reference

declaration of classes [Fl_Printer](#), [Fl_System_Printer](#) and [Fl_PostScript_Printer](#).

```

#include <FL/x.H>
#include <FL/Fl_Paged_Device.H>
#include <FL/fl_draw.H>
#include <FL/Fl_Pixmap.H>
#include <FL/Fl_RGB_Image.H>
#include <FL/Fl_Bitmap.H>
#include <stdio.h>
#include <FL/Fl_PostScript.H>

```

Classes

- class [Fl_PostScript_Printer](#)
Print support under Unix/Linux.
- class [Fl_Printer](#)
OS-independent print support.
- class [Fl_System_Printer](#)
Print support under MSWindows and Mac OS.

32.95.1 Detailed Description

declaration of classes [Fl_Printer](#), [Fl_System_Printer](#) and [Fl_PostScript_Printer](#).

32.96 Fl_Printer.H

[Go to the documentation of this file.](#)

```

1 //
2 // "$Id$"
3 //
4 // Printing support for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 2010-2014 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 // http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 // http://www.fltk.org/str.php
17 //
18
23 #ifndef Fl_Printer_H
24 #define Fl_Printer_H
25
26 #include <FL/x.H>
27 #include <FL/Fl_Paged_Device.H>
28 #include <FL/fl_draw.H>
29 #include <FL/Fl_Pixmap.H>
30 #include <FL/Fl_RGB_Image.H>
31 #include <FL/Fl_Bitmap.H>
32 #include <stdio.h>
33 #if !(defined(__APPLE__) || defined(WIN32))
34 #include <FL/Fl_PostScript.H>
35 #elif defined(WIN32)
36 #include <commdlg.h>
37 #endif
38
39 #if defined(__APPLE__) || defined(WIN32) || defined(FL_DOXYGEN)
47 class Fl_System_Printer : public Fl_Paged_Device {
48     friend class Fl_Printer;
49 private:
51     void *gc;
52     void set_current(void);
53 #ifdef __APPLE__
54     float scale_x;
55     float scale_y;
56     float angle; // rotation angle in radians
57     PMPrintSession printSession;
58     PMPageFormat pageFormat;
59     PMPrintSettings printSettings;
60 #elif defined(WIN32)
61     int abortPrint;
62     PRINTDLG pd;
63     HDC hDC;
64     int prerr;
65     int left_margin;
66     int top_margin;
67     void absolute_printable_rect(int *x, int *y, int *w, int *h);
68 #endif
69 protected:
71     Fl_System_Printer(void);
72 public:
73     static const char *class_id;
74     const char *class_name() {return class_id;};
75     int start_job(int pagecount, int *frompage = NULL, int *topage = NULL);
76     int start_page(void);
77     int printable_rect(int *w, int *h);
78     void margins(int *left, int *top, int *right, int *bottom);
79     void origin(int *x, int *y);
80     void origin(int x, int y);
81     void scale(float scale_x, float scale_y = 0.);
82     void rotate(float angle);
83     void translate(int x, int y);
84     void untranslate(void);
85     int end_page(void);
86     void end_job(void);
87 #ifdef __APPLE__
88     void print_window_part(Fl_Window *win, int x, int y, int w, int h, int delta_x, int delta_y);
89 #endif
91     ~Fl_System_Printer(void);
92 }; // class Fl_System_Printer
93
94 #endif
95
96 #if !(defined(__APPLE__) || defined(WIN32))
104 class Fl_PostScript_Printer : public Fl_PostScript_File_Device {

```

```

105 friend class Fl_Printer;
106 protected:
107     Fl_PostScript_Printer(void) {};
108 public:
109     static const char *class_id;
110     const char *class_name() {return class_id;};
111     int start_job(int pages, int *firstpage = NULL, int *lastpage = NULL);
112 };
113 };
114
115 #endif
116
117 class FL_EXPORT Fl_Printer : public Fl_Paged_Device {
118 public:
119     static const char *class_id;
120     const char *class_name() {return class_id;};
121     Fl_Printer(void);
122     int start_job(int pagecount, int *frompage = NULL, int *topage = NULL);
123     int start_page(void);
124     int printable_rect(int *w, int *h);
125     void margins(int *left, int *top, int *right, int *bottom);
126     void origin(int *x, int *y);
127     void origin(int x, int y);
128     void scale(float scale_x, float scale_y = 0.);
129     void rotate(float angle);
130     void translate(int x, int y);
131     void untranslate(void);
132     int end_page (void);
133     void end_job (void);
134     void print_widget(Fl_Widget* widget, int delta_x=0, int delta_y=0);
135     void print_window_part(Fl_Window *win, int x, int y, int w, int h, int delta_x=0, int delta_y=0);
136     void set_current(void);
137     Fl_Graphics_Driver* driver(void);
138     ~Fl_Printer(void);
139
140     static const char *dialog_title;
141     static const char *dialog_printer;
142     static const char *dialog_range;
143     static const char *dialog_copies;
144     static const char *dialog_all;
145     static const char *dialog_pages;
146     static const char *dialog_from;
147     static const char *dialog_to;
148     static const char *dialog_properties;
149     static const char *dialog_copyNo;
150     static const char *dialog_print_button;
151     static const char *dialog_cancel_button;
152     static const char *dialog_print_to_file;
153     static const char *property_title;
154     static const char *property_pagesize;
155     static const char *property_mode;
156     static const char *property_use;
157     static const char *property_save;
158     static const char *property_cancel;
159 private:
160     #if defined(WIN32) || defined(__APPLE__)
161         Fl_System_Printer *printer;
162     #else
163         Fl_PostScript_Printer *printer;
164     #endif
165 };
166
167 #endif // Fl_Printer_H
168
169 //
170 // End of "$Id$"
171 //

```

32.97 Fl_Progress.H

```

1 //
2 // "$Id$"
3 //
4 // Progress bar widget definitions.
5 //
6 // Copyright 2000-2010 by Michael Sweet.
7 //
8 // This library is free software.  Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file.  If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php

```

```

17 //
18
19 /* \file
20 Fl_Progress widget . */
21
22 #ifndef _Fl_Progress_H_
23 # define _Fl_Progress_H_
24
25 //
26 // Include necessary headers.
27 //
28
29 #include "Fl_Widget.H"
30
31
32 //
33 // Progress class...
34 //
35
36 class FL_EXPORT Fl_Progress : public Fl_Widget {
37
38     float value_,
39           minimum_,
40           maximum_;
41
42     protected:
43
44     virtual void draw();
45
46     public:
47
48     Fl_Progress(int x, int y, int w, int h, const char *l = 0);
49
50     void maximum(float v) { maximum_ = v; redraw(); }
51     float maximum()const { return (maximum_); }
52
53     void minimum(float v) { minimum_ = v; redraw(); }
54     float minimum()const { return (minimum_); }
55
56     void value(float v) { value_ = v; redraw(); }
57     float value()const { return (value_); }
58 };
59
60 #endif // !_Fl_Progress_H_
61
62 //
63 // End of "$Id$".
64 //

```

32.98 Fl_Radio_Button.H

```

1 //
2 // "$Id$"
3 //
4 // Radio button header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2014 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Radio_Button widget . */
21
22 #ifndef Fl_Radio_Button_H
23 #define Fl_Radio_Button_H
24
25 #include "Fl_Button.H"
26
27 class FL_EXPORT Fl_Radio_Button : public Fl_Button {
28     public:
29     Fl_Radio_Button(int X,int Y,int W,int H,const char *L=0);
30 };
31
32 #endif
33
34 //
35 // End of "$Id$".

```

```
36 //
```

32.99 Fl_Radio_Light_Button.H

```
1 //
2 // "$Id$"
3 //
4 // Radio light button header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2014 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Radio_Light_Button widget . */
21
22 #ifndef Fl_Radio_Light_Button_H
23 #define Fl_Radio_Light_Button_H
24
25 #include "Fl_Light_Button.H"
26
27 class FL_EXPORT Fl_Radio_Light_Button : public Fl_Light_Button {
28 public:
29     Fl_Radio_Light_Button(int X,int Y,int W,int H,const char *l=0);
30 };
31
32 #endif
33
34 //
35 // End of "$Id$".
36 //
```

32.100 Fl_Radio_Round_Button.H

```
1 //
2 // "$Id$"
3 //
4 // Radio round button header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2014 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Radio_Round_Button widget . */
21
22 #ifndef Fl_Radio_Round_Button_H
23 #define Fl_Radio_Round_Button_H
24
25 #include "Fl_Round_Button.H"
26
27 class FL_EXPORT Fl_Radio_Round_Button : public Fl_Round_Button {
28 public:
29     Fl_Radio_Round_Button(int X,int Y,int W,int H,const char *L=0);
30 };
31
32 #endif
33
34 //
35 // End of "$Id$".
36 //
```

32.101 Fl_Repeat_Button.H

```
1 //
2 // "$Id$"
3 //
4 // Repeat button header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Repeat_Button widget . */
21
22 #ifndef Fl_Repeat_Button_H
23 #define Fl_Repeat_Button_H
24 #include "Fl.H"
25 #include "Fl_Button.H"
26
27 class FL_EXPORT Fl_Repeat_Button : public Fl_Button {
28     static void repeat_callback(void *);
29 public:
30     int handle(int);
31     Fl_Repeat_Button(int X,int Y,int W,int H,const char *l=0);
32
33     void deactivate() {
34         Fl::remove_timeout(repeat_callback,this);
35         Fl_Button::deactivate();
36     }
37 };
38
39 #endif
40
41 // End of "$Id$".
42 //
```

32.102 Fl_Return_Button.H

```
1 //
2 // "$Id$"
3 //
4 // Return button header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Return_Button widget . */
21
22 #ifndef Fl_Return_Button_H
23 #define Fl_Return_Button_H
24 #include "Fl_Button.H"
25
26 class FL_EXPORT Fl_Return_Button : public Fl_Button {
27     protected:
28         void draw();
29     public:
30         int handle(int);
31         Fl_Return_Button(int X, int Y, int W, int H,const char *l=0);
32 };
33
34 #endif
35
36 //
```

```
49 // End of "$Id$".
50 //
```

32.103 FI_RGB_Image.H

```
1 //
2 // "$Id$"
3 //
4 // RGB Image header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 #ifndef Fl_RGB_Image_H
20 #define Fl_RGB_Image_H
21 #include "Fl_Image.H"
22 #endif // !Fl_RGB_Image_H
23
24 //
25 // End of "$Id$".
26 //
```

32.104 FI_Roller.H

```
1 //
2 // "$Id$"
3 //
4 // Roller header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Roller widget . */
21
22 #ifndef Fl_Roller_H
23 #define Fl_Roller_H
24
25 #ifndef Fl_Valuator_H
26 #include "Fl_Valuator.H"
27 #endif
28
29 class FL_EXPORT Fl_Roller : public Fl_Valuator {
30 protected:
31     void draw();
32 public:
33     int handle(int);
34     Fl_Roller(int X,int Y,int W,int H,const char* L=0);
35 };
36 #endif
37
38 //
39 // End of "$Id$".
40 //
```

32.105 FI_Round_Button.H

```
1 //
```

```

2 // "$Id$"
3 //
4 // Round button header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2014 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 #ifndef Fl_Round_Button_H
20 #define Fl_Round_Button_H
21
22 #include "Fl_Light_Button.H"
23
24 class FL_EXPORT Fl_Round_Button : public Fl_Light_Button {
25 public:
26     Fl_Round_Button(int x,int y,int w,int h,const char *l = 0);
27 };
28 #endif
29 //
30 // End of "$Id$".
31 //

```

32.106 Fl_Round_Clock.H

```

1 //
2 // "$Id$"
3 //
4 // Round clock header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Round_Clock widget . */
21
22 #ifndef Fl_Round_Clock_H
23 #define Fl_Round_Clock_H
24
25 #include "Fl_Clock.H"
26
27 class FL_EXPORT Fl_Round_Clock : public Fl_Clock {
28 public:
29     Fl_Round_Clock(int X,int Y,int W,int H, const char *L = 0);
30 };
31 #endif
32 //
33 // End of "$Id$".
34 //

```

32.107 Fl_Scroll.H

```

1 //
2 // Scroll header file for the Fast Light Tool Kit (FLTK).
3 //
4 // Copyright 1998-2021 by Bill Spitzak and others.
5 //
6 // This library is free software. Distribution and use rights are outlined in
7 // the file "COPYING" which should have been included with this file. If this
8 // file is missing or damaged, see the license at:

```

```

9 //
10 //      https://www.fltk.org/COPYING.php
11 //
12 // Please see the following page on how to report bugs and issues:
13 //
14 //      https://www.fltk.org/bugs.php
15 //
16
17 /* \file
18 Fl_Scroll widget . */
19
20 #ifndef Fl_Scroll_H
21 #define Fl_Scroll_H
22
23 #include "Fl_Group.H"
24 #include "Fl_Scrollbar.H"
25
26 class FL_EXPORT Fl_Scroll : public Fl_Group {
27
28     int xposition_, yposition_;
29     int oldx, oldy;
30     int scrollbar_size_;
31     static void hscrollbar_cb(Fl_Widget*, void*);
32     static void scrollbar_cb(Fl_Widget*, void*);
33     void fix_scrollbar_order();
34     static void draw_clip(void*,int,int,int,int);
35
36 #if FLTK_ABI_VERSION >= 10303
37 protected:          // NEW (STR#1895)
38 #else
39 private:            // OLD
40 #endif
41
42     typedef struct { int x,y,w,h; } Fl_Region_XYWH;
43
44     typedef struct {
45         int l;
46         int r;
47         int t;
48         int b;
49     } Fl_Region_LRTB;
50
51     typedef struct {
52         int x,y,w,h;
53         int pos;
54         int size;
55         int first;
56         int total;
57     } Fl_Scrollbar_Data;
58
59     typedef struct {
60         int scrollsize;
61         Fl_Region_XYWH innerbox;
62         Fl_Region_XYWH innerchild;
63         Fl_Region_LRTB child;
64         int hneeded;
65         int vneeded;
66         Fl_Scrollbar_Data hscroll;
67         Fl_Scrollbar_Data vscroll;
68     } ScrollInfo;
69
70     void recalc_scrollbars(ScrollInfo &si);
71
72 protected:
73
74     void bbox(int&,int&,int&,int&);
75     void draw();
76
77 public:
78
79     Fl_Scrollbar scrollbar;
80     Fl_Scrollbar hscrollbar;
81
82     void resize(int X, int Y, int W, int H);
83     int handle(int);
84
85     Fl_Scroll(int X,int Y,int W,int H,const char*l=0);
86
87     enum { // values for type()
88         HORIZONTAL = 1,
89         VERTICAL = 2,
90         BOTH = 3,
91         ALWAYS_ON = 4,
92         HORIZONTAL_ALWAYS = 5,
93         VERTICAL_ALWAYS = 6,
94         BOTH_ALWAYS = 7
95     };
96 };

```



```

164
166 int xposition()const {return xposition_;}
168 int yposition()const {return yposition_;}
169 void scroll_to(int, int);
170 void clear();
180 int scrollbar_size()const {
181     return(scrollbar_size_);
182 }
202 void scrollbar_size(int newSize) {
203     if ( newSize != scrollbar_size_ ) redraw();
204     scrollbar_size_ = newSize;
205 }
206 };
207
208 #endif

```

32.108 Fl_Scrollbar.H

```

1 //
2 // "$Id$"
3 //
4 // Scroll bar header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software.  Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file.  If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Scrollbar widget . */
21
22 #ifndef Fl_Scrollbar_H
23 #define Fl_Scrollbar_H
24
25 #include "Fl_Slider.H"
26
43 class FL_EXPORT Fl_Scrollbar : public Fl_Slider {
44
45     int linesize_;
46     int pushed_;
47     static void timeout_cb(void*);
48     void increment_cb();
49 protected:
50     void draw();
51
52 public:
53
54     Fl_Scrollbar(int X,int Y,int W,int H, const char *L = 0);
55     ~Fl_Scrollbar();
56     int handle(int);
57
65     int value()const {return int(Fl_Slider::value());}
66
73     int value(int p) {return int(Fl_Slider::value((double)p));}
74
89     int value(int pos, int windowSize, int first, int total) {
90         return scrollvalue(pos, windowSize, first, total);
91     }
92
96     int linesize()const {return linesize_;}
97
103     void linesize(int i) {linesize_ = i;}
104
105 };
106
107 #endif
108
109 //
110 // End of "$Id$".
111 //

```

32.109 Fl_Secret_Input.H

```

1 //

```

```

2 // "$Id$"
3 //
4 // Secret input header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2011 by Bill Spitzak and others.
7 //
8 // This library is free software.  Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file.  If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18 //
19 /* \file
20 Fl_Secret_Input widget . */
21
22 #ifndef Fl_Secret_Input_H
23 #define Fl_Secret_Input_H
24
25 #include "Fl_Input.H"
26
27 class FL_EXPORT Fl_Secret_Input : public Fl_Input {
28 public:
29     Fl_Secret_Input(int X,int Y,int W,int H,const char *l = 0);
30     int handle(int);
31 };
32
33 #endif
34 // End of "$Id$".
35 //

```

32.110 Fl_Select_Browser.H

```

1 //
2 // "$Id$"
3 //
4 // Select browser header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software.  Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file.  If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18 //
19 /* \file
20 Fl_Select_Browser widget . */
21
22 #ifndef Fl_Select_Browser_H
23 #define Fl_Select_Browser_H
24
25 #include "Fl_Browser.H"
26
27 class FL_EXPORT Fl_Select_Browser : public Fl_Browser {
28 public:
29     Fl_Select_Browser(int X,int Y,int W,int H,const char *L=0);
30 };
31
32 #endif
33 // End of "$Id$".
34 //

```

32.111 Fl_Shared_Image.H File Reference

[Fl_Shared_Image](#) class.

```
#include "Fl_Image.H"
```

Classes

- class [Fl_Shared_Image](#)

This class supports caching, loading, scaling, and drawing of image files.

Typedefs

- typedef [Fl_Image](#) `*(Fl_Shared_Handler) (const char *name, uchar *header, int headerlen)`

Functions

- FL_EXPORT void [fl_register_images](#) ()

Register the image formats.

32.111.1 Detailed Description

[Fl_Shared_Image](#) class.

32.111.2 Function Documentation

32.111.2.1 fl_register_images()

```
FL_EXPORT void fl_register_images ( )
```

Register the image formats.

This function is provided in the fltk_images library and registers all of the "extra" image file formats that are not part of the core FLTK library.

32.112 Fl_Shared_Image.H

[Go to the documentation of this file.](#)

```
1 //
2 // "$Id$"
3 //
4 // Shared image header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2017 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 #ifndef Fl_Shared_Image_H
20 #define Fl_Shared_Image_H
21
22 #include "Fl_Image.H"
23
24 // Test function for adding new formats
25 typedef Fl_Image *(Fl_Shared_Handler)(const char *name, uchar *header,
26                                     int headerlen);
27
28 // Shared images class.
29 class FL_EXPORT Fl_Shared_Image : public Fl_Image {
30
31     friend class Fl_JPEG_Image;
32     friend class Fl_PNG_Image;
33
34 private:
35     static Fl_RGB_Scaling scaling_algorithm_; // method used to rescale RGB source images
36 #if FLTK_ABI_VERSION >= 10304
37     Fl_Image *scaled_image_;
38 #endif
39 };
```

```

59 #endif
60 protected:
61
62 static Fl_Shared_Image **images_; // Shared images
63 static int num_images_; // Number of shared images
64 static int alloc_images_; // Allocated shared images
65 static Fl_Shared_Handler *handlers_; // Additional format handlers
66 static int num_handlers_; // Number of format handlers
67 static int alloc_handlers_; // Allocated format handlers
68
69 const char *name_; // Name of image file
70 int original_; // Original image?
71 int refcount_; // Number of times this image has been used
72 Fl_Image *image_; // The image that is shared
73 int alloc_image_; // Was the image allocated?
74
75 static int compare(Fl_Shared_Image **i0, Fl_Shared_Image **i1);
76
77 // Use get() and release() to load/delete images in memory...
78 Fl_Shared_Image();
79 Fl_Shared_Image(const char *n, Fl_Image *img = 0);
80 virtual ~Fl_Shared_Image();
81 void add();
82 void update();
83
84 public:
85 const char *name() { return name_; }
86
87
88
89 int refcount() { return refcount_; }
90
91
92
93
94 int original() { return original_; }
95
96 void release();
97 void reload();
98
99 virtual Fl_Image *copy(int W, int H);
100 Fl_Image *copy() { return copy(w(), h()); }
101 virtual void color_average(Fl_Color c, float i);
102 virtual void desaturate();
103 virtual void draw(int X, int Y, int W, int H, int cx, int cy);
104 void draw(int X, int Y) { draw(X, Y, w(), h(), 0, 0); }
105 void scale(int width, int height, int proportional = 1, int can_expand = 0);
106 virtual void uncache();
107
108 static Fl_Shared_Image *find(const char *name, int W = 0, int H = 0);
109 static Fl_Shared_Image *get(const char *name, int W = 0, int H = 0);
110 static Fl_Shared_Image *get(Fl_RGB_Image *rgb, int own_it = 1);
111 static Fl_Shared_Image **images();
112 static int num_images();
113 static void add_handler(Fl_Shared_Handler f);
114 static void remove_handler(Fl_Shared_Handler f);
115 static void scaling_algorithm(Fl_RGB_Scaling algorithm) {scaling_algorithm_ = algorithm; }
116 };
117
118 //
119 // The following function is provided in the fltk_images library and
120 // registers all of the "extra" image file formats that are not part
121 // of the core FLTK library...
122 //
123 FL_EXPORT extern void fl_register_images();
124
125 #endif // !Fl_Shared_Image_H
126
127 //
128 // End of "$Id$"
129 //

```

32.113 fl_show_colormap.H File Reference

The `fl_show_colormap()` function hides the implementation classes used to provide the popup window and color selection mechanism.

Functions

- FL_EXPORT `Fl_Color fl_show_colormap(Fl_Color oldcol)`

Pops up a window to let the user pick a colormap entry.

32.113.1 Detailed Description

The `fl_show_colormap()` function hides the implementation classes used to provide the popup window and color selection mechanism.

32.114 fl_show_colormap.H

[Go to the documentation of this file.](#)

```

1 //
2 // "$Id$"
3 //
4 // Colormap picker header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software.  Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file.  If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
24 #ifndef fl_show_colormap_H
25 #define fl_show_colormap_H
26
27 /* doxygen comment here to avoid exposing ColorMenu in fl_show_colormap.cxx
28 */
29
41 FL_EXPORT Fl_Color fl_show_colormap(Fl_Color oldcol);
42
45 #endif
46
47 //
48 // End of "$Id$".
49 //

```

32.115 fl_show_input.H

```

1 //
2 // "$Id$"
3 //
4 // Standard input dialog header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software.  Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file.  If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 #include "fl_ask.H"
20
21 //
22 // End of "$Id$".
23 //

```

32.116 Fl_Simple_Counter.H

```

1 //
2 // "$Id$"
3 //
4 // Simple counter header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software.  Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file.  If this
10 // file is missing or damaged, see the license at:

```

```

11 //
12 //      http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //      http://www.fltk.org/str.php
17 //
18 //
19 /* \file
20 Fl_Simple_Counter widget . */
21
22 #ifndef Fl_Simple_Counter_H
23 #define Fl_Simple_Counter_H
24
25 #include "Fl_Counter.H"
26
27 class FL_EXPORT Fl_Simple_Counter : public Fl_Counter {
28 public:
29     Fl_Simple_Counter(int X,int Y,int W,int H, const char *L = 0);
30 };
31
32 #endif
33 //
34 // End of "$Id$".
35 //

```

32.117 Fl_Single_Window.H

```

1 //
2 // "$Id$"
3 //
4 // Single-buffered window header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2015 by Bill Spitzak and others.
7 //
8 // This library is free software.  Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file.  If this
10 // file is missing or damaged, see the license at:
11 //
12 //      http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //      http://www.fltk.org/str.php
17 //
18 //
19 /* \file
20 Fl_Single_Window class . */
21
22 #ifndef Fl_Single_Window_H
23 #define Fl_Single_Window_H
24
25 #include "Fl_Window.H"
26
27 class FL_EXPORT Fl_Single_Window : public Fl_Window {
28 public:
29     void show();
30     void show(int a, char **b) {Fl_Window::show(a,b);}
31     void flush();
32     Fl_Single_Window(int W, int H, const char *l=0);
33     Fl_Single_Window(int X, int Y, int W, int H, const char *l=0);
34     int make_current();
35 };
36
37 #endif
38 //
39 // End of "$Id$".
40 //

```

32.118 Fl_Slider.H

```

1 //
2 // "$Id$"
3 //
4 // Slider header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software.  Distribution and use rights are outlined in

```

```

9 // the file "COPYING" which should have been included with this file.   If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Slider widget . */
21
22 #ifndef Fl_Slider_H
23 #define Fl_Slider_H
24
25 #ifndef Fl_Valuator_H
26 #include "Fl_Valuator.H"
27 #endif
28
29 // values for type(), lowest bit indicate horizontal:
30 #define FL_VERT_SLIDER      0
31 #define FL_HOR_SLIDER      1
32 #define FL_VERT_FILL_SLIDER 2
33 #define FL_HOR_FILL_SLIDER 3
34 #define FL_VERT_NICE_SLIDER 4
35 #define FL_HOR_NICE_SLIDER 5
36
37 class FL_EXPORT Fl_Slider : public Fl_Valuator {
38
39     float slider_size_;
40     uchar slider_;
41     void _Fl_Slider();
42     void draw_bg(int, int, int, int);
43
44 protected:
45
46     // these allow subclasses to put the slider in a smaller area:
47     void draw(int, int, int, int);
48     int handle(int, int, int, int, int);
49     void draw();
50
51 public:
52
53     int handle(int);
54     Fl_Slider(int X,int Y,int W,int H, const char *L = 0);
55     Fl_Slider(uchar t,int X,int Y,int W,int H, const char *L);
56
57     int scrollvalue(int pos,int size,int first,int total);
58     void bounds(double a, double b);
59
60     float slider_size()const {return slider_size_;}
61     void slider_size(double v);
62
63     Fl_Boxtype slider()const {return (Fl_Boxtype)slider_;}
64
65     void slider(Fl_Boxtype c) {slider_ = c;}
66 };
67 #endif
68 //
69 // End of "$Id$".
70 //

```

32.119 Fl_Spinner.H

```

1 //
2 // "$Id$"
3 //
4 // Spinner widget for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software.  Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file.   If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //

```

```

18
19 /* \file
20 Fl_Spinner widget . */
21
22 #ifndef Fl_Spinner_H
23 # define Fl_Spinner_H
24
25 //
26 // Include necessary headers...
27 //
28
29 # include <FL/Enumerations.H>
30 # include <FL/Fl_Group.H>
31 # include <FL/Fl_Input.H>
32 # include <FL/Fl_Repeat_Button.H>
33 # include <stdio.h>
34 # include <stdlib.h>
35
36
37 class FL_EXPORT Fl_Spinner : public Fl_Group {
38
39     double      value_;           // Current value
40     double      minimum_;        // Minimum value
41     double      maximum_;        // Maximum value
42     double      step_;           // Amount to add/subtract for up/down
43     const char  *format_;        // Format string
44
45 #if FLTK_ABI_VERSION >= 10301
46 // NEW
47 protected:
48 #endif
49     Fl_Input    input_;          // Input field for the value
50     Fl_Repeat_Button repeat_;    // Repeat button
51     up_button_, // Up button
52     down_button_; // Down button
53
54 private:
55     static void sb_cb(Fl_Widget *w, Fl_Spinner *sb) {
56         double v; // New value
57
58         if (w == &(sb->input_)) {
59             // Something changed in the input field...
60             v = atof(sb->input_.value());
61
62             if (v < sb->minimum_) {
63                 sb->value_ = sb->minimum_;
64                 sb->update();
65             } else if (v > sb->maximum_) {
66                 sb->value_ = sb->maximum_;
67                 sb->update();
68             } else sb->value_ = v;
69         } else if (w == &(sb->up_button_)) {
70             // Up button pressed...
71             v = sb->value_ + sb->step_;
72
73             if (v > sb->maximum_) sb->value_ = sb->minimum_;
74             else sb->value_ = v;
75
76             sb->update();
77         } else if (w == &(sb->down_button_)) {
78             // Down button pressed...
79             v = sb->value_ - sb->step_;
80
81             if (v < sb->minimum_) sb->value_ = sb->maximum_;
82             else sb->value_ = v;
83
84             sb->update();
85         }
86         sb->set_changed();
87         sb->do_callback();
88     }
89
90     void update() {
91         char s[255]; // Value string
92
93         if (format_[0]!='%'&&format_[1]!='.'&&format_[2]!='*') { // precision argument
94             // this code block is a simplified version of
95             // Fl_Valuator::format() and works well (but looks ugly)
96             int c = 0;
97             char temp[64], *sp = temp;
98             sprintf(temp, "%.12f", step_);
99             while (*sp) sp++;
100            sp--;
101            while (sp>temp && *sp=='0') sp--;
102            while (sp>temp && (*sp>='0' && *sp<='9')) { sp--; c++; }
103            sprintf(s, format_, c, value_);
104        } else {

```



```

113         sprintf(s, format_, value_);
114     }
115     input_.value(s);
116 }
117
118 public:
119
120 Fl_Spinner(int X, int Y, int W, int H, const char *L = 0);
121
122 const char *format() { return (format_); }
123 void format(const char *f) { format_ = f; update(); }
124
125 int handle(int event) {
126     switch (event) {
127     case FL_KEYDOWN :
128     case FL_SHORTCUT :
129         if (Fl::event_key() == FL_Up) {
130             up_button_.do_callback();
131             return 1;
132         } else if (Fl::event_key() == FL_Down) {
133             down_button_.do_callback();
134             return 1;
135         } else return 0;
136
137     case FL_FOCUS :
138         if (input_.take_focus()) return 1;
139         else return 0;
140     }
141
142     return Fl_Group::handle(event);
143 }
144
145 double maximum()const { return (maximum_); }
146 double maximum(const) { return (maximum_); }
147 void maximum(double m) { maximum_ = m; }
148 double minimum()const { return (minimum_); }
149 double minimum(const) { return (minimum_); }
150 void minimum(double m) { minimum_ = m; }
151 void range(double a, double b) { minimum_ = a; maximum_ = b; }
152 void resize(int X, int Y, int W, int H) {
153     Fl_Group::resize(X, Y, W, H);
154
155     input_.resize(X, Y, W - H / 2 - 2, H);
156     up_button_.resize(X + W - H / 2 - 2, Y, H / 2 + 2, H / 2);
157     down_button_.resize(X + W - H / 2 - 2, Y + H - H / 2,
158                         H / 2 + 2, H / 2);
159 }
160 double step()const { return (step_); }
161 void step(double s) {
162     step_ = s;
163     if (step_ != (int)step_) input_.type(FL_FLOAT_INPUT);
164     else input_.type(FL_INT_INPUT);
165     update();
166 }
167 Fl_Color textcolor()const {
168     return (input_.textcolor());
169 }
170 void textcolor(Fl_Color c) {
171     input_.textcolor(c);
172 }
173 Fl_Font textfont()const {
174     return (input_.textfont());
175 }
176 void textfont(Fl_Font f) {
177     input_.textfont(f);
178 }
179 Fl_Fontsize textsize()const {
180     return (input_.textsize());
181 }
182 void textsize(Fl_Fontsize s) {
183     input_.textsize(s);
184 }
185
186 uchar type()const { return (input_.type()); }
187 void type(uchar v) {
188     if (v==FL_FLOAT_INPUT) {
189         format("%.*f");
190     } else {
191         format("%.0f");
192     }
193     input_.type(v);
194 }
195 double value()const { return (value_); }
196 void value(double v) { value_ = v; update(); }
197 void color(Fl_Color v) { input_.color(v); }
198 Fl_Color color()const { return (input_.color()); }
199 void selection_color(Fl_Color val) { input_.selection_color(val); }
200 Fl_Color selection_color()const { return (input_.selection_color()); }

```

```

253 };
254
255 #endif // !Fl_Spinner_H
256
257 //
258 // End of "$Id$".
259 //

```

32.120 Fl_Sys_Menu_Bar.H

```

1 //
2 // "$Id$"
3 //
4 // MacOS system menu bar header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 #ifndef Fl_Sys_Menu_Bar_H
20 #define Fl_Sys_Menu_Bar_H
21
22 #include "Fl_Menu_Bar.H"
23 #include "x.H"
24
25 #if defined(__APPLE__) || defined(FL_DOXYGEN)
26
27 class FL_EXPORT Fl_Sys_Menu_Bar : public Fl_Menu_Bar {
28
29 #if FLTK_ABI_VERSION >= 10304
30 // NEW -- update() public (STR#3317)
31 public:
32 void update();
33 protected:
34 void draw();
35 #else
36 // OLD -- update() protected
37 protected:
38 void update();
39 void draw();
40 #endif
41 #endif
42
43 public:
44 Fl_Sys_Menu_Bar(int x,int y,int w,int h,const char *l=0);
45 ~Fl_Sys_Menu_Bar();
46 const Fl_Menu_Item *menu()const {return Fl_Menu_::menu();}
47 void menu(const Fl_Menu_Item *m);
48 int add(const char* label, int shortcut, Fl_Callback*, void *user_data=0, int flags=0);
49 int add(const char* label, const char* shortcut, Fl_Callback* cb, void *user_data=0, int flags=0) {
50     return add(label, fl_old_shortcut(shortcut), cb, user_data, flags);
51 }
52 int add(const char* str);
53 int insert(int index, const char* label, int shortcut, Fl_Callback *cb, void *user_data=0, int
54     flags=0);
55 int insert(int index, const char* label, const char* shortcut, Fl_Callback *cb, void *user_data=0, int
56     flags=0) {
57     return insert(index, label, fl_old_shortcut(shortcut), cb, user_data, flags);
58 }
59 void remove(int n);
60 void replace(int index, const char *name);
61 void clear();
62 int clear_submenu(int index);
63 void global() {};
64 void mode (int i, int fl) {
65     Fl_Menu_::mode(i, fl);
66     update();
67 }
68 int mode(int i)const { return Fl_Menu_::mode(i); }
69 void shortcut (int i, int s) { Fl_Menu_::shortcut(i, s); update(); }
70 void setonly (Fl_Menu_Item *item) { Fl_Menu_::setonly(item); update(); }
71 };
72 #else
73
74 #if FLTK_ABI_VERSION >= 10304
75 // NEW -- small class for update()

```

```

118 class FL_EXPORT Fl_Sys_Menu_Bar : public Fl_Menu_Bar {
119 public:
120   Fl_Sys_Menu_Bar(int x,int y,int w,int h,const char *l=0) : Fl_Menu_Bar(x,y,w,h,l) {}
121   inline void update() {}
122 };
123 #else
124 // OLD -- simple typedef
125 typedef Fl_Menu_Bar Fl_Sys_Menu_Bar;
126 #endif
127
128 #endif // defined(__APPLE__) || defined(FL_DOXYGEN)
129
130 #endif // Fl_Sys_Menu_Bar_H
131
132 //
133 // End of "$Id$".
134 //

```

32.121 Fl_Table.H

```

1 //
2 // "$Id$"
3 //
4 // Fl_Table -- A table widget
5 //
6 // Copyright 2002 by Greg Ercolano.
7 // Copyright (c) 2004 O'ksi'D
8 //
9 // This library is free software. Distribution and use rights are outlined in
10 // the file "COPYING" which should have been included with this file. If this
11 // file is missing or damaged, see the license at:
12 //
13 //   http://www.fltk.org/COPYING.php
14 //
15 // Please report all bugs and problems on the following page:
16 //
17 //   http://www.fltk.org/str.php
18 //
19
20 #ifndef _FL_TABLE_H
21 #define _FL_TABLE_H
22
23 #include <sys/types.h>
24 #include <string.h> // memcpy
25 #ifdef WIN32
26 #include <malloc.h> // WINDOWS: malloc/realloc
27 #else /*WIN32*/
28 #include <stdlib.h> // UNIX: malloc/realloc
29 #endif /*WIN32*/
30
31 #include <FL/Fl.H>
32 #include <FL/Fl_Group.H>
33 #include <FL/Fl_Scroll.H>
34 #include <FL/Fl_Box.H>
35 #include <FL/Fl_Scrollbar.H>
36
170 class FL_EXPORT Fl_Table : public Fl_Group {
171 public:
172   enum TableContext {
173     CONTEXT_NONE = 0,
174     CONTEXT_STARTPAGE = 0x01,
175     CONTEXT_ENDPAGE = 0x02,
176     CONTEXT_ROW_HEADER = 0x04,
177     CONTEXT_COL_HEADER = 0x08,
178     CONTEXT_CELL = 0x10,
179     CONTEXT_TABLE = 0x20,
180     CONTEXT_RC_RESIZE = 0x40
181   };
182
183 private:
184   int _rows, _cols; // total rows/cols
185   int _row_header_w; // width of row header
186   int _col_header_h; // height of column header
187   int _row_position; // last row_position set (not necessarily == toprow!)
188   int _col_position; // last col_position set (not necessarily == leftcol!)
189
190   char _row_header; // row header enabled?
191   char _col_header; // col header enabled?
192   char _row_resize; // row resizing enabled?
193   char _col_resize; // col resizing enabled?
194   int _row_resize_min; // row minimum resizing height (default=1)
195   int _col_resize_min; // col minimum resizing width (default=1)
196
197   // OPTIMIZATION: partial row/column redraw variables
198   int _redraw_toprow;

```

```

204 int _redraw_botrow;
205 int _redraw_leftcol;
206 int _redraw_rightcol;
207 Fl_Color _row_header_color;
208 Fl_Color _col_header_color;
209
210 int _auto_drag;
211 int _selecting;
212 #if FLTK_ABI_VERSION >= 10301
213 int _scrollbar_size;
214 #endif
215 #if FLTK_ABI_VERSION >= 10303
216 enum {
217     TABCELLNAV = 1<<0,
218 };
219 unsigned int flags_;
220 #endif
221
222 // An STL-ish vector without templates
223 class FL_EXPORT IntVector {
224     int *arr;
225     unsigned int _size;
226     void init() {
227         arr = NULL;
228         _size = 0;
229     }
230     void copy(int *newarr, unsigned int newsz) {
231         size(newsz);
232         memcpy(arr, newarr, newsz * sizeof(int));
233     }
234 public:
235     IntVector() { init(); } // CTOR
236     ~IntVector() { if ( arr ) free(arr); arr = NULL; } // DTOR
237     IntVector(IntVector&o) { init(); copy(o.arr, o._size); } // COPY CTOR
238     IntVector& operator=(IntVector&o) { // ASSIGN
239         init();
240         copy(o.arr, o._size);
241         return(*this);
242     }
243     int operator[](int x) const { return(arr[x]); }
244     int& operator[](int x) { return(arr[x]); }
245     unsigned int size() { return(_size); }
246     void size(unsigned int count) {
247         if ( count != _size ) {
248             arr = (int*)realloc(arr, count * sizeof(int));
249             _size = count;
250         }
251     }
252     int pop_back() { int tmp = arr[_size-1]; _size--; return(tmp); }
253     void push_back(int val) { unsigned int x = _size; size(_size+1); arr[x] = val; }
254     int back() { return(arr[_size-1]); }
255 };
256
257 IntVector _colwidths; // column widths in pixels
258 IntVector _rowheights; // row heights in pixels
259
260 Fl_Cursor _last_cursor; // last mouse cursor before changed to 'resize' cursor
261
262 // EVENT CALLBACK DATA
263 TableContext _callback_context; // event context
264 int _callback_row, _callback_col; // event row/col
265
266 // handle() state variables.
267 // Put here instead of local statics in handle(), so more
268 // than one Fl_Table can exist without crosstalk between them.
269 //
270 int _resizing_col; // column being dragged
271 int _resizing_row; // row being dragged
272 int _dragging_x; // starting x position for horiz drag
273 int _dragging_y; // starting y position for vert drag
274 int _last_row; // last row we FL_PUSH'ed
275
276 // Redraw single cell
277 void _redraw_cell(TableContext context, int R, int C);
278
279 void _start_auto_drag();
280 void _stop_auto_drag();
281 void _auto_drag_cb();
282 static void _auto_drag_cb2(void *d);
283
284 protected:
285 enum ResizeFlag {
286     RESIZE_NONE = 0,
287     RESIZE_COL_LEFT = 1,
288     RESIZE_COL_RIGHT = 2,
289     RESIZE_ROW_ABOVE = 3,
290     RESIZE_ROW_BELOW = 4

```

```

291 };
292
293 int table_w, table_h;           // table's virtual size (in pixels)
294 int toprow, botrow, leftcol, rightcol; // four corners of viewable table
295
296 // selection
297 int current_row, current_col;
298 int select_row, select_col;
299
300 // OPTIMIZATION: Precomputed scroll positions for the toprow/leftcol
301 int toprow_scrollpos;
302 int leftcol_scrollpos;
303
304 // Dimensions
305 int tix, tiy, tiw, tih;       // data table inner dimension xywh
306 int tox, toy, tow, toh;       // data table outer dimension xywh
307 int wix, wiy, wiw, wih;       // widget inner dimension xywh
308
309 Fl_Scroll *table;             // container for child fltk widgets (if any)
310 Fl_Scrollbar *vscrollbar;     // vertical scrollbar
311 Fl_Scrollbar *hscrollbar;     // horizontal scrollbar
312
313 // Fltk
314 int handle(int e);           // fltk handle() override
315
316 // Class maintenance
317 void recalc_dimensions();
318 void table_resized();        // table resized; recalc
319 void table_scrolled();       // table scrolled; recalc
320 void get_bounds(TableContext context, // return x/y/w/h bounds for context
321                int &X, int &Y, int &W, int &H);
322 void change_cursor(Fl_Cursor newcursor); // change mouse cursor to some other shape
323 TableContext cursor2rowcol(int &R, int &C, ResizeFlag &resizeflag);
324 // find r/c given current x/y event
325 int find_cell(TableContext context, // find cell's x/y/w/h given r/c
326               int R, int C, int &X, int &Y, int &W, int &H);
327 int row_col_clamp(TableContext context, int &R, int &C);
328 // clamp r/c to known universe
329
340 virtual void draw_cell(TableContext context, int R=0, int C=0,
341                        int X=0, int Y=0, int W=0, int H=0)
342 { } // overridden by deriving class
343
344 long row_scroll_position(int row); // find scroll position of row (in pixels)
345 long col_scroll_position(int col); // find scroll position of col (in pixels)
346
347 int is_fltk_container() { // does table contain fltk widgets?
348     return( Fl_Group::children() > 3 ); // (ie. more than box and 2 scrollbars?)
349 }
350
351 static void scroll_cb(Fl_Widget*,void*); // h/v scrollbar callback
352
353 void damage_zone(int r1, int c1, int r2, int c2, int r3 = 0, int c3 = 0);
354
355 void redraw_range(int topRow, int botRow, int leftCol, int rightCol) {
356     if ( _redraw_topleft == -1 ) {
357         // Initialize redraw range
358         _redraw_topleft = topRow;
359         _redraw_botrow = botRow;
360         _redraw_leftcol = leftCol;
361         _redraw_rightcol = rightCol;
362     } else {
363         // Extend redraw range
364         if ( topRow < _redraw_topleft ) _redraw_topleft = topRow;
365         if ( botRow > _redraw_botrow ) _redraw_botrow = botRow;
366         if ( leftCol < _redraw_leftcol ) _redraw_leftcol = leftCol;
367         if ( rightCol > _redraw_rightcol ) _redraw_rightcol = rightCol;
368     }
369
370     // Indicate partial redraw needed of some cells
371     damage(FL_DAMAGE_CHILD);
372 }
373
374 public:
375 Fl_Table(int X, int Y, int W, int H, const char *l=0);
376
377 ~Fl_Table();
378
379 virtual void clear() { rows(0); cols(0); table->clear(); }
380
381 // \todo: add topline(), middleline(), bottomline()
382
383 inline void table_box(Fl_Boxtype val) {
384     table->box(val);
385     table_resized();
386 }
387
388

```

```

510 inline Fl_Boxtype table_box( void ) {
511     return(table->box());
512 }
513
514 virtual void rows(int val); // set/get number of rows
515
516 inline int rows() {
517     return(_rows);
518 }
519
520 virtual void cols(int val); // set/get number of columns
521
522 inline int cols() {
523     return(_cols);
524 }
525
526 inline void visible_cells(int& r1, int& r2, int& c1, int& c2) {
527     r1 = toprow;
528     r2 = botrow;
529     c1 = leftcol;
530     c2 = rightcol;
531 }
532
533 int is_interactive_resize() {
534     return(_resizing_row != -1 || _resizing_col != -1);
535 }
536
537 inline int row_resize() {
538     return(_row_resize);
539 }
540
541 void row_resize(int flag) { // enable row resizing
542     _row_resize = flag;
543 }
544
545 inline int col_resize() {
546     return(_col_resize);
547 }
548
549 void col_resize(int flag) { // enable col resizing
550     _col_resize = flag;
551 }
552
553 inline int col_resize_min() { // column minimum resizing width
554     return(_col_resize_min);
555 }
556
557 void col_resize_min(int val) {
558     _col_resize_min = ( val < 1 ) ? 1 : val;
559 }
560
561 inline int row_resize_min() { // column minimum resizing width
562     return(_row_resize_min);
563 }
564
565 void row_resize_min(int val) {
566     _row_resize_min = ( val < 1 ) ? 1 : val;
567 }
568
569 inline int row_header() { // set/get row header enable flag
570     return(_row_header);
571 }
572
573 void row_header(int flag) {
574     _row_header = flag;
575     table_resized();
576     redraw();
577 }
578
579 inline int col_header() { // set/get col header enable flag
580     return(_col_header);
581 }
582
583 void col_header(int flag) {
584     _col_header = flag;
585     table_resized();
586     redraw();
587 }
588
589 inline void col_header_height(int height) { // set/get col header height
590     _col_header_h = height;
591     table_resized();
592     redraw();
593 }
594
595 inline int col_header_height() {
596     return(_col_header_h);
597 }

```

```

695
696 inline void row_header_width(int width) { // set/get row header width
697     _row_header_w = width;
700     table_resized();
701     redraw();
702 }
703
704
705 inline int row_header_width() {
706     return(_row_header_w);
707 }
708
709 inline void row_header_color(Fl_Color val) { // set/get row header color
710     _row_header_color = val;
711     redraw();
712 }
713
714
715 inline Fl_Color row_header_color() {
716     return(_row_header_color);
717 }
718
719
720 inline void col_header_color(Fl_Color val) { // set/get col header color
721     _col_header_color = val;
722     redraw();
723 }
724
725
726 inline Fl_Color col_header_color() {
727     return(_col_header_color);
728 }
729
730
731 void row_height(int row, int height); // set/get row height
732
733
734 inline int row_height(int row) {
735     return((row<0 || row>=(int)_rowheights.size()) ? 0 : _rowheights[row]);
736 }
737
738 void col_width(int col, int width); // set/get a column's width
739
740
741 inline int col_width(int col) {
742     return((col<0 || col>=(int)_colwidths.size()) ? 0 : _colwidths[col]);
743 }
744
745
746 void row_height_all(int height) { // set all row/col heights
747     for ( int r=0; r<rows(); r++ ) {
748         row_height(r, height);
749     }
750 }
751
752
753 void col_width_all(int width) {
754     for ( int c=0; c<cols(); c++ ) {
755         col_width(c, width);
756     }
757 }
758
759
760 void row_position(int row); // set/get table's current scroll position
761
762
763 void col_position(int col);
764
765
766 int row_position() { // current row position
767     return(_row_position);
768 }
769
770
771 int col_position() { // current col position
772     return(_col_position);
773 }
774
775
776 inline void top_row(int row) { // set/get top row (deprecated)
777     row_position(row);
778 }
779
780
781 inline int top_row() {
782     return(row_position());
783 }
784
785
786 int is_selected(int r, int c); // selected cell
787 void get_selection(int &row_top, int &col_left, int &row_bot, int &col_right);
788 void set_selection(int row_top, int col_left, int row_bot, int col_right);
789 int move_cursor(int R, int C, int shiftselect);
790 int move_cursor(int R, int C);
791
792
793 void resize(int X, int Y, int W, int H); // fltk resize() override
794 void draw(void); // fltk draw() override
795
796
797 // This crashes sortapp() during init.
798 // void box(Fl_Boxtype val) {
799 //     Fl_Group::box(val);
800 //     if ( table ) {
801 //         resize(x(), y(), w(), h());
802 //     }
803 // }

```

```

849 // }
850 // Fl_Boxtype box(void) const {
851 //     return(Fl_Group::box());
852 // }
853
854 // Child group
855 void init_sizes() {
856     table->init_sizes();
857     table->redraw();
858 }
859 void add(Fl_Widget& wgt) {
860     table->add(wgt);
861     if ( table->children() > 2 ) {
862         table->show();
863     } else {
864         table->hide();
865     }
866 }
867 void add(Fl_Widget* wgt) {
868     add(*wgt);
869 }
870 void insert(Fl_Widget& wgt, int n) {
871     table->insert(wgt,n);
872 }
873 void insert(Fl_Widget& wgt, Fl_Widget* w2) {
874     table->insert(wgt,w2);
875 }
876 void remove(Fl_Widget& wgt) {
877     table->remove(wgt);
878 }
879 void begin() {
880     table->begin();
881 }
882 void end() {
883     table->end();
884     // HACK: Avoid showing Fl_Scroll; seems to erase screen
885     //         causing unnecessary flicker, even if its box() is FL_NO_BOX.
886     //
887     if ( table->children() > 2 ) {
888         table->show();
889     } else {
890         table->hide();
891     }
892     Fl_Group::current(Fl_Group::parent());
893 }
894 Fl_Widget * const *array() {
895     return(table->array());
896 }
897
898 Fl_Widget *child(int n)const {
899     return(table->child(n));
900 }
901
902 int children()const {
903     return(table->children()-2);    // -2: skip Fl_Scroll's h/v scrollbar widgets
904 }
905 int find(const Fl_Widget *wgt)const {
906     return(table->find(wgt));
907 }
908 int find(const Fl_Widget &wgt)const {
909     return(table->find(wgt));
910 }
911
912 // CALLBACKS
913
914 int callback_row() {
915     return(_callback_row);
916 }
917
918 int callback_col() {
919     return(_callback_col);
920 }
921
922 TableContext callback_context() {
923     return(_callback_context);
924 }
925
926 void do_callback(TableContext context, int row, int col) {
927     _callback_context = context;
928     _callback_row = row;
929     _callback_col = col;
930     Fl_Widget::do_callback();
931 }
932
933 #ifdef FL_DOXYGEN
934 void when(Fl_When flags);
935 #endif
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```



```

1001 #ifdef FL_DOXYGEN
1079 void callback(Fl_Widget*, void*);
1080 #endif
1081
1082 #if FLTK_ABI_VERSION >= 10301
1083 // NEW
1093 int scrollbar_size()const {
1094     return(_scrollbar_size);
1095 }
1114 void scrollbar_size(int newSize) {
1115     if ( newSize != _scrollbar_size ) redraw();
1116     _scrollbar_size = newSize;
1117 }
1118 #endif
1119 #if FLTK_ABI_VERSION >= 10303
1133 void tab_cell_nav(int val) {
1134     if ( val ) flags_ |= TABCELLNAV;
1135     else flags_ &= ~TABCELLNAV;
1136 }
1137
1145 int tab_cell_nav()const {
1146     return(flags_ & TABCELLNAV ? 1 : 0);
1147 }
1148 #endif
1149 };
1150
1151 #endif /*_FL_TABLE_H*/
1152
1153 //
1154 // End of "$Id$".
1155 //

```

32.122 Fl_Table_Row.H

```

1 //
2 // "$Id$"
3 //
4
5 #ifndef _FL_TABLE_ROW_H
6 #define _FL_TABLE_ROW_H
7
8 //
9 // Fl_Table_Row -- A row oriented table widget
10 //
11 //     A class specializing in a table of rows.
12 //     Handles row-specific selection behavior.
13 //
14 // Copyright 2002 by Greg Ercolano.
15 //
16 // This library is free software. Distribution and use rights are outlined in
17 // the file "COPYING" which should have been included with this file. If this
18 // file is missing or damaged, see the license at:
19 //
20 //     http://www.fltk.org/COPYING.php
21 //
22 // Please report all bugs and problems to "erco at seriss dot com".
23 //
24
25 #include "Fl_Table.H"
26
44 class FL_EXPORT Fl_Table_Row : public Fl_Table {
45 public:
46     enum TableRowSelectMode {
47         SELECT_NONE,           // no selection allowed
48         SELECT_SINGLE,        // single row selection
49         SELECT_MULTI          // multiple row selection (default)
50     };
51 private:
52     // An STL-ish vector without templates
53     class FL_EXPORT CharVector {
54     char *arr;
55     int _size;
56     void init() {
57         arr = NULL;
58         _size = 0;
59     }
60     void copy(char *newarr, int newsz) {
61         size(newsz);
62         memcpy(arr, newarr, newsz * sizeof(char));
63     }
64 public:
65     CharVector() { // CTOR
66         init();
67     }
68     ~CharVector() { // DTOR

```

```

69     if ( arr ) free(arr);
70     arr = NULL;
71 }
72 CharVector(CharVector&o) {           // COPY CTOR
73     init();
74     copy(o.arr, o._size);
75 }
76 CharVector& operator=(CharVector&o) { // ASSIGN
77     init();
78     copy(o.arr, o._size);
79     return(*this);
80 }
81 char operator[](int x)const {
82     return(arr[x]);
83 }
84 char& operator[](int x) {
85     return(arr[x]);
86 }
87 int size() {
88     return(_size);
89 }
90 void size(int count) {
91     if ( count != _size ) {
92         arr = (char*)realloc(arr, count * sizeof(char));
93         _size = count;
94     }
95 }
96 char pop_back() {
97     char tmp = arr[_size-1];
98     _size--;
99     return(tmp);
100 }
101 void push_back(char val) {
102     int x = _size;
103     size(_size+1);
104     arr[x] = val;
105 }
106 char back() {
107     return(arr[_size-1]);
108 }
109 };
110 CharVector _rowselect;           // selection flag for each row
111
112 // handle() state variables.
113 // Put here instead of local statics in handle(), so more
114 // than one instance can exist without crosstalk between.
115 //
116 int _dragging_select;           // dragging out a selection?
117 int _last_row;
118 int _last_y;                    // last event's Y position
119 int _last_push_x;              // last PUSH event's X position
120 int _last_push_y;              // last PUSH event's Y position
121
122 TableRowSelectMode _selectmode;
123
124 protected:
125 int handle(int event);
126 int find_cell(TableContext context,           // find cell's x/y/w/h given r/c
127     int R, int C, int &X, int &Y, int &W, int &H) {
128     return(Fl_Table::find_cell(context, R, C, X, Y, W, H));
129 }
130
131 public:
132 Fl_Table_Row(int X, int Y, int W, int H, const char *l=0) : Fl_Table(X,Y,W,H,l) {
133     _dragging_select = 0;
134     _last_row        = -1;
135     _last_y          = -1;
136     _last_push_x     = -1;
137     _last_push_y     = -1;
138     _selectmode      = SELECT_MULTI;
139 }
140
141 ~Fl_Table_Row() { }
142
143 void rows(int val);             // set number of rows
144 int rows() {                    // get number of rows
145     return(Fl_Table::rows());
146 }
147
148 void type(TableRowSelectMode val); // set selection mode
149 TableRowSelectMode type()const { // get selection mode
150     return(_selectmode);
151 }
152
153 int row_selected(int row);      // is row selected? (0=no, 1=yes, -1=range err)
154
155

```

```

181 int select_row(int row, int flag=1); // select state for row: flag:0=off, 1=on, 2=toggle
182 // returns: 0=no change, 1=changed, -1=range err
183
184 void select_all_rows(int flag=1); // all rows to a known state
185
186 void clear() {
187     rows(0); // implies clearing selection
188     cols(0);
189     Fl_Table::clear(); // clear the table
190 }
191 };
192
193 #endif /*_FL_TABLE_ROW_H*/
194
195 //
196 // End of "$Id$".
197 //

```

32.123 Fl_Tabs.H

```

1 //
2 // "$Id$"
3 //
4 // Tab header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Tabs widget . */
21
22 #ifndef Fl_Tabs_H
23 #define Fl_Tabs_H
24
25 #include "Fl_Group.H"
26
27 class FL_EXPORT Fl_Tabs : public Fl_Group {
28 #if FLTK_ABI_VERSION >= 10304
29     // NEW (nothing)
30 #else
31     // OLD (maintained for ABI compat)
32     Fl_Widget *value_; // NOTE: this member no longer used -- STR #3169
33 #endif
34     Fl_Widget *push_;
35     int *tab_pos; // array of x-offsets of tabs per child + 1
36     int *tab_width; // array of widths of tabs per child + 1
37     int tab_count; // array size
38     int tab_positions(); // allocate and calculate tab positions
39     void clear_tab_positions();
40     int tab_height();
41     void draw_tab(int x1, int x2, int W, int H, Fl_Widget* o, int sel=0);
42 protected:
43     void redraw_tabs();
44     void draw();
45
46 public:
47     int handle(int);
48     Fl_Widget *value();
49     int value(Fl_Widget *);
50     Fl_Widget *push()const {return push_;}
51     int push(Fl_Widget *);
52     Fl_Tabs(int,int,int,int,const char * = 0);
53     Fl_Widget *which(int event_x, int event_y);
54     ~Fl_Tabs();
55     void client_area(int &rx, int &ry, int &rw, int &rh, int tabh=0);
56 };
57
58 #endif
59
60 //
61 // End of "$Id$".
62 //

```

32.124 Fl_Text_Buffer.H

```

1 //
2 // "$Id$"
3 //
4 // Header file for Fl_Text_Buffer class.
5 //
6 // Copyright 2001-2016 by Bill Spitzak and others.
7 // Original code Copyright Mark Edel. Permission to distribute under
8 // the LGPL for the FLTK library granted by Mark Edel.
9 //
10 // Please report all bugs and problems on the following page:
11 //
12 //     http://www.fltk.org/str.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Text_Buffer, Fl_Text_Selection widget . */
21
22 #ifndef FL_TEXT_BUFFER_H
23 #define FL_TEXT_BUFFER_H
24
25
26 #undef ASSERT_UTF8
27
28 #ifdef ASSERT_UTF8
29 # include <assert.h>
30 # define IS_UTF8_ALIGNED(a) if (a && *a) assert(fl_utf8len(*(a))>0);
31 # define IS_UTF8_ALIGNED2(a, b) if (b>=0 && b<a->length()) assert(fl_utf8len(a->byte_at(b))>0);
32 #else
33 # define IS_UTF8_ALIGNED(a)
34 # define IS_UTF8_ALIGNED2(a, b)
35 #endif
36
37
38 /*
39 "character size" is the size of a UTF-8 character in bytes
40 "character width" is the width of a Unicode character in pixels
41 "column" was originally defined as a character offset from the left margin.
42 It was identical to the byte offset. In UTF-8, we have neither a byte offset
43 nor truly fixed width fonts (*). Column could be a pixel value multiplied with
44 an average character width (which is a bearable approximation).
45
46 * in Unicode, there are no fixed width fonts! Even if the ASCII characters may
47 happen to be all the same width in pixels, Chinese characters surely are not.
48 There are plenty of exceptions, like ligatures, that make special handling of
49 "fixed" character widths a nightmare. I decided to remove all references to
50 fixed fonts and see "columns" as a multiple of the average width of a
51 character in the main font.
52 - Matthias
53 */
54
55
56 /* Maximum length in characters of a tab or control character expansion
57 of a single buffer character */
58 #define FL_TEXT_MAX_EXP_CHAR_LEN 20
59
60 #include "Fl_Export.H"
61
62
63 class FL_EXPORT Fl_Text_Selection {
64     friend class Fl_Text_Buffer;
65
66 public:
67     void set(int start, int end);
68
69     void update(int pos, int nDeleted, int nInserted);
70
71     int start()const { return mStart; }
72
73     int end()const { return mEnd; }
74
75     bool selected()const { return mSelected; }
76
77     void selected(bool b) { mSelected = b; }
78
79     int includes(int pos) const;
80
81     int position(int* start, int* end) const;
82
83 protected:
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131

```

```

132 int mStart;
133 int mEnd;
134 bool mSelected;
135 };
136
137
138 typedef void (*Fl_Text_Modify_Cb)(int pos, int nInserted, int nDeleted,
139                                 int nRestyled, const char* deletedText,
140                                 void* cbArg);
141
142
143 typedef void (*Fl_Text_Predelete_Cb)(int pos, int nDeleted, void* cbArg);
144
145
146 class FL_EXPORT Fl_Text_Buffer {
147 public:
148     Fl_Text_Buffer(int requestedSize = 0, int preferredGapSize = 1024);
149
150     ~Fl_Text_Buffer();
151
152     int length()const { return mLength; }
153
154     char* text() const;
155
156     void text(const char* text);
157
158     char* text_range(int start, int end) const;
159
160     unsigned int char_at(int pos) const;
161
162     char byte_at(int pos) const;
163
164     const char *address(int pos)const
165 { return (pos < mGapStart) ? mBuf+pos : mBuf+pos+mGapEnd-mGapStart; }
166
167     char *address(int pos)
168 { return (pos < mGapStart) ? mBuf+pos : mBuf+pos+mGapEnd-mGapStart; }
169
170     void insert(int pos, const char* text);
171
172     void append(const char* t) { insert(length(), t); }
173
174     void remove(int start, int end);
175
176     void replace(int start, int end, const char *text);
177
178     void copy(Fl_Text_Buffer* fromBuf, int fromStart, int fromEnd, int toPos);
179
180     int undo(int *cp=0);
181
182     void canUndo(char flag=1);
183
184     int insertfile(const char *file, int pos, int buflen = 128*1024);
185
186     int appendfile(const char *file, int buflen = 128*1024)
187 { return insertfile(file, length(), buflen); }
188
189     int loadfile(const char *file, int buflen = 128*1024)
190 { select(0, length()); remove_selection(); return appendfile(file, buflen); }
191
192     int outputfile(const char *file, int start, int end, int buflen = 128*1024);
193
194     int savefile(const char *file, int buflen = 128*1024)
195 { return outputfile(file, 0, length(), buflen); }
196
197     int tab_distance()const { return mTabDist; }
198
199     void tab_distance(int tabDist);
200
201     void select(int start, int end);
202
203     int selected()const { return mPrimary.selected(); }
204
205     void unselect();
206
207     int selection_position(int* start, int* end);
208
209     char* selection_text();
210
211     void remove_selection();
212
213     void replace_selection(const char* text);
214
215     void secondary_select(int start, int end);
216
217     int secondary_selected() { return mSecondary.selected(); }

```

```

404
408 void secondary_unselect();
409
413 int secondary_selection_position(int* start, int* end);
414
420 char* secondary_selection_text();
421
426 void remove_secondary_selection();
427
432 void replace_secondary_selection(const char* text);
433
437 void highlight(int start, int end);
438
444 int highlight() { return mHighlight.selected(); }
445
449 void unhighlight();
450
454 int highlight_position(int* start, int* end);
455
461 char* highlight_text();
462
474 void add_modify_callback(Fl_Text_Modify_Cb bufModifiedCB, void* cbArg);
475
479 void remove_modify_callback(Fl_Text_Modify_Cb bufModifiedCB, void* cbArg);
480
485 void call_modify_callbacks() { call_modify_callbacks(0, 0, 0, 0, 0); }
486
490 void add_predelete_callback(Fl_Text_Predelete_Cb bufPredelCB, void* cbArg);
491
496 void remove_predelete_callback(Fl_Text_Predelete_Cb predelCB, void* cbArg);
497
502 void call_predelete_callbacks() { call_predelete_callbacks(0, 0); }
503
512 char* line_text(int pos) const;
513
519 int line_start(int pos) const;
520
528 int line_end(int pos) const;
529
535 int word_start(int pos) const;
536
542 int word_end(int pos) const;
543
551 int count_displayed_characters(int lineStartPos, int targetPos) const;
552
562 int skip_displayed_characters(int lineStartPos, int nChars);
563
568 int count_lines(int startPos, int endPos) const;
569
574 int skip_lines(int startPos, int nLines);
575
582 int rewind_lines(int startPos, int nLines);
583
598 int findchar_forward(int startPos, unsigned searchChar, int* foundPos) const;
599
613 int findchar_backward(int startPos, unsigned int searchChar, int* foundPos) const;
614
626 int search_forward(int startPos, const char* searchString, int* foundPos,
627                    int matchCase = 0) const;
628
640 int search_backward(int startPos, const char* searchString, int* foundPos,
641                    int matchCase = 0) const;
642
646 const Fl_Text_Selection* primary_selection()const { return &mPrimary; }
647
651 Fl_Text_Selection* primary_selection() { return &mPrimary; }
652
656 const Fl_Text_Selection* secondary_selection()const { return &mSecondary; }
657
661 const Fl_Text_Selection* highlight_selection()const { return &mHighlight; }
662
667 int prev_char(int ix) const;
668 int prev_char_clipped(int ix) const;
669
674 int next_char(int ix) const;
675 int next_char_clipped(int ix) const;
676
680 int utf8_align(int) const;
681
685 int input_file_was_transcoded;
686
690 static const char* file_encoding_warning_message;
691
701 void (*transcoding_warning_action)(Fl_Text_Buffer*);
702
703 protected:
704

```

```

709 void call_modify_callbacks(int pos, int nDeleted, int nInserted,
710                           int nRestyled, const char* deletedText) const;
711
716 void call_predelete_callbacks(int pos, int nDeleted) const;
717
727 int insert_(int pos, const char* text);
728
735 void remove_(int start, int end);
736
741 void redisplay_selection(Fl_Text_Selection* oldSelection,
742                         Fl_Text_Selection* newSelection) const;
743
747 void move_gap(int pos);
748
753 void reallocate_with_gap(int newGapStart, int newGapLen);
754
755 char* selection_text_(Fl_Text_Selection* sel) const;
756
760 void remove_selection_(Fl_Text_Selection* sel);
761
765 void replace_selection_(Fl_Text_Selection* sel, const char* text);
766
770 void update_selections(int pos, int nDeleted, int nInserted);
771
772 Fl_Text_Selection mPrimary;
773 Fl_Text_Selection mSecondary;
774 Fl_Text_Selection mHighlight;
775 int mLength;
776 char* mBuf;
777 int mGapStart;
778 int mGapEnd;
779 // The hardware tab distance used by all displays for this buffer,
780 // and used in computing offsets for rectangular selection operations.
781 int mTabDist;
782 int mNModifyProcs;
783 Fl_Text_Modify_Cb *mModifyProcs;
784 void** mCbArgs;
785 int mNPredelateProcs;
786 Fl_Text_Predelate_Cb *mPredelateProcs;
787 void **mPredelateCbArgs;
788 int mCursorPosHint;
789 char mCanUndo;
790 int mPreferredGapSize;
791 };
800
801 #endif
802
803 //
804 // End of "$Id$".
805 //

```

32.125 Fl_Text_Display.H

```

1 //
2 // "$Id$"
3 //
4 // Header file for Fl_Text_Display class.
5 //
6 // Copyright 2001-2016 by Bill Spitzak and others.
7 // Original code Copyright Mark Edel. Permission to distribute under
8 // the LGPL for the FLTK library granted by Mark Edel.
9 //
10 // This library is free software. Distribution and use rights are outlined in
11 // the file "COPYING" which should have been included with this file. If this
12 // file is missing or damaged, see the license at:
13 //
14 // http://www.fltk.org/COPYING.php
15 //
16 // Please report all bugs and problems on the following page:
17 //
18 // http://www.fltk.org/str.php
19 //
20
21 /* \file
22 Fl_Text_Display widget . */
23
24 #ifndef FL_TEXT_DISPLAY_H
25 #define FL_TEXT_DISPLAY_H
26
27 #include "fl_draw.H"
28 #include "Fl_Group.H"
29 #include "Fl_Widget.H"
30 #include "Fl_Scrollbar.H"
31 #include "Fl_Text_Buffer.H"
32

```

```

82 class FL_EXPORT Fl_Text_Display: public Fl_Group {
83
84 public:
85
86     enum {
87         NORMAL_CURSOR,
88         CARET_CURSOR,
89         DIM_CURSOR,
90         BLOCK_CURSOR,
91         HEAVY_CURSOR,
92         SIMPLE_CURSOR
93     };
94
95     enum {
96         CURSOR_POS,
97         CHARACTER_POS
98     };
99
100    enum {
101        DRAG_NONE = -2,
102        DRAG_START_DND = -1,
103        DRAG_CHAR = 0,
104        DRAG_WORD = 1,
105        DRAG_LINE = 2
106    };
107
108    enum {
109        WRAP_NONE,
110        WRAP_AT_COLUMN,
111        WRAP_AT_PIXEL,
112        WRAP_AT_BOUNDS
113    };
114
115    friend void fl_text_drag_me(int pos, Fl_Text_Display* d);
116
117    typedef void (*Unfinished_Style_Cb)(int, void *);
118
119    struct Style_Table_Entry {
120        Fl_Color    color;
121        Fl_Font     font;
122        Fl_Fontsize size;
123        unsigned    attr;
124    };
125
126    Fl_Text_Display(int X, int Y, int W, int H, const char *l = 0);
127    ~Fl_Text_Display();
128
129    virtual int handle(int e);
130
131    void buffer(Fl_Text_Buffer* buf);
132
133    void buffer(Fl_Text_Buffer& buf) { buffer(&buf); }
134
135    Fl_Text_Buffer* buffer()const { return mBuffer; }
136
137    void redisplay_range(int start, int end);
138    void scroll(int topLineNum, int horizOffset);
139    void insert(const char* text);
140    void overstrike(const char* text);
141    void insert_position(int newPos);
142
143    int insert_position()const { return mCursorPos; }
144    int position_to_xy(int pos, int* x, int* y) const;
145
146    int in_selection(int x, int y) const;
147    void show_insert_position();
148
149    int move_right();
150    int move_left();
151    int move_up();
152    int move_down();
153    int count_lines(int start, int end, bool start_pos_is_line_start) const;
154    int line_start(int pos) const;
155    int line_end(int startPos, bool startPosIsLineStart) const;
156    int skip_lines(int startPos, int nLines, bool startPosIsLineStart);
157    int rewind_lines(int startPos, int nLines);
158    void next_word(void);
159    void previous_word(void);
160
161    void show_cursor(int b = 1);
162
163    void hide_cursor() { show_cursor(0); }
164
165    void cursor_style(int style);
166
167    Fl_Color cursor_color()const {return mCursor_color;}
168
169

```



```

223 void cursor_color(Fl_Color n) {mCursor_color = n;}
224
229 int scrollbar_width()const { return scrollbar_width_; }
230
235 void scrollbar_width(int W) { scrollbar_width_ = W; }
236
241 Fl_Align scrollbar_align()const { return scrollbar_align_; }
242
247 void scrollbar_align(Fl_Align a) { scrollbar_align_ = a; }
248
254 int word_start(int pos)const { return buffer()->word_start(pos); }
255
261 int word_end(int pos)const { return buffer()->word_end(pos); }
262
263
264 void highlight_data(Fl_Text_Buffer *styleBuffer,
265                    const Style_Table_Entry *styleTable,
266                    int nStyles, char unfinishedStyle,
267                    Unfinished_Style_Cb unfinishedHighlightCB,
268                    void *cbArg);
269
270 int position_style(int lineStartPos, int lineLen, int lineIndex) const;
271
277 int shortcut()const {return shortcut_;}
278
284 void shortcut(int s) {shortcut_ = s;}
285
290 Fl_Font textfont()const {return textfont_;}
291
296 void textfont(Fl_Font s) {textfont_ = s; mColumnScale = 0;}
297
302 Fl_Fontsize textsize()const {return textsize_;}
303
308 void textsize(Fl_Fontsize s) {textsize_ = s; mColumnScale = 0;}
309
314 Fl_Color textcolor()const {return textcolor_;}
315
320 void textcolor(Fl_Color n) {textcolor_ = n;}
321
322 int wrapped_column(int row, int column) const;
323 int wrapped_row(int row) const;
324 void wrap_mode(int wrap, int wrap_margin);
325
326 virtual void resize(int X, int Y, int W, int H);
327
333 double x_to_col(double x) const;
334
341 double col_to_x(double col) const;
342
343 void linenummer_width(int width);
344 int linenummer_width() const;
345 void linenummer_font(Fl_Font val);
346 Fl_Font linenummer_font() const;
347 void linenummer_size(Fl_Fontsize val);
348 Fl_Fontsize linenummer_size() const;
349 void linenummer_fgcolor(Fl_Color val);
350 Fl_Color linenummer_fgcolor() const;
351 void linenummer_bgcolor(Fl_Color val);
352 Fl_Color linenummer_bgcolor() const;
353 void linenummer_align(Fl_Align val);
354 Fl_Align linenummer_align() const;
355 void linenummer_format(const char* val);
356 const char* linenummer_format() const;
357
358 protected:
359 // Most (all?) of this stuff should only be called from resize() or
360 // draw().
361 // Anything with "vline" indicates that it deals with currently
362 // visible lines.
363
364 virtual void draw();
365 void draw_text(int X, int Y, int W, int H);
366 void draw_range(int start, int end);
367 void draw_cursor(int, int);
368
369 void draw_string(int style, int x, int y, int toX, const char *string,
370                int nChars) const;
371
372 void draw_vline(int visLineNum, int leftClip, int rightClip,
373               int leftCharIndex, int rightCharIndex);
374
375 int find_x(const char *s, int len, int style, int x) const;
376
377 enum {
378     DRAW_LINE,
379     FIND_INDEX,
380     FIND_INDEX_FROM_ZERO,

```

```

381     GET_WIDTH
382 };
383
384 int handle_vline(int mode,
385                 int lineStart, int lineLen, int leftChar, int rightChar,
386                 int topClip, int bottomClip,
387                 int leftClip, int rightClip) const;
388
389 void draw_line_numbers(bool clearAll);
390
391 void clear_rect(int style, int x, int y, int width, int height) const;
392 void display_insert();
393
394 void offset_line_starts(int newTopLineNum);
395
396 void calc_line_starts(int startLine, int endLine);
397
398 void update_line_starts(int pos, int charsInserted, int charsDeleted,
399                       int linesInserted, int linesDeleted, int *scrolled);
400
401 void calc_last_char();
402
403 int position_to_line( int pos, int* lineNum ) const;
404 double string_width(const char* string, int length, int style) const;
405
406 static void scroll_timer_cb(void*);
407
408 static void buffer_predelete_cb(int pos, int nDeleted, void* cbArg);
409 static void buffer_modified_cb(int pos, int nInserted, int nDeleted,
410                               int nRestyled, const char* deletedText,
411                               void* cbArg);
412
413 static void h_scrollbar_cb(Fl_Scrollbar* w, Fl_Text_Display* d);
414 static void v_scrollbar_cb( Fl_Scrollbar* w, Fl_Text_Display* d);
415 void update_v_scrollbar();
416 void update_h_scrollbar();
417 int measure_vline(int visLineNum) const;
418 int longest_vline() const;
419 int empty_vlines() const;
420 int vline_length(int visLineNum) const;
421 int xy_to_position(int x, int y, int PosType = CHARACTER_POS) const;
422
423 void xy_to_rowcol(int x, int y, int* row, int* column,
424                 int PosType = CHARACTER_POS) const;
425 void maintain_absolute_top_line_number(int state);
426 int get_absolute_top_line_number() const;
427 void absolute_top_line_number(int oldFirstChar);
428 int maintaining_absolute_top_line_number() const;
429 void reset_absolute_top_line_number();
430 int position_to_linecol(int pos, int* lineNum, int* column) const;
431 int scroll_(int topLineNum, int horizOffset);
432
433 void extend_range_for_styles(int* start, int* end);
434
435 void find_wrap_range(const char *deletedText, int pos, int nInserted,
436                    int nDeleted, int *modRangeStart, int *modRangeEnd,
437                    int *linesInserted, int *linesDeleted);
438 void measure_deleted_lines(int pos, int nDeleted);
439 void wrapped_line_counter(Fl_Text_Buffer *buf, int startPos, int maxPos,
440                          int maxLines, bool startPosIsLineStart,
441                          int styleBufOffset, int *retPos, int *retLines,
442                          int *retLineStart, int *retLineEnd,
443                          bool countLastLineMissingNewLine = true) const;
444 void find_line_end(int pos, bool start_pos_is_line_start, int *lineEnd,
445                  int *nextLineStart) const;
446 double measure_proportional_character(const char *s, int colNum, int pos) const;
447 int wrap_uses_character(int lineEndPos) const;
448
449 int damage_range1_start, damage_range1_end;
450 int damage_range2_start, damage_range2_end;
451 int mCursorPos;
452 int mCursorOn;
453 int mCursorOldY;           /* Y pos.  of cursor for blanking */
454 int mCursorToHint;        /* Tells the buffer modified callback
455 where to move the cursor, to reduce
456 the number of redraw calls */
457 int mCursorStyle;         /* One of enum cursorStyles above */
458 int mCursorPreferredXPos; /* Pixel position for vert.  cursor movement */
459 int mNVisibleLines;       /* # of visible (displayed) lines */
460 int mNBufferLines;        /* # of newlines in the buffer */
461 Fl_Text_Buffer* mBuffer;   /* Contains text to be displayed */
462 Fl_Text_Buffer* mStyleBuffer; /* Optional parallel buffer containing
463 color and font information */
464 int mFirstChar, mLastChar; /* Buffer positions of first and last
465 displayed character (lastChar points
466 either to a newline or one character
467 beyond the end of the buffer) */

```

```

468 int mContinuousWrap;          /* Wrap long lines when displaying */
469 int mWrapMarginPix;          /* Margin in # of pixels for
470 wrapping in continuousWrap mode */
471 int* mLineStarts;
472 int mTopLineNum;             /* Line number of top displayed line
473 of file (first line of file is 1) */
474 int mAbsTopLineNum;         /* In continuous wrap mode, the line
475 number of the top line if the text
476 were not wrapped (note that this is
477 only maintained as needed). */
478 int mNeedAbsTopLineNum;     /* Externally settable flag to continue
479 maintaining absTopLineNum even if
480 it isn't needed for line # display */
481 int mHorizOffset;           /* Horizontal scroll pos. in pixels */
482 int mTopLineNumHint;       /* Line number of top displayed line
483 of file (first line of file is 1) */
484 int mHorizOffsetHint;       /* Horizontal scroll pos. in pixels */
485 int mNStyles;               /* Number of entries in styleTable */
486 const Style_Table_Entry *mStyleTable; /* Table of fonts and colors for
487 coloring/syntax-highlighting */
488 char mUnfinishedStyle;     /* Style buffer entry which triggers
489 on-the-fly reparsing of region */
490 Unfinished_Style_Cb mUnfinishedHighlightCB; /* Callback to parse "unfinished" */
491 /* regions */
492 void* mHighlightCBArg;     /* Arg to unfinishedHighlightCB */
493
494 int mMaxsize;
495
496 int mSuppressResync;        /* Suppress resynchronization of line
497 starts during buffer updates */
498 int mNLinesDeleted;        /* Number of lines deleted during
499 buffer modification (only used
500 when resynchronization is suppressed) */
501 int mModifyingTabDistance; /* Whether tab distance is being
502 modified */
503
504 mutable double mColumnScale; /* Width in pixels of an average character. This
505 value is calculated as needed (lazy eval); it
506 needs to be mutable so that it can be calculated
507 within a method marked as "const" */
508
509 Fl_Color mCursor_color;
510
511 Fl_Scrollbar* mHScrollBar;
512 Fl_Scrollbar* mVScrollBar;
513 int scrollbar_width_;
514 Fl_Align scrollbar_align_;
515 int dragPos, dragType, dragging;
516 int display_insert_position_hint;
517 struct { int x, y, w, h; } text_area;
518
519 int shortcut_;
520
521 Fl_Font textfont_;
522 Fl_Fontsize textsize_;
523 Fl_Color textcolor_;
524
525 // Line number margin and width
526 int mLineNumLeft, mLineNumWidth;
527
528 // Line number font/colors
529 #if FLTK_ABI_VERSION >= 10303
530 Fl_Font    linenum_font_;
531 Fl_Fontsize linenum_size_;
532 Fl_Color   linenum_fgcolor_;
533 Fl_Color   linenum_bgcolor_;
534 Fl_Align   linenum_align_;
535 const char* linenum_format_;
536 #endif
537 };
538
539 #endif
540
541 //
542 // End of "$Id$".
543 //

```

32.126 Fl_Text_Editor.H

```

1 //
2 // "$Id$"
3 //
4 // Header file for Fl_Text_Editor class.
5 //
6 // Copyright 2001-2010 by Bill Spitzak and others.

```

```

7 // Original code Copyright Mark Edel.  Permission to distribute under
8 // the LGPL for the FLTK library granted by Mark Edel.
9 //
10 // This library is free software.  Distribution and use rights are outlined in
11 // the file "COPYING" which should have been included with this file.  If this
12 // file is missing or damaged, see the license at:
13 //
14 //     http://www.fltk.org/COPYING.php
15 //
16 // Please report all bugs and problems on the following page:
17 //
18 //     http://www.fltk.org/str.php
19 //
20
21 /* \file
22 Fl_Text_Editor widget . */
23
24
25 #ifndef FL_TEXT_EDITOR_H
26 #define FL_TEXT_EDITOR_H
27
28 #include "Fl_Text_Display.H"
29
30 // key will match in any state
31 #define FL_TEXT_EDITOR_ANY_STATE (-1L)
32
33 class FL_EXPORT Fl_Text_Editor : public Fl_Text_Display {
34 public:
35     typedef int (*Key_Func)(int key, Fl_Text_Editor* editor);
36
37     struct Key_Binding {
38         int         key;
39         int         state;
40         Key_Func    function;
41         Key_Binding* next;
42     };
43
44     Fl_Text_Editor(int X, int Y, int W, int H, const char* l = 0);
45     ~Fl_Text_Editor() { remove_all_key_bindings(); }
46     virtual int handle(int e);
47     void insert_mode(int b) { insert_mode_ = b; }
48     int insert_mode() { return insert_mode_; }
49
50 #if FLTK_ABI_VERSION >= 10304
51     void tab_nav(int val);
52     int tab_nav() const;
53 #endif
54
55     void add_key_binding(int key, int state, Key_Func f, Key_Binding** list);
56     void add_key_binding(int key, int state, Key_Func f)
57     { add_key_binding(key, state, f, &key_bindings); }
58     void remove_key_binding(int key, int state, Key_Binding** list);
59     void remove_key_binding(int key, int state)
60     { remove_key_binding(key, state, &key_bindings); }
61     void remove_all_key_bindings(Key_Binding** list);
62     void remove_all_key_bindings() { remove_all_key_bindings(&key_bindings); }
63     void add_default_key_bindings(Key_Binding** list);
64 #if FLTK_ABI_VERSION < 10304
65     // OLD: non-const
66     Key_Func bound_key_function(int key, int state, Key_Binding* list);
67     Key_Func bound_key_function(int key, int state)
68     { return bound_key_function(key, state, key_bindings); }
69 #else
70     // NEW: const (STR#3306)
71     Key_Func bound_key_function(int key, int state, Key_Binding* list) const;
72     Key_Func bound_key_function(int key, int state) const
73     { return bound_key_function(key, state, key_bindings); }
74 #endif
75     void default_key_function(Key_Func f) { default_key_function_ = f; }
76
77     // functions for the built in default bindings
78     static int kf_default(int c, Fl_Text_Editor* e);
79     static int kf_ignore(int c, Fl_Text_Editor* e);
80     static int kf_backspace(int c, Fl_Text_Editor* e);
81     static int kf_enter(int c, Fl_Text_Editor* e);
82     static int kf_move(int c, Fl_Text_Editor* e);
83     static int kf_shift_move(int c, Fl_Text_Editor* e);
84     static int kf_ctrl_move(int c, Fl_Text_Editor* e);
85     static int kf_c_s_move(int c, Fl_Text_Editor* e);
86     static int kf_meta_move(int c, Fl_Text_Editor* e);
87     static int kf_m_s_move(int c, Fl_Text_Editor* e);
88     static int kf_home(int, Fl_Text_Editor* e);
89     static int kf_end(int c, Fl_Text_Editor* e);
90     static int kf_left(int c, Fl_Text_Editor* e);
91     static int kf_up(int c, Fl_Text_Editor* e);
92     static int kf_right(int c, Fl_Text_Editor* e);
93     static int kf_down(int c, Fl_Text_Editor* e);

```

```

119     static int kf_page_up(int c, Fl_Text_Editor* e);
120     static int kf_page_down(int c, Fl_Text_Editor* e);
121     static int kf_insert(int c, Fl_Text_Editor* e);
122     static int kf_delete(int c, Fl_Text_Editor* e);
123     static int kf_copy(int c, Fl_Text_Editor* e);
124     static int kf_cut(int c, Fl_Text_Editor* e);
125     static int kf_paste(int c, Fl_Text_Editor* e);
126     static int kf_select_all(int c, Fl_Text_Editor* e);
127     static int kf_undo(int c, Fl_Text_Editor* e);
128
129     protected:
130         int handle_key();
131         void maybe_do_callback();
132
133 #ifndef FL_DOXYGEN
134     int insert_mode_;
135     Key_Binding* key_bindings;
136 #endif
137
138     static Key_Binding* global_key_bindings;
139
140 #ifndef FL_DOXYGEN
141     Key_Func default_key_function_;
142 #endif
143 };
144 #endif
145
146 //
147 // End of "$Id$".
148 //
149

```

32.127 Fl_Tile.H

```

1 //
2 // "$Id$"
3 //
4 // Tile header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2016 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 #ifndef Fl_Tile_H
20 #define Fl_Tile_H
21
22 #include "Fl_Group.H"
23
24 /*
25 The Fl_Tile class lets you resize its children by dragging
26 the border between them.
27 */
28
29 class FL_EXPORT Fl_Tile : public Fl_Group {
30 public:
31     int handle(int event);
32     Fl_Tile(int X, int Y, int W, int H, const char *L=0);
33     void resize(int X, int Y, int W, int H);
34     void position(int oldx, int oldy, int newx, int newy);
35 };
36
37 #endif
38
39 //
40 // End of "$Id$".
41 //

```

32.128 Fl_Tiled_Image.H

```

1 //
2 // "$Id$"
3 //

```

```

4 // Tiled image header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2015 by Bill Spitzak and others.
7 //
8 // This library is free software.  Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file.  If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Tiled_Image widget . */
21
22 #ifndef Fl_Tiled_Image_H
23 # define Fl_Tiled_Image_H
24
25 # include "Fl_Image.H"
26
27
28
29
30
31
32
33
34 class FL_EXPORT Fl_Tiled_Image : public Fl_Image {
35 protected:
36
37     Fl_Image      *image_;           // The image that is tiled
38     int           alloc_image_;     // Did we allocate this image?
39
40 public:
41
42     Fl_Tiled_Image(Fl_Image *i, int W = 0, int H = 0);
43     virtual ~Fl_Tiled_Image();
44
45     virtual Fl_Image *copy(int W, int H);
46     Fl_Image *copy() { return copy(w(), h()); }
47     virtual void color_average(Fl_Color c, float i);
48     virtual void desaturate();
49     virtual void draw(int X, int Y, int W, int H, int cx, int cy);
50     void draw(int X, int Y) { draw(X, Y, w(), h(), 0, 0); }
51     Fl_Image *image() { return image_; }
52 };
53
54
55 #endif // !Fl_Tiled_Image_H
56
57 //
58 // End of "$Id$"
59 //

```

32.129 Fl_Timer.H

```

1 //
2 // "$Id$"
3 //
4 // Timer header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software.  Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file.  If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Timer widget . */
21
22 #ifndef Fl_Timer_H
23 #define Fl_Timer_H
24
25 #ifndef Fl_Widget_H
26 #include "Fl_Widget.H"
27 #endif
28
29 // values for type():
30 #define FL_NORMAL_TIMER      0
31 #define FL_VALUE_TIMER      1
32 #define FL_HIDDEN_TIMER     2
33

```

```

41 class FL_EXPORT Fl_Timer : public Fl_Widget {
42     static void stepcb(void *);
43     void step();
44     char on, direction_;
45     double delay, total;
46     long lastsec,lastusec;
47 protected:
48     void draw();
49 public:
50     int handle(int);
51     Fl_Timer(uchar t,int x,int y,int w,int h, const char *l);
52     ~Fl_Timer();
53     void value(double);
54     double value()const {return delay>0.0?delay:0.0;}
55     char direction()const {return direction_;}
56     void direction(char d) {direction_ = d;}
57     char suspended()const {return !on;}
58     void suspended(char d);
59 };
60 #endif
61 //
62 // End of "$Id$".
63 //
64 //

```

32.130 Fl_Toggle_Button.H

```

1 //
2 // "$Id$"
3 //
4 // Toggle button header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software.  Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file.  If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Toggle_Button widget . */
21
22 #ifndef Fl_Toggle_Button_H
23 #define Fl_Toggle_Button_H
24
25 #include "Fl_Button.H"
26
27 class FL_EXPORT Fl_Toggle_Button : public Fl_Button {
28 public:
29     Fl_Toggle_Button(int X,int Y,int W,int H,const char *l=0);
30 };
31 #endif
32 //
33 // End of "$Id$".
34 //

```

32.131 Fl_Toggle_Light_Button.H

```

1 //
2 // "$Id$"
3 //
4 // Toggle light button header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software.  Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file.  If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:

```

```

15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 // provided for back-compatibility only
20
21 #ifndef Fl_Toggle_Light_Button
22 #include "Fl_Light_Button.H"
23 #define Fl_Toggle_Light_Button Fl_Light_Button
24 #endif
25
26 //
27 // End of "$Id$".
28 //

```

32.132 Fl_Toggle_Round_Button.H

```

1 //
2 // "$Id$"
3 //
4 // Toggle round button header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 // provided for back-compatibility only
20
21 #ifndef Fl_Toggle_Round_Button
22 #include "Fl_Round_Button.H"
23 #define Fl_Toggle_Round_Button Fl_Round_Button
24 #endif
25
26 //
27 // End of "$Id$".
28 //

```

32.133 Fl_Tooltip.H

```

1 //
2 // "$Id$"
3 //
4 // Tooltip header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2011 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Tooltip widget . */
21
22 #ifndef Fl_Tooltip_H
23 #define Fl_Tooltip_H
24
25 #include <FL/Fl.H>
26 #include <FL/Fl_Widget.H>
27
28 class FL_EXPORT Fl_Tooltip {
29 public:
30     static float delay() { return delay_; }
31     static void delay(float f) { delay_ = f; }
32     static float hoverdelay() { return hoverdelay_; }
33     static void hoverdelay(float f) { hoverdelay_ = f; }
34     static int enabled() { return Fl::option(Fl::OPTION_SHOW_TOOLTIPS); }

```



```

55 static void enable(int b = 1) { Fl::option(Fl::OPTION_SHOW_TOOLTIPS, (b!=0)); }
57 static void disable() { enable(0); }
58 static void (*enter)(Fl_Widget* w);
59 static void enter_area(Fl_Widget* w, int X, int Y, int W, int H, const char* tip);
60 static void (*exit)(Fl_Widget *w);
62 static Fl_Widget* current() {return widget_;}
63 static void current(Fl_Widget*);
64
66 static Fl_Font font() { return font_; }
68 static void font(Fl_Font i) { font_ = i; }
70 static Fl_Fontsize size() { return (size_ == -1 ? FL_NORMAL_SIZE : size_); }
72 static void size(Fl_Fontsize s) { size_ = s; }
74 static Fl_Color color() { return color_; }
76 static void color(Fl_Color c) { color_ = c; }
78 static Fl_Color textcolor() { return textcolor_; }
80 static void textcolor(Fl_Color c) { textcolor_ = c; }
81 #if FLTK_ABI_VERSION >= 10301
83 static int margin_width() { return margin_width_; }
85 static void margin_width(int v) { margin_width_ = v; }
87 static int margin_height() { return margin_height_; }
89 static void margin_height(int v) { margin_height_ = v; }
91 static int wrap_width() { return wrap_width_; }
93 static void wrap_width(int v) { wrap_width_ = v; }
94 #else
95 static int margin_width() { return 3; }
96 static int margin_height() { return 3; }
97 static int wrap_width() { return 400; }
98 #endif
99
100 #ifdef __APPLE__
101 // the unique tooltip window
102 static Fl_Window* current_window(void);
103 #endif
104
105 // These should not be public, but Fl_Widget::tooltip() needs them...
106 // fabien: made it private with only a friend function access
107 private:
108 friend void Fl_Widget::tooltip(const char *);
109 friend void Fl_Widget::copy_tooltip(const char *);
110 static void enter_(Fl_Widget* w);
111 static void exit_(Fl_Widget *w);
112 static void set_enter_exit_once_();
113
114 private:
115 static float delay_;
116 static float hoverdelay_;
117 static Fl_Color color_;
118 static Fl_Color textcolor_;
119 static Fl_Font font_;
120 static Fl_Fontsize size_;
121 static Fl_Widget* widget_;
122 #if FLTK_ABI_VERSION >= 10301
123 static int margin_width_;
124 static int margin_height_;
125 static int wrap_width_;
126 #endif
127 };
128
129 #endif
130
131 //
132 // End of "$Id$".
133 //

```

32.134 Fl_Tree.H File Reference

This file contains the definitions of the [Fl_Tree](#) class.

```

#include <FL/Fl.H>
#include <FL/Fl_Group.H>
#include <FL/Fl_Scrollbar.H>
#include <FL/fl_draw.H>
#include <FL/Fl_Tree_Item.H>
#include <FL/Fl_Tree_Prefs.H>

```

Classes

- class [Fl_Tree](#)

Tree widget.

Enumerations

- enum `Fl_Tree_Reason` {
`FL_TREE_REASON_NONE = 0`, `FL_TREE_REASON_SELECTED`, `FL_TREE_REASON_DESELECTED`,
`FL_TREE_REASON_RESELECTED`,
`FL_TREE_REASON_OPENED`, `FL_TREE_REASON_CLOSED`, `FL_TREE_REASON_DRAGGED` }

The reason the callback was invoked.

32.134.1 Detailed Description

This file contains the definitions of the `Fl_Tree` class.

32.134.2 Enumeration Type Documentation

32.134.2.1 `Fl_Tree_Reason`

enum `Fl_Tree_Reason`

The reason the callback was invoked.

Enumerator

<code>FL_TREE_REASON_NONE</code>	unknown reason
<code>FL_TREE_REASON_SELECTED</code>	an item was selected
<code>FL_TREE_REASON_DESELECTED</code>	an item was de-selected
<code>FL_TREE_REASON_RESELECTED</code>	an item was re-selected (e.g. double-clicked)
<code>FL_TREE_REASON_OPENED</code>	an item was opened
<code>FL_TREE_REASON_CLOSED</code>	an item was closed
<code>FL_TREE_REASON_DRAGGED</code>	an item was dragged into a new place

32.135 `Fl_Tree.H`

[Go to the documentation of this file.](#)

```

1 //
2 // "$Id$"
3 //
4
5 #ifndef FL_TREE_H
6 #define FL_TREE_H
7
8 #include <FL/Fl.H>
9 #include <FL/Fl_Group.H>
10 #include <FL/Fl_Scrollbar.H>
11 #include <FL/fl_draw.H>
12
13 #include <FL/Fl_Tree_Item.H>
14 #include <FL/Fl_Tree_Prefs.H>
15
16 // FL/Fl_Tree.H
17 //
18 //
19 //
20 // Fl_Tree -- This file is part of the Fl_Tree widget for FLTK
21 // Copyright (C) 2009-2010 by Greg Ercolano.
22 //
23 // This library is free software. Distribution and use rights are outlined in
24 // the file "COPYING" which should have been included with this file. If this
25 // file is missing or damaged, see the license at:
26 //
27 //     http://www.fltk.org/COPYING.php
28 //
29 // Please report all bugs and problems on the following page:
30 //
31 //     http://www.fltk.org/str.php

```

```

32 //
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401

```

```

enum Fl_Tree_Reason {
    FL_TREE_REASON_NONE=0,
    FL_TREE_REASON_SELECTED,
    FL_TREE_REASON_DESELECTED,
#ifdef FLTK_ABI_VERSION >= 10301
    FL_TREE_REASON_RESELECTED,
#endif /*FLTK_ABI_VERSION*/
    FL_TREE_REASON_OPENED,
    FL_TREE_REASON_CLOSED,
    FL_TREE_REASON_DRAGGED
};

class FL_EXPORT Fl_Tree : public Fl_Group {
    friend class Fl_Tree_Item;
    Fl_Tree_Item *_root; // can be null!
    Fl_Tree_Item *_item_focus; // item that has focus box
    Fl_Tree_Item *_callback_item; // item invoked during callback (can be NULL)
    Fl_Tree_Reason _callback_reason; // reason for the callback
    Fl_Tree_Prefs _prefs; // all the tree's settings
    int _scrollbar_size; // size of scrollbar trough
#ifdef FLTK_ABI_VERSION >= 10301
    // NEW:
    Fl_Tree_Item *_lastselect; // used to extend selections
#else /*FLTK_ABI_VERSION*/
    // OLD: static data inside handle() method
#endif /*FLTK_ABI_VERSION*/
    void fix_scrollbar_order();
protected:
    Fl_Scrollbar *_vscroll;
#ifdef FLTK_ABI_VERSION >= 10303
    Fl_Scrollbar *_hscroll;
    int _tox, _toy, _tow, _toy;
    int _tix, _tiy, _tiw, _tih;
#endif
    int _tree_w;
    int _tree_h;
#ifdef FLTK_ABI_VERSION >= 10303
    void item_clicked(Fl_Tree_Item* val);
    void do_callback_for_item(Fl_Tree_Item* item, Fl_Tree_Reason reason);
#endif
    // next_visible_item() and extend_selection() moved to 'public' in ABI 1.3.3
    // undocumented draw_tree() dropped -- draw() does all the work now
    void extend_selection(Fl_Tree_Item *from, Fl_Tree_Item *to);
    int draw_tree();
public:
    Fl_Tree(int X, int Y, int W, int H, const char *L=0);
    ~Fl_Tree();
    int handle(int e);
    void draw();
    void show_self();
    void resize(int,int,int,int);

    // root methods
    void root_label(const char *new_label);
    Fl_Tree_Item* root();
    void root(Fl_Tree_Item *newitem);
    const Fl_Tree_Prefs& prefs()const { return _prefs; }

    // Item creation/removal methods
#ifdef FLTK_ABI_VERSION >= 10303
    Fl_Tree_Item *add(const char *path, Fl_Tree_Item *newitem=0);
#else
    Fl_Tree_Item *add(const char *path);
    Fl_Tree_Item *add(const char *path, Fl_Tree_Item *newitem);
#endif
    Fl_Tree_Item* add(Fl_Tree_Item *parent_item, const char *name);
    Fl_Tree_Item *insert_above(Fl_Tree_Item *above, const char *name);
    Fl_Tree_Item* insert(Fl_Tree_Item *item, const char *name, int pos);
    int remove(Fl_Tree_Item *item);
    void clear();
    void clear_children(Fl_Tree_Item *item);

    // Item lookup methods
    Fl_Tree_Item *find_item(const char *path);
    const Fl_Tree_Item *find_item(const char *path) const;
    int item_pathname(char *pathname, int pathnamelen, const Fl_Tree_Item *item) const;
#ifdef FLTK_ABI_VERSION >= 10303
    const Fl_Tree_Item* find_clicked(int yonly=0) const;

```

```

402 Fl_Tree_Item* find_clicked(int yonly=0);
403 #else
404 const Fl_Tree_Item *find_clicked() const;
405 Fl_Tree_Item *find_clicked();
406 #endif
407 Fl_Tree_Item *item_clicked();
408 Fl_Tree_Item *first();
409 Fl_Tree_Item *first_visible(); // deprecated in ABI 10303
410 Fl_Tree_Item *first_visible_item();
411 Fl_Tree_Item *next(Fl_Tree_Item *item=0);
412 Fl_Tree_Item *prev(Fl_Tree_Item *item=0);
413 Fl_Tree_Item *last();
414 Fl_Tree_Item *last_visible(); // deprecated in ABI 10303
415 Fl_Tree_Item *last_visible_item();
416 #if FLTK_ABI_VERSION >= 10303
417 Fl_Tree_Item *next_visible_item(Fl_Tree_Item *start, int dir); // made public in 1.3.3 ABI
418 #endif
419 Fl_Tree_Item *first_selected_item();
420 Fl_Tree_Item *last_selected_item();
421 Fl_Tree_Item *next_item(Fl_Tree_Item *item, int dir=FL_Down, bool visible=false);
422 #if FLTK_ABI_VERSION >= 10303
423 Fl_Tree_Item *next_selected_item(Fl_Tree_Item *item=0, int dir=FL_Down);
424 int get_selected_items(Fl_Tree_Item_Array &ret_items);
425 #else
426 Fl_Tree_Item *next_selected_item(Fl_Tree_Item *item=0);
427 Fl_Tree_Item *next_selected_item(Fl_Tree_Item *item, int dir);
428 #endif
429
431 // Item open/close methods
432 int open(Fl_Tree_Item *item, int docallback=1);
433 int open(const char *path, int docallback=1);
434 void open_toggle(Fl_Tree_Item *item, int docallback=1);
435 int close(Fl_Tree_Item *item, int docallback=1);
436 int close(const char *path, int docallback=1);
437 int is_open(Fl_Tree_Item *item) const;
438 int is_open(const char *path) const;
439 int is_close(Fl_Tree_Item *item) const;
440 int is_close(const char *path) const;
441
442 // Item selection methods
443 int select(Fl_Tree_Item *item, int docallback=1);
444 int select(const char *path, int docallback=1);
445 void select_toggle(Fl_Tree_Item *item, int docallback=1);
446 int deselect(Fl_Tree_Item *item, int docallback=1);
447 int deselect(const char *path, int docallback=1);
448 int deselect_all(Fl_Tree_Item *item=0, int docallback=1);
449 int select_only(Fl_Tree_Item *selitem, int docallback=1);
450 int select_all(Fl_Tree_Item *item=0, int docallback=1);
451 int extend_selection_dir(Fl_Tree_Item *from,
452                          Fl_Tree_Item *to,
453                          int dir,
454                          int val,
455                          bool visible);
456 #if FLTK_ABI_VERSION >= 10303
457 int extend_selection(Fl_Tree_Item *from,
458                    Fl_Tree_Item *to,
459                    int val=1,
460                    bool visible=false);
461 #else
462 private:
463 // Adding overload if not at least one overload breaks ABI, so avoid
464 // by keeping private until we can break ABI. ref: http://www.rst.org/rep/rep-0009.html
465 int extend_selection__(Fl_Tree_Item *from,
466                       Fl_Tree_Item *to,
467                       int val,
468                       bool visible);
469 public:
470 #endif
471 void set_item_focus(Fl_Tree_Item *item);
472 Fl_Tree_Item *get_item_focus() const;
473 int is_selected(Fl_Tree_Item *item) const;
474 int is_selected(const char *path);
475
476 // Item attribute related methods
477 Fl_Font item_labelfont() const;
478 void item_labelfont(Fl_Font val);
479 Fl_Fontsize item_labelsize() const;
480 void item_labelsize(Fl_Fontsize val);
481 Fl_Color item_labelfgcolor(void) const;
482 void item_labelfgcolor(Fl_Color val);
483 Fl_Color item_labelbgcolor(void) const;
484 void item_labelbgcolor(Fl_Color val);
485 Fl_Color connectorcolor() const;
486 void connectorcolor(Fl_Color val);
487 int marginleft() const;
488 void marginleft(int val);
489 int margintop() const;

```

```

495 void margintop(int val);
496 #if FLTK_ABI_VERSION >= 10301
497 int marginbottom() const;
498 void marginbottom(int val);
499 #endif /*FLTK_ABI_VERSION*/
500 int linespacing() const;
501 void linespacing(int val);
502 int openchild_marginbottom() const;
503 void openchild_marginbottom(int val);
504 int usericonmarginleft() const;
505 void usericonmarginleft(int val);
506 int labelmarginleft() const;
507 void labelmarginleft(int val);
508 #if FLTK_ABI_VERSION >= 10301
509 int widgetmarginleft() const;
510 void widgetmarginleft(int val);
511 #endif /*FLTK_ABI_VERSION*/
512 int connectorwidth() const;
513 void connectorwidth(int val);
514 Fl_Image* usericon() const;
515 void usericon(Fl_Image *val);
516 Fl_Image* openicon() const;
517 void openicon(Fl_Image *val);
518 Fl_Image* closeicon() const;
519 void closeicon(Fl_Image *val);
520 int showcollapse() const;
521 void showcollapse(int val);
522 int showroot() const;
523 void showroot(int val);
524 Fl_Tree_Connector connectorstyle() const;
525 void connectorstyle(Fl_Tree_Connector val);
526 Fl_Tree_Sort sortorder() const;
527 void sortorder(Fl_Tree_Sort val);
528 Fl_Boxtype selectbox() const;
529 void selectbox(Fl_Boxtype val);
530 Fl_Tree_Select selectmode() const;
531 void selectmode(Fl_Tree_Select val);
532 #if FLTK_ABI_VERSION >= 10301
533 Fl_Tree_Item_Reselect_Mode item_reselect_mode() const;
534 void item_reselect_mode(Fl_Tree_Item_Reselect_Mode mode);
535 Fl_Tree_Item_Draw_Mode item_draw_mode() const;
536 void item_draw_mode(Fl_Tree_Item_Draw_Mode mode);
537 void item_draw_mode(int mode);
538 #endif
539 #if FLTK_ABI_VERSION >= 10303
540 void calc_dimensions();
541 void calc_tree();
542 #endif
543 void recalc_tree();
544 int displayed(Fl_Tree_Item *item);
545 void show_item(Fl_Tree_Item *item, int yoff);
546 void show_item(Fl_Tree_Item *item);
547 void show_item_top(Fl_Tree_Item *item);
548 void show_item_middle(Fl_Tree_Item *item);
549 void show_item_bottom(Fl_Tree_Item *item);
550 void display(Fl_Tree_Item *item);
551 int vposition() const;
552 void vposition(int pos);
553 int hposition() const;
554 void hposition(int pos);
555
556 int is_scrollbar(Fl_Widget *w);
557 int scrollbar_size() const;
558 void scrollbar_size(int size);
559 int is_vscroll_visible() const;
560 int is_hscroll_visible() const;
561
562 // callback related
563 void callback_item(Fl_Tree_Item* item);
564 Fl_Tree_Item* callback_item();
565 void callback_reason(Fl_Tree_Reason reason);
566 Fl_Tree_Reason callback_reason() const;
567
571 void load(class Fl_Preferences&);
572 };
573
574 #endif /*FL_TREE_H*/
575
576 //
577 // End of "$Id$".
578 //

```

32.136 Fl_Tree_Item.H File Reference

This file contains the definitions for [Fl_Tree_Item](#).

```
#include <FL/Fl.H>
#include <FL/Fl_Widget.H>
#include <FL/Fl_Image.H>
#include <FL/fl_draw.H>
#include <FL/Fl_Tree_Item_Array.H>
#include <FL/Fl_Tree_Prefs.H>
```

Classes

- class [Fl_Tree_Item](#)

Tree widget item.

32.136.1 Detailed Description

This file contains the definitions for [Fl_Tree_Item](#).

32.137 Fl_Tree_Item.H

[Go to the documentation of this file.](#)

```
1 //
2 // "$Id$"
3 //
4
5 #ifndef FL_TREE_ITEM_H
6 #define FL_TREE_ITEM_H
7
8 #include <FL/Fl.H>
9 #include <FL/Fl_Widget.H>
10 #include <FL/Fl_Image.H>
11 #include <FL/fl_draw.H>
12
13 #include <FL/Fl_Tree_Item_Array.H>
14 #include <FL/Fl_Tree_Prefs.H>
15
17 // FL/Fl_Tree_Item.H
19 //
20 // Fl_Tree -- This file is part of the Fl_Tree widget for FLTK
21 // Copyright (C) 2009-2010 by Greg Ercolano.
22 //
23 // This library is free software. Distribution and use rights are outlined in
24 // the file "COPYING" which should have been included with this file. If this
25 // file is missing or damaged, see the license at:
26 //
27 // http://www.fltk.org/COPYING.php
28 //
29 // Please report all bugs and problems on the following page:
30 //
31 // http://www.fltk.org/str.php
32 //
33
34
36 class Fl_Tree;
37 class FL_EXPORT Fl_Tree_Item {
38 #if FLTK_ABI_VERSION >= 10303
39     Fl_Tree *_tree; // parent tree
40 #endif
41     const char *_label; // label (memory managed)
42     Fl_Font _labelfont; // label's font face
43     Fl_Fontsize _labelsize; // label's font size
44     Fl_Color _labelfgcolor; // label's fg color
45     Fl_Color _labelbgcolor; // label's bg color (0xffffffff is 'transparent')
46 #if FLTK_ABI_VERSION >= 10303
47     enum Fl_Tree_Item_Flags {
48 #else
49     enum {
50 #endif
51     OPEN = 1<<0,
52     VISIBLE = 1<<1,
53     ACTIVE = 1<<2,
54     SELECTED = 1<<3
55 };
```

```

88 #if FLTK_ABI_VERSION >= 10301
89 // NEW
90 unsigned short _flags; // misc flags
91 #else /*FLTK_ABI_VERSION*/
92 // OLD: this will go away after 1.3.x
93 char _open; // item is open?
94 char _visible; // item is visible?
95 char _active; // item activated?
96 char _selected; // item selected?
97 #endif /*FLTK_ABI_VERSION*/
98 int _xywh[4]; // xywh of this widget (if visible)
99 int _collapse_xywh[4]; // xywh of collapse icon (if visible)
100 int _label_xywh[4]; // xywh of label
101 Fl_Widget *_widget; // item's label widget (optional)
102 Fl_Image *_usericon; // item's user-specific icon (optional)
103 #if FLTK_ABI_VERSION >= 10304
104 Fl_Image *_userdeicon; // deactivated usericon
105 #endif
106 Fl_Tree_Item_Array _children; // array of child items
107 Fl_Tree_Item *_parent; // parent item (=0 if root)
108 void *_userdata; // user data that can be associated with an item
109 #if FLTK_ABI_VERSION >= 10301
110 Fl_Tree_Item *_prev_sibling; // previous sibling (same level)
111 Fl_Tree_Item *_next_sibling; // next sibling (same level)
112 #endif /*FLTK_ABI_VERSION*/
113 // Protected methods
114 protected:
115 void _Init(const Fl_Tree_Prefs &prefs, Fl_Tree *tree);
116 void show_widgets();
117 void hide_widgets();
118 void draw_vertical_connector(int x, int y1, int y2, const Fl_Tree_Prefs &prefs);
119 void draw_horizontal_connector(int x1, int x2, int y, const Fl_Tree_Prefs &prefs);
120 void recalc_tree();
121 int calc_item_height(const Fl_Tree_Prefs &prefs) const;
122 #if FLTK_ABI_VERSION >= 10303
123 Fl_Color drawfgcolor() const;
124 Fl_Color drawbgcolor() const;
125 #endif
126
127 public:
128 Fl_Tree_Item(const Fl_Tree_Prefs &prefs); // CTOR -- backwards compatible
129 #if FLTK_ABI_VERSION >= 10303
130 Fl_Tree_Item(Fl_Tree *tree); // CTOR -- ABI 1.3.3+
131 virtual ~Fl_Tree_Item(); // DTOR -- ABI 1.3.3+
132 #else
133 ~Fl_Tree_Item(); // DTOR -- backwards compatible
134 #endif
135 Fl_Tree_Item(const Fl_Tree_Item *o); // COPY CTOR
136 int x()const { return(_xywh[0]); }
137 int y()const { return(_xywh[1]); }
138 int w()const { return(_xywh[2]); }
139 int h()const { return(_xywh[3]); }
140 int label_x()const { return(_label_xywh[0]); }
141 int label_y()const { return(_label_xywh[1]); }
142 int label_w()const { return(_label_xywh[2]); }
143 int label_h()const { return(_label_xywh[3]); }
144 #if FLTK_ABI_VERSION >= 10303
145 virtual int draw_item_content(int render);
146 void draw(int X, int &Y, int W, Fl_Tree_Item *itemfocus,
147 int &tree_item_xmax, int lastchild=1, int render=1);
148 #else
149 void draw(int X, int &Y, int W, Fl_Widget *tree,
150 Fl_Tree_Item *itemfocus, const Fl_Tree_Prefs &prefs, int lastchild=1);
151 #endif
152 void show_self(const char *indent = "") const;
153 void label(const char *val);
154 const char *label() const;
155
156 inline void user_data( void* data ) { _userdata = data; }
157
158 inline void* user_data()const { return _userdata; }
159
160 void labelfont(Fl_Font val) {
161 _labelfont = val;
162 recalc_tree(); // may change tree geometry
163 }
164 Fl_Font labelfont()const {
165 return(_labelfont);
166 }
167 void labelsizes(Fl_Fontsize val) {
168 _labelsizes = val;
169 recalc_tree(); // may change tree geometry
170 }
171 Fl_Fontsize labelsizes()const {
172 return(_labelsizes);
173 }
174 void labelfgcolor(Fl_Color val) {

```

```

196     _labelfgcolor = val;
197 }
198 Fl_Color labelfgcolor()const {
199     return(_labelfgcolor);
200 }
201 void labelcolor(Fl_Color val) {
202     labelfgcolor(val);
203 }
204 Fl_Color labelcolor()const {
205     return labelfgcolor();
206 }
207 void labelbgcolor(Fl_Color val) {
208     _labelbgcolor = val;
209 }
210 Fl_Color labelbgcolor()const {
211     return(_labelbgcolor);
212 }
213 void widget(Fl_Widget *val) {
214     _widget = val;
215     recalc_tree(); // may change tree geometry
216 }
217 Fl_Widget *widget()const {
218     return(_widget);
219 }
220 int children()const {
221     return(_children.total());
222 }
223 Fl_Tree_Item *child(int index) {
224     return(_children[index]);
225 }
226 const Fl_Tree_Item *child(int t) const;
227 int has_children()const {
228     return(children());
229 }
230 int find_child(const char *name);
231 int find_child(Fl_Tree_Item *item);
232 int remove_child(Fl_Tree_Item *item);
233 int remove_child(const char *new_label);
234 void clear_children();
235 void swap_children(int ax, int bx);
236 int swap_children(Fl_Tree_Item *a, Fl_Tree_Item *b);
237 const Fl_Tree_Item *find_child_item(const char *name) const;
238 Fl_Tree_Item *find_child_item(const char *name);
239 const Fl_Tree_Item *find_child_item(char **arr) const;
240 Fl_Tree_Item *find_child_item(char **arr);
241 const Fl_Tree_Item *find_item(char **arr) const;
242 Fl_Tree_Item *find_item(char **arr);
243 // Adding items
244 Fl_Tree_Item *add(const Fl_Tree_Prefs &prefs,
245                 const char *new_label,
246                 Fl_Tree_Item *newitem);
247 Fl_Tree_Item *add(const Fl_Tree_Prefs &prefs,
248                 const char *new_label);
249 Fl_Tree_Item *add(const Fl_Tree_Prefs &prefs,
250                 char **arr,
251                 Fl_Tree_Item *newitem);
252 Fl_Tree_Item *add(const Fl_Tree_Prefs &prefs,
253                 char **arr);
254 #if FLTK_ABI_VERSION >= 10303
255     Fl_Tree_Item *replace(Fl_Tree_Item *new_item);
256     Fl_Tree_Item *replace_child(Fl_Tree_Item *olditem, Fl_Tree_Item *newitem);
257 #endif
258 Fl_Tree_Item *insert(const Fl_Tree_Prefs &prefs, const char *new_label, int pos=0);
259 Fl_Tree_Item *insert_above(const Fl_Tree_Prefs &prefs, const char *new_label);
260 Fl_Tree_Item *deparent(int index);
261 int reparent(Fl_Tree_Item *newchild, int index);
262 int move(int to, int from);
263 int move(Fl_Tree_Item *item, int op=0, int pos=0);
264 int move_above(Fl_Tree_Item *item);
265 int move_below(Fl_Tree_Item *item);
266 int move_into(Fl_Tree_Item *item, int pos=0);
267 int depth() const;
268 Fl_Tree_Item *prev();
269 Fl_Tree_Item *next();
270 Fl_Tree_Item *next_sibling();
271 Fl_Tree_Item *prev_sibling();
272 void update_prev_next(int index);
273 Fl_Tree_Item *next_displayed(Fl_Tree_Prefs &prefs); // deprecated
274 Fl_Tree_Item *prev_displayed(Fl_Tree_Prefs &prefs); // deprecated
275 Fl_Tree_Item *next_visible(Fl_Tree_Prefs &prefs);
276 Fl_Tree_Item *prev_visible(Fl_Tree_Prefs &prefs);
277
278 Fl_Tree_Item *parent() {
279     return(_parent);
280 }
281 const Fl_Tree_Item *parent()const {
282     return(_parent);

```



```

302 }
306 void parent(Fl_Tree_Item *val) {
307     _parent = val;
308 }
309 #if FLTK_ABI_VERSION >= 10303
310     const Fl_Tree_Prefs& prefs() const;
313     const Fl_Tree *tree() const {
314         return(_tree);
315     }
316 #endif
317 #if FLTK_ABI_VERSION >= 10304
320     Fl_Tree *tree() {
321         return(_tree);
322     }
323 #endif
325     // State
327     void open();
328     void close();
330     int is_open()const {
331         return(is_flag(OPEN));
332     }
334     int is_close()const {
335         return(is_flag(OPEN)?0:1);
336     }
338     void open_toggle() {
339         is_open()?close():open();    // handles calling recalc_tree()
340     }
344     void select(int val=1) {
345         set_flag(SELECTED, val);
346     }
348     void select_toggle() {
349         if ( is_selected() ) {
350             deselect();    // deselect if selected
351         } else {
352             select();    // select if deselected
353         }
354     }
359     int select_all() {
360         int count = 0;
361         if ( ! is_selected() ) {
362             select();
363             ++count;
364         }
365         for ( int t=0; t<children(); t++ ) {
366             count += child(t)->select_all();
367         }
368         return(count);
369     }
371     void deselect() {
372         set_flag(SELECTED, 0);
373     }
378     int deselect_all() {
379         int count = 0;
380         if ( is_selected() ) {
381             deselect();
382             ++count;
383         }
384         for ( int t=0; t<children(); t++ ) {
385             count += child(t)->deselect_all();
386         }
387         return(count);
388     }
390     char is_selected()const {
391         return(is_flag(SELECTED));
392     }
401     void activate(int val=1) {
402         set_flag(ACTIVE, val);
403         if ( _widget && val != (int)_widget->active() ) {
404             if ( val ) {
405                 _widget->activate();
406             } else {
407                 _widget->deactivate();
408             }
409             _widget->redraw();
410         }
411     }
415     void deactivate() {
416         activate(0);
417     }
419     char is_activated()const {
420         return(is_flag(ACTIVE));
421     }
423     char is_active()const {
424         return(is_activated());
425     }
427     int visible()const {
428         return(is_visible());

```

```

429 }
431 int is_visible()const {
432     return(is_flag(VISIBLE));
433 }
434 int visible_r() const;
435
445 void usericon(Fl_Image *val) {
446     _usericon = val;
447     recalc_tree();           // may change tree geometry
448 }
450 Fl_Image *usericon()const {
451     return(_usericon);
452 }
479 #if FLTK_ABI_VERSION >= 10304
480 void userdeicon(Fl_Image* val) {
481     _userdeicon = val;
482 }
485 Fl_Image* userdeicon()const {
486     return _userdeicon;
487 }
488 #endif
490 // Events
492 #if FLTK_ABI_VERSION >= 10303
493     const Fl_Tree_Item* find_clicked(const Fl_Tree_Prefs &prefs, int yonly=0) const;
494     Fl_Tree_Item* find_clicked(const Fl_Tree_Prefs &prefs, int yonly=0);
495 #else
496     const Fl_Tree_Item* find_clicked(const Fl_Tree_Prefs &prefs) const;
497     Fl_Tree_Item* find_clicked(const Fl_Tree_Prefs &prefs);
498 #endif
499     int event_on_collapse_icon(const Fl_Tree_Prefs &prefs) const;
500     int event_on_label(const Fl_Tree_Prefs &prefs) const;
502     int is_root()const {
503         return(_parent==0?1:0);
504     }
505
506     // Protected methods
507     // TODO: move these to top 'protected:' section
508 protected:
509 #if FLTK_ABI_VERSION >= 10301
511     inline void set_flag(unsigned short flag,int val) {
512         if ( flag==OPEN || flag==VISIBLE ) {
513             recalc_tree();           // may change tree geometry
514         }
515         if ( val ) _flags |= flag; else _flags &= ~flag;
516     }
518     inline int is_flag(unsigned short val)const {
519         return(_flags & val ? 1 : 0);
520     }
521 #else /*FLTK_ABI_VERSION*/
523     void set_flag(unsigned short flag,int val) {
524         switch (flag) {
525             case OPEN: _open = val; break;
526             case VISIBLE: _visible = val; break;
527             case ACTIVE: _active = val; break;
528             case SELECTED: _selected = val; break;
529         }
530     }
532     int is_flag(unsigned short flag)const {
533         switch (flag) {
534             case OPEN: return(_open ? 1 : 0);
535             case VISIBLE: return(_visible ? 1 : 0);
536             case ACTIVE: return(_active ? 1 : 0);
537             case SELECTED: return(_selected ? 1 : 0);
538             default: return(0);
539         }
540     }
541 #endif /*FLTK_ABI_VERSION*/
542
543 };
544
545 #endif /*FL_TREE_ITEM_H*/
546
547 //
548 // End of "$Id$".
549 //

```

32.138 Fl_Tree_Item_Array.H File Reference

This file defines a class that manages an array of [Fl_Tree_Item](#) pointers.

```
#include <FL/Fl.H>
```

```
#include "Fl_Export.H"
```

Classes

- class [Fl_Tree_Item_Array](#)
Manages an array of [Fl_Tree_Item](#) pointers.

32.138.1 Detailed Description

This file defines a class that manages an array of [Fl_Tree_Item](#) pointers.

32.139 Fl_Tree_Item_Array.H

[Go to the documentation of this file.](#)

```

1 //
2 // "$Id$"
3 //
4
5 #ifndef _FL_TREE_ITEM_ARRAY_H
6 #define _FL_TREE_ITEM_ARRAY_H
7
8 #include <FL/Fl.H>
9 #include "Fl_Export.H"
10
11 class Fl_Tree_Item;          // forward decl must *precede* first doxygen comment block
12                             // or doxygen will not document our class..
13
14 // FL/Fl_Tree_Item_Array.H
15 //
16 // Fl_Tree -- This file is part of the Fl_Tree widget for FLTK
17 // Copyright (C) 2009-2010 by Greg Ercolano.
18 //
19 // This library is free software.  Distribution and use rights are outlined in
20 // the file "COPYING" which should have been included with this file.  If this
21 // file is missing or damaged, see the license at:
22 //
23 // http://www.fltk.org/COPYING.php
24 //
25 // Please report all bugs and problems on the following page:
26 //
27 // http://www.fltk.org/str.php
28 //
29 //
30 //
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47 class FL_EXPORT Fl_Tree_Item_Array {
48     Fl_Tree_Item **_items;    // items array
49     int _total;              // #items in array
50     int _size;               // #items *allocated* for array
51     int _chunksize;         // #items to enlarge mem allocation
52 #if FLTK_ABI_VERSION >= 10303
53     enum {
54         MANAGE_ITEM = 1,
55     };
56     char _flags;             // flags to control behavior
57 #endif
58     void enlarge(int count);
59 public:
60     Fl_Tree_Item_Array(int new_chunksize = 10);    // CTOR
61     ~Fl_Tree_Item_Array();                        // DTOR
62     Fl_Tree_Item_Array(const Fl_Tree_Item_Array *o); // COPY CTOR
63     Fl_Tree_Item *operator[](int i) {
64         return(_items[i]);
65     }
66     const Fl_Tree_Item *operator[](int i) const {
67         return(_items[i]);
68     }
69     int total() const {
70         return(_total);
71     }
72 #if FLTK_ABI_VERSION >= 10301
73     // NEW -- code moved to .cxx
74     void swap(int ax, int bx);
75 #else /*FLTK_ABI_VERSION*/
76     // OLD
77     void swap(int ax, int bx) {
78         Fl_Tree_Item *asave = _items[ax];
79         _items[ax] = _items[bx];
80         _items[bx] = asave;
81     }
82 #endif /*FLTK_ABI_VERSION*/
83     int move(int to, int from);
84     int deparent(int pos);

```

```

89  int reparent(Fl_Tree_Item *item, Fl_Tree_Item *newparent, int pos);
90  void clear();
91  void add(Fl_Tree_Item *val);
92  void insert(int pos, Fl_Tree_Item *new_item);
93  void replace(int pos, Fl_Tree_Item *new_item);
94  void remove(int index);
95  int  remove(Fl_Tree_Item *item);
96  #if FLTK_ABI_VERSION >= 10303
100 void manage_item_destroy(int val) {
101     if ( val ) _flags |= MANAGE_ITEM; else _flags &= ~MANAGE_ITEM;
102 }
103 int manage_item_destroy() const {
104     return _flags & MANAGE_ITEM ? 1 : 0;
105 }
106 #endif
107 };
108
109 #endif /*_FL_TREE_ITEM_ARRAY_H*/
110
111 //
112 // End of "$Id$".
113 //

```

32.140 Fl_Tree_Prefs.H File Reference

This file contains the definitions for `Fl_Tree`'s preferences.

```
#include <FL/Fl.H>
```

Classes

- class `Fl_Tree_Prefs`
Tree widget's preferences.

Typedefs

- typedef void() `Fl_Tree_Item_Draw_Callback(Fl_Tree_Item *, void *)`

Enumerations

- enum `Fl_Tree_Connector` { `FL_TREE_CONNECTOR_NONE` =0 , `FL_TREE_CONNECTOR_DOTTED` =1 , `FL_TREE_CONNECTOR_SOLID` =2 }
- Defines the style of connection lines between items.*
- enum `Fl_Tree_Item_Draw_Mode` { `FL_TREE_ITEM_DRAW_DEFAULT` =0 , `FL_TREE_ITEM_DRAW_LABEL_AND_WIDGET` =1 , `FL_TREE_ITEM_HEIGHT_FROM_WIDGET` =2 }
- Bit flags that control how item's labels and widget(s) are drawn in the tree via `item_draw_mode()`.*
- enum `Fl_Tree_Item_Reselect_Mode` { `FL_TREE_SELECTABLE_ONCE` =0 , `FL_TREE_SELECTABLE_ALWAYS` }
- Defines the ways an item can be (re) selected via `item_reselect_mode()`.*
- enum `Fl_Tree_Select` { `FL_TREE_SELECT_NONE` =0 , `FL_TREE_SELECT_SINGLE` =1 , `FL_TREE_SELECT_MULTI` =2 , `FL_TREE_SELECT_SINGLE_DRAGGABLE` =3 }
- Tree selection style.*
- enum `Fl_Tree_Sort` { `FL_TREE_SORT_NONE` =0 , `FL_TREE_SORT_ASCENDING` =1 , `FL_TREE_SORT_DESCENDING` =2 }
- Sort order options for items added to the tree.*

32.140.1 Detailed Description

This file contains the definitions for `Fl_Tree`'s preferences.

```

Fl_Tree_Prefs
:
:
:
:
Fl_Tree :
|_____ Fl_Tree_Item

```

32.140.2 Enumeration Type Documentation

32.140.2.1 Fl_Tree_Connector

enum [Fl_Tree_Connector](#)

Defines the style of connection lines between items.

Enumerator

FL_TREE_CONNECTOR_NONE	Use no lines connecting items.
FL_TREE_CONNECTOR_DOTTED	Use dotted lines connecting items (default)
FL_TREE_CONNECTOR_SOLID	Use solid lines connecting items.

32.140.2.2 Fl_Tree_Item_Draw_Mode

enum [Fl_Tree_Item_Draw_Mode](#)

Bit flags that control how item's labels and widget(s) are drawn in the tree via `item_draw_mode()`.

Enumerator

FL_TREE_ITEM_DRAW_DEFAULT	If widget() defined, draw in place of label, and widget() tracks item height (default)
FL_TREE_ITEM_DRAW_LABEL_AND_WIDGET	If widget() defined, include label to the left of the widget.
FL_TREE_ITEM_HEIGHT_FROM_WIDGET	If widget() defined, widget()'s height controls item's height.

32.140.2.3 Fl_Tree_Item_Reselect_Mode

enum [Fl_Tree_Item_Reselect_Mode](#)

Defines the ways an item can be (re) selected via `item_reselect_mode()`.

Enumerator

FL_TREE_SELECTABLE_ONCE	Item can only be selected once (default)
FL_TREE_SELECTABLE_ALWAYS	Enables FL_TREE_REASON_RESELECTED events for callbacks.

32.140.2.4 Fl_Tree_Select

enum [Fl_Tree_Select](#)

Tree selection style.

Enumerator

FL_TREE_SELECT_NONE	Nothing selected when items are clicked.
FL_TREE_SELECT_SINGLE	Single item selected when item is clicked (default)
FL_TREE_SELECT_MULTI	Multiple items can be selected by clicking with SHIFT, CTRL or mouse drags.
FL_TREE_SELECT_SINGLE_DRAGGABLE	Single items may be selected, and they may be. reordered by mouse drag.

32.140.2.5 Fl_Tree_Sort

enum [Fl_Tree_Sort](#)

Sort order options for items added to the tree.

Enumerator

FL_TREE_SORT_NONE	No sorting; items are added in the order defined (default).
FL_TREE_SORT_ASCENDING	Add items in ascending sort order.
FL_TREE_SORT_DESCENDING	Add items in descending sort order.

32.141 Fl_Tree_Prefs.H

[Go to the documentation of this file.](#)

```

1 //
2 // "$Id$"
3 //
4
5 #ifndef FL_TREE_PREFS_H
6 #define FL_TREE_PREFS_H
7
8 #include <FL/Fl.H> // needed for ABI version features (via Enumerations.H)
9
10 // FL/Fl_Tree_Prefs.H
11 //
12 //
13 // Fl_Tree_Prefs -- This file is part of the Fl_Tree widget for FLTK
14 // Copyright (C) 2009-2010 by Greg Ercolano.
15 //
16 //
17 // This library is free software. Distribution and use rights are outlined in
18 // the file "COPYING" which should have been included with this file. If this
19 // file is missing or damaged, see the license at:
20 //
21 // http://www.fltk.org/COPYING.php
22 //
23 // Please report all bugs and problems on the following page:
24 //
25 // http://www.fltk.org/str.php
26 //
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49 enum Fl_Tree_Sort {
50     FL_TREE_SORT_NONE=0,
51     FL_TREE_SORT_ASCENDING=1,
52     FL_TREE_SORT_DESCENDING=2
53 };
54
55
56
57
58 enum Fl_Tree_Connector {
59     FL_TREE_CONNECTOR_NONE=0,
60     FL_TREE_CONNECTOR_DOTTED=1,
61     FL_TREE_CONNECTOR_SOLID=2
62 };
63
64
65
66
67 enum Fl_Tree_Select {
68     FL_TREE_SELECT_NONE=0,
69     FL_TREE_SELECT_SINGLE=1,
70     FL_TREE_SELECT_MULTI=2,
71     FL_TREE_SELECT_SINGLE_DRAGGABLE=3
72 };
73
74
75
76 #if FLTK_ABI_VERSION >= 10301
77 enum Fl_Tree_Item_Reselect_Mode {
78     FL_TREE_SELECTABLE_ONCE=0,
79     FL_TREE_SELECTABLE_ALWAYS=1
80 };
81
82
83
84
85
86
87
88
89
90 enum Fl_Tree_Item_Draw_Mode {
91     FL_TREE_ITEM_DRAW_DEFAULT=0,
92     FL_TREE_ITEM_DRAW_LABEL_AND_WIDGET=1,
93     FL_TREE_ITEM_HEIGHT_FROM_WIDGET=2
94 };
95
96 #endif /*FLTK_ABI_VERSION*/
97
98 #if FLTK_ABI_VERSION >= 10303

```

```

99 class Fl_Tree_Item;
100 typedef void (Fl_Tree_Item_Draw_Callback)(Fl_Tree_Item*, void*);
101 #endif
102
103 class FL_EXPORT Fl_Tree_Prefs {
104     Fl_Font _labelfont;           // label's font face
105     Fl_Fontsize _labelsize;      // label's font size
106     int _margintop;              // --
107     int _marginleft;             // |- tree's controllable margins
108 #if FLTK_ABI_VERSION >= 10301
109     int _marginbottom;          // --
110 #endif
111     int _openchild_marginbottom;  // extra space below an open child tree
112     int _usericonmarginleft;     // space to left of user icon (if any)
113     int _labelmarginleft;       // space to left of label
114 #if FLTK_ABI_VERSION >= 10301
115     int _widgetmarginleft;      // space to left of widget
116 #endif
117     int _connectorwidth;         // connector width (right of open/close icon)
118     int _linespacing;           // vertical space between lines
119     // Colors
120     Fl_Color _labelfgcolor;     // label's foreground color
121     Fl_Color _labelbgcolor;     // label's background color
122     Fl_Color _connectorcolor;   // connector dotted line color
123     Fl_Tree_Connector _connectorstyle; // connector line style
124     Fl_Image *_openimage;       // the 'open' icon [+]
125     Fl_Image *_closeimage;      // the 'close' icon [-]
126     Fl_Image *_userimage;       // user's own icon
127 #if FLTK_ABI_VERSION >= 10304
128     Fl_Image *_opendeimage;     // deactivated 'open' icon
129     Fl_Image *_closedeimage;    // deactivated 'close' icon
130     Fl_Image *_userdeimage;     // deactivated user icon
131 #endif
132     char _showcollapse;         // 1=show collapse icons, 0=don't
133     char _showroot;            // show the root item as part of the tree
134     Fl_Tree_Sort _sortorder;    // none, ascending, descending, etc.
135     Fl_Boxtype _selectbox;      // selection box type
136     Fl_Tree_Select _selectmode; // selection mode
137 #if FLTK_ABI_VERSION >= 10301
138     Fl_Tree_Item_Reselect_Mode _itemreselectmode; // controls item selection callback() behavior
139     Fl_Tree_Item_Draw_Mode _itemdrawmode;        // controls how items draw label + widget()
140 #endif /*FLTK_ABI_VERSION*/
141 #if FLTK_ABI_VERSION >= 10303
142     Fl_Tree_Item_Draw_Callback *_itemdrawcallback; // callback to handle drawing items (0=none)
143     void *_itemdrawuserdata; // data for drawing items (0=none)
144 #endif
145 public:
146     Fl_Tree_Prefs();
147 #if FLTK_ABI_VERSION >= 10304
148     ~Fl_Tree_Prefs();
149 #endif
150
151     // Labels
152     inline Fl_Font item_labelfont()const { return(_labelfont); }
153     inline void item_labelfont(Fl_Font val) { _labelfont = val; }
154     inline Fl_Fontsize item_labelsize()const { return(_labelsize); }
155     inline void item_labelsize(Fl_Fontsize val) { _labelsize = val; }
156     inline Fl_Color item_labelfgcolor()const { return(_labelfgcolor); }
157     inline void item_labelfgcolor(Fl_Color val) { _labelfgcolor = val; }
158 #if FLTK_ABI_VERSION >= 10301
159     inline Fl_Color item_labelbgcolor()const {
160         return _labelbgcolor;
161     }
162     inline void item_labelbgcolor(Fl_Color val) {
163         _labelbgcolor = val;
164     }
165 #else /*FLTK_ABI_VERSION*/
166     inline Fl_Color item_labelbgcolor()const {
167         return(_labelbgcolor);
168     }
169     inline void item_labelbgcolor(Fl_Color val) {
170         _labelbgcolor = val;
171     }
172 #endif /*FLTK_ABI_VERSION*/
173
174     // Obsolete names - for 1.3.0 backwards compat
175     inline Fl_Font labelfont()const { return(_labelfont); }
176     inline void labelfont(Fl_Font val) { _labelfont = val; }
177     inline Fl_Fontsize labelsize()const { return(_labelsize); }
178     inline void labelsize(Fl_Fontsize val) { _labelsize = val; }
179     inline Fl_Color labelfgcolor()const { return(_labelfgcolor); }
180     inline void labelfgcolor(Fl_Color val) { _labelfgcolor = val; }
181     inline Fl_Color labelbgcolor()const { return(item_labelbgcolor()); }
182     inline void labelbgcolor(Fl_Color val) { item_labelbgcolor(val); }
183
184     // Margins
185     inline int marginleft()const {

```

```

224     return(_marginleft);
225 }
227 inline void marginleft(int val) {
228     _marginleft = val;
229 }
231 inline int margintop()const {
232     return(_margintop);
233 }
235 inline void margintop(int val) {
236     _margintop = val;
237 }
238 #if FLTK_ABI_VERSION >= 10301
241 inline int marginbottom()const {
242     return(_marginbottom);
243 }
246 inline void marginbottom(int val) {
247     _marginbottom = val;
248 }
249 #endif /*FLTK_ABI_VERSION*/
251 inline int openchild_marginbottom()const {
252     return(_openchild_marginbottom);
253 }
255 inline void openchild_marginbottom(int val) {
256     _openchild_marginbottom = val;
257 }
259 inline int usericonmarginleft()const {
260     return(_usericonmarginleft);
261 }
263 inline void usericonmarginleft(int val) {
264     _usericonmarginleft = val;
265 }
267 inline int labelmarginleft()const {
268     return(_labelmarginleft);
269 }
271 inline void labelmarginleft(int val) {
272     _labelmarginleft = val;
273 }
274 #if FLTK_ABI_VERSION >= 10301
276 inline int widgetmarginleft()const {
277     return(_widgetmarginleft);
278 }
280 inline void widgetmarginleft(int val) {
281     _widgetmarginleft = val;
282 }
283 #endif /*FLTK_ABI_VERSION*/
285 inline int linespacing()const {
286     return(_linespacing);
287 }
289 inline void linespacing(int val) {
290     _linespacing = val;
291 }
292
294 // Colors and Styles
297 inline Fl_Color connectorcolor()const {
298     return(_connectorcolor);
299 }
301 inline void connectorcolor(Fl_Color val) {
302     _connectorcolor = val;
303 }
305 inline Fl_Tree_Connector connectorstyle()const {
306     return(_connectorstyle);
307 }
309 inline void connectorstyle(Fl_Tree_Connector val) {
310     _connectorstyle = val;
311 }
313 inline void connectorstyle(int val) {
314     _connectorstyle = Fl_Tree_Connector(val);
315 }
317 inline int connectorwidth()const {
318     return(_connectorwidth);
319 }
321 inline void connectorwidth(int val) {
322     _connectorwidth = val;
323 }
324
326 // Icons
331 inline Fl_Image *openicon()const {
332     return(_openimage);
333 }
334 void openicon(Fl_Image *val);
338 inline Fl_Image *closeicon()const {
339     return(_closeimage);
340 }
341 void closeicon(Fl_Image *val);
343 inline Fl_Image *usericon()const {
344     return(_userimage);
345 }

```



```

349 inline void usericon(Fl_Image *val) {
350     _userimage = val;
351 #if FLTK_ABI_VERSION >= 10304
352     // Update deactivated version of icon..
353     if ( _userdeimage ) delete _userdeimage;
354     if ( _userimage ) {
355         _userdeimage = _userimage->copy();
356         _userdeimage->inactive();
357     } else {
358         _userdeimage = 0;
359     }
360 #endif
361 }
362
363 #if FLTK_ABI_VERSION >= 10304
364 inline Fl_Image *opendeicon()const {
365     return _opendeimage;
366 }
367 inline Fl_Image *closedeicon()const {
368     return _closedeimage;
369 }
370 inline Fl_Image *userdeicon()const {
371     return _userdeimage;
372 }
373 #endif
374
375 // Options
376 inline char showcollapse()const {
377     return(_showcollapse);
378 }
379 inline void showcollapse(int val) {
380     _showcollapse = val;
381 }
382 inline Fl_Tree_Sort sortorder()const {
383     return(_sortorder);
384 }
385 inline void sortorder(Fl_Tree_Sort val) {
386     _sortorder = val;
387 }
388 inline Fl_Boxtype selectbox()const {
389     return(_selectbox);
390 }
391 inline void selectbox(Fl_Boxtype val) {
392     _selectbox = val;
393 }
394 inline int showroot()const {
395     return(int(_showroot));
396 }
397 inline void showroot(int val) {
398     _showroot = char(val);
399 }
400 inline Fl_Tree_Select selectmode()const {
401     return(_selectmode);
402 }
403 inline void selectmode(Fl_Tree_Select val) {
404     _selectmode = val;
405 }
406 #if FLTK_ABI_VERSION >= 10301
407 Fl_Tree_Item_Reselect_Mode item_reselect_mode()const {
408     return _itemreselectmode;
409 }
410 void item_reselect_mode(Fl_Tree_Item_Reselect_Mode mode) {
411     _itemreselectmode = mode;
412 }
413 inline Fl_Tree_Item_Draw_Mode item_draw_mode()const {
414     return(_itemdrawmode);
415 }
416 inline void item_draw_mode(Fl_Tree_Item_Draw_Mode val) {
417     _itemdrawmode = val;
418 }
419 #endif
420 #if FLTK_ABI_VERSION >= 10303
421 void item_draw_callback(Fl_Tree_Item_Draw_Callback *cb, void *data=0) {
422     _itemdrawcallback = cb;
423     _itemdrawuserdata = data;
424 }
425 Fl_Tree_Item_Draw_Callback* item_draw_callback()const {
426     return(_itemdrawcallback);
427 }
428 void* item_draw_user_data()const {
429     return(_itemdrawuserdata);
430 }
431 void do_item_draw_callback(Fl_Tree_Item *o)const {
432     _itemdrawcallback(o, _itemdrawuserdata);
433 }
434 #endif
435 };

```

```

479
480 #endif /*FL_TREE_PREFS_H*/
481
482 //
483 // End of "$Id$".
484 //

```

32.142 fl_types.h File Reference

This file contains simple "C"-style type definitions.

Typedefs

Miscellaneous

- typedef unsigned int **FI_Char**
24-bit Unicode character - upper 8 bits are unused
- typedef const char * **FI_CString**
Flexible length UTF-8 Unicode read-only string.
- typedef unsigned int **FI_Shortcut**
24-bit Unicode character + 8-bit indicator for keyboard flags
- typedef char * **FI_String**
Flexible length UTF-8 Unicode text.
- typedef unsigned char **uchar**
unsigned char
- typedef unsigned long **ulong**
unsigned long

32.142.1 Detailed Description

This file contains simple "C"-style type definitions.

32.142.2 Typedef Documentation

32.142.2.1 FI_CString

```
typedef const char* FI_CString
```

Flexible length UTF-8 Unicode read-only string.

See also

[FI_String](#)

32.142.2.2 FI_String

```
typedef char* FI_String
```

Flexible length UTF-8 Unicode text.

Todo FIXME: temporary (?) typedef to mark UTF-8 and Unicode conversions

32.143 fl_types.h

[Go to the documentation of this file.](#)

```

1 /*
2 * "$Id$"
3 *
4 * Simple "C"-style types for the Fast Light Tool Kit (FLTK).
5 *

```

```

6 * Copyright 1998-2015 by Bill Spitzak and others.
7 *
8 * This library is free software.  Distribution and use rights are outlined in
9 * the file "COPYING" which should have been included with this file.  If this
10 * file is missing or damaged, see the license at:
11 *
12 *     http://www.fltk.org/COPYING.php
13 *
14 * Please report all bugs and problems on the following page:
15 *
16 *     http://www.fltk.org/str.php
17 */
18
19 #ifndef FL_TYPES_H
20 #define FL_TYPES_H
21     /* group: Miscellaneous */
22
23 typedef unsigned char uchar;
24 typedef unsigned long ulong;
25
26 typedef char *Fl_String;
27
28 typedef const char *Fl_CString;
29
30 typedef unsigned int Fl_Shortcut;
31
32 typedef unsigned int Fl_Char;
33     /* group: Miscellaneous */
34 #endif
35
36 /*
37 * End of "$Id$".
38 */

```

32.144 fl_utf8.h File Reference

header for Unicode and UTF-8 character handling

```

#include "Fl_Export.H"
#include "fl_types.h"
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <locale.h>

```

Macros

- `#define xchar` unsigned short

Functions

- FL_EXPORT int [fl_access](#) (const char *f, int mode)
Cross-platform function to test a files access() with a UTF-8 encoded name or value.
- FL_EXPORT int [fl_chmod](#) (const char *f, int mode)
Cross-platform function to set a files mode() with a UTF-8 encoded name or value.
- FL_EXPORT int [fl_execvp](#) (const char *file, char *const *argv)
- FL_EXPORT FILE * [fl_fopen](#) (const char *f, const char *mode)
Cross-platform function to open files with a UTF-8 encoded name.
- FL_EXPORT char * [fl_getcwd](#) (char *b, int l)
Cross-platform function to get the current working directory as a UTF-8 encoded value.
- FL_EXPORT char * [fl_getenv](#) (const char *v)
Cross-platform function to get environment variables with a UTF-8 encoded name or value.
- FL_EXPORT char [fl_make_path](#) (const char *path)
Cross-platform function to recursively create a path in the file system.

- FL_EXPORT void [fl_make_path_for_file](#) (const char *path)
Cross-platform function to create a path for the file in the file system.
- FL_EXPORT int [fl_mkdir](#) (const char *f, int mode)
Cross-platform function to create a directory with a UTF-8 encoded name.
- FL_EXPORT unsigned int [fl_nonspacing](#) (unsigned int ucs)
Returns true if the Unicode character `ucs` is non-spacing.
- FL_EXPORT int [fl_open](#) (const char *f, int oflags,...)
Cross-platform function to open files with a UTF-8 encoded name.
- FL_EXPORT int [fl_rename](#) (const char *f, const char *n)
Cross-platform function to rename a filesystem object using UTF-8 encoded names.
- FL_EXPORT int [fl_rmdir](#) (const char *f)
Cross-platform function to remove a directory with a UTF-8 encoded name.
- FL_EXPORT int [fl_stat](#) (const char *f, struct stat *b)
Cross-platform function to `stat()` a file using a UTF-8 encoded name or value.
- FL_EXPORT int [fl_system](#) (const char *cmd)
Cross-platform function to run a system command with a UTF-8 encoded string.
- FL_EXPORT int [fl_tolower](#) (unsigned int ucs)
Returns the Unicode lower case value of `ucs`.
- FL_EXPORT int [fl_toupper](#) (unsigned int ucs)
Returns the Unicode upper case value of `ucs`.
- FL_EXPORT unsigned [fl_ucs_to_Utf16](#) (const unsigned ucs, unsigned short *dst, const unsigned dstlen)
- FL_EXPORT int [fl_unlink](#) (const char *f)
Cross-platform function to `unlink()` (that is, delete) a file using a UTF-8 encoded filename.
- FL_EXPORT char * [fl_utf2mbcs](#) (const char *s)
Converts UTF-8 string `s` to a local multi-byte character string.
- FL_EXPORT const char * [fl_utf8back](#) (const char *p, const char *start, const char *end)
- FL_EXPORT int [fl_utf8bytes](#) (unsigned ucs)
Return the number of bytes needed to encode the given UCS4 character in UTF-8.
- FL_EXPORT unsigned [fl_utf8decode](#) (const char *p, const char *end, int *len)
- FL_EXPORT int [fl_utf8encode](#) (unsigned ucs, char *buf)
- FL_EXPORT unsigned [fl_utf8from_mb](#) (char *dst, unsigned dstlen, const char *src, unsigned srclen)
- FL_EXPORT unsigned [fl_utf8froma](#) (char *dst, unsigned dstlen, const char *src, unsigned srclen)
- FL_EXPORT unsigned [fl_utf8fromwlc](#) (char *dst, unsigned dstlen, const wchar_t *src, unsigned srclen)
- FL_EXPORT const char * [fl_utf8fwd](#) (const char *p, const char *start, const char *end)
- FL_EXPORT int [fl_utf8len](#) (char c)
Returns the byte length of the UTF-8 sequence with first byte `c`, or -1 if `c` is not valid.
- FL_EXPORT int [fl_utf8len1](#) (char c)
Returns the byte length of the UTF-8 sequence with first byte `c`, or 1 if `c` is not valid.
- FL_EXPORT int [fl_utf8locale](#) (void)
- FL_EXPORT int [fl_utf8test](#) (const char *src, unsigned len)
- FL_EXPORT unsigned [fl_utf8to_mb](#) (const char *src, unsigned srclen, char *dst, unsigned dstlen)
- FL_EXPORT unsigned [fl_utf8toa](#) (const char *src, unsigned srclen, char *dst, unsigned dstlen)
- FL_EXPORT unsigned [fl_utf8toUtf16](#) (const char *src, unsigned srclen, unsigned short *dst, unsigned dstlen)
- FL_EXPORT unsigned [fl_utf8towc](#) (const char *src, unsigned srclen, wchar_t *dst, unsigned dstlen)
Converts a UTF-8 string into a wide character string.
- FL_EXPORT int [fl_utf_nb_char](#) (const unsigned char *buf, int len)
Returns the number of Unicode chars in the UTF-8 string.
- FL_EXPORT int [fl_utf_strcasecmp](#) (const char *s1, const char *s2)
UTF-8 aware `strcasecmp` - converts to Unicode and tests.
- FL_EXPORT int [fl_utf_strncasecmp](#) (const char *s1, const char *s2, int n)
UTF-8 aware `strncasecmp` - converts to lower case Unicode and tests.
- FL_EXPORT int [fl_utf_tolower](#) (const unsigned char *str, int len, char *buf)

Converts the string `str` to its lower case equivalent into `buf`.

- FL_EXPORT int `fl_utf_toupper` (const unsigned char *str, int len, char *buf)

Converts the string `str` to its upper case equivalent into `buf`.

- FL_EXPORT int `fl_wcwidth` (const char *src)
extended wrapper around `fl_wcwidth_(unsigned int ucs)` function.
- FL_EXPORT int `fl_wcwidth_` (unsigned int ucs)
wrapper to adapt Markus Kuhn's implementation of `wcwidth()` for FLTK

32.144.1 Detailed Description

header for Unicode and UTF-8 character handling

32.145 fl_utf8.h

[Go to the documentation of this file.](#)

```

1 /*
2  * "$Id$"
3  *
4  * Author: Jean-Marc Lienher ( http://oksid.ch )
5  * Copyright 2000-2010 by O'ksi'D.
6  *
7  * This library is free software. Distribution and use rights are outlined in
8  * the file "COPYING" which should have been included with this file. If this
9  * file is missing or damaged, see the license at:
10 *
11 * http://www.fltk.org/COPYING.php
12 *
13 * Please report all bugs and problems on the following page:
14 *
15 * http://www.fltk.org/str.php
16 */
17
18 /* Merged in some functionality from the fltk-2 version. IMM.
19 * The following code is an attempt to merge the functions incorporated in FLTK2
20 * with the functions provided in Oksid's fltk-1.1.6-utf8 port
21 */
22
23 #ifndef _HAVE_FL_UTF8_HDR_
24 #define _HAVE_FL_UTF8_HDR_
25
26 #include "Fl_Export.H"
27 #include "fl_types.h"
28
29 #include <stdio.h>
30 #include <string.h>
31 #include <stdlib.h>
32
33 #ifdef WIN32
34 # include <sys/types.h>
35 # include <sys/stat.h>
36 # include <locale.h>
37 # include <ctype.h>
38 # define xchar wchar_t
39 # if !defined(FL_DLL) && !defined(__CYGWIN__)
40 # undef strdup
41 # define strdup _strdup
42 # undef putenv
43 # define putenv _putenv
44 # undef stricmp
45 # define stricmp _stricmp
46 # undef strnicmp
47 # define strnicmp _strnicmp
48 # undef chdir
49 # define chdir _chdir
50 # endif
51 #elif defined(__APPLE__)
52 # include <wchar.h>
53 # include <sys/stat.h>
54 # define xchar wchar_t
55 #else /* X11 */
56 # include <sys/types.h>
57 # include <sys/stat.h>
58 # if defined(FL_LIBRARY) /* don't expose X11 headers in user space */
59 # include <X11/Xlocale.h>
60 # include <X11/Xlib.h>
61 # endif /* defined(FL_LIBRARY) -- don't expose X11 headers in user space */
62 # include <locale.h>
63 # define xchar unsigned short

```

```

69 #endif
70
71 #ifdef __cplusplus
72 extern "C" {
73 #endif
74
75 /* F2: comes from FLTK2 */
76 /* OD: comes from Oksid */
77
78 FL_EXPORT int fl_utf8bytes(unsigned ucs);
79
80 /* OD: returns the byte length of the first UTF-8 char sequence (returns -1 if not valid) */
81 FL_EXPORT int fl_utf8len(char c);
82
83 /* OD: returns the byte length of the first UTF-8 char sequence (returns +1 if not valid) */
84 FL_EXPORT int fl_utf8len1(char c);
85
86 /* OD: returns the number of Unicode chars in the UTF-8 string */
87 FL_EXPORT int fl_utf_nb_char(const unsigned char *buf, int len);
88
89 /* F2: Convert the next UTF-8 char-sequence into a Unicode value (and say how many bytes were used) */
90 FL_EXPORT unsigned fl_utf8decode(const char* p, const char* end, int* len);
91
92 /* F2: Encode a Unicode value into a UTF-8 sequence, return the number of bytes used */
93 FL_EXPORT int fl_utf8encode(unsigned ucs, char* buf);
94
95 /* F2: Move forward to the next valid UTF-8 sequence start between start and end */
96 FL_EXPORT const char* fl_utf8fwd(const char* p, const char* start, const char* end);
97
98 /* F2: Move backward to the previous valid UTF-8 sequence start */
99 FL_EXPORT const char* fl_utf8back(const char* p, const char* start, const char* end);
100
101 /* XX: Convert a single 32-bit Unicode value into UTF16 */
102 FL_EXPORT unsigned fl_ucs_to_Utf16(const unsigned ucs, unsigned short *dst, const unsigned dstlen);
103
104 /* F2: Convert a UTF-8 string into UTF16 */
105 FL_EXPORT unsigned fl_utf8toUtf16(const char* src, unsigned srclen, unsigned short* dst, unsigned
dstlen);
106
107 /* F2: Convert a UTF-8 string into a wide character string - makes UTF16 on win32, "UCS4" elsewhere */
108 FL_EXPORT unsigned fl_utf8towc(const char *src, unsigned srclen, wchar_t *dst, unsigned dstlen);
109
110 /* F2: Convert a wide character string to UTF-8 - takes in UTF16 on win32, "UCS4" elsewhere */
111 FL_EXPORT unsigned fl_utf8fromwc(char *dst, unsigned dstlen, const wchar_t *src, unsigned srclen);
112
113 /* F2: Convert a UTF-8 string into ASCII, eliding untranslatable glyphs */
114 FL_EXPORT unsigned fl_utf8toa (const char *src, unsigned srclen, char *dst, unsigned dstlen);
115
116 /* F2: Convert 8859-1 string to UTF-8 */
117 FL_EXPORT unsigned fl_utf8froma (char *dst, unsigned dstlen, const char *src, unsigned srclen);
118
119 /* F2: Returns true if the current O/S locale is UTF-8 */
120 FL_EXPORT int fl_utf8locale(void);
121
122 /* F2: Examine the first len characters of src, to determine if the input text is UTF-8 or not
123 * NOTE: The value returned is not simply boolean - it contains information about the probable
124 * type of the src text. */
125 FL_EXPORT int fl_utf8test(const char *src, unsigned len);
126
127 /* XX: return width of "raw" ucs character in columns.
128 * for internal use only */
129 FL_EXPORT int fl_wcwidth_(unsigned int ucs);
130
131 /* XX: return width of utf-8 character string in columns.
132 * NOTE: this may also do C1 control character (0x80 to 0x9f) to CP1252 mapping,
133 * depending on original build options */
134 FL_EXPORT int fl_wcwidth(const char *src);
135
136 /* OD: Return true if the character is non-spacing */
137 FL_EXPORT unsigned int fl_nonspacing(unsigned int ucs);
138
139 /* F2: Convert UTF-8 to a local multi-byte encoding - mainly for win32? */
140 FL_EXPORT unsigned fl_utf8to_mb(const char *src, unsigned srclen, char *dst, unsigned dstlen);
141 /* OD: Convert UTF-8 to a local multi-byte encoding */
142 FL_EXPORT char* fl_utf2mbcs(const char *src);
143
144 /* F2: Convert a local multi-byte encoding to UTF-8 - mainly for win32? */
145 FL_EXPORT unsigned fl_utf8from_mb(char *dst, unsigned dstlen, const char *src, unsigned srclen);
146
147 /*****
148 #ifdef WIN32
149 /* OD: Attempt to convert the UTF-8 string to the current locale */
150 FL_EXPORT char *fl_utf8_to_locale(const char *s, int len, unsigned int codepage);
151
152 /* OD: Attempt to convert a string in the current locale to UTF-8 */
153 FL_EXPORT char *fl_locale_to_utf8(const char *s, int len, unsigned int codepage);
154 #endif
155 */

```

```

164
165 /*****
166 * The following functions are intended to provide portable, UTF-8 aware
167 * versions of standard functions
168 */
169
170 /* OD: UTF-8 aware strncasecmp - converts to lower case Unicode and tests */
171 FL_EXPORT int fl_utf_strncasecmp(const char *s1, const char *s2, int n);
172
173 /* OD: UTF-8 aware strcasecmp - converts to Unicode and tests */
174 FL_EXPORT int fl_utf_strcasecmp(const char *s1, const char *s2);
175
176 /* OD: return the Unicode lower case value of ucs */
177 FL_EXPORT int fl_tolower(unsigned int ucs);
178
179 /* OD: return the Unicode upper case value of ucs */
180 FL_EXPORT int fl_toupper(unsigned int ucs);
181
182 /* OD: converts the UTF-8 string to the lower case equivalent */
183 FL_EXPORT int fl_utf_tolower(const unsigned char *str, int len, char *buf);
184
185 /* OD: converts the UTF-8 string to the upper case equivalent */
186 FL_EXPORT int fl_utf_toupper(const unsigned char *str, int len, char *buf);
187
188 /* OD: Portable UTF-8 aware chmod wrapper */
189 FL_EXPORT int fl_chmod(const char* f, int mode);
190
191 /* OD: Portable UTF-8 aware access wrapper */
192 FL_EXPORT int fl_access(const char* f, int mode);
193
194 /* OD: Portable UTF-8 aware stat wrapper */
195 FL_EXPORT int fl_stat( const char *path, struct stat *buffer );
196
197 /* OD: Portable UTF-8 aware getcwd wrapper */
198 FL_EXPORT char* fl_getcwd( char *buf, int maxlen);
199
200 /* OD: Portable UTF-8 aware fopen wrapper */
201 FL_EXPORT FILE *fl_fopen(const char *f, const char *mode);
202
203 /* OD: Portable UTF-8 aware system wrapper */
204 FL_EXPORT int fl_system(const char* f);
205
206 /* OD: Portable UTF-8 aware execvp wrapper */
207 FL_EXPORT int fl_execvp(const char *file, char *const *argv);
208
209 /* OD: Portable UTF-8 aware open wrapper */
210 FL_EXPORT int fl_open(const char* f, int o, ...);
211
212 /* OD: Portable UTF-8 aware unlink wrapper */
213 FL_EXPORT int fl_unlink(const char *f);
214
215 /* OD: Portable UTF-8 aware rmdir wrapper */
216 FL_EXPORT int fl_rmdir(const char *f);
217
218 /* OD: Portable UTF-8 aware getenv wrapper */
219 FL_EXPORT char* fl_getenv(const char *name);
220
221 /* OD: Portable UTF-8 aware execvp wrapper */
222 FL_EXPORT int fl_mkdir(const char* f, int mode);
223
224 /* OD: Portable UTF-8 aware rename wrapper */
225 FL_EXPORT int fl_rename(const char* f, const char *t);
226
227
228 /* OD: Given a full pathname, this will create the directory path needed to hold the file named */
229 FL_EXPORT void fl_make_path_for_file( const char *path );
230
231 /* OD: recursively create a path in the file system */
232 FL_EXPORT char fl_make_path( const char *path );
233
234
235 /*****/
236
237 #ifdef __cplusplus
238 }
239 #endif /* __cplusplus */
240
241 #endif /* _HAVE_FL_UTF8_HDR_ */
242
243
244 /*
245 * End of "$Id$".
246 */

```

32.146 Fl_Valuator.H

```

1 //
2 // "$Id$"
3 //
4 // Valuator header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2016 by Bill Spitzak and others.
7 //
8 // This library is free software.  Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file.  If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Valuator widget . */
21
22 #ifndef Fl_Valuator_H
23 #define Fl_Valuator_H
24
25 #ifndef Fl_Widget_H
26 #include "Fl_Widget.H"
27 #endif
28
29 // shared type() values for classes that work in both directions:
30 #define FL_VERTICAL      0
31 #define FL_HORIZONTAL    1
32
33
34 class FL_EXPORT Fl_Valuator : public Fl_Widget {
35
36     double value_;
37     double previous_value_;
38     double min, max; // truncates to this range *after* rounding
39     double A; int B; // rounds to multiples of A/B, or no rounding if A is zero
40
41 protected:
42     int horizontal()const {return type() & FL_HORIZONTAL;}
43     Fl_Valuator(int X, int Y, int W, int H, const char* L);
44
45     double previous_value()const {return previous_value_;}
46     void handle_push() {previous_value_ = value_;}
47     double softclamp(double);
48     void handle_drag(double newvalue);
49     void handle_release(); // use drag() value
50     virtual void value_damage(); // cause damage() due to value() changing
51     void set_value(double v) {value_ = v;}
52
53 public:
54     void bounds(double a, double b) {min=a; max=b;}
55     double minimum()const {return min;}
56     void minimum(double a) {min = a;}
57     double maximum()const {return max;}
58     void maximum(double a) {max = a;}
59     void range(double a, double b) {min = a; max = b;}
60     void step(int a) {A = a; B = 1;}
61     void step(double a, int b) {A = a; B = b;}
62     void step(double s);
63     double step()const {return A/B;}
64     void precision(int digits);
65
66     double value()const {return value_;}
67     int value(double);
68
69     virtual int format(char*);
70     double round(double); // round to nearest multiple of step
71     double clamp(double); // keep in range
72     double increment(double, int); // add n*step to value
73 };
74
75 #endif
76
77 //
78 // End of "$Id$".
79 //

```


32.147 Fl_Value_Input.H

```

1 //
2 // "$Id$"
3 //
4 // Value input header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software.  Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file.  If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Value_Input widget . */
21
22 #ifndef Fl_Value_Input_H
23 #define Fl_Value_Input_H
24
25 #include "Fl_Valuator.H"
26 #include "Fl_Input.H"
27
28 class FL_EXPORT Fl_Value_Input : public Fl_Valuator {
29 public:
30     /* This is the encapsulated Fl_input attribute to which
31     this class delegates the value font, color and shortcut */
32     Fl_Input input;
33 private:
34     char soft_;
35     static void input_cb(Fl_Widget*,void*);
36     virtual void value_damage(); // cause damage() due to value() changing
37 public:
38     int handle(int);
39 protected:
40     void draw();
41 public:
42     void resize(int,int,int,int);
43     Fl_Value_Input(int x,int y,int w,int h,const char *l=0);
44     ~Fl_Value_Input();
45
46     void soft(char s) {soft_ = s;}
47     char soft()const {return soft_;}
48     int shortcut()const {return input.shortcut();}
49     void shortcut(int s) {input.shortcut(s);}
50
51     Fl_Font textfont()const {return input.textfont();}
52     void textfont(Fl_Font s) {input.textfont(s);}
53     Fl_Fontsize textsize()const {return input.textsize();}
54     void textsize(Fl_Fontsize s) {input.textsize(s);}
55     Fl_Color textcolor()const {return input.textcolor();}
56     void textcolor(Fl_Color n) {input.textcolor(n);}
57     Fl_Color cursor_color()const {return input.cursor_color();}
58     void cursor_color(Fl_Color n) {input.cursor_color(n);}
59 };
60
61 #endif
62 //
63 // End of "$Id$".
64 //

```

32.148 Fl_Value_Output.H

```

1 //
2 // "$Id$"
3 //
4 // Value output header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software.  Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file.  If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //

```

```

14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Value_Output widget . */
21
22 #ifndef Fl_Value_Output_H
23 #define Fl_Value_Output_H
24
25 #ifndef Fl_Valuator_H
26 #include "Fl_Valuator.H"
27 #endif
28
29 class FL_EXPORT Fl_Value_Output : public Fl_Valuator {
30     Fl_Font textfont_;
31     Fl_Fontsize textsize_;
32     uchar soft_;
33     Fl_Color textcolor_;
34
35 protected:
36     void draw();
37
38 public:
39     int handle(int);
40     Fl_Value_Output(int x,int y,int w,int h,const char *l=0);
41
42     void soft(uchar s) {soft_ = s;}
43     uchar soft()const {return soft_;}
44
45     Fl_Font textfont()const {return textfont_;}
46     void textfont(Fl_Font s) {textfont_ = s;}
47     Fl_Fontsize textsize()const {return textsize_;}
48     void textsize(Fl_Fontsize s) {textsize_ = s;}
49     Fl_Color textcolor()const {return textcolor_;}
50     void textcolor(Fl_Color s) {textcolor_ = s;}
51 };
52
53 #endif
54
55 //
56 // End of "$Id$".
57 //

```

32.149 Fl_Value_Slider.H

```

1 //
2 // "$Id$"
3 //
4 // Value slider header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Value_Slider widget . */
21
22 #ifndef Fl_Value_Slider_H
23 #define Fl_Value_Slider_H
24
25 #include "Fl_Slider.H"
26
27 class FL_EXPORT Fl_Value_Slider : public Fl_Slider {
28     Fl_Font textfont_;
29     Fl_Fontsize textsize_;
30     Fl_Color textcolor_;
31
32 protected:
33     void draw();
34
35 public:
36     int handle(int);
37     Fl_Value_Slider(int x,int y,int w,int h, const char *l = 0);
38     Fl_Font textfont()const {return textfont_;}
39     void textfont(Fl_Font s) {textfont_ = s;}
40     Fl_Fontsize textsize()const {return textsize_;}

```

```

49     void textsize(Fl_Fontsize s) {textsize_ = s;}
51     Fl_Color textcolor()const {return textcolor_;}
53     void textcolor(Fl_Color s) {textcolor_ = s;}
54 };
55
56 #endif
57
58 //
59 // End of "$Id$".
60 //

```

32.150 Fl_Widget.H File Reference

[Fl_Widget](#), [Fl_Label](#) classes .

```
#include "Enumerations.H"
```

Classes

- struct [Fl_Label](#)
This struct stores all information for a text or mixed graphics label.
- class [Fl_Widget](#)
Fl_Widget is the base class for all widgets in FLTK.

Macros

- #define [FL_RESERVED_TYPE](#) 100
Reserved type numbers (necessary for my cheapo RTTI) start here.

Typedefs

- typedef void() [Fl_Callback](#)([Fl_Widget](#) *, void *)
Default callback type definition for all fltk widgets (by far the most used)
- typedef void() [Fl_Callback0](#)([Fl_Widget](#) *)
One parameter callback type definition passing only the widget.
- typedef void() [Fl_Callback1](#)([Fl_Widget](#) *, long)
Callback type definition passing the widget and a long data value.
- typedef [Fl_Callback](#) * [Fl_Callback_p](#)
Default callback type pointer definition for all fltk widgets.
- typedef long [fl_intptr_t](#)
- typedef unsigned long [fl_uintptr_t](#)

32.150.1 Detailed Description

[Fl_Widget](#), [Fl_Label](#) classes .

32.150.2 Macro Definition Documentation

32.150.2.1 FL_RESERVED_TYPE

```
#define FL_RESERVED_TYPE 100
```

Reserved type numbers (necessary for my cheapo RTTI) start here.

Grep the header files for "RESERVED_TYPE" to find the next available number.

32.150.3 Typedef Documentation

32.150.3.1 fl_intptr_t

```
typedef long fl_intptr_t
```

Todo typedef's fl_intptr_t and fl_uintptr_t should be documented.

32.151 Fl_Widget.H

Go to the documentation of this file.

```
1 //
2 // "$Id$"
3 //
4 // Widget header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2015 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
22 #ifndef Fl_Widget_H
23 #define Fl_Widget_H
24
25 #include "Enumerations.H"
26
30 #ifdef _WIN64
31 #if defined(__GNUC__) || defined(__clang__)
32 #include <stdint.h>
33 #else
34 #include <stddef.h> // MSVC
35 #endif
36 typedef intptr_t fl_intptr_t;
37 typedef uintptr_t fl_uintptr_t;
38 #else
39 typedef long fl_intptr_t;
40 typedef unsigned long fl_uintptr_t;
41 #endif
42
43 class Fl_Widget;
44 class Fl_Window;
45 class Fl_Group;
46 class Fl_Image;
47
49 typedef void (Fl_Callback) (Fl_Widget*, void*);
51 typedef Fl_Callback* Fl_Callback_p; // needed for BORLAND
53 typedef void (Fl_Callback0) (Fl_Widget*);
55 typedef void (Fl_Callback1) (Fl_Widget*, long);
56
65 struct FL_EXPORT Fl_Label {
67     const char* value;
69     Fl_Image* image;
71     Fl_Image* deimage;
73     Fl_Font font;
75     Fl_Fontsize size;
77     Fl_Color color;
79     Fl_Align align_;
81     uchar type;
82
84     void draw(int,int,int,int, Fl_Align) const ;
85     void measure(int &w, int &h) const ;
86 };
87
88
101 class FL_EXPORT Fl_Widget {
102     friend class Fl_Group;
103
104     Fl_Group* parent_;
105     Fl_Callback* callback_;
106     void* user_data_;
107     int x_, y_, w_, h_;
108     Fl_Label label_;
109     unsigned int flags_;
110     Fl_Color color_;
111     Fl_Color color2_;
112     uchar type_;
113     uchar damage_;
```

```

114  uchar box_;
115  uchar when_;
116
117  const char *tooltip_;
118
120  Fl_Widget(const Fl_Widget &);
122  Fl_Widget& operator=(const Fl_Widget &);
123
124 protected:
125
136  Fl_Widget(int x, int y, int w, int h, const char *label=0L);
137
139  void x(int v) {x_ = v;}
141  void y(int v) {y_ = v;}
143  void w(int v) {w_ = v;}
145  void h(int v) {h_ = v;}
147  unsigned int flags()const {return flags_;}
149  void set_flag(unsigned int c) {flags_ |= c;}
151  void clear_flag(unsigned int c) {flags_ &= ~c;}
155  enum {
156      INACTIVE          = 1<<0,
157      INVISIBLE         = 1<<1,
158      OUTPUT           = 1<<2,
159      NOBORDER         = 1<<3,
160      FORCE_POSITION    = 1<<4,
161      NON_MODAL        = 1<<5,
162      SHORTCUT_LABEL   = 1<<6,
163      CHANGED          = 1<<7,
164      OVERRIDE         = 1<<8,
165      VISIBLE_FOCUS    = 1<<9,
166      COPIED_LABEL     = 1<<10,
167      CLIP_CHILDREN    = 1<<11,
168      MENU_WINDOW      = 1<<12,
169      TOOLTIP_WINDOW   = 1<<13,
170      MODAL            = 1<<14,
171      NO_OVERLAY       = 1<<15,
172      GROUP_RELATIVE   = 1<<16,
173      COPIED_TOOLTIP   = 1<<17,
174      FULLSCREEN       = 1<<18,
175      MAC_USE_ACCENTS_MENU = 1<<19,
176      // (space for more flags)
177      USERFLAG3        = 1<<29,
178      USERFLAG2        = 1<<30,
179      USERFLAG1        = 1<<31
180  };
181  void draw_box() const;
182  void draw_box(Fl_Boxtype t, Fl_Color c) const;
183  void draw_box(Fl_Boxtype t, int x,int y,int w,int h, Fl_Color c) const;
184  void draw_backdrop() const;
186  void draw_focus() {draw_focus(box(),x(),y(),w(),h());}
187  void draw_focus(Fl_Boxtype t, int x,int y,int w,int h) const;
188  void draw_label() const;
189  void draw_label(int, int, int, int) const;
190
191 public:
192
201  virtual ~Fl_Widget();
202
219  virtual void draw() = 0;
220
237  virtual int handle(int event);
238
247  int is_label_copied()const {return ((flags_ & COPIED_LABEL) ? 1 : 0);}
248
254  Fl_Group* parent()const {return parent_;}
255
264  void parent(Fl_Group* p) {parent_ = p;} // for hacks only, use Fl_Group::add()
265
274  uchar type()const {return type_;}
275
279  void type(uchar t) {type_ = t;}
280
284  int x()const {return x_;}
285
289  int y()const {return y_;}
290
294  int w()const {return w_;}
295
299  int h()const {return h_;}
300
320  virtual void resize(int x, int y, int w, int h);
321
323  int damage_resize(int,int,int,int);
324
332  void position(int X,int Y) {resize(X,Y,w_,h_);}
333
341  void size(int W,int H) {resize(x_,y_,W,H);}

```

```

342
348 Fl_Align align()const {return label_.align_;}
349
357 void align(Fl_Align alignment) {label_.align_ = alignment;}
358
363 Fl_Boxtype box()const {return (Fl_Boxtype)box_;}
364
372 void box(Fl_Boxtype new_box) {box_ = new_box;}
373
378 Fl_Color color()const {return color_;}
379
390 void color(Fl_Color bg) {color_ = bg;}
391
396 Fl_Color selection_color()const {return color2_;}
397
406 void selection_color(Fl_Color a) {color2_ = a;}
407
415 void color(Fl_Color bg, Fl_Color sel) {color_=bg; color2_=sel;}
416
421 const char* label()const {return label_.value;}
422
433 void label(const char* text);
434
445 void copy_label(const char *new_label);
446
450 void label(Fl_Labeltype a, const char* b) {label_.type = a; label_.value = b;}
451
456 Fl_Labeltype labeltype()const {return (Fl_Labeltype)label_.type;}
457
466 void labeltype(Fl_Labeltype a) {label_.type = a;}
467
472 Fl_Color labelcolor()const {return label_.color;}
473
478 void labelcolor(Fl_Color c) {label_.color=c;}
479
487 Fl_Font labelfont()const {return label_.font;}
488
496 void labelfont(Fl_Font f) {label_.font=f;}
497
502 Fl_Fontsize labelsize()const {return label_.size;}
503
508 void labelsize(Fl_Fontsize pix) {label_.size=pix;}
509
514 Fl_Image* image() {return label_.image;}
515 const Fl_Image* image()const {return label_.image;}
516
521 void image(Fl_Image* img) {label_.image=img;}
522
527 void image(Fl_Image& img) {label_.image=&img;}
528
533 Fl_Image* deimage() {return label_.deimage;}
534 const Fl_Image* deimage()const {return label_.deimage;}
535
540 void deimage(Fl_Image* img) {label_.deimage=img;}
541
546 void deimage(Fl_Image& img) {label_.deimage=&img;}
547
552 const char *tooltip()const {return tooltip_;}
553
554 void tooltip(const char *text); // see Fl_Tooltip
555 void copy_tooltip(const char *text); // see Fl_Tooltip
556
561 Fl_Callback_p callback()const {return callback_;}
562
568 void callback(Fl_Callback* cb, void* p) {callback_=cb; user_data_=p;}
569
574 void callback(Fl_Callback* cb) {callback_=cb;}
575
580 void callback(Fl_Callback0*cb) {callback_=(Fl_Callback*)cb;}
581
587 void callback(Fl_Callback1*cb, long p=0) {callback_=(Fl_Callback*)cb;
user_data_=(void*)(fl_intptr_t)p;}
588
593 void* user_data()const {return user_data_;}
594
599 void user_data(void* v) {user_data_ = v;}
600
605 long argument()const {return (long)(fl_intptr_t)user_data_;}
606
611 void argument(long v) {user_data_ = (void*)(fl_intptr_t)v;}
612
621 Fl_When when()const {return (Fl_When)when_;}
622
654 void when(uchar i) {when_ = i;}
655
660 unsigned int visible()const {return !(flags_&INVISIBLE);}
661

```

```

666 int visible_r() const;
667
685 virtual void show();
686
690 virtual void hide();
691
696 void set_visible() {flags_ &= ~INVISIBLE;}
697
702 void clear_visible() {flags_ |= INVISIBLE;}
703
708 unsigned int active()const {return !(flags_&INACTIVE);}
709
714 int active_r() const;
715
721 void activate();
722
737 void deactivate();
738
747 unsigned int output()const {return (flags_&OUTPUT);}
748
752 void set_output() {flags_ |= OUTPUT;}
753
757 void clear_output() {flags_ &= ~OUTPUT;}
758
764 unsigned int takeevents()const {return !(flags_&(INACTIVE|INVISIBLE|OUTPUT));}
765
781 unsigned int changed()const {return flags_&CHANGED;}
782
786 void set_changed() {flags_ |= CHANGED;}
787
791 void clear_changed() {flags_ &= ~CHANGED;}
792
797 void clear_active() {flags_ |= INACTIVE;}
798
803 void set_active() {flags_ &= ~INACTIVE;}
804
812 int take_focus();
813
820 void set_visible_focus() { flags_ |= VISIBLE_FOCUS; }
821
826 void clear_visible_focus() { flags_ &= ~VISIBLE_FOCUS; }
827
832 void visible_focus(int v) { if (v) set_visible_focus(); else clear_visible_focus(); }
833
838 unsigned int visible_focus() { return flags_ & VISIBLE_FOCUS; }
839
855 static void default_callback(Fl_Widget *cb, void *d);
856
861 void do_callback() {do_callback(this,user_data_);}
862
869 void do_callback(Fl_Widget* o,long arg) {do_callback(o,(void*)(fl_intptr_t)arg);}
870
871 // Causes a widget to invoke its callback function with arbitrary arguments.
872 // Documentation and implementation in Fl_Widget.cxx
873 void do_callback(Fl_Widget* o,void* arg=0);
874
875 /* Internal use only. */
876 int test_shortcut();
877 /* Internal use only. */
878 static unsigned int label_shortcut(const char *t);
879 /* Internal use only. */
880 static int test_shortcut(const char*, const bool require_alt = false);
881 /* Internal use only. */
882 void _set_fullscreen() {flags_ |= FULLSCREEN;}
883 void _clear_fullscreen() {flags_ &= ~FULLSCREEN;}
884
890 int contains(const Fl_Widget *w) const ;
891
898 int inside(const Fl_Widget* wgt)const {return wgt ? wgt->contains(this) : 0;}
899
903 void redraw();
904
909 void redraw_label();
910
917 uchar damage()const {return damage_;}
918
931 void clear_damage(uchar c = 0) {damage_ = c;}
932
938 void damage(uchar c);
939
946 void damage(uchar c, int x, int y, int w, int h);
947
948 void draw_label(int, int, int, int, Fl_Align) const;
949
957 void measure_label(int& ww, int& hh)const {label_.measure(ww, hh);}
958
959 Fl_Window* window() const ;

```

```

960 Fl_Window* top_window() const;
961 Fl_Window* top_window_offset(int& xoff, int& yoff) const;
962
986 virtual Fl_Group* as_group() {return 0;}
987
1000 virtual Fl_Window* as_window() {return 0;}
1001
1012 virtual class Fl_Gl_Window* as_gl_window() {return 0;}
1013
1016 int use_accents_menu() { return flags() & MAC_USE_ACCENTS_MENU; }
1017
1021 Fl_Color color2()const {return (Fl_Color)color2_;}
1022
1026 void color2(unsigned a) {color2_ = a;}
1027 };
1028
1034 #define FL_RESERVED_TYPE 100
1035
1036 #endif
1037
1038 //
1039 // End of "$Id$".
1040 //

```

32.152 FI_Window.H File Reference

[Fl_Window](#) widget .

```

#include "Fl_Group.H"
#include "Fl_Bitmap.H"
#include <stdlib.h>

```

Classes

- class [Fl_Window](#)
This widget produces an actual window.
- struct [Fl_Window::shape_data_type](#)
Data supporting a non-rectangular window shape.

Macros

- #define [FL_DOUBLE_WINDOW](#) 0xF1
double window type id
- #define [FL_WINDOW](#) 0xF0
window type id all subclasses have type() >= this

32.152.1 Detailed Description

[Fl_Window](#) widget .

32.153 FI_Window.H

[Go to the documentation of this file.](#)

```

1 //
2 // "$Id$"
3 //
4 // Window header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2012 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 // http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //

```



```

16 //      http://www.fltk.org/str.php
17 //
18
22 #ifndef Fl_Window_H
23 #define Fl_Window_H
24
25 #ifdef WIN32
26 #include <windows.h>
27 #endif
28
29 #include "Fl_Group.H"
30 #include "Fl_Bitmap.H"
31 #include <stdlib.h>
32
33 #define FL_WINDOW 0xF0
34 #define FL_DOUBLE_WINDOW 0xF1
35
36 class Fl_X;
37 class Fl_RGB_Image;
38
39
57 class FL_EXPORT Fl_Window : public Fl_Group {
58
59     static char *default_xclass_;
60     // Note: we must use separate statements for each of the following 8 variables,
61     // with the static attribute, otherwise MS VC++ 2008/2010 complains :-()
62     // AlbrechtS 04/2012
63     #if FLTK_ABI_VERSION < 10301
64     static // when these members are static, ABI compatibility with 1.3.0 is respected
65     #endif
66     int no_fullscreen_x;
67     #if FLTK_ABI_VERSION < 10301
68     static // when these members are static, ABI compatibility with 1.3.0 is respected
69     #endif
70     int no_fullscreen_y;
71     #if FLTK_ABI_VERSION < 10301
72     static // when these members are static, ABI compatibility with 1.3.0 is respected
73     #endif
74     int no_fullscreen_w;
75     #if FLTK_ABI_VERSION < 10301
76     static // when these members are static, ABI compatibility with 1.3.0 is respected
77     #endif
78     int no_fullscreen_h;
79     #if FLTK_ABI_VERSION < 10303
80     static // when these members are static, ABI compatibility with 1.3.0 is respected
81     #endif
82     int fullscreen_screen_top;
83     #if FLTK_ABI_VERSION < 10303
84     static // when these members are static, ABI compatibility with 1.3.0 is respected
85     #endif
86     int fullscreen_screen_bottom;
87     #if FLTK_ABI_VERSION < 10303
88     static // when these members are static, ABI compatibility with 1.3.0 is respected
89     #endif
90     int fullscreen_screen_left;
91     #if FLTK_ABI_VERSION < 10303
92     static // when these members are static, ABI compatibility with 1.3.0 is respected
93     #endif
94     int fullscreen_screen_right;
95
96     friend class Fl_X;
97     Fl_X *i; // points at the system-specific stuff
98
99     struct icon_data {
100         const void *legacy_icon;
101         Fl_RGB_Image **icons;
102         int count;
103     #ifdef WIN32
104         HICON big_icon;
105         HICON small_icon;
106     #endif
107     };
108
109     const char* iconlabel_;
110     char* xclass_;
111     struct icon_data *icon_;
112     // size_range stuff:
113     int minw, minh, maxw, maxh;
114     int dw, dh, aspect;
115     uchar size_range_set;
116     // cursor stuff
117     Fl_Cursor cursor_default;
118     #if FLTK_ABI_VERSION < 10303
119     // legacy, not used
120     Fl_Color cursor_fg, cursor_bg;
121     #endif
122

```

```

123 protected:
124     struct shape_data_type {
125         int lw_;
126         int lh_;
127         Fl_Image* shape_;
128     #if defined(__APPLE__)
129         typedef struct CGImage* CGImageRef;
130         CGImageRef mask;
131     #endif
132     #endif
133     Fl_Bitmap *todelete_;
134     };
135
136 #if FLTK_ABI_VERSION < 10303 && !defined(FL_DOXYGEN)
137     static
138 #endif
139     shape_data_type *shape_data_;
140 private:
141     void shape_bitmap_(Fl_Image* b);
142     void shape_alpha_(Fl_Image* img, int offset);
143     void shape_pixmap_(Fl_Image* pixmap);
144 public:
145     void shape(const Fl_Image* img);
146     inline void shape(const Fl_Image& b) { shape(&b); }
147 #if ! (defined(WIN32) || defined(__APPLE__) || defined(FL_DOXYGEN))
148     void combine_mask(void);
149 #endif
150 private:
151
152     void size_range_();
153     void _Fl_Window(); // constructor innards
154     void fullscreen_x(); // platform-specific part of sending a window to full screen
155     void fullscreen_off_x(int X, int Y, int W, int H); // platform-specific part of leaving full screen
156
157     // unimplemented copy ctor and assignment operator
158     Fl_Window(const Fl_Window&);
159     Fl_Window& operator=(const Fl_Window&);
160
161 protected:
162
163     static Fl_Window *current_;
164     virtual void draw();
165     virtual void flush();
166
167     void force_position(int force) {
168         if (force) set_flag(FORCE_POSITION);
169         else clear_flag(FORCE_POSITION);
170     }
171     int force_position()const { return ((flags() & FORCE_POSITION)?1:0); }
172
173     void free_icons();
174 public:
175
176     Fl_Window(int w, int h, const char* title= 0);
177     Fl_Window(int x, int y, int w, int h, const char* title = 0);
178     virtual ~Fl_Window();
179
180     virtual int handle(int);
181
182     virtual void resize(int X,int Y,int W,int H);
183     void border(int b);
184     void clear_border() {set_flag(NO_BORDER);}
185     unsigned int border()const {return !(flags() & NO_BORDER);}
186     void set_override() {set_flag(NO_BORDER|OVERRIDE);}
187     unsigned int override() const { return flags()&OVERRIDE; }
188     void set_modal() {set_flag(MODAL);}
189     unsigned int modal()const {return flags() & MODAL;}
190     void set_non_modal() {set_flag(NON_MODAL);}
191     unsigned int non_modal()const {return flags() & (NON_MODAL|MODAL);}
192
193     void clear_modal_states() {clear_flag(NON_MODAL | MODAL);}
194
195     void set_menu_window() {set_flag(MENU_WINDOW);}
196     unsigned int menu_window()const {return flags() & MENU_WINDOW;}
197
198     void set_tooltip_window() { set_flag(TOOLTIP_WINDOW);
199         clear_flag(MENU_WINDOW); }
200     unsigned int tooltip_window()const {return flags() & TOOLTIP_WINDOW;}
201
202     void hotspot(int x, int y, int offscreen = 0);
203     void hotspot(const Fl_Widget*, int offscreen = 0);
204     void hotspot(const Fl_Widget& p, int offscreen = 0) {hotspot(&p,offscreen);}
205
206     void free_position() {clear_flag(FORCE_POSITION);}
207     void size_range(int minw, int minh, int maxw=0, int maxh=0, int dw=0, int dh=0, int aspect=0) {

```

```

439     this->minw   = minw;
440     this->minh   = minh;
441     this->maxw   = maxw;
442     this->maxh   = maxh;
443     this->dw     = dw;
444     this->dh     = dh;
445     this->aspect = aspect;
446     size_range_();
447 }
448
449 const char* label()const {return Fl_Widget::label();}
450 const char* iconlabel()const {return iconlabel_;}
451 void label(const char*);
452 void iconlabel(const char*);
453 void label(const char* label, const char* iconlabel); // platform dependent
454 void copy_label(const char* a);
455
456 static void default_xclass(const char*);
457 static const char *default_xclass();
458 const char* xclass() const;
459 void xclass(const char* c);
460
461 static void default_icon(const Fl_RGB_Image*);
462 static void default_icons(const Fl_RGB_Image*[], int);
463 void icon(const Fl_RGB_Image*);
464 void icons(const Fl_RGB_Image*[], int);
465
466 #ifdef WIN32
467     static void default_icons(HICON big_icon, HICON small_icon);
468     void icons(HICON big_icon, HICON small_icon);
469 #endif
470
471 /* for legacy compatibility */
472 const void* icon() const;
473 void icon(const void * ic);
474
475 int shown() {return i != 0;}
476 virtual void show();
477 virtual void hide();
478 void show(int argc, char **argv);
479
480 // Enables synchronous show(), docs in Fl_Window.cxx
481 void wait_for_expose();
482
483 void fullscreen();
484 void fullscreen_off();
485 void fullscreen_off(int X,int Y,int W,int H);
486 unsigned int fullscreen_active()const { return flags() & FULLSCREEN; }
487 void fullscreen_screens(int top, int bottom, int left, int right);
488 void iconize();
489
490 int x_root() const ;
491 int y_root() const ;
492
493 static Fl_Window *current();
494 void make_current();
495
496 // Note: Doxygen docs in Fl_Widget.H to avoid redundancy.
497 virtual Fl_Window* as_window() { return this; }
498
499 void cursor(Fl_Cursor);
500 void cursor(const Fl_RGB_Image*, int, int);
501 void default_cursor(Fl_Cursor);
502
503 /* for legacy compatibility */
504 void cursor(Fl_Cursor c, Fl_Color, Fl_Color=FL_WHITE);
505 void default_cursor(Fl_Cursor c, Fl_Color, Fl_Color=FL_WHITE);
506
507 static void default_callback(Fl_Window*, void* v);
508
509 int decorated_w();
510 int decorated_h();
511 };
512 #endif
513
514 //
515 // End of "$Id$".
516 //

```

32.154 Fl_Wizard.H

```

1 //
2 // "$Id$"

```

```

3 //
4 // Fl_Wizard widget definitions.
5 //
6 // Copyright 1999-2010 by Easy Software Products.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_Wizard widget . */
21
22 //
23 // Include necessary header files...
24 //
25
26 #ifndef _Fl_Wizard_H_
27 # define _Fl_Wizard_H_
28
29 # include <FL/Fl_Group.H>
30
31
32 class FL_EXPORT Fl_Wizard : public Fl_Group {
33
34     Fl_Widget *value_;
35
36     void draw();
37
38     public:
39
40     Fl_Wizard(int, int, int, int, const char * = 0);
41
42     void next();
43     void prev();
44     Fl_Widget *value();
45     void value(Fl_Widget *);
46 };
47
48 #endif // !_Fl_Wizard_H_
49
50 //
51 // End of "$Id$".
52 //

```

32.155 Fl_XBM_Image.H

```

1 //
2 // "$Id$"
3 //
4 // XBM image header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 /* \file
20 Fl_XBM_Image class . */
21
22 #ifndef Fl_XBM_Image_H
23 #define Fl_XBM_Image_H
24 # include "Fl_Bitmap.H"
25
26 class FL_EXPORT Fl_XBM_Image : public Fl_Bitmap {
27
28     public:
29
30     Fl_XBM_Image(const char* filename);
31 };
32
33

```

```
37 #endif // !Fl_XBM_Image_H
38
39 //
40 // End of "$Id$".
41 //
```

32.156 Fl_XPM_Image.H

```
1 //
2 // "$Id$"
3 //
4 // XPM image header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 /* \file
20
21 Fl_XPM_Image class . */
22
23 #ifndef Fl_XPM_Image_H
24 #define Fl_XPM_Image_H
25 # include "Fl_Pixmap.H"
26
27 class FL_EXPORT Fl_XPM_Image : public Fl_Pixmap {
28 public:
29     Fl_XPM_Image(const char* filename);
30 };
31
32 #endif // !Fl_XPM_Image
33
34 //
35 // End of "$Id$".
36 //
```

32.157 forms.H

```
1 //
2 // "$Id$"
3 //
4 // Forms emulation header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2011 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 #ifndef __FORMS_H__
20 #define __FORMS_H__
21
22 #include "Fl.H"
23 #include "Fl_Group.H"
24 #include "Fl_Window.H"
25 #include "fl_draw.H"
26
27 typedef Fl_Widget FL_OBJECT;
28 typedef Fl_Window FL_FORM;
29
30 // Random constants & symbols defined by forms.h file:
31
32
33 #ifndef NULL
34 #define NULL 0
35 #endif
```

```

35 #endif
36 #ifndef FALSE
37 #define FALSE 0
38 #define TRUE 1
39 #endif
40
41 #define FL_ON 1
42 #define FL_OK 1
43 #define FL_VALID 1
44 #define FL_PREEMPT 1
45 #define FL_AUTO 2
46 #define FL_WHEN_NEEDED FL_AUTO
47 #define FL_OFF 0
48 #define FL_NONE 0
49 #define FL_CANCEL 0
50 #define FL_INVALID 0
51 #define FL_IGNORE -1
52 //#define FL_CLOSE -2 // this variable is never used in FLTK Forms. It is removed because it
    conflicts with the window FL_CLOSE event
53
54 #define FL_LCOL FL_BLACK
55 #define FL_COL1 FL_GRAY
56 #define FL_MCOL FL_LIGHT1
57 #define FL_LEFT_BCOL FL_LIGHT3 // 53 is better match
58 #define FL_TOP_BCOL FL_LIGHT2 // 51
59 #define FL_BOTTOM_BCOL FL_DARK2 // 40
60 #define FL_RIGHT_BCOL FL_DARK3 // 36
61 #define FL_INACTIVE FL_INACTIVE_COLOR
62 #define FL_INACTIVE_COL FL_INACTIVE_COLOR
63 #define FL_FREE_COL1 FL_FREE_COLOR
64 #define FL_FREE_COL2 ((Fl_Color)(FL_FREE_COLOR+1))
65 #define FL_FREE_COL3 ((Fl_Color)(FL_FREE_COLOR+2))
66 #define FL_FREE_COL4 ((Fl_Color)(FL_FREE_COLOR+3))
67 #define FL_FREE_COL5 ((Fl_Color)(FL_FREE_COLOR+4))
68 #define FL_FREE_COL6 ((Fl_Color)(FL_FREE_COLOR+5))
69 #define FL_FREE_COL7 ((Fl_Color)(FL_FREE_COLOR+6))
70 #define FL_FREE_COL8 ((Fl_Color)(FL_FREE_COLOR+7))
71 #define FL_FREE_COL9 ((Fl_Color)(FL_FREE_COLOR+8))
72 #define FL_FREE_COL10 ((Fl_Color)(FL_FREE_COLOR+9))
73 #define FL_FREE_COL11 ((Fl_Color)(FL_FREE_COLOR+10))
74 #define FL_FREE_COL12 ((Fl_Color)(FL_FREE_COLOR+11))
75 #define FL_FREE_COL13 ((Fl_Color)(FL_FREE_COLOR+12))
76 #define FL_FREE_COL14 ((Fl_Color)(FL_FREE_COLOR+13))
77 #define FL_FREE_COL15 ((Fl_Color)(FL_FREE_COLOR+14))
78 #define FL_FREE_COL16 ((Fl_Color)(FL_FREE_COLOR+15))
79 #define FL_TOMATO ((Fl_Color)(131))
80 #define FL_INDIANRED ((Fl_Color)(164))
81 #define FL_SLATEBLUE ((Fl_Color)(195))
82 #define FL_DARKGOLD ((Fl_Color)(84))
83 #define FL_PALEGREEN ((Fl_Color)(157))
84 #define FL_ORCHID ((Fl_Color)(203))
85 #define FL_DARKCYAN ((Fl_Color)(189))
86 #define FL_DARKTOMATO ((Fl_Color)(113))
87 #define FL_WHEAT ((Fl_Color)(174))
88
89 #define FL_ALIGN_BESIDE FL_ALIGN_INSIDE
90
91 #define FL_PUP_TOGGLE 2 // FL_MENU_TOGGLE
92 #define FL_PUP_INACTIVE 1 // FL_MENU_INACTIVE
93 #define FL_NO_FRAME FL_NO_BOX
94 #define FL_ROUNDED3D_UPBOX FL_ROUND_UP_BOX
95 #define FL_ROUNDED3D_DOWNBOX FL_ROUND_DOWN_BOX
96 #define FL_OVAL3D_UPBOX FL_ROUND_UP_BOX
97 #define FL_OVAL3D_DOWNBOX FL_ROUND_DOWN_BOX
98
99 #define FL_MBUTTON1 1
100 #define FL_LEFTMOUSE 1
101 #define FL_MBUTTON2 2
102 #define FL_MIDDLEMOUSE 2
103 #define FL_MBUTTON3 3
104 #define FL_RIGHTMOUSE 3
105 #define FL_MBUTTON4 4
106 #define FL_MBUTTON5 5
107
108 #define FL_INVALID_STYLE 255
109 #define FL_NORMAL_STYLE FL_HELVETICA
110 #define FL_BOLD_STYLE FL_HELVETICA_BOLD
111 #define FL_ITALIC_STYLE FL_HELVETICA_ITALIC
112 #define FL_BOLDITALIC_STYLE FL_HELVETICA_BOLD_ITALIC
113 #define FL_FIXED_STYLE FL_COURIER
114 #define FL_FIXEDBOLD_STYLE FL_COURIER_BOLD
115 #define FL_FIXEDITALIC_STYLE FL_COURIER_ITALIC
116 #define FL_FIXEDBOLDITALIC_STYLE FL_COURIER_BOLD_ITALIC
117 #define FL_TIMES_STYLE FL_TIMES
118 #define FL_TIMESBOLD_STYLE FL_TIMES_BOLD
119 #define FL_TIMESITALIC_STYLE FL_TIMES_ITALIC
120 #define FL_TIMESBOLDITALIC_STYLE FL_TIMES_BOLD_ITALIC

```

```

121
122 // hacks to change the labeltype() when passed to fl_set_object_lstyle():
123 #define FL_SHADOW_STYLE      (FL_SHADOW_LABEL<<8)
124 #define FL_ENGRAVED_STYLE    (FL_ENGRAVED_LABEL<<8)
125 #define FL_EMBOSSSED_STYLE   (FL_EMBOSSSED_LABEL<<0)
126
127 // size values are different from XForms, match older Forms:
128 #define FL_TINY_SIZE        8
129 #define FL_SMALL_SIZE      11 // 10
130 // #define FL_NORMAL_SIZE    14 // 12
131 #define FL_MEDIUM_SIZE     18 // 14
132 #define FL_LARGE_SIZE      24 // 18
133 #define FL_HUGE_SIZE        32 // 24
134 #define FL_DEFAULT_SIZE    FL_SMALL_SIZE
135 #define FL_TINY_FONT        FL_TINY_SIZE
136 #define FL_SMALL_FONT       FL_SMALL_SIZE
137 #define FL_NORMAL_FONT      FL_NORMAL_SIZE
138 #define FL_MEDIUM_FONT      FL_MEDIUM_SIZE
139 #define FL_LARGE_FONT       FL_LARGE_SIZE
140 #define FL_HUGE_FONT        FL_HUGE_SIZE
141 #define FL_NORMAL_FONT1     FL_SMALL_FONT
142 #define FL_NORMAL_FONT2     FL_NORMAL_FONT
143 #define FL_DEFAULT_FONT     FL_SMALL_FONT
144
145 #define FL_RETURN_END_CHANGED  FL_WHEN_RELEASE
146 #define FL_RETURN_CHANGED     FL_WHEN_CHANGED
147 #define FL_RETURN_END         FL_WHEN_RELEASE_ALWAYS
148 #define FL_RETURN_ALWAYS      (FL_WHEN_CHANGED|FL_WHEN_NOT_CHANGED)
149
150 #define FL_BOUND_WIDTH        3
151
152 typedef int FL_Coord;
153 typedef int FL_COLOR;
154
155 // fltk interaction:
156
157 #define FL_CMD_OPT void
158 extern FL_EXPORT void fl_initialize(int*, char*[], const char*, FL_CMD_OPT*, int);
159 inline void fl_finish() {}
160
161 typedef void (*FL_IO_CALLBACK) (FL_SOCKET, void*);
162 inline void fl_add_io_callback(int fd, short w, FL_IO_CALLBACK cb, void* v) {
163     Fl::add_fd(fd, w, cb, v);}
164 inline void fl_remove_io_callback(int fd, short, FL_IO_CALLBACK) {
165     Fl::remove_fd(fd);} // removes all the callbacks!
166
167 // type of callback is different and no "id" number is returned:
168 inline void fl_add_timeout(long msec, void (*cb)(void*), void* v) {
169     Fl::add_timeout(msec*.001, cb, v);}
170 inline void fl_remove_timeout(int) {}
171
172 // type of callback is different!
173 inline void fl_set_idle_callback(void (*cb)()) {Fl::set_idle(cb);}
174
175 FL_EXPORT Fl_Widget* fl_do_forms(void);
176 FL_EXPORT Fl_Widget* fl_check_forms();
177 inline Fl_Widget* fl_do_only_forms(void) {return fl_do_forms();}
178 inline Fl_Widget* fl_check_only_forms(void) {return fl_check_forms();}
179
180 // because of new redraw behavior, these are no-ops:
181 inline void fl_freeze_object(Fl_Widget*) {}
182 inline void fl_unfreeze_object(Fl_Widget*) {}
183 inline void fl_freeze_form(Fl_Window*) {}
184 inline void fl_unfreeze_form(Fl_Window*) {}
185 inline void fl_freeze_all_forms() {}
186 inline void fl_unfreeze_all_forms() {}
187
188 inline void fl_set_focus_object(Fl_Window*, Fl_Widget* o) {Fl::focus(o);}
189 inline void fl_reset_focus_object(Fl_Widget* o) {Fl::focus(o);}
190 #define fl_set_object_focus fl_set_focus_object
191
192 // void fl_set_form_atclose(Fl_Window*w,int (*cb)(Fl_Window*,void*),void* v)
193 // void fl_set_atclose(int (*cb)(Fl_Window*,void*),void*)
194 // fl_set_form_atactivate/atdeactivate not implemented!
195
196 // Fl_Widget:
197
198 inline void fl_set_object_boxtype(Fl_Widget* o, Fl_Boxtype a) {o->box(a);}
199 inline void fl_set_object_lsize(Fl_Widget* o,int s) {o->labelsize(s);}
200
201 /* forms lib font indexes must be byte sized - extract correct byte from style word */
202 inline void fl_set_object_lstyle(Fl_Widget* o,int a) {
203     o->labelfont((Fl_Font)(a&0xff)); o->labeltype((Fl_Labeltype)(a>>8));}
204 inline void fl_set_object_lcol(Fl_Widget* o, Fl_Color a) {o->labelcolor(a);}
205 #define fl_set_object_lcolor fl_set_object_lcol
206 inline void fl_set_object_lalign(Fl_Widget* o, Fl_Align a) {o->align(a);}
207 #define fl_set_object_align fl_set_object_lalign

```

```

210 inline void fl_set_object_color(Fl_Widget* o, Fl_Color a, Fl_Color b) {o->color(a,b);}
211 inline void fl_set_object_label(Fl_Widget* o, const char* a) {o->label(a); o->redraw();}
212 inline void fl_set_object_position(Fl_Widget* o, int x, int y) {o->position(x,y);}
213 inline void fl_set_object_size(Fl_Widget* o, int w, int h) {o->size(w,h);}
214 inline void fl_set_object_geometry(Fl_Widget* o, int x, int y, int w, int h) {o->resize(x,y,w,h);}
215
216 inline void fl_get_object_geometry(Fl_Widget* o, int*x, int*y, int*w, int*h) {
217     *x = o->x(); *y = o->y(); *w = o->w(); *h = o->h();}
218 inline void fl_get_object_position(Fl_Widget* o, int*x, int*y) {
219     *x = o->x(); *y = o->y();}
220
221 typedef void (*Forms_CB)(Fl_Widget*, long);
222 inline void fl_set_object_callback(Fl_Widget* o, Forms_CB c, long a) {o->callback(c,a);}
223 #define fl_set_call_back      fl_set_object_callback
224 inline void fl_call_object_callback(Fl_Widget* o) {o->do_callback();}
225 inline void fl_trigger_object(Fl_Widget* o) {o->do_callback();}
226 inline void fl_set_object_return(Fl_Widget* o, int v) {
227     o->when((Fl_When)(v|FL_WHEN_RELEASE));}
228
229 inline void fl_redraw_object(Fl_Widget* o) {o->redraw();}
230 inline void fl_show_object(Fl_Widget* o) {o->show();}
231 inline void fl_hide_object(Fl_Widget* o) {o->hide();}
232 inline void fl_free_object(Fl_Widget* x) {delete x;}
233 inline void fl_delete_object(Fl_Widget* o) {o->parent()->remove(*o);}
234 inline void fl_activate_object(Fl_Widget* o) {o->activate();}
235 inline void fl_deactivate_object(Fl_Widget* o) {o->deactivate();}
236
237 inline void fl_add_object(Fl_Window* f, Fl_Widget* x) {f->add(x);}
238 inline void fl_insert_object(Fl_Widget* o, Fl_Widget* b) {b->parent()->insert(*o,b);}
239
240 inline Fl_Window* FL_ObjWin(Fl_Widget* o) {return o->window();}
241
242 // things that appered in the demos a lot that I don't emulate, but
243 // I did not want to edit out of all the demos...
244
245 inline int fl_get_border_width() {return 3;}
246 inline void fl_set_border_width(int) {}
247 inline void fl_set_object_dblbuffer(Fl_Widget*, int) {}
248 inline void fl_set_form_dblbuffer(Fl_Window*, int) {}
249
250 // Fl_Window:
251
252 inline void fl_free_form(Fl_Window* x) {delete x;}
253 inline void fl_redraw_form(Fl_Window* f) {f->redraw();}
254
255 inline Fl_Window* fl_bgn_form(Fl_Boxtype b, int w, int h) {
256     Fl_Window* g = new Fl_Window(w,h,0);
257     g->box(b);
258     return g;
259 }
260 FL_EXPORT void fl_end_form();
261 inline void fl_addto_form(Fl_Window* f) {f->begin();}
262 inline Fl_Group* fl_bgn_group() {return new Fl_Group(0,0,0,0);}
263 inline void fl_end_group() {Fl_Group::current()->forms_end();}
264 inline void fl_addto_group(Fl_Widget* o) {((Fl_Group*) o)->begin();}
265 #define resizebox__ddfdesign_kludge()
266
267 inline void fl_scale_form(Fl_Window* f, double x, double y) {
268     f->resizable(f); f->size(int(f->w()*x),int(f->h()*y));}
269 inline void fl_set_form_position(Fl_Window* f, int x, int y) {f->position(x,y);}
270 inline void fl_set_form_size(Fl_Window* f, int w, int h) {f->size(w,h);}
271 inline void fl_set_form_geometry(Fl_Window* f, int x, int y, int w, int h) {
272     f->resize(x,y,w,h);}
273 #define fl_set_initial_placement fl_set_form_geometry
274 inline void fl_adjust_form_size(Fl_Window*) {}
275
276 FL_EXPORT void fl_show_form(Fl_Window* f, int p, int b, const char* n);
277 enum { // "p" argument values:
278     FL_PLACE_FREE = 0, // make resizable
279     FL_PLACE_MOUSE = 1, // mouse centered on form
280     FL_PLACE_CENTER = 2, // center of the screen
281     FL_PLACE_POSITION = 4, // fixed position, resizable
282     FL_PLACE_SIZE = 8, // fixed size, normal fltk behavior
283     FL_PLACE_GEOMETRY = 16, // fixed size and position
284     FL_PLACE_ASPECT = 32, // keep aspect ratio (ignored)
285     FL_PLACE_FULLSCREEN = 64, // fill screen
286     FL_PLACE_HOTSPOT = 128, // enables hotspot
287     FL_PLACE_ICONIC = 256, // iconic (ignored)
288     FL_FREE_SIZE = (1<14), // force resizable
289     FL_FIX_SIZE = (1<15) // force off resizable
290 };
291 #define FL_PLACE_FREE_CENTER (FL_PLACE_CENTER|FL_FREE_SIZE)
292 #define FL_PLACE_CENTERFREE (FL_PLACE_CENTER|FL_FREE_SIZE)
293 enum { // "b" argument values:
294     FL_NOBORDER = 0,
295     FL_FULLBORDER,
296     FL_TRANSIENT
297 };

```



```

299 //FL_MODAL = (1<<8)      // not implemented yet in Forms
300 };
301 inline void fl_set_form_hotspot(Fl_Window* w,int x,int y) {w->hotspot(x,y);}
302 inline void fl_set_form_hotobject(Fl_Window* w, Fl_Widget* o) {w->hotspot(o);}
303 extern FL_EXPORT char fl_flip; // in forms.C
304 inline void fl_flip_yorigin() {fl_flip = 1;}
305
306 #define fl_prepare_form_window fl_show_form
307 inline void fl_show_form_window(Fl_Window*) {}
308
309 inline void fl_raise_form(Fl_Window* f) {f->show();}
310
311 inline void fl_hide_form(Fl_Window* f) {f->hide();}
312 inline void fl_pop_form(Fl_Window* f) {f->show();}
313
314 extern FL_EXPORT char fl_modal_next; // in forms.C
315 inline void fl_activate_all_forms() {}
316 inline void fl_deactivate_all_forms() {fl_modal_next = 1;}
317 inline void fl_deactivate_form(Fl_Window*w) {w->deactivate();}
318 inline void fl_activate_form(Fl_Window*w) {w->activate();}
319
320 inline void fl_set_form_title(Fl_Window* f, const char* s) {f->label(s);}
321 inline void fl_title_form(Fl_Window* f, const char* s) {f->label(s);}
322
323 typedef void (*Forms_FormCB)(Fl_Widget*);
324 inline void fl_set_form_callback(Fl_Window* f,Forms_FormCB c) {f->callback(c);}
325 #define fl_set_form_call_back fl_set_form_callback
326
327 inline void fl_init() {}
328 FL_EXPORT void fl_set_graphics_mode(int,int);
329
330 inline int fl_form_is_visible(Fl_Window* f) {return f->visible();}
331
332 inline int fl_mouse_button() {return Fl::event_button();}
333 #define fl_mousebutton fl_mouse_button
334
335 #define fl_free      free
336 #define fl_malloc    malloc
337 #define fl_calloc    calloc
338 #define fl_realloc   realloc
339
340 // Drawing functions.    Only usable inside an Fl_Free object?
341
342
343 inline void fl_drw_box(Fl_Boxtype b,int x,int y,int w,int h,Fl_Color bgc,int=3) {
344     fl_draw_box(b,x,y,w,h,bgc);}
345 inline void fl_drw_frame(Fl_Boxtype b,int x,int y,int w,int h,Fl_Color bgc,int=3) {
346     fl_draw_box(b,x,y,w,h,bgc);}
347
348 inline void fl_drw_text(Fl_Align align, int x, int y, int w, int h,
349     Fl_Color fgcolor, int size, Fl_Font style,
350     const char* s) {
351     fl_font(style,size);
352     fl_color(fgcolor);
353     fl_draw(s,x,y,w,h,align);
354 }
355
356 // this does not work except for CENTER...
357 inline void fl_drw_text_beside(Fl_Align align, int x, int y, int w, int h,
358     Fl_Color fgcolor, int size, Fl_Font style,
359     const char* s) {
360     fl_font(style,size);
361     fl_color(fgcolor);
362     fl_draw(s,x,y,w,h,align);
363 }
364
365 inline void fl_set_font_name(Fl_Font n,const char* s) {Fl::set_font(n,s);}
366
367 inline void fl_mapcolor(Fl_Color c, uchar r, uchar g, uchar b) {Fl::set_color(c,r,g,b);}
368
369 #define fl_set_clipping(x,y,w,h) fl_push_clip(x,y,w,h)
370 #define fl_unset_clipping() fl_pop_clip()
371
372 // Forms classes:
373
374
375 inline Fl_Widget* fl_add_new(Fl_Widget* p) {return p;}
376 inline Fl_Widget* fl_add_new(uchar t,Fl_Widget* p) {p->type(t); return p;}
377
378 #define forms_constructor(type,name) \
379 inline type* name(uchar t,int x,int y,int w,int h,const char* l) { \
380 return (type*)(fl_add_new(t, new type(x,y,w,h,l)));}
381 #define forms_constructort(type,name) \
382 inline type* name(uchar t,int x,int y,int w,int h,const char* l) { \
383 return (type*)(fl_add_new(new type(t,x,y,w,h,l)));}
384 #define forms_constructorb(type,name) \
385 inline type* name(Fl_Boxtype t,int x,int y,int w,int h,const char* l) { \
386 return (type*)(fl_add_new(new type(t,x,y,w,h,l)));}
387

```

```

388 #include "Fl_FormsBitmap.H"
389 #define FL_NORMAL_BITMAP FL_NO_BOX
390 forms_constructorb(Fl_FormsBitmap, fl_add_bitmap)
391 inline void fl_set_bitmap_data(Fl_Widget* o, int w, int h, const uchar* b) {
392     ((Fl_FormsBitmap*)o)->set(w,h,b);
393 }
394
395 #include "Fl_FormsPixmap.H"
396 #define FL_NORMAL_PIXMAP FL_NO_BOX
397 forms_constructorb(Fl_FormsPixmap, fl_add_pixmap)
398 inline void fl_set_pixmap_data(Fl_Widget* o, char*const* b) {
399     ((Fl_FormsPixmap*)o)->set(b);
400 }
401 //inline void fl_set_pixmap_file(Fl_Widget*, const char*);
402 inline void fl_set_pixmap_align(Fl_Widget* o, Fl_Align a, int, int) {o->align(a);}
403 //inline void fl_set_pixmap_colorcloseness(int, int, int);
404
405 #include "Fl_Box.H"
406 forms_constructorb(Fl_Box, fl_add_box)
407
408 #include "Fl_Browser.H"
409 forms_constructor(Fl_Browser, fl_add_browser)
410
411 inline void fl_clear_browser(Fl_Widget* o) {
412     ((Fl_Browser*)o)->clear();}
413 inline void fl_add_browser_line(Fl_Widget* o, const char* s) {
414     ((Fl_Browser*)o)->add(s);}
415 inline void fl_addto_browser(Fl_Widget* o, const char* s) {
416     ((Fl_Browser*)o)->add(s);} /* should also scroll to bottom */
417 //inline void fl_addto_browser_chars(Fl_Widget*, const char*)
418 //define fl_append_browser fl_addto_browser_chars
419 inline void fl_insert_browser_line(Fl_Widget* o, int n, const char* s) {
420     ((Fl_Browser*)o)->insert(n,s);}
421 inline void fl_delete_browser_line(Fl_Widget* o, int n) {
422     ((Fl_Browser*)o)->remove(n);}
423 inline void fl_replace_browser_line(Fl_Widget* o, int n, const char* s) {
424     ((Fl_Browser*)o)->replace(n,s);}
425 inline char* fl_get_browser_line(Fl_Widget* o, int n) {
426     return (char*)((Fl_Browser*)o)->text(n);}
427 inline int fl_load_browser(Fl_Widget* o, const char* f) {
428     return ((Fl_Browser*)o)->load(f);}
429 inline void fl_select_browser_line(Fl_Widget* o, int n) {
430     ((Fl_Browser*)o)->select(n,1);}
431 inline void fl_deselect_browser_line(Fl_Widget* o, int n) {
432     ((Fl_Browser*)o)->select(n,0);}
433 inline void fl_deselect_browser(Fl_Widget* o) {
434     ((Fl_Browser*)o)->deselect();}
435 inline int fl_isspace_browser_line(Fl_Widget* o, int n) {
436     return ((Fl_Browser*)o)->selected(n);}
437 inline int fl_get_browser_topline(Fl_Widget* o) {
438     return ((Fl_Browser*)o)->topline();}
439 inline int fl_get_browser(Fl_Widget* o) {
440     return ((Fl_Browser*)o)->value();}
441 inline int fl_get_browser_maxline(Fl_Widget* o) {
442     return ((Fl_Browser*)o)->size();}
443 //inline int fl_get_browser_screenlines(Fl_Widget*);
444 inline void fl_set_browser_topline(Fl_Widget* o, int n) {
445     ((Fl_Browser*)o)->topline(n);}
446 inline void fl_set_browser_fontsize(Fl_Widget* o, int s) {
447     ((Fl_Browser*)o)->textsize(s);}
448 inline void fl_set_browser_fontstyle(Fl_Widget* o, Fl_Font s) {
449     ((Fl_Browser*)o)->textfont(s);}
450 inline void fl_set_browser_specialkey(Fl_Widget* o, char c) {
451     ((Fl_Browser*)o)->format_char(c);}
452 //inline void fl_set_browser_vscrollbar(Fl_Widget*, int);
453 //inline void fl_set_browser_hscrollbar(Fl_Widget*, int);
454 //inline void fl_set_browser_leftslider(Fl_Widget*, int);
455 //define fl_set_browser_leftscrollbar fl_set_browser_leftslider
456 //inline void fl_set_browser_line_selectable(Fl_Widget*, int, int);
457 //inline void fl_get_browser_dimension(Fl_Widget*,int*,int*,int*,int*);
458 //inline void fl_set_browser_dbclick_callback(Fl_Widget*,FL_CALLBACKPTR,long);
459 //inline void fl_set_browser_xoffset(Fl_Widget*, FL_Coord);
460 //inline void fl_set_browser_scrollbarsize(Fl_Widget*, int, int);
461 inline void fl_setdisplayed_browser_line(Fl_Widget* o, int n, int i) {
462     ((Fl_Browser*)o)->display(n,i);}
463 inline int fl_isdisplayed_browser_line(Fl_Widget* o, int n) {
464     return ((Fl_Browser*)o)->displayed(n);}
465
466 #include "Fl_Button.H"
467
468 #define FL_NORMAL_BUTTON 0
469 #define FL_TOUCH_BUTTON 4
470 #define FL_INOUT_BUTTON 5
471 #define FL_RETURN_BUTTON 6
472 #define FL_HIDDEN_RET_BUTTON 7
473 #define FL_PUSH_BUTTON FL_TOGGLE_BUTTON
474 #define FL_MENU_BUTTON 9

```

```

475
476 FL_EXPORT Fl_Button* fl_add_button(uchar t,int x,int y,int w,int h,const char* l);
477 inline int fl_get_button(Fl_Widget* b) {return ((Fl_Button*)b)->value();}
478 inline void fl_set_button(Fl_Widget* b, int v) {((Fl_Button*)b)->value(v);}
479 inline int fl_get_button_num(Fl_Widget*) {return Fl::event_button();}
480 inline void fl_set_button_shortcut(Fl_Widget* b, const char* s,int=0) {
481     ((Fl_Button*)b)->shortcut(s);}
482 // #define fl_set_object_shortcut(b,s) fl_set_button_shortcut(b,s)
483
484 #include "Fl_Light_Button.H"
485 forms_constructor(Fl_Light_Button, fl_add_lightbutton)
486
487 #include "Fl_Round_Button.H"
488 forms_constructor(Fl_Round_Button, fl_add_roundbutton)
489 forms_constructor(Fl_Round_Button, fl_add_round3dbutton)
490
491 #include "Fl_Check_Button.H"
492 forms_constructor(Fl_Check_Button, fl_add_checkbutton)
493
494 inline Fl_Widget* fl_add_bitmapbutton(int t,int x,int y,int w,int h,const char* l) {Fl_Widget* o =
    fl_add_button(t,x,y,w,h,l); return o;}
495 inline void fl_set_bitmapbutton_data(Fl_Widget* o,int a,int b,uchar* c) {
496     (new Fl_Bitmap(c,a,b))->label(o);} // does not delete old Fl_Bitmap!
497
498 inline Fl_Widget* fl_add_pixmapbutton(int t,int x,int y,int w,int h,const char* l) {Fl_Widget* o =
    fl_add_button(t,x,y,w,h,l); return o;}
499 inline void fl_set_pixmapbutton_data(Fl_Widget* o, const char*const* c) {
500     (new Fl_Pixmap(c))->label(o);} // does not delete old Fl_Pixmap!
501
502 // Fl_Canvas object not yet implemented!
503
504 #include "Fl_Chart.H"
505
506 forms_constructor(Fl_Chart, fl_add_chart)
507 inline void fl_clear_chart(Fl_Widget* o) {
508     ((Fl_Chart*)o)->clear();}
509 inline void fl_add_chart_value(Fl_Widget* o,double v,const char* s,uchar c){
510     ((Fl_Chart*)o)->add(v,s,c);}
511 inline void fl_insert_chart_value(Fl_Widget* o, int i, double v, const char* s, uchar c) {
512     ((Fl_Chart*)o)->insert(i,v,s,c);}
513 inline void fl_replace_chart_value(Fl_Widget* o, int i, double v, const char* s, uchar c) {
514     ((Fl_Chart*)o)->replace(i,v,s,c);}
515 inline void fl_set_chart_bounds(Fl_Widget* o, double a, double b) {
516     ((Fl_Chart*)o)->bounds(a,b);}
517 inline void fl_set_chart_maxnumb(Fl_Widget* o, int v) {
518     ((Fl_Chart*)o)->maxsize(v);}
519 inline void fl_set_chart_autosize(Fl_Widget* o, int v) {
520     ((Fl_Chart*)o)->autosize(v);}
521 inline void fl_set_chart_lstyle(Fl_Widget* o, Fl_Font v) {
522     ((Fl_Chart*)o)->textfont(v);}
523 inline void fl_set_chart_lsize(Fl_Widget* o, int v) {
524     ((Fl_Chart*)o)->textsize(v);}
525 inline void fl_set_chart_lcolor(Fl_Widget* o, Fl_Color v) {
526     ((Fl_Chart*)o)->textcolor(v);}
527 #define fl_set_chart_lcol    fl_set_chart_lcolor
528
529 #include "Fl_Choice.H"
530
531 #define FL_NORMAL_CHOICE      0
532 #define FL_NORMAL_CHOICE2    0
533 #define FL_DROPLIST_CHOICE   0
534
535 forms_constructor(Fl_Choice, fl_add_choice)
536 inline void fl_clear_choice(Fl_Widget* o) {
537     ((Fl_Choice*)o)->clear();}
538 inline void fl_addto_choice(Fl_Widget* o, const char* s) {
539     ((Fl_Choice*)o)->add(s);}
540 inline void fl_replace_choice(Fl_Widget* o, int i, const char* s) {
541     ((Fl_Choice*)o)->replace(i-1,s);}
542 inline void fl_delete_choice(Fl_Widget* o, int i) {
543     ((Fl_Choice*)o)->remove(i-1);}
544 inline void fl_set_choice(Fl_Widget* o, int i) {
545     ((Fl_Choice*)o)->value(i-1);}
546 // inline void fl_set_choice_text(Fl_Widget*, const char*);
547 inline int fl_get_choice(Fl_Widget* o) {
548     return ((Fl_Choice*)o)->value()+1;}
549 // inline const char* fl_get_choice_item_text(Fl_Widget*, int);
550 // inline int fl_get_choice_maxitems(Fl_Widget*);
551 inline const char* fl_get_choice_text(Fl_Widget* o) {
552     return ((Fl_Choice*)o)->text();}
553 inline void fl_set_choice_fontsize(Fl_Widget* o, int x) {
554     ((Fl_Choice*)o)->textsize(x);}
555 inline void fl_set_choice_fontstyle(Fl_Widget* o, Fl_Font x) {
556     ((Fl_Choice*)o)->textfont(x);}
557 // inline void fl_set_choice_item_mode(Fl_Widget*, int, unsigned);
558 // inline void fl_set_choice_item_shortcut(Fl_Widget*, int, const char*);
559

```

```

560 #include "Fl_Clock.H"
561 forms_constructor(Fl_Clock, fl_add_clock)
562 inline void fl_get_clock(Fl_Widget* o, int* h, int* m, int* s) {
563     *h = ((Fl_Clock*)o)->hour();
564     *m = ((Fl_Clock*)o)->minute();
565     *s = ((Fl_Clock*)o)->second();
566 }
567
568 #include "Fl_Counter.H"
569 forms_constructor(Fl_Counter, fl_add_counter)
570 inline void fl_set_counter_value(Fl_Widget* o, double v) {
571     ((Fl_Counter*)o)->value(v);}
572 inline void fl_set_counter_bounds(Fl_Widget* o, double a, double b) {
573     ((Fl_Counter*)o)->bounds(a,b);}
574 inline void fl_set_counter_step(Fl_Widget* o, double a, double b) {
575     ((Fl_Counter*)o)->step(a,b);}
576 inline void fl_set_counter_precision(Fl_Widget* o, int v) {
577     ((Fl_Counter*)o)->precision(v);}
578 inline void fl_set_counter_return(Fl_Widget* o, int v) {
579     ((Fl_Counter*)o)->when((Fl_When)(v|FL_WHEN_RELEASE));}
580 inline double fl_get_counter_value(Fl_Widget* o) {
581     return ((Fl_Counter*)o)->value();}
582 inline void fl_get_counter_bounds(Fl_Widget* o, float* a, float* b) {
583     *a = float(((Fl_Counter*)o)->minimum());
584     *b = float(((Fl_Counter*)o)->maximum());
585 }
586 //inline void fl_set_counter_filter(Fl_Widget*,const char* (*)(Fl_Widget*,double,int));
587
588 // Cursor stuff cannot be emulated because it uses X stuff
589 inline void fl_set_cursor(Fl_Window* w, Fl_Cursor c) {w->cursor(c);}
590 #define FL_INVISIBLE_CURSOR FL_CURSOR_NONE
591 #define FL_DEFAULT_CURSOR FL_CURSOR_DEFAULT
592
593 #include "Fl_Dial.H"
594
595 #define FL_DIAL_COL1 FL_GRAY
596 #define FL_DIAL_COL2 37
597
598 forms_constructor(Fl_Dial, fl_add_dial)
599 inline void fl_set_dial_value(Fl_Widget* o, double v) {
600     ((Fl_Dial*)o)->value(v);}
601 inline double fl_get_dial_value(Fl_Widget* o) {
602     return ((Fl_Dial*)o)->value();}
603 inline void fl_set_dial_bounds(Fl_Widget* o, double a, double b) {
604     ((Fl_Dial*)o)->bounds(a, b);}
605 inline void fl_get_dial_bounds(Fl_Widget* o, float* a, float* b) {
606     *a = float(((Fl_Dial*)o)->minimum());
607     *b = float(((Fl_Dial*)o)->maximum());
608 }
609 inline void fl_set_dial_return(Fl_Widget* o, int i) {
610     ((Fl_Dial*)o)->when((Fl_When)(i|FL_WHEN_RELEASE));}
611 inline void fl_set_dial_angles(Fl_Widget* o, int a, int b) {
612     ((Fl_Dial*)o)->angles((short)a, (short)b);}
613 //inline void fl_set_dial_cross(Fl_Widget* o, int);
614 // inline void fl_set_dial_direction(Fl_Widget* o, uchar d) {
615 //     ((Fl_Dial*)o)->direction(d);}
616 inline void fl_set_dial_step(Fl_Widget* o, double v) {
617     ((Fl_Dial*)o)->step(v);}
618
619 // Frames:
620
621 inline Fl_Widget* fl_add_frame(Fl_Boxtype i,int x,int y,int w,int h,const char* l) {
622     return fl_add_box(i,x-3,y-3,w+6,h+6,l);}
623
624 // labelframe nyi
625 inline Fl_Widget* fl_add_labelframe(Fl_Boxtype i,int x,int y,int w,int h,const char* l) {
626     Fl_Widget* o = fl_add_box(i,x-3,y-3,w+6,h+6,l);
627     o->align(FL_ALIGN_TOP_LEFT);
628     return o;
629 }
630
631 #include "Fl_Free.H"
632 inline Fl_Free*
633 fl_add_free(int t,double x,double y,double w,double h,const char* l,
634             FL_HANDLEPTR hdl) {
635     return (Fl_Free*)(fl_add_new(
636         new Fl_Free(t, int(x), int(y), int(w), int(h), l, hdl)));
637 }
638
639 #include "fl_ask.H"
640 #include "fl_show_colormap.H"
641
642 inline int fl_show_question(const char* c, int = 0) {return fl_choice("%s",fl_no,fl_yes,0L,c);}
643 FL_EXPORT void fl_show_message(const char *,const char *,const char *);
644 FL_EXPORT void fl_show_alert(const char *,const char *,const char *,int=0);
645 FL_EXPORT int fl_show_question(const char *,const char *,const char *);
646 inline const char *fl_show_input(const char *l,const char*d=0) {return fl_input("%s",d,l);}

```

```

647 FL_EXPORT /*const*/ char *fl_show_simple_input(const char *label, const char *deflt = 0);
648 FL_EXPORT int fl_show_choice(
649     const char *m1,
650     const char *m2,
651     const char *m3,
652     int numb,
653     const char *b0,
654     const char *b1,
655     const char *b2);
656
657 inline void fl_set_goodies_font(Fl_Font a, Fl_Fontsize b) {fl_message_font(a,b);}
658 #define fl_show_messages fl_message
659 inline int fl_show_choices(const char* c,int n,const char* b1,const char* b2,
660     const char* b3, int) {
661     return fl_show_choice(0,c,0,n,b1,b2,b3);
662 }
663
664 #include "filename.H"
665 #include "Fl_File_Chooser.H"
666 inline int do_matching(char* a, const char* b) {return fl_filename_match(a,b);}
667
668 // Forms-compatible file chooser (implementation in fselect.C):
669 FL_EXPORT char* fl_show_file_selector(const char* message,const char* dir,
670     const char* pat,const char* fname);
671 FL_EXPORT char* fl_get_directory();
672 FL_EXPORT char* fl_get_pattern();
673 FL_EXPORT char* fl_get_filename();
674
675 #include "Fl_Input.H"
676 forms_constructor(Fl_Input, fl_add_input)
677 inline void fl_set_input(Fl_Widget* o, const char* v) {
678     ((Fl_Input*)o)->value(v);}
679 inline void fl_set_input_return(Fl_Widget* o, int x) {
680     ((Fl_Input*)o)->when((Fl_When)(x | FL_WHEN_RELEASE));}
681 inline void fl_set_input_color(Fl_Widget* o, Fl_Color a, Fl_Color b) {
682     ((Fl_Input*)o)->textcolor(a);
683     ((Fl_Input*)o)->cursor_color(b);
684 }
685 // inline void fl_set_input_scroll(Fl_Widget*, int);
686 inline void fl_set_input_cursorpos(Fl_Widget* o, int x, int /*y*/) {
687     ((Fl_Input*)o)->position(x);}
688 // inline void fl_set_input_selected(Fl_Widget*, int);
689 // inline void fl_set_input_selected_range(Fl_Widget*, int, int);
690 // inline void fl_set_input_maxchars(Fl_Widget*, int);
691 // inline void fl_set_input_format(Fl_Widget*, int, int);
692 // inline void fl_set_input_hscrollbar(Fl_Widget*, int);
693 // inline void fl_set_input_vscrollbar(Fl_Widget*, int);
694 // inline void fl_set_input_xoffset(Fl_Widget*, int);
695 // inline void fl_set_input_topline(Fl_Widget*, int);
696 // inline void fl_set_input_scrollbarsize(Fl_Widget*, int, int);
697 // inline int fl_get_input_topline(Fl_Widget*);
698 // inline int fl_get_input_screenlines(Fl_Widget*);
699 inline int fl_get_input_cursorpos(Fl_Widget* o, int*x, int*y) {
700     *x = ((Fl_Input*)o)->position(); *y = 0; return *x;}
701 // inline int fl_get_input_numberoflines(Fl_Widget*);
702 // inline void fl_get_input_format(Fl_Widget*, int*, int*);
703 inline const char* fl_get_input(Fl_Widget* o) {return ((Fl_Input*)o)->value();}
704
705 #include "Fl_Menu_Button.H"
706
707 // types are not implemented, they all act like FL_PUSH_MENU:
708 #define FL_TOUCH_MENU 0
709 #define FL_PUSH_MENU 1
710 #define FL_PULLDOWN_MENU 2
711 forms_constructor(Fl_Menu_Button, fl_add_menu)
712
713 inline void fl_clear_menu(Fl_Widget* o) {
714     ((Fl_Menu_Button*)o)->clear();}
715 inline void fl_set_menu(Fl_Widget* o, const char* s) {
716     ((Fl_Menu_Button*)o)->clear(); ((Fl_Menu_Button*)o)->add(s);}
717 inline void fl_addto_menu(Fl_Widget* o, const char* s) {
718     ((Fl_Menu_Button*)o)->add(s);}
719 inline void fl_replace_menu_item(Fl_Widget* o, int i, const char* s) {
720     ((Fl_Menu_Button*)o)->replace(i-1,s);}
721 inline void fl_delete_menu_item(Fl_Widget* o, int i) {
722     ((Fl_Menu_Button*)o)->remove(i-1);}
723 inline void fl_set_menu_item_shortcut(Fl_Widget* o, int i, const char* s) {
724     ((Fl_Menu_Button*)o)->shortcut(i-1,fl_old_shortcut(s));}
725 inline void fl_set_menu_item_mode(Fl_Widget* o, int i, long x) {
726     ((Fl_Menu_Button*)o)->mode(i-1,x);}
727 inline void fl_show_menu_symbol(Fl_Widget*, int) {
728     /* ((Fl_Menu_Button*)o)->show_menu_symbol(i); */}
729 // inline void fl_set_menu_popup(Fl_Widget*, int);
730 inline int fl_get_menu(Fl_Widget* o) {
731     return ((Fl_Menu_Button*)o)->value()+1;}
732 inline const char* fl_get_menu_item_text(Fl_Widget* o, int i) {
733     return ((Fl_Menu_Button*)o)->text(i);}

```

```

734 inline int fl_get_menu_maxitems(Fl_Widget* o) {
735     return ((Fl_Menu_Button*)o)->size();}
736 inline int fl_get_menu_item_mode(Fl_Widget* o, int i) {
737     return ((Fl_Menu_Button*)o)->mode(i);}
738 inline const char* fl_get_menu_text(Fl_Widget* o) {
739     return ((Fl_Menu_Button*)o)->text();}
740
741 #include "Fl_Positioner.H"
742 #define FL_NORMAL_POSITIONER 0
743 forms_constructor(Fl_Positioner, fl_add_positioner)
744 inline void fl_set_positioner_xvalue(Fl_Widget* o, double v) {
745     ((Fl_Positioner*)o)->xvalue(v);}
746 inline double fl_get_positioner_xvalue(Fl_Widget* o) {
747     return ((Fl_Positioner*)o)->xvalue();}
748 inline void fl_set_positioner_xbounds(Fl_Widget* o, double a, double b) {
749     ((Fl_Positioner*)o)->xbounds(a,b);}
750 inline void fl_get_positioner_xbounds(Fl_Widget* o, float* a, float* b) {
751     *a = float(((Fl_Positioner*)o)->xminimum());
752     *b = float(((Fl_Positioner*)o)->xmaximum());
753 }
754 inline void fl_set_positioner_yvalue(Fl_Widget* o, double v) {
755     ((Fl_Positioner*)o)->yvalue(v);}
756 inline double fl_get_positioner_yvalue(Fl_Widget* o) {
757     return ((Fl_Positioner*)o)->yvalue();}
758 inline void fl_set_positioner_ybounds(Fl_Widget* o, double a, double b) {
759     ((Fl_Positioner*)o)->ybounds(a,b);}
760 inline void fl_get_positioner_ybounds(Fl_Widget* o, float* a, float* b) {
761     *a = float(((Fl_Positioner*)o)->yminimum());
762     *b = float(((Fl_Positioner*)o)->ymaximum());
763 }
764 inline void fl_set_positioner_xstep(Fl_Widget* o, double v) {
765     ((Fl_Positioner*)o)->xstep(v);}
766 inline void fl_set_positioner_ystep(Fl_Widget* o, double v) {
767     ((Fl_Positioner*)o)->ystep(v);}
768 inline void fl_set_positioner_return(Fl_Widget* o, int v) {
769     ((Fl_Positioner*)o)->when((Fl_When)(v|FL_WHEN_RELEASE));}
770
771 #include "Fl_Slider.H"
772
773 #define FL_HOR_BROWSER_SLIDER FL_HOR_SLIDER
774 #define FL_VERT_BROWSER_SLIDER FL_VERT_SLIDER
775
776 forms_constructort(Fl_Slider, fl_add_slider)
777 #define FL_SLIDER_COL1 FL_GRAY
778 inline void fl_set_slider_value(Fl_Widget* o, double v) {
779     ((Fl_Slider*)o)->value(v);}
780 inline double fl_get_slider_value(Fl_Widget* o) {
781     return ((Fl_Slider*)o)->value();}
782 inline void fl_set_slider_bounds(Fl_Widget* o, double a, double b) {
783     ((Fl_Slider*)o)->bounds(a, b);}
784 inline void fl_get_slider_bounds(Fl_Widget* o, float* a, float* b) {
785     *a = float(((Fl_Slider*)o)->minimum());
786     *b = float(((Fl_Slider*)o)->maximum());
787 }
788 inline void fl_set_slider_return(Fl_Widget* o, int i) {
789     ((Fl_Slider*)o)->when((Fl_When)(i|FL_WHEN_RELEASE));}
790 inline void fl_set_slider_step(Fl_Widget* o, double v) {
791     ((Fl_Slider*)o)->step(v);}
792 // inline void fl_set_slider_increment(Fl_Widget* o, double v, double);
793 inline void fl_set_slider_size(Fl_Widget* o, double v) {
794     ((Fl_Slider*)o)->slider_size(v);}
795
796 #include "Fl_Value_Slider.H"
797 forms_constructor(Fl_Value_Slider, fl_add_valslider)
798
799 inline void fl_set_slider_precision(Fl_Widget* o, int i) {
800     ((Fl_Value_Slider*)o)->precision(i);}
801 // filter function!
802
803 // The forms text object was the same as an Fl_Box except it inverted the
804 // meaning of FL_ALIGN_INSIDE. Implementation in forms.cxx
805 class FL_EXPORT Fl_FormsText : public Fl_Widget {
806 protected:
807     void draw();
808 public:
809     Fl_FormsText(Fl_Boxtype b, int X, int Y, int W, int H, const char* l=0)
810         : Fl_Widget(X,Y,W,H,l) {box(b); align(FL_ALIGN_LEFT);}
811 };
812 #define FL_NORMAL_TEXT FL_NO_BOX
813 forms_constructorb(Fl_FormsText, fl_add_text)
814
815 #include "Fl_Timer.H"
816 forms_constructort(Fl_Timer, fl_add_timer)
817 inline void fl_set_timer(Fl_Widget* o, double v) {((Fl_Timer*)o)->value(v);}
818 inline double fl_get_timer(Fl_Widget* o) {return ((Fl_Timer*)o)->value();}
819 inline void fl_suspend_timer(Fl_Widget* o) {((Fl_Timer*)o)->suspended(1);}
820 inline void fl_resume_timer(Fl_Widget* o) {((Fl_Timer*)o)->suspended(0);}

```

```

821 inline void fl_set_timer_countup(Fl_Widget* o, char d) {((Fl_Timer*)o)->direction(d);}
822 void FL_EXPORT fl_gettime(long* sec, long* usec);
823
824 // Fl_XYPlot nyl
825
826
827 // stuff from DDForms:
828
829 inline int fl_double_click() {return Fl::event_clicks();}
830 inline void fl_draw() {Fl::flush();}
831
832 #endif /* define __FORMS_H__ */
833
834 //
835 // End of "$Id$".
836 //

```

32.158 gl.h File Reference

This file defines wrapper functions for OpenGL in FLTK.

```

#include "Enumerations.H"
#include <GL/gl.h>

```

Functions

- FL_EXPORT void [gl_color](#) (Fl_Color i)
 - Sets the current OpenGL color to an FLTK color.*
- void [gl_color](#) (int c)
 - back compatibility*
- FL_EXPORT int [gl_descent](#) ()
 - Returns the current font's descent.*
- FL_EXPORT void [gl_draw](#) (const char *)
 - Draws a nul-terminated string in the current font at the current position.*
- FL_EXPORT void [gl_draw](#) (const char *, float x, float y)
 - Draws a nul-terminated string in the current font at the given position.*
- FL_EXPORT void [gl_draw](#) (const char *, int n)
 - Draws an array of n characters of the string in the current font at the current position.*
- FL_EXPORT void [gl_draw](#) (const char *, int n, float x, float y)
 - Draws n characters of the string in the current font at the given position.*
- FL_EXPORT void [gl_draw](#) (const char *, int n, int x, int y)
 - Draws n characters of the string in the current font at the given position.*
- FL_EXPORT void [gl_draw](#) (const char *, int x, int y)
 - Draws a nul-terminated string in the current font at the given position.*
- FL_EXPORT void [gl_draw](#) (const char *, int x, int y, int w, int h, Fl_Align)
 - Draws a string formatted into a box, with newlines and tabs expanded, other control characters changed to ^X.*
- FL_EXPORT void [gl_draw_image](#) (const uchar *, int x, int y, int w, int h, int d=3, int ld=0)
- FL_EXPORT void [gl_finish](#) ()
 - Releases an OpenGL context.*
- FL_EXPORT void [gl_font](#) (int fontid, int size)
 - Sets the current OpenGL font to the same font as calling [fl_font\(\)](#)*
- FL_EXPORT int [gl_height](#) ()
 - Returns the current font's height.*
- FL_EXPORT void [gl_measure](#) (const char *, int &x, int &y)
 - Measure how wide and tall the string will be when drawn by the [gl_draw\(\)](#) function.*
- FL_EXPORT void [gl_rect](#) (int x, int y, int w, int h)
 - Outlines the given rectangle with the current color.*
- void [gl_rectf](#) (int x, int y, int w, int h)

- Fills the given rectangle with the current color.*

 - FL_EXPORT void **gl_start** ()
Creates an OpenGL context.
 - FL_EXPORT double **gl_width** (const char *)
Returns the width of the string in the current font.
 - FL_EXPORT double **gl_width** (const char *, int n)
Returns the width of n characters of the string in the current font.
 - FL_EXPORT double **gl_width** (uchar)
Returns the width of the character in the current font.

32.158.1 Detailed Description

This file defines wrapper functions for OpenGL in FLTK.

To use OpenGL from within an FLTK application you MUST use `gl_visual()` to select the default visual before doing `show()` on any windows. Mesa will crash if you try to use a visual not returned by `glxChooseVisual`.

This does not work with [Fl_Double_Window](#)'s! It will try to draw into the front buffer. Depending on the system this will either crash or do nothing (when pixmaps are being used as back buffer and GL is being done by hardware), work correctly (when GL is done with software, such as Mesa), or draw into the front buffer and be erased when the buffers are swapped (when double buffer hardware is being used)

32.158.2 Function Documentation

32.158.2.1 `gl_color()`

```
FL_EXPORT void gl_color (
    Fl_Color i )
```

Sets the current OpenGL color to an FLTK color.

For color-index modes it will use `fl_xpixel(c)`, which is only right if the window uses the default colormap!

32.158.2.2 `gl_draw()` [1/7]

```
FL_EXPORT void gl_draw (
    const char * str )
```

Draws a nul-terminated string in the current font at the current position.

See also

On the Mac OS X platform, see [gl_texture_pile_height\(int\)](#)

32.158.2.3 `gl_draw()` [2/7]

```
FL_EXPORT void gl_draw (
    const char * str,
    float x,
    float y )
```

Draws a nul-terminated string in the current font at the given position.

See also

On the Mac OS X platform, see [gl_texture_pile_height\(int\)](#)

32.158.2.4 gl_draw() [3/7]

```
FL_EXPORT void gl_draw (
    const char * str,
    int n )
```

Draws an array of n characters of the string in the current font at the current position.

See also

On the Mac OS X platform, see [gl_texture_pile_height\(int\)](#)

32.158.2.5 gl_draw() [4/7]

```
FL_EXPORT void gl_draw (
    const char * str,
    int n,
    float x,
    float y )
```

Draws n characters of the string in the current font at the given position.

See also

On the Mac OS X platform, see [gl_texture_pile_height\(int\)](#)

32.158.2.6 gl_draw() [5/7]

```
FL_EXPORT void gl_draw (
    const char * str,
    int n,
    int x,
    int y )
```

Draws n characters of the string in the current font at the given position.

See also

On the Mac OS X platform, see [gl_texture_pile_height\(int\)](#)

32.158.2.7 gl_draw() [6/7]

```
FL_EXPORT void gl_draw (
    const char * str,
    int x,
    int y )
```

Draws a nul-terminated string in the current font at the given position.

See also

On the Mac OS X platform, see [gl_texture_pile_height\(int\)](#)

32.158.2.8 gl_draw() [7/7]

```
FL_EXPORT void gl_draw (
    const char * str,
    int x,
    int y,
    int w,
```

```

    int h,
    FL_Align align )

```

Draws a string formatted into a box, with newlines and tabs expanded, other control characters changed to ^X. and aligned with the edges or center. Exactly the same output as [fl_draw\(\)](#).

32.158.2.9 gl_rect()

```

FL_EXPORT void gl_rect (
    int x,
    int y,
    int w,
    int h )

```

Outlines the given rectangle with the current color.

If [FL_Gl_Window::ortho\(\)](#) has been called, then the rectangle will exactly fill the given pixel rectangle.

32.158.2.10 gl_rectf()

```

void gl_rectf (
    int x,
    int y,
    int w,
    int h ) [inline]

```

Fills the given rectangle with the current color.

See also

[gl_rect\(int x, int y, int w, int h\)](#)

32.159 gl.h

[Go to the documentation of this file.](#)

```

1 //
2 // "$Id$"
3 //
4 // OpenGL header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2011 by Bill Spitzak and others.
7 //
8 // You must include this instead of GL/gl.h to get the Microsoft
9 // APIENTRY stuff included (from <windows.h>) prior to the OpenGL
10 // header files.
11 //
12 // This file also provides "missing" OpenGL functions, and
13 // gl_start() and gl_finish() to allow OpenGL to be used in any window
14 //
15 // This library is free software. Distribution and use rights are outlined in
16 // the file "COPYING" which should have been included with this file. If this
17 // file is missing or damaged, see the license at:
18 //
19 //     http://www.fltk.org/COPYING.php
20 //
21 // Please report all bugs and problems on the following page:
22 //
23 //     http://www.fltk.org/str.php
24 //
25
42 #ifndef FL_gl_H
43 # define FL_gl_H
44
45 # include "Enumerations.H" // for color names
46 # ifdef WIN32
47 #     include <windows.h>
48 # endif
49 # ifndef APIENTRY
50 #     if defined(__CYGWIN__)
51 #         define APIENTRY __attribute__((__stdcall__))
52 #     else
53 #         define APIENTRY
54 #     endif
55 # endif
56
57 # ifdef __APPLE__
58 #     include <OpenGL/gl.h>

```

```

59 # else
60 #     include <GL/gl.h>
61 # endif // __APPLE__
62
63 FL_EXPORT void gl_start();
64 FL_EXPORT void gl_finish();
65
66 FL_EXPORT void gl_color(Fl_Color i);
67 inline void gl_color(int c) {gl_color((Fl_Color)c);}
68
69
70 FL_EXPORT void gl_rect(int x,int y,int w,int h);
71 inline void gl_rectf(int x,int y,int w,int h) {glRecti(x,y,x+w,y+h);}
72
73
74 FL_EXPORT void gl_font(int fontid, int size);
75 FL_EXPORT int gl_height();
76 FL_EXPORT int gl_descent();
77 FL_EXPORT double gl_width(const char *);
78 FL_EXPORT double gl_width(const char *, int n);
79 FL_EXPORT double gl_width(uchar);
80
81
82 FL_EXPORT void gl_draw(const char*);
83 FL_EXPORT void gl_draw(const char*, int n);
84 FL_EXPORT void gl_draw(const char*, int x, int y);
85 FL_EXPORT void gl_draw(const char*, float x, float y);
86 FL_EXPORT void gl_draw(const char*, int n, int x, int y);
87 FL_EXPORT void gl_draw(const char*, int n, float x, float y);
88 FL_EXPORT void gl_draw(const char*, int x, int y, int w, int h, Fl_Align);
89 FL_EXPORT void gl_measure(const char*, int& x, int& y);
90 #ifdef __APPLE__
91 extern FL_EXPORT void gl_texture_pile_height(int max);
92 extern FL_EXPORT int gl_texture_pile_height();
93 #endif
94
95 FL_EXPORT void gl_draw_image(const uchar *, int x,int y,int w,int h, int d=3, int ld=0);
96
97 #endif // !FL_gl_H
98
99 //
100 // End of "$Id$".
101 //

```

32.160 gl2opengl.h

```

1 /*      gl.h
2
3 GL to OpenGL translator.
4 If you include this, you might be able to port old GL programs.
5 There are also much better emulators available on the net.
6
7 */
8
9 #include <FL/gl.h>
10 #include "gl_draw.H"
11
12 inline void clear() {glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);}
13 #define RGBcolor(r,g,b) glColor3ub(r,g,b)
14 #define bgnline() glBegin(GL_LINE_STRIP)
15 #define bgnpolygon() glBegin(GL_POLYGON)
16 #define bgnclosedline() glBegin(GL_LINE_LOOP)
17 #define endline() glEnd()
18 #define endpolygon() glEnd()
19 #define endclosedline() glEnd()
20 #define v2f(v) glVertex2fv(v)
21 #define v2s(v) glVertex2sv(v)
22 #define cmov(x,y,z) glRasterPos3f(x,y,z)
23 #define charstr(s) gl_draw(s)
24 #define fmprstr(s) gl_draw(s)
25 typedef float Matrix[4][4];
26 inline void pushmatrix() {glPushMatrix();}
27 inline void popmatrix() {glPopMatrix();}
28 inline void multmatrix(Matrix m) {glMultMatrixf((float *)m);}
29 inline void color(int n) {glIndexi(n);}
30 inline void rect(int x,int y,int r,int t) {gl_rect(x,y,r-x,t-y);}
31 inline void rectf(int x,int y,int r,int t) {glRectf(x,y,r+1,t+1);}
32 inline void recti(int x,int y,int r,int t) {gl_rect(x,y,r-x,t-y);}
33 inline void rectfi(int x,int y,int r,int t) {glRecti(x,y,r+1,t+1);}
34 inline void rects(int x,int y,int r,int t) {gl_rect(x,y,r-x,t-y);}
35 inline void rectfs(int x,int y,int r,int t) {glRects(x,y,r+1,t+1);}

```

32.161 gl_draw.H

```
1 //
```

```

2 // "$Id$"
3 //
4 // OpenGL header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software.  Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file.  If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 #include "gl.h"
20
21 extern FL_EXPORT void gl_remove_displaylist_fonts();
22
23
24 //
25 // End of "$Id$".
26 //

```

32.162 glu.h

```

1 //
2 // "$Id$"
3 //
4 // GLu header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // You must include this instead of GL/gl.h to get the Microsoft
9 // APIENTRY stuff included (from <windows.h>) prior to the OpenGL
10 // header files.
11 //
12 // This file also provides "missing" OpenGL functions, and
13 // gl_start() and gl_finish() to allow OpenGL to be used in any window
14 //
15 // This library is free software.  Distribution and use rights are outlined in
16 // the file "COPYING" which should have been included with this file.  If this
17 // file is missing or damaged, see the license at:
18 //
19 //     http://www.fltk.org/COPYING.php
20 //
21 // Please report all bugs and problems on the following page:
22 //
23 //     http://www.fltk.org/str.php
24 //
25
26 #ifndef FL_glu_H
27 # define FL_glu_H
28
29 # include "Enumerations.H" // for color names
30 # ifdef WIN32
31 #   include <windows.h>
32 # endif
33 # ifndef APIENTRY
34 #   if defined(__CYGWIN__)
35 #     define APIENTRY __attribute__((__stdcall__))
36 #   else
37 #     define APIENTRY
38 #   endif
39 # endif
40
41 # ifdef __APPLE__
42 #   include <OpenGL/glu.h>
43 # else
44 #   include <GL/glu.h>
45 # endif
46
47 #endif // !FL_glu_H
48
49 //
50 // End of "$Id$".
51 //

```

32.163 glut.H

```

1 //
2 // "$Id$"
3 //
4 // GLUT emulation header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2015 by Bill Spitzak and others.
7 //
8 // This library is free software.  Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file.  If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 // Emulation of GLUT using fltk.
20
21 // GLUT is Copyright (c) Mark J. Kilgard, 1994, 1995, 1996:
22 // "This program is freely distributable without licensing fees and is
23 // provided without guarantee or warranty expressed or implied.  This
24 // program is -not- in the public domain."
25
26 // Although I have copied the GLUT API, none of my code is based on
27 // any GLUT implementation details and is therefore covered by the LGPL.
28
29 // Commented out lines indicate parts of GLUT that are not emulated.
30
31 #ifndef Fl_glut_H
32 #   define Fl_glut_H
33
34 #   include "gl.h"
35
36
37 #   include "Fl.H"
38 #   include "Fl_Gl_Window.H"
39
40 class FL_EXPORT Fl_Glut_Window : public Fl_Gl_Window {
41     void _init();
42     int mouse_down;
43 protected:
44     void draw();
45     void draw_overlay();
46     int handle(int);
47 public: // so the inline functions work
48     int number;
49     int menu[3];
50     void make_current();
51     void (*display)();
52     void (*overlaydisplay)();
53     void (*reshape)(int w, int h);
54     void (*keyboard)(uchar, int x, int y);
55     void (*mouse)(int b, int state, int x, int y);
56     void (*motion)(int x, int y);
57     void (*passivemotion)(int x, int y);
58     void (*entry)(int);
59     void (*visibility)(int);
60     void (*special)(int, int x, int y);
61     Fl_Glut_Window(int w, int h, const char *);
62     Fl_Glut_Window(int x, int y, int w, int h, const char *);
63     ~Fl_Glut_Window();
64 };
65
66 extern FL_EXPORT Fl_Glut_Window *glut_window; // the current window
67 extern FL_EXPORT int glut_menu; // the current menu
68
69
70 // function pointers that are not per-window:
71 extern FL_EXPORT void (*glut_idle_function)();
72 extern FL_EXPORT void (*glut_menustate_function)(int);
73 extern FL_EXPORT void (*glut_menustatus_function)(int,int,int);
74
75
76 //# define GLUT_API_VERSION This does not match any version of GLUT exactly...
77
78 FL_EXPORT void glutInit(int *argcp, char **argv); // creates first window
79
80 FL_EXPORT void glutInitDisplayMode(unsigned int mode);
81 // the FL_ symbols have the same value as the GLUT ones:
82 #   define GLUT_RGB         FL_RGB
83 #   define GLUT_RGBA        FL_RGBA
84 #   define GLUT_INDEX       FL_INDEX
85 #   define GLUT_SINGLE      FL_SINGLE
86 #   define GLUT_DOUBLE      FL_DOUBLE

```

```

91 # define GLUT_ACCUM      FL_ACCUM
92 # define GLUT_ALPHA      FL_ALPHA
93 # define GLUT_DEPTH      FL_DEPTH
94 # define GLUT_STENCIL    FL_STENCIL
95 # define GLUT_MULTISAMPLE FL_MULTISAMPLE
96 # define GLUT_STEREO     FL_STEREO
97 // # define GLUT_LUMINANCE      512
98
99 FL_EXPORT void glutInitWindowPosition(int x, int y);
100
101 FL_EXPORT void glutInitWindowSize(int w, int h);
102
103 FL_EXPORT void glutMainLoop();
104
105 FL_EXPORT int glutCreateWindow(char *title);
106 FL_EXPORT int glutCreateWindow(const char *title);
107
108 FL_EXPORT int glutCreateSubWindow(int win, int x, int y, int width, int height);
109
110 FL_EXPORT void glutDestroyWindow(int win);
111
112 inline void glutPostRedisplay() {glut_window->redraw();}
113
114 FL_EXPORT void glutPostWindowRedisplay(int win);
115
116 FL_EXPORT void glutSwapBuffers();
117
118 inline int glutGetWindow() {return glut_window->number;}
119
120 FL_EXPORT void glutSetWindow(int win);
121
122 inline void glutSetWindowTitle(char *t) {glut_window->label(t);}
123
124 inline void glutSetIconTitle(char *t) {glut_window->iconlabel(t);}
125
126 inline void glutPositionWindow(int x, int y) {glut_window->position(x,y);}
127
128 inline void glutReshapeWindow(int w, int h) {glut_window->size(w,h);}
129
130 inline void glutPopWindow() {glut_window->show();}
131
132 inline void glutPushWindow() { /* do nothing */ }
133
134 inline void glutIconifyWindow() {glut_window->iconize();}
135
136 inline void glutShowWindow() {glut_window->show();}
137
138 inline void glutHideWindow() {glut_window->hide();}
139
140 inline void glutFullScreen() {glut_window->fullscreen();}
141
142 inline void glutSetCursor(Fl_Cursor cursor) {glut_window->cursor(cursor);}
143 // notice that the numeric values are different than glut:
144 # define GLUT_CURSOR_RIGHT_ARROW      ((Fl_Cursor)2)
145 # define GLUT_CURSOR_LEFT_ARROW      ((Fl_Cursor)67)
146 # define GLUT_CURSOR_INFO            FL_CURSOR_HAND
147 # define GLUT_CURSOR_DESTROY        ((Fl_Cursor)45)
148 # define GLUT_CURSOR_HELP           FL_CURSOR_HELP
149 # define GLUT_CURSOR_CYCLE          ((Fl_Cursor)26)
150 # define GLUT_CURSOR_SPRAY          ((Fl_Cursor)63)
151 # define GLUT_CURSOR_WAIT           FL_CURSOR_WAIT
152 # define GLUT_CURSOR_TEXT           FL_CURSOR_INSERT
153 # define GLUT_CURSOR_CROSSHAIR     FL_CURSOR_CROSS
154 # define GLUT_CURSOR_UP_DOWN       FL_CURSOR_NS
155 # define GLUT_CURSOR_LEFT_RIGHT    FL_CURSOR_WE
156 # define GLUT_CURSOR_TOP_SIDE      FL_CURSOR_N
157 # define GLUT_CURSOR_BOTTOM_SIDE   FL_CURSOR_S
158 # define GLUT_CURSOR_LEFT_SIDE     FL_CURSOR_W
159 # define GLUT_CURSOR_RIGHT_SIDE    FL_CURSOR_E
160 # define GLUT_CURSOR_TOP_LEFT_CORNER FL_CURSOR_NW
161 # define GLUT_CURSOR_TOP_RIGHT_CORNER FL_CURSOR_NE
162 # define GLUT_CURSOR_BOTTOM_RIGHT_CORNER FL_CURSOR_SE
163 # define GLUT_CURSOR_BOTTOM_LEFT_CORNER FL_CURSOR_SW
164 # define GLUT_CURSOR_INHERIT       FL_CURSOR_DEFAULT
165 # define GLUT_CURSOR_NONE          FL_CURSOR_NONE
166 # define GLUT_CURSOR_FULL_CROSSHAIR FL_CURSOR_CROSS
167
168 inline void glutWarpPointer(int, int) { /* do nothing */ }
169
170 inline void glutEstablishOverlay() {glut_window->make_overlay_current();}
171
172 inline void glutRemoveOverlay() {glut_window->hide_overlay();}
173
174 inline void glutUseLayer(GLenum layer) {
175     layer ? glut_window->make_overlay_current() : glut_window->make_current();}
176 enum {GLUT_NORMAL, GLUT_OVERLAY};
177

```

```

178 inline void glutPostOverlayRedisplay() {glut_window->redraw_overlay();}
179
180 inline void glutShowOverlay() {glut_window->redraw_overlay();}
181
182 inline void glutHideOverlay() {glut_window->hide_overlay();}
183
184 FL_EXPORT int glutCreateMenu(void (*)(int));
185
186 FL_EXPORT void glutDestroyMenu(int menu);
187
188 inline int glutGetMenu() {return glut_menu;}
189
190 inline void glutSetMenu(int m) {glut_menu = m;}
191
192 FL_EXPORT void glutAddMenuEntry(char *label, int value);
193
194 FL_EXPORT void glutAddSubMenu(char *label, int submenu);
195
196 FL_EXPORT void glutChangeToMenuEntry(int item, char *labela, int value);
197
198 FL_EXPORT void glutChangeToSubMenu(int item, char *label, int submenu);
199
200 FL_EXPORT void glutRemoveMenuItem(int item);
201
202 inline void glutAttachMenu(int b) {glut_window->menu[b] = glut_menu;}
203
204 inline void glutDetachMenu(int b) {glut_window->menu[b] = 0;}
205
206 inline void glutDisplayFunc(void (*)( )) {glut_window->display = f;}
207
208 inline void glutReshapeFunc(void (*)(int w, int h)) {glut_window->reshape=f;}
209
210 inline void glutKeyboardFunc(void (*)(uchar key, int x, int y)) {
211     glut_window->keyboard = f;}
212
213 inline void glutMouseFunc(void (*)(int b, int state, int x, int y)) {
214     glut_window->mouse = f;}
215 # define GLUT_LEFT_BUTTON          0
216 # define GLUT_MIDDLE_BUTTON        1
217 # define GLUT_RIGHT_BUTTON         2
218 # define GLUT_DOWN                 0
219 # define GLUT_UP                   1
220
221 inline void glutMotionFunc(void (*)(int x, int y)) {glut_window->motion= f;}
222
223 inline void glutPassiveMotionFunc(void (*)(int x, int y)) {
224     glut_window->passivemotion= f;}
225
226 inline void glutEntryFunc(void (*)(int s)) {glut_window->entry = f;}
227 enum {GLUT_LEFT, GLUT_ENTERED};
228
229 inline void glutVisibilityFunc(void (*)(int s)) {glut_window->visibility=f;}
230 enum {GLUT_NOT_VISIBLE, GLUT_VISIBLE};
231
232 FL_EXPORT void glutIdleFunc(void (*)( ));
233
234 inline void glutTimerFunc(unsigned int msec, void (*)(int), int value) {
235     Fl::add_timeout(msec*.001, (void (*)(void *))f, (void *) (fl_intptr_t)value);
236 }
237
238 inline void glutMenuStateFunc(void (*)(int state)) {
239     glut_menustate_function = f;}
240
241 inline void glutMenuStatusFunc(void (*)(int status, int x, int y)) {
242     glut_menustatus_function = f;}
243 enum {GLUT_MENU_NOT_IN_USE, GLUT_MENU_IN_USE};
244
245 inline void glutSpecialFunc(void (*)(int key, int x, int y)) {
246     glut_window->special = f;}
247 # define GLUT_KEY_F1                1
248 # define GLUT_KEY_F2                2
249 # define GLUT_KEY_F3                3
250 # define GLUT_KEY_F4                4
251 # define GLUT_KEY_F5                5
252 # define GLUT_KEY_F6                6
253 # define GLUT_KEY_F7                7
254 # define GLUT_KEY_F8                8
255 # define GLUT_KEY_F9                9
256 # define GLUT_KEY_F10               10
257 # define GLUT_KEY_F11               11
258 # define GLUT_KEY_F12               12
259 // WARNING: Different values than GLUT uses:
260 # define GLUT_KEY_LEFT               FL_Left
261 # define GLUT_KEY_UP                 FL_Up
262 # define GLUT_KEY_RIGHT              FL_Right
263 # define GLUT_KEY_DOWN               FL_Down
264 # define GLUT_KEY_PAGE_UP            FL_Page_Up

```

```

265 # define GLUT_KEY_PAGE_DOWN          FL_Page_Down
266 # define GLUT_KEY_HOME                FL_Home
267 # define GLUT_KEY_END                 FL_End
268 # define GLUT_KEY_INSERT              FL_Insert
269
270 //inline void glutSpaceballMotionFunc(void (*)(int x, int y, int z));
271
272 //inline void glutSpaceballRotateFunc(void (*)(int x, int y, int z));
273
274 //inline void glutSpaceballButtonFunc(void (*)(int button, int state));
275
276 //inline void glutButtonBoxFunc(void (*)(int button, int state));
277
278 //inline void glutDialsFunc(void (*)(int dial, int value));
279
280 //inline void glutTabletMotionFunc(void (*)(int x, int y));
281
282 //inline void glutTabletButtonFunc(void (*)(int button, int state, int x, int y));
283
284 inline void glutOverlayDisplayFunc(void (*)(void)) {
285     glut_window->overlaydisplay = f;}
286
287 //inline void glutWindowStatusFunc(void (*)(int state));
288 //enum {GLUT_HIDDEN, GLUT_FULLY_RETAINED, GLUT_PARTIALLY_RETAINED,
289 //       GLUT_FULLY_COVERED};
290
291 //inline void glutSetColor(int, GLfloat red, GLfloat green, GLfloat blue);
292
293 //inline GLfloat glutGetColor(int ndx, int component);
294 //define GLUT_RED          0
295 //define GLUT_GREEN        1
296 //define GLUT_BLUE         2
297
298 //inline void glutCopyColormap(int win);
299
300 // Warning:  values are changed from GLUT!
301 // Also relies on the GL_ symbols having values greater than 100
302 FL_EXPORT int glutGet(GLenum type);
303 enum {
304     GLUT_RETURN_ZERO = 0,
305     GLUT_WINDOW_X,
306     GLUT_WINDOW_Y,
307     GLUT_WINDOW_WIDTH,
308     GLUT_WINDOW_HEIGHT,
309     GLUT_WINDOW_PARENT,
310     GLUT_SCREEN_WIDTH,
311     GLUT_SCREEN_HEIGHT,
312     GLUT_MENU_NUM_ITEMS,
313     GLUT_DISPLAY_MODE_POSSIBLE,
314     GLUT_INIT_WINDOW_X,
315     GLUT_INIT_WINDOW_Y,
316     GLUT_INIT_WINDOW_WIDTH,
317     GLUT_INIT_WINDOW_HEIGHT,
318     GLUT_INIT_DISPLAY_MODE,
319     GLUT_WINDOW_BUFFER_SIZE,
320     GLUT_VERSION
321 //GLUT_WINDOW_NUM_CHILDREN,
322 //GLUT_WINDOW_CURSOR,
323 //GLUT_SCREEN_WIDTH_MM,
324 //GLUT_SCREEN_HEIGHT_MM,
325 //GLUT_ELAPSED_TIME,
326 };
327
328 # define GLUT_WINDOW_STENCIL_SIZE      GL_STENCIL_BITS
329 # define GLUT_WINDOW_DEPTH_SIZE       GL_DEPTH_BITS
330 # define GLUT_WINDOW_RED_SIZE         GL_RED_BITS
331 # define GLUT_WINDOW_GREEN_SIZE       GL_GREEN_BITS
332 # define GLUT_WINDOW_BLUE_SIZE        GL_BLUE_BITS
333 # define GLUT_WINDOW_ALPHA_SIZE       GL_ALPHA_BITS
334 # define GLUT_WINDOW_ACCUM_RED_SIZE   GL_ACCUM_RED_BITS
335 # define GLUT_WINDOW_ACCUM_GREEN_SIZE GL_ACCUM_GREEN_BITS
336 # define GLUT_WINDOW_ACCUM_BLUE_SIZE  GL_ACCUM_BLUE_BITS
337 # define GLUT_WINDOW_ACCUM_ALPHA_SIZE GL_ACCUM_ALPHA_BITS
338 # define GLUT_WINDOW_DOUBLEBUFFER     GL_DOUBLEBUFFER
339 # define GLUT_WINDOW_RGBA              GL_RGBA
340 # define GLUT_WINDOW_COLORMAP_SIZE    GL_INDEX_BITS
341 # ifdef GL_SAMPLES_SGIS
342 #     define GLUT_WINDOW_NUM_SAMPLES   GL_SAMPLES_SGIS
343 # else
344 #     define GLUT_WINDOW_NUM_SAMPLES   GLUT_RETURN_ZERO
345 # endif
346 # define GLUT_WINDOW_STEREO            GL_STEREO
347
348 # define GLUT_HAS_KEYBOARD             600
349 # define GLUT_HAS_MOUSE                601
350 # define GLUT_HAS_SPACEBALL            602
351 # define GLUT_HAS_DIAL_AND_BUTTON_BOX 603

```



```

352 # define GLUT_HAS_TABLET          604
353 # define GLUT_NUM_MOUSE_BUTTONS    605
354 # define GLUT_NUM_SPACEBALL_BUTTONS 606
355 # define GLUT_NUM_BUTTON_BOX_BUTTONS 607
356 # define GLUT_NUM_DIALS            608
357 # define GLUT_NUM_TABLET_BUTTONS    609
358 FL_EXPORT int glutDeviceGet(GLenum type);
359
360 // WARNING: these values are different than GLUT uses:
361 # define GLUT_ACTIVE_SHIFT          FL_SHIFT
362 # define GLUT_ACTIVE_CTRL          FL_CTRL
363 # define GLUT_ACTIVE_ALT           FL_ALT
364 inline int glutGetModifiers() {return Fl::event_state() & (GLUT_ACTIVE_SHIFT | GLUT_ACTIVE_CTRL |
    GLUT_ACTIVE_ALT);}
365
366 FL_EXPORT int glutLayerGet(GLenum);
367 # define GLUT_OVERLAY_POSSIBLE      800
368 // #define GLUT_LAYER_IN_USE        801
369 // #define GLUT_HAS_OVERLAY         802
370 # define GLUT_TRANSPARENT_INDEX     803
371 # define GLUT_NORMAL_DAMAGED        804
372 # define GLUT_OVERLAY_DAMAGED       805
373
374 extern "C" {
375 typedef void (*GLUTproc)();
376 }
377
378 FL_EXPORT GLUTproc glutGetProcAddress(const char *procName);
379
380 // inline int glutVideoResizeGet(GLenum param);
381 // #define GLUT_VIDEO_RESIZE_POSSIBLE 900
382 // #define GLUT_VIDEO_RESIZE_IN_USE   901
383 // #define GLUT_VIDEO_RESIZE_X_DELTA  902
384 // #define GLUT_VIDEO_RESIZE_Y_DELTA  903
385 // #define GLUT_VIDEO_RESIZE_WIDTH_DELTA 904
386 // #define GLUT_VIDEO_RESIZE_HEIGHT_DELTA 905
387 // #define GLUT_VIDEO_RESIZE_X        906
388 // #define GLUT_VIDEO_RESIZE_Y        907
389 // #define GLUT_VIDEO_RESIZE_WIDTH    908
390 // #define GLUT_VIDEO_RESIZE_HEIGHT   909
391
392 // inline void glutSetupVideoResizing();
393
394 // inline void glutStopVideoResizing();
395
396 // inline void glutVideoResize(int x, int y, int width, int height);
397
398 // inline void glutVideoPan(int x, int y, int width, int height);
399
400 // Font argument must be a void* for compatibility, so...
401 struct Fl_Glut_Bitmap_Font {Fl_Font font; Fl_Fontsize size;};
402
403
404 extern FL_EXPORT struct Fl_Glut_Bitmap_Font
405 glutBitmap9By15, glutBitmap8By13, glutBitmapTimesRoman10,
406 glutBitmapTimesRoman24, glutBitmapHelvetica10, glutBitmapHelvetica12,
407 glutBitmapHelvetica18;
408 # define GLUT_BITMAP_9_BY_15        (&glutBitmap9By15)
409 # define GLUT_BITMAP_8_BY_13        (&glutBitmap8By13)
410 # define GLUT_BITMAP_TIMES_ROMAN_10 (&glutBitmapTimesRoman10)
411 # define GLUT_BITMAP_TIMES_ROMAN_24 (&glutBitmapTimesRoman24)
412 # define GLUT_BITMAP_HELVETICA_10   (&glutBitmapHelvetica10)
413 # define GLUT_BITMAP_HELVETICA_12   (&glutBitmapHelvetica12)
414 # define GLUT_BITMAP_HELVETICA_18   (&glutBitmapHelvetica18)
415
416 FL_EXPORT void glutBitmapCharacter(void *font, int character);
417 FL_EXPORT int glutBitmapHeight(void *font);
418 FL_EXPORT int glutBitmapLength(void *font, const unsigned char *string);
419 FL_EXPORT void glutBitmapString(void *font, const unsigned char *string);
420 FL_EXPORT int glutBitmapWidth(void *font, int character);
421
422 FL_EXPORT int glutExtensionSupported(char *name);
423
424 /* GLUT stroked font sub-API */
425 struct Fl_Glut_StrokeVertex {
426     GLfloat X, Y;
427 };
428
429 struct Fl_Glut_StrokeStrip {
430     int Number;
431     const Fl_Glut_StrokeVertex* Vertices;
432 };
433
434 struct Fl_Glut_StrokeChar {
435     GLfloat Right;
436     int Number;
437     const Fl_Glut_StrokeStrip* Strips;
438 };

```

```

439
440 struct Fl_Glut_StrokeFont {
441     char* Name; // The source font name
442     int Quantity; // Number of chars in font
443     GLfloat Height; // Height of the characters
444     const Fl_Glut_StrokeChar** Characters; // The characters mapping
445 };
446 extern FL_EXPORT Fl_Glut_StrokeFont glutStrokeRoman;
447 extern FL_EXPORT Fl_Glut_StrokeFont glutStrokeMonoRoman;
448 # define GLUT_STROKE_ROMAN (&glutStrokeRoman)
449 # define GLUT_STROKE_MONO_ROMAN (&glutStrokeMonoRoman)
450
451 FL_EXPORT void glutStrokeCharacter(void *font, int character);
452 FL_EXPORT GLfloat glutStrokeHeight(void *font);
453 FL_EXPORT int glutStrokeLength(void *font, const unsigned char *string);
454 FL_EXPORT void glutStrokeString(void *font, const unsigned char *string);
455 FL_EXPORT int glutStrokeWidth(void *font, int character);
456
457 /* GLUT pre-built models sub-API */
458 FL_EXPORT void glutWireSphere(GLdouble radius, GLint slices, GLint stacks);
459 FL_EXPORT void glutSolidSphere(GLdouble radius, GLint slices, GLint stacks);
460 FL_EXPORT void glutWireCone(GLdouble base, GLdouble height, GLint slices, GLint stacks);
461 FL_EXPORT void glutSolidCone(GLdouble base, GLdouble height, GLint slices, GLint stacks);
462 FL_EXPORT void glutWireCube(GLdouble size);
463 FL_EXPORT void glutSolidCube(GLdouble size);
464 FL_EXPORT void glutWireTorus(GLdouble innerRadius, GLdouble outerRadius, GLint sides, GLint rings);
465 FL_EXPORT void glutSolidTorus(GLdouble innerRadius, GLdouble outerRadius, GLint sides, GLint rings);
466 FL_EXPORT void glutWireDodecahedron();
467 FL_EXPORT void glutSolidDodecahedron();
468 FL_EXPORT void glutWireTeapot(GLdouble size);
469 FL_EXPORT void glutSolidTeapot(GLdouble size);
470 FL_EXPORT void glutWireOctahedron();
471 FL_EXPORT void glutSolidOctahedron();
472 FL_EXPORT void glutWireTetrahedron();
473 FL_EXPORT void glutSolidTetrahedron();
474 FL_EXPORT void glutWireIcosahedron();
475 FL_EXPORT void glutSolidIcosahedron();
476
477 #endif // !Fl_glut_H
478
479 //
480 // End of "$Id$".
481 //

```

32.164 mac.H File Reference

Mac OS X-specific symbols.

Classes

- class [Fl_Mac_App_Menu](#)
Mac OS-specific class allowing to customize and localize the application menu.

Functions

- void [fl_mac_set_about](#) ([Fl_Callback](#) *cb, void *user_data, int shortcut=0)
Attaches a callback to the "About myprog" item of the system application menu.
- void [fl_open_callback](#) (void(*cb)(const char *))
Register a function called for each file dropped onto an application icon.

Variables

- int [fl_mac_os_version](#)
The version number of the running Mac OS X (e.g., 100604 for 10.6.4)
- int [fl_mac_quit_early](#)
Determines whether cmd-Q or the "Quit xxx" item of application menu terminates the app or only the event loop.
- class [Fl_Sys_Menu_Bar](#) * [fl_sys_menu_bar](#)
The system menu bar.

32.164.1 Detailed Description

Mac OS X-specific symbols.

32.165 mac.H

[Go to the documentation of this file.](#)

```

1 //
2 // "$Id$"
3 //
4 // Mac header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2016 by Bill Spitzak and others.
7 //
8 // This library is free software.  Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file.  If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 // Do not directly include this file, instead use <FL/x.H>.  It will
20 // include this file if "__APPLE__" is defined.  This is to encourage
21 // portability of even the system-specific code...
22 #ifndef FL_DOXYGEN
23
24 #if !defined(FL_X_H)
25 # error "Never use <FL/mac.H> directly; include <FL/x.H> instead."
26 #endif // !FL_X_H
27
28 #ifdef __OBJC__
29 @class FLWindow; // a subclass of the NSWindow Cocoa class
30 typedef FLWindow *Window;
31 #else
32 typedef class FLWindow *Window; // pointer to the FLWindow objective-c class
33 #endif // __OBJC__
34
35 #if !(defined(FL_LIBRARY) || defined(FL_INTERNALS)) // this part is used when compiling an application
36 # program
37 # include <FL/Fl_Widget.H>
38
39 typedef struct flCocoaRegion* Fl_Region;
40 typedef struct CGContext* CGContextRef;
41 typedef struct OpaquePMPrintSettings* PMPrintSettings;
42 typedef struct OpaquePMPageFormat* PMPageFormat;
43 typedef struct OpaquePMPrintSession* PMPrintSession;
44 typedef struct CGImage* CGImageRef;
45 typedef struct __CFData* CFMutableDataRef; // used in Fl_Copy_Surface.H
46
47 #else // this part must be compiled when building the FLTK libraries
48
49 // Standard MacOS C/C++ includes...
50 #include <ApplicationServices/ApplicationServices.h>
51 #undef check // because of Fl::check()
52
53 #ifndef MAC_OS_X_VERSION_10_4
54 #define MAC_OS_X_VERSION_10_4 1040
55 #endif
56 #ifndef MAC_OS_X_VERSION_10_5
57 #define MAC_OS_X_VERSION_10_5 1050
58 #endif
59 #ifndef MAC_OS_X_VERSION_10_6
60 #define MAC_OS_X_VERSION_10_6 1060
61 #endif
62 #ifndef MAC_OS_X_VERSION_10_7
63 #define MAC_OS_X_VERSION_10_7 1070
64 #endif
65 #ifndef MAC_OS_X_VERSION_10_8
66 #define MAC_OS_X_VERSION_10_8 1080
67 #endif
68 #ifndef MAC_OS_X_VERSION_10_9
69 #define MAC_OS_X_VERSION_10_9 1090
70 #endif
71 #ifndef MAC_OS_X_VERSION_10_10
72 #define MAC_OS_X_VERSION_10_10 101000
73 #endif
74 #ifndef MAC_OS_X_VERSION_10_11
75 #define MAC_OS_X_VERSION_10_11 101100
76 #endif

```

```

77 #ifndef MAC_OS_X_VERSION_10_12
78 #define MAC_OS_X_VERSION_10_12 101200
79 #endif
80 #ifndef MAC_OS_X_VERSION_10_13
81 #define MAC_OS_X_VERSION_10_13 101300
82 #endif
83 #ifndef MAC_OS_X_VERSION_10_14
84 #define MAC_OS_X_VERSION_10_14 101400
85 #endif
86
87 #ifndef NSINTEGER_DEFINED // appears with 10.5 in NSObjCRuntime.h
88 #if defined(__LP64__) && __LP64__
89 typedef long NSInteger;
90 typedef unsigned long NSUInteger;
91 #else
92 typedef int NSInteger;
93 typedef unsigned int NSUInteger;
94 #endif
95 #endif
96
97 #ifdef __OBJC__
98 @class NSCursor;
99 @class NSOpenGLPixelFormat;
100 @class NSOpenGLContext;
101 #else
102 class NSCursor;
103 class NSOpenGLPixelFormat;
104 class NSOpenGLContext;
105 #endif // __OBJC__
106
107 typedef CGContextRef Fl_Offscreen;
108 #if MAC_OS_X_VERSION_MAX_ALLOWED < MAC_OS_X_VERSION_10_4
109 typedef CGImageAlphaInfo CGContextAlphaInfo;
110 #endif
111
112 typedef struct flCocoaRegion {
113     int count;
114     CGRect *rects;
115 } *Fl_Region; // a region is the union of a series of rectangles
116
117 # include "Fl_Window.H"
118 # include "../src/Fl_Font.H"
119
120 // Some random X equivalents
121 struct XPoint { int x, y; };
122 struct XRectangle {int x, y, width, height;};
123 #ifndef CGFLOAT_DEFINED //appears with 10.5 in CGBase.h
124 #if defined(__LP64__) && __LP64__
125 typedef double CGFloat;
126 #else
127 typedef float CGFloat;
128 #endif
129 #endif // CGFLOAT_DEFINED
130
131 extern CGRect fl_cgrectmake_cocoa(int x, int y, int w, int h);
132 inline Fl_Region XRectangleRegion(int x, int y, int w, int h) {
133     Fl_Region R = (Fl_Region)malloc(sizeof(*R));
134     R->count = 1;
135     R->rects = (CGRect *)malloc(sizeof(CGRect));
136     *(R->rects) = fl_cgrectmake_cocoa(x, y, w, h);
137     return R;
138 }
139 inline void XDestroyRegion(Fl_Region r) {
140     if(r) {
141         free(r->rects);
142         free(r);
143     }
144 }
145 extern NSCursor *fl_default_cursor;
146
147 // This object contains all mac-specific stuff about a window:
148 // WARNING: this object is highly subject to change!
149 class Fl_X {
150 public:
151     Window xid; // pointer to the Cocoa window object (FLWindow*)
152     Fl_Offscreen other_xid; // pointer for offscreen bitmaps (overlay window)
153     Fl_Window *w; // FLTK window for
154     Fl_Region region;
155     #if FLTK_ABI_VERSION < 10304
156     Fl_Region subRegion; // for ABI compatibility, recycled to replace subRect_
157     #endif
158     Fl_X *next; // chain of mapped windows
159     #if FLTK_ABI_VERSION < 10304
160     Fl_X *xidChildren; // useless with true subwindows, recycled to replace mapped_to_retina_
161     Fl_X *xidNext; // useless with true subwindows
162     #endif
163 };

```

```

164 int wait_for_expose;
165 NSCursor *cursor;
166 static Fl_X* first;
167 static Fl_X* i(const Fl_Window* w) {return w->i;}
168 static int fake_X_wm(const Fl_Window*, int&, int&, int&, int&, int&, int, int, int, int);
169 static void make(Fl_Window*);
170 void flush();
171 static void set_high_resolution(bool);
172 #if FLTK_ABI_VERSION >= 10304
173 CGRect* subRect() { return subRect_; } // getter
174 void subRect(CGRect *r) { subRect_ = r; } // setter
175 #else
176 CGRect* subRect() { return (CGRect*)subRegion; } // getter
177 void subRect(CGRect *r) { subRegion = (Fl_Region)r; } // setter
178 #endif
179 bool mapped_to_retina(); // is window mapped to retina display?
180 void mapped_to_retina(bool); // sets whether window is mapped to retina display
181 bool changed_resolution(); // did window just moved to display with another resolution?
182 void changed_resolution(bool); // sets whether window just moved to display with another resolution
183 bool in_windowDidResize(); // is window performing windowDidResize?
184 void in_windowDidResize(bool); // sets whether window is performing windowDidResize
185 // Quartz additions:
186 CGContextRef gc; // graphics context (NULL when using QD)
187 static void q_fill_context(); // fill a Quartz context with current FLTK state
188 static void q_clear_clipping(); // remove all clipping from a Quartz context
189 static void q_release_context(Fl_X *x=0); // free all resources associated with fl_gc
190 static void q_begin_image(CGRect&, int x, int y, int w, int h);
191 static void q_end_image();
192 // Cocoa additions
193 static NSOpenGLPixelFormat *mode_to_NSOpenGLPixelFormat(int mode, const int*); // computes
NSOpenGLPixelFormat from Gl window's mode
194 static NSOpenGLContext* create_GLcontext_for_window(NSOpenGLPixelFormat *pixelformat, NSOpenGLContext
*shared_ctx, Fl_Window *window);
195 static void GLcontext_update(NSOpenGLContext*);
196 static void GLcontext_flushbuffer(NSOpenGLContext*);
197 static void GLcontext_release(NSOpenGLContext*);
198 static void GLcontext_makecurrent(NSOpenGLContext*);
199 static void GL_cleardrawable(void);
200 static void gl_start(NSOpenGLContext*);
201 void destroy(void);
202 void map(void);
203 void unmap(void);
204 void collapse(void);
205 WindowRef window_ref(void); // useless with cocoa GL windows
206 void set_key_window(void);
207 // OS X doesn't have per window icons
208 static void set_default_icons(const Fl_RGB_Image*[], int) {};
209 void set_icons() {};
210 int set_cursor(Fl_Cursor);
211 int set_cursor(const Fl_RGB_Image*, int, int);
212 static CGImageRef CGImage_from_window_rect(Fl_Window *win, int x, int y, int w, int h);
213 static unsigned char *bitmap_from_window_rect(Fl_Window *win, int x, int y, int w, int h, int
*bytesPerPixel);
214 static Fl_Region intersect_region_and_rect(Fl_Region current, int x, int y, int w, int h);
215 static void *get_carbon_function(const char *name);
216 static void screen_work_area(int &X, int &Y, int &W, int &H, int n); // compute work area of a given
screen
217 static int next_marked_length; // next length of marked text after current marked text will have been
replaced
218 static int insertion_point_location(int *px, int *py, int *pheight); // computes window coordinates &
height of insertion point
219 static const int CoreText_threshold; // Mac OS version from which the Core Text API is used to
display text
220 static Fl_Fontdesc* calc_fl_fonts(void); // computes the fl_fonts global variable
221 static int dnd(int use_selection); // call Fl_X::dnd(1) to support text dragging
222 static int calc_mac_os_version(void); // computes the fl_mac_os_version global variable
223 static void clip_to_rounded_corners(CGContextRef gc, int w, int h);
224 static void *get_titlebar_layer(Fl_Window *win);
225 static void draw_layer_to_context(void *layer, CGContextRef ctxt, int w, int h);
226 private:
227 #if FLTK_ABI_VERSION >= 10304
228 CGRect* subRect_; // makes sure subwindow remains inside its parent window
229 // stores 3 binary flags: whether window is mapped to retina display; whether resolution just
changed;
230 // whether window is OpenGL and is currently being resized.
231 unsigned mapped_to_retina_;
232 #else
233 bool subwindow; // for ABI compatibility, useless with true subwindows
234 #endif
235 };
236
237 extern Window fl_window;
238
239 #endif // FL_LIBRARY || FL_INTERNALS
240
241 typedef CGImageRef Fl_Bitmap;
242

```

```

243 extern CGContextRef fl_gc;
244
245 extern Window fl_xid(const Fl_Window*);
246 extern Fl_Window* fl_find(Window xid);
247 void fl_clip_region(Fl_Region);
248
249 extern Fl_Bitmask fl_create_bitmask(int w, int h, const uchar *data);
250 extern Fl_Bitmask fl_create_alphamask(int w, int h, int d, int ld, const uchar *data);
251 extern void fl_delete_bitmask(Fl_Bitmask bm);
252 extern Fl_Offscreen fl_create_offscreen(int w, int h);
253 extern void fl_copy_offscreen(int x,int y,int w,int h, Fl_Offscreen gWorld, int srcx,int srcy);
254 extern void fl_delete_offscreen(Fl_Offscreen gWorld);
255 extern void fl_begin_offscreen(Fl_Offscreen gWorld);
256 extern void fl_end_offscreen();
257
258 extern int fl_parse_color(const char* p, uchar& r, uchar& g, uchar& b);
259 extern void fl_open_display();
260
261 #endif // FL_DOXYGEN
275 extern void fl_open_callback(void (*cb)(const char *));
276
285 extern void fl_mac_set_about( Fl_Callback *cb, void *user_data, int shortcut = 0);
286
289 extern int fl_mac_os_version;
290
297 extern int fl_mac_quit_early;
298
301 extern class Fl_Sys_Menu_Bar *fl_sys_menu_bar;
302
303 struct Fl_Menu_Item;
304
305 class Fl_Mac_App_Menu {
306 public:
307     static const char *about;
308     static const char *print;
309     static const char *services;
310     static const char *hide;
311     static const char *hide_others;
312     static const char *show;
313     static const char *quit;
314     static void custom_application_menu_items(const Fl_Menu_Item *m);
315 };
316
319 //
320 // End of "$Id$".
321 //
322

```

32.166 math.h

```

1 //
2 // "$Id$"
3 //
4 // Math header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 // Xcode on OS X includes files by recursing down into directories.
20 // This code catches the cycle and directly includes the required file.
21 #ifdef fl_math_h_cyclic_include
22 # include "/usr/include/math.h"
23 #endif
24
25 #ifndef fl_math_h
26 # define fl_math_h
27
28 # define fl_math_h_cyclic_include
29 # include <math.h>
30 # undef fl_math_h_cyclic_include
31
32 # ifdef __EMX__
33 # include <float.h>
34 # endif
35

```

```

36
37 #  ifndef M_PI
38 #      define M_PI           3.14159265358979323846
39 #      define M_PI_2        1.57079632679489661923
40 #      define M_PI_4        0.78539816339744830962
41 #      define M_1_PI        0.31830988618379067154
42 #      define M_2_PI        0.63661977236758134308
43 #  endif // !M_PI
44
45 #  ifndef M_SQRT2
46 #      define M_SQRT2        1.41421356237309504880
47 #      define M_SQRT1_2     0.70710678118654752440
48 #  endif // !M_SQRT2
49
50 #  if (defined(WIN32) || defined(CRAY)) && !defined(__MINGW32__) && !defined(__MWERKS__)
51
52 inline double rint(double v) {return floor(v+.5);}
53 inline double copysign(double a, double b) {return b<0 ? -a : a;}
54
55 #  endif // (WIN32 || CRAY) && !__MINGW32__ && !__MWERKS__
56
57 #endif // !fl_math_h
58
59
60 //
61 // End of "$Id$".
62 //

```

32.167 names.h

```

1 //
2 // "$Id$"
3 //
4 // Event names header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software.  Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file.  If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 // Thanks to Greg Ercolano for this addition.
20
21 #ifndef FL_NAMES_H
22 #define FL_NAMES_H
23
24 const char * const fl_eventnames[] =
25 {
26     "FL_NO_EVENT",
27     "FL_PUSH",
28     "FL_RELEASE",
29     "FL_ENTER",
30     "FL_LEAVE",
31     "FL_DRAG",
32     "FL_FOCUS",
33     "FL_UNFOCUS",
34     "FL_KEYDOWN",
35     "FL_KEYUP",
36     "FL_CLOSE",
37     "FL_MOVE",
38     "FL_SHORTCUT",
39     "FL_DEACTIVATE",
40     "FL_ACTIVATE",
41     "FL_HIDE",
42     "FL_SHOW",
43     "FL_PASTE",
44     "FL_SELECTIONCLEAR",
45     "FL_MOUSEWHEEL",
46     "FL_DND_ENTER",
47     "FL_DND_DRAG",
48     "FL_DND_LEAVE",
49     "FL_DND_RELEASE",
50     "FL_SCREEN_CONFIGURATION_CHANGED",
51     "FL_FULLSCREEN",
52     "FL_ZOOM_GESTURE",
53     "FL_EVENT_27", // not yet defined, just in case they /will/ be defined ...
54     "FL_EVENT_28",
55     "FL_EVENT_29",

```

```

75  "FL_EVENT_30"
76 };
77
95 const char * const fl_fontnames[] =
96 {
97  "FL_HELVETICA",
98  "FL_HELVETICA_BOLD",
99  "FL_HELVETICA_ITALIC",
100 "FL_HELVETICA_BOLD_ITALIC",
101  "FL_COURIER",
102  "FL_COURIER_BOLD",
103  "FL_COURIER_ITALIC",
104  "FL_COURIER_BOLD_ITALIC",
105  "FL_TIMES",
106  "FL_TIMES_BOLD",
107  "FL_TIMES_ITALIC",
108  "FL_TIMES_BOLD_ITALIC",
109  "FL_SYMBOL",
110  "FL_SCREEN",
111  "FL_SCREEN_BOLD",
112  "FL_ZAPF_DINGBATS",
113 };
114
117 #endif /* FL_NAMES_H */
118
119 //
120 // End of "$Id$".
121 //

```

32.168 platform.H

```

1 //
2 // "$Id$"
3 //
4 // Platform abstraction header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2018 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 // This file is present for compatibility with FLTK 1.4 and later.
20
21 // In FLTK 1.4 FL/platform.H replaces FL/x.H. FLTK 1.4 code that
22 // includes FL/platform.H instead of FL/x.H can now be compiled with
23 // FLTK 1.3.5 and later versions.
24
25 #if !defined(FL_X_H)
26 #include <FL/x.H>
27 #endif
28
29 //
30 // End of "$Id$".
31 //

```

32.169 win32.H

```

1 //
2 // "$Id$"
3 //
4 // WIN32 header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2012 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //

```



```

18
19 // Do not directly include this file, instead use <FL/x.H>. It will
20 // include this file if WIN32 is defined. This is to encourage
21 // portability of even the system-specific code...
22
23 #ifndef FL_DOXYGEN
24 #ifndef FL_X_H
25 # error "Never use <FL/win32.H> directly; include <FL/x.H> instead."
26 #endif // !FL_X_H
27
28 #include <windows.h>
29 typedef HRGN Fl_Region;
30 typedef HWND Window;
31 typedef POINT XPoint;
32
33 #include <FL/Fl_Window.H>
34
35 // this part is included only when compiling the FLTK library or if requested explicitly
36 #if defined(FL_LIBRARY) || defined(FL_INTERNALS)
37
38 // In some of the distributions, the gcc header files are missing some stuff:
39 #ifndef LPMINMAXINFO
40 #define LPMINMAXINFO MINMAXINFO*
41 #endif
42 #ifndef VK_LWIN
43 #define VK_LWIN 0x5B
44 #define VK_RWIN 0x5C
45 #define VK_APPS 0x5D
46 #endif
47
48 // some random X equivalents
49 struct XRectangle {int x, y, width, height;};
50 extern Fl_Region XRectangleRegion(int x, int y, int w, int h);
51 inline void XDestroyRegion(Fl_Region r) {DeleteObject(r);}
52 inline void XClipBox(Fl_Region r, XRectangle* rect) {
53     RECT win_rect; GetRgnBox(r, &win_rect);
54     rect->x=win_rect.left;
55     rect->y=win_rect.top;
56     rect->width=win_rect.right-win_rect.left;
57     rect->height=win_rect.bottom-win_rect.top;
58 }
59 #define XDestroyWindow(a,b) DestroyWindow(b)
60 #define XMapWindow(a,b) ShowWindow(b, SW_RESTORE)
61 #define XUnmapWindow(a,b) ShowWindow(b, SW_HIDE)
62
63 // this object contains all win32-specific stuff about a window:
64 // Warning: this object is highly subject to change!
65 class FL_EXPORT Fl_X {
66 public:
67     // member variables - add new variables only at the end of this block
68     Window xid;
69     HBITMAP other_xid; // for double-buffered windows
70     Fl_Window* w;
71     Fl_Region region;
72     Fl_X *next;
73     int wait_for_expose;
74     HDC private_dc; // used for OpenGL
75     HCURSOR cursor;
76     int custom_cursor;
77     HDC saved_hdc; // saves the handle of the DC currently loaded
78     // static variables, static functions and member functions
79     static Fl_X* first;
80     static Fl_X* i(const Fl_Window* w) {return w->i;}
81     static int fake_X_wm(const Fl_Window* w, int &X, int &Y,
82         int &bt, int &bx, int &by);
83     void make_fullscreen(int X, int Y, int W, int H);
84     void setwindow(Fl_Window* wi) {w=wi; wi->i=this;}
85     void flush() {w->flush();}
86     void set_minmax(LPMINMAXINFO minmax);
87     void mapraise();
88     static void set_default_icons(const Fl_RGB_Image*[], int);
89     static void set_default_icons(HICON, HICON);
90     void set_icons();
91     int set_cursor(Fl_Cursor);
92     int set_cursor(const Fl_RGB_Image*, int, int);
93     static Fl_X* make(Fl_Window*);
94 };
95 extern FL_EXPORT UINT fl_wake_msg;
96 extern FL_EXPORT char fl_override_redirect; // hack into Fl_Window::make_xid()
97 extern FL_EXPORT int fl_background_pixel; // hack into Fl_Window::make_xid()
98 extern FL_EXPORT HPALETTE fl_palette; // non-zero only on 8-bit displays!
99 extern FL_EXPORT void fl_release_dc(HWND w, HDC dc);
100 extern FL_EXPORT void fl_save_dc(HWND w, HDC dc);
101
102 inline Window fl_xid(const Fl_Window* w) { Fl_X *temp = Fl_X::i(w); return temp ? temp->xid : 0; }
103
104 extern FL_EXPORT void fl_open_display();

```

```

105
106 #else
107 FL_EXPORT Window fl_xid(const Fl_Window* w);
108 #define fl_xid(w) fl_xid_(w)
109 #endif // FL_LIBRARY || FL_INTERNALS
110
111 FL_EXPORT Fl_Window* fl_find(Window xid);
112 void fl_clip_region(Fl_Region);
113
114 // most recent fl_color() or fl_rgbcolor() points at one of these:
115 extern FL_EXPORT struct Fl_XMap {
116     COLORREF rgb; // this should be the type the RGB() macro returns
117     HPEN pen; // pen, 0 if none created yet
118     int brush; // ref to solid brush, 0 if none created yet
119 } *fl_current_xmap;
120 inline COLORREF fl_RGB() {return fl_current_xmap->rgb;}
121 inline HPEN fl_pen() {return fl_current_xmap->pen;}
122 FL_EXPORT HBRUSH fl_brush(); // allocates a brush if necessary
123 FL_EXPORT HBRUSH fl_brush_action(int); // now does the real work
124
125 extern FL_EXPORT HINSTANCE fl_display;
126 extern FL_EXPORT Window fl_window;
127 extern FL_EXPORT HDC fl_gc;
128 extern FL_EXPORT MSG fl_msg;
129 extern FL_EXPORT HDC fl_GetDC(Window);
130 extern FL_EXPORT HDC fl_makeDC(HBITMAP);
131
132 // off-screen pixmaps: create, destroy, draw into, copy to window
133 typedef HBITMAP Fl_Offscreen;
134 #define fl_create_offscreen(w, h) \
135 CreateCompatibleBitmap( (fl_gc ? fl_gc : fl_GetDC(0) ) , w, h)
136
137 # define fl_begin_offscreen(b) \
138 HDC _sgc=fl_gc; Window _sw=fl_window; \
139 Fl_Surface_Device *_ss = Fl_Surface_Device::surface(); \
140 Fl_Display_Device::display_device()->set_current(); \
141 fl_gc=fl_makeDC(b); int _savedc = SaveDC(fl_gc); fl_window=(HWND)b; fl_push_no_clip()
142 # define fl_end_offscreen() \
143 fl_pop_clip(); RestoreDC(fl_gc, _savedc); DeleteDC(fl_gc); _ss->set_current(); fl_window=_sw; fl_gc =
144 _sgc
145
146 FL_EXPORT void fl_copy_offscreen(int x,int y,int w,int h,HBITMAP pixmap,int srcx,int srcy);
147 #define fl_delete_offscreen(bitmap) DeleteObject(bitmap)
148
149 // Bitmap masks
150 typedef HBITMAP Fl_Bitmask;
151
152 extern FL_EXPORT Fl_Bitmask fl_create_bitmask(int w, int h, const uchar *data);
153 extern FL_EXPORT Fl_Bitmask fl_create_alphamask(int w, int h, int d, int ld, const uchar *data);
154 extern FL_EXPORT void fl_delete_bitmask(Fl_Bitmask bm);
155
156 // Dummy function to register a function for opening files via the window manager...
157 inline void fl_open_callback(void (*)(const char *)) {}
158
159 extern FL_EXPORT int fl_parse_color(const char* p, uchar& r, uchar& g, uchar& b);
160 #endif // FL_DOXYGEN
161 //
162 // End of "$Id$".
163 //

```

32.170 x.H

```

1 //
2 // "$Id$"
3 //
4 // X11 header file for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2012 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 // These are internal fltk symbols that are necessary or useful for
20 // calling Xlib. You should include this file if (and ONLY if) you
21 // need to call Xlib directly. These symbols may not exist on non-X

```

```

22 // systems.
23
24 #if !defined(FL_X_H) && !defined(FL_DOXYGEN)
25 # define FL_X_H
26
27 # include "Enumerations.H"
28
29 # ifdef WIN32
30 #   include "win32.H"
31 # elif defined(__APPLE__)
32 #   include "mac.H"
33 # else
34 #   if defined(_ABIN32) || defined(_ABI64) // fix for broken SGI Irix X .h files
35 #     pragma set woff 3322
36 #   endif
37 #   include <X11/Xlib.h>
38 #   include <X11/Xutil.h>
39 #   if defined(_ABIN32) || defined(_ABI64)
40 #     pragma reset woff 3322
41 #   endif
42 #   include <X11/Xatom.h>
43 #   include "Fl_Window.H"
44 // Mirror X definition of Region to Fl_Region, for portability...
45 typedef Region Fl_Region;
46
47 FL_EXPORT void fl_open_display();
48 FL_EXPORT void fl_open_display(Display*);
49 FL_EXPORT void fl_close_display();
50
51 // constant info about the X server connection:
52 extern FL_EXPORT Display *fl_display;
53 extern FL_EXPORT int fl_screen;
54 extern FL_EXPORT XVisualInfo *fl_visual;
55 extern FL_EXPORT Colormap fl_colormap;
56
57
58 // drawing functions:
59 extern FL_EXPORT GC fl_gc;
60 extern FL_EXPORT Window fl_window;
61 FL_EXPORT ulong fl_xpixel(Fl_Color i);
62 FL_EXPORT ulong fl_xpixel(uchar r, uchar g, uchar b);
63 FL_EXPORT void fl_clip_region(Fl_Region);
64 FL_EXPORT Fl_Region fl_clip_region();
65
66 // feed events into fltk:
67 FL_EXPORT int fl_handle(const XEvent&);
68
69 // you can use these in Fl::add_handler() to look at events:
70 extern FL_EXPORT const XEvent* fl_xevent;
71 extern FL_EXPORT ulong fl_event_time;
72
73 // off-screen pixmaps: create, destroy, draw into, copy to window:
74 typedef ulong Fl_Offscreen;
75 # define fl_create_offscreen(w,h) XCreatePixmap(fl_display, RootWindow(fl_display, fl_screen), w, h,
76   fl_visual->depth)
76 # define fl_create_offscreen_with_alpha(w,h) XCreatePixmap(fl_display, RootWindow(fl_display,
77   fl_screen), w, h, 32)
77 // begin/end are macros that save the old state in local variables:
78 # define fl_begin_offscreen(pixmap) \
79 Window _sw=fl_window; fl_window=pixmap; \
80 GC _sgc = fl_gc; if (!sgc) fl_gc = XCreateGC(fl_display, pixmap, 0, 0); \
81 Fl_Surface_Device *_ss = Fl_Surface_Device::surface();
82   Fl_Display_Device::display_device()->set_current(); \
82 fl_push_no_clip()
83 # define fl_end_offscreen() \
84 fl_pop_clip(); fl_window = _sw; _ss->set_current(); \
85 if (!sgc) XFreeGC(fl_display, fl_gc); fl_gc = _sgc
86
87 extern FL_EXPORT void fl_copy_offscreen(int x, int y, int w, int h, Fl_Offscreen pixmap, int srcx, int
88   srcy);
88 # define fl_delete_offscreen(pixmap) XFreePixmap(fl_display, pixmap)
89
90 // Bitmap masks
91 typedef ulong Fl_Bitmask;
92
93 extern FL_EXPORT Fl_Bitmask fl_create_bitmask(int w, int h, const uchar *data);
94 extern FL_EXPORT Fl_Bitmask fl_create_alphamask(int w, int h, int d, int ld, const uchar *data);
95 extern FL_EXPORT void fl_delete_bitmask(Fl_Bitmask bm);
96
97 #if defined(FL_LIBRARY) || defined(FL_INTERNALS)
98 extern FL_EXPORT Window fl_message_window;
99 extern FL_EXPORT void *fl_xfont;
100 FL_EXPORT Fl_Region XRectangleRegion(int x, int y, int w, int h); // in fl_rect.cxx
101
102 // access to core fonts:
103 // This class provides a "smart pointer" that returns a pointer to an XFontStruct.
104 // The global variable fl_xfont can be called wherever a bitmap "core" font is

```

```

105 // needed, e.g. when rendering to a GL context under X11.
106 // With Xlib / X11 fonts, fl_xfont will return the current selected font.
107 // With XFT / X11 fonts, fl_xfont will attempt to return the bitmap "core" font most
108 // similar to (usually the same as) the current XFT font.
109 class Fl_XFont_On_Demand
110 {
111 public:
112   Fl_XFont_On_Demand(XFontStruct* p = NULL) : ptr(p) { }
113   Fl_XFont_On_Demand& operator=(const Fl_XFont_On_Demand& x)
114   { ptr = x.ptr; return *this; }
115   Fl_XFont_On_Demand& operator=(XFontStruct* p)
116   { ptr = p; return *this; }
117   XFontStruct* value();
118   operator XFontStruct*() { return value(); }
119   XFontStruct& operator*() { return *value(); }
120   XFontStruct* operator->() { return value(); }
121   bool operator==(const Fl_XFont_On_Demand& x) { return ptr == x.ptr; }
122   bool operator!=(const Fl_XFont_On_Demand& x) { return ptr != x.ptr; }
123 private:
124   XFontStruct *ptr;
125 };
126 extern FL_EXPORT Fl_XFont_On_Demand fl_xfont;
127 extern FL_EXPORT XFontStruct* fl_X_core_font();
128
129 // this object contains all X-specific stuff about a window:
130 // Warning: this object is highly subject to change!
131 // FL_LIBRARY or FL_INTERNALS must be defined to access this class.
132 class FL_EXPORT Fl_X {
133 public:
134   Window xid;
135   Window other_xid;
136   Fl_Window *w;
137   Fl_Region region;
138   Fl_X *next;
139   char wait_for_expose;
140   char backbuffer_bad; // used for XDBE
141   static Fl_X* first;
142   static Fl_X* i(const Fl_Window* wi) {return wi->i;}
143   void setwindow(Fl_Window* wi) {w=wi; wi->i=this;}
144   void sendxjunk();
145   static void set_default_icons(const Fl_RGB_Image*[], int);
146   void set_icons();
147   int set_cursor(Fl_Cursor);
148   int set_cursor(const Fl_RGB_Image*, int, int);
149   static void make_xid(Fl_Window*, XVisualInfo* =fl_visual, Colormap=fl_colormap);
150   static Fl_X* set_xid(Fl_Window*, Window);
151   // kludges to get around protection:
152   void flush() {w->flush();}
153   static void x(Fl_Window* wi, int X) {wi->x(X);}
154   static void y(Fl_Window* wi, int Y) {wi->y(Y);}
155   static int ewmh_supported();
156   static int xrender_supported();
157   static void activate_window(Window w);
158 };
159
160 extern FL_EXPORT char fl_override_redirect; // hack into Fl_X::make_xid()
161 extern FL_EXPORT int fl_background_pixel; // hack into Fl_X::make_xid()
162
163 inline Window fl_xid(const Fl_Window* w) { Fl_X *xTemp = Fl_X::i(w); return xTemp ? xTemp->xid : 0; }
164
165 #else
166
167 extern FL_EXPORT Window fl_xid_(const Fl_Window* w);
168 #define fl_xid(w) fl_xid_(w)
169
170 #endif // FL_LIBRARY || FL_INTERNALS
171
172 FL_EXPORT Fl_Window* fl_find(Window xid);
173
174
175 // Dummy function to register a function for opening files via the window manager...
176 inline void fl_open_callback(void (*) (const char *)) {}
177
178 extern FL_EXPORT int fl_parse_color(const char* p, uchar& r, uchar& g, uchar& b);
179
180 # endif
181 #endif
182
183 //
184 // End of "$Id$".
185 //

```

32.171 cgdebug.h

```
1 //
```

```
2 // "$Id$"
3 //
4 // OS X Core Graphics debugging help for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18 //
19 // This file allows easier debugging of Mac OS X Core Graphics
20 // code. This file is normally not included into any FLTK builds,
21 // but since it has proven to be tremendously useful in debugging
22 // the FLTK port to "Quartz", I decided to add this file in case
23 // more bugs show up.
24 //
25 // This header is activated by adding the following
26 // line to "config.h"
27 //     #include "src/cgdebug.h"
28 //
29 // Running "./configure" will remove this line from "config.h".
30 //
31 // When used erroneously, Core Graphics prints warnings to
32 // stderr. This is helpful, however it is not possible to
33 // associate a line number or source file with the warning message.
34 // This header file outputs a trace of CG calls, interweaving
35 // them with CG warnings.
36 //
37 // Matthias
38 //
39 #ifndef CGDEBUG
40 #define CGDEBUG
41 //
42 #include <stdio.h>
43 #include <Carbon/Carbon.h>
44 //
45 //+BitmapContextCreate
46 //+BitmapContextGetData
47 // ClipCGContextToRegion
48 // QDBeginCGContext
49 // QDEndCGContext
50 //
51 //+AddArc
52 //+AddLineToPoint
53 // ClipToRect
54 // ClosePath
55 //+ConcatCTM
56 //+DrawImage
57 // FillPath
58 // FillRect
59 // Flush
60 //+GetCTM
61 // MoveToPoint
62 //+Release
63 // RestoreGState
64 // SaveGState
65 //+ScaleCTM
66 //+SetLineCap
67 //+SetLineDash
68 //+SetLineJoin
69 //+SetLineWidth
70 //+SetRGBFillColor
71 //+SetRGBStrokeColor
72 //+SetShouldAntialias
73 //+SetTextMatrix
74 //+StrokePath
75 //+TranslateCTM
76 //
77 inline OSStatus dbgLocation(const char *file, int line)
78 {
79     fprintf(stderr, "%s:%d ", file, line);
80     return 0;
81 }
82 //
83 inline OSStatus dbgEndl()
84 {
85     fprintf(stderr, "\n");
86     return 0;
87 }
88 //
```

```

89
90 inline void dbgCGContextClipToRect(CGContextRef a, CGRect b)
91 {
92     CGContextClipToRect(a, b);
93 }
94
95 #define CGContextClipToRect(a, b) { \
96     fprintf(stderr, "%s:%d ", __FILE__, __LINE__); \
97     dbgCGContextClipToRect(a, b); \
98     fprintf(stderr, "\n"); }
99
100 inline void dbgCGContextFillRect(CGContextRef a, CGRect b)
101 {
102     CGContextFillRect(a, b);
103 }
104
105 #define CGContextFillRect(a, b) { \
106     fprintf(stderr, "%s:%d ", __FILE__, __LINE__); \
107     dbgCGContextFillRect(a, b); \
108     fprintf(stderr, "\n"); }
109
110 inline OSStatus dbgQDEndCGContext(CGrafPtr a, CGContextRef *b)
111 {
112     return QDEndCGContext(a, b);
113 }
114
115 #define QDEndCGContext(a, b) ( \
116     dbgLocation(__FILE__, __LINE__) + \
117     dbgQDEndCGContext(a, b) + \
118     dbgEndl() )
119
120 inline OSStatus dbgQDBeginCGContext(CGrafPtr a, CGContextRef *b)
121 {
122     return QDBeginCGContext(a, b);
123 }
124
125 #define QDBeginCGContext(a, b) ( \
126     dbgLocation(__FILE__, __LINE__) + \
127     dbgQDBeginCGContext(a, b) + \
128     dbgEndl() )
129
130 inline void dbgClipCGContextToRegion(CGContextRef a, const Rect *b, RgnHandle c)
131 {
132     ClipCGContextToRegion(a, b, c);
133 }
134
135 #define ClipCGContextToRegion(a, b, c) { \
136     fprintf(stderr, "%s:%d ", __FILE__, __LINE__); \
137     dbgClipCGContextToRegion(a, b, c); \
138     fprintf(stderr, "\n"); }
139
140 inline void dbgCGContextMoveToPoint(CGContextRef context, float x, float y)
141 {
142     CGContextMoveToPoint(context, x, y);
143 }
144
145 #define CGContextMoveToPoint(a, b, c) { \
146     fprintf(stderr, "%s:%d ", __FILE__, __LINE__); \
147     dbgCGContextMoveToPoint(a, b, c); \
148     fprintf(stderr, "\n"); }
149
150 inline void dbgCGContextFillPath(CGContextRef context)
151 {
152     CGContextFillPath(context);
153 }
154
155 #define CGContextFillPath(a) { \
156     fprintf(stderr, "%s:%d ", __FILE__, __LINE__); \
157     dbgCGContextFillPath(a); \
158     fprintf(stderr, "\n"); }
159
160 inline void dbgCGContextClosePath(CGContextRef context)
161 {
162     CGContextClosePath(context);
163 }
164
165 #define CGContextClosePath(a) { \
166     fprintf(stderr, "%s:%d ", __FILE__, __LINE__); \
167     dbgCGContextClosePath(a); \
168     fprintf(stderr, "\n"); }
169
170 inline void dbgCGContextFlush(CGContextRef context)
171 {
172     CGContextFlush(context);
173 }
174
175 #define CGContextFlush(a) { \

```

```

176 fprintf(stderr, "%s:%d ", __FILE__, __LINE__); \
177     dbgCGContextFlush(a); \
178     fprintf(stderr, "\n"); }
179
180 inline void dbgCGContextSaveGState(CGContextRef context)
181 {
182     CGContextSaveGState(context);
183 }
184
185 #define CGContextSaveGState(a) { \
186     fprintf(stderr, "%s:%d ", __FILE__, __LINE__); \
187     dbgCGContextSaveGState(a); \
188     fprintf(stderr, "\n"); }
189
190 inline void dbgCGContextRestoreGState(CGContextRef context)
191 {
192     CGContextRestoreGState(context);
193 }
194
195 #define CGContextRestoreGState(a) { \
196     fprintf(stderr, "%s:%d ", __FILE__, __LINE__); \
197     dbgCGContextRestoreGState(a); \
198     fprintf(stderr, "\n"); }
199
200
201 #endif
202
203 //
204 // End of "$Id$".
205 //
206

```

32.172 fastarrow.h

```

1 #define fastarrow_width 16
2 #define fastarrow_height 16
3 static const unsigned char fastarrow_bits[] = {
4     0x00, 0x00, 0x00, 0x07, 0xe0, 0x07, 0xfc, 0x03, 0xff, 0xff, 0xfc, 0x03,
5     0xe0, 0x07, 0x00, 0x07, 0xe0, 0x00, 0xe0, 0x07, 0xc0, 0x3f, 0xff, 0xff,
6     0xc0, 0x3f, 0xe0, 0x07, 0xe0, 0x00, 0x00, 0x00};

```

32.173 fl_arc.cxx File Reference

Utility functions for drawing arcs and circles.

```

#include <FL/fl_draw.H>
#include <FL/math.h>

```

32.173.1 Detailed Description

Utility functions for drawing arcs and circles.

32.174 fl_arci.cxx File Reference

Utility functions for drawing circles using integers.

```

#include <FL/fl_draw.H>
#include <FL/x.H>
#include <config.h>

```

32.174.1 Detailed Description

Utility functions for drawing circles using integers.

32.175 fl_ask.cxx File Reference

Utility Functions for Common Dialogs.

```
#include <stdio.h>
#include <stdarg.h>
#include "flstring.h"
#include <FL/Fl.H>
#include <FL/fl_ask.H>
#include <FL/Fl_Box.H>
#include <FL/Fl_Button.H>
#include <FL/Fl_Return_Button.H>
#include <FL/Fl_Window.H>
#include <FL/Fl_Input.H>
#include <FL/Fl_Secret_Input.H>
#include <FL/x.H>
#include <FL/fl_draw.H>
```

Functions

- void [fl_alert](#) (const char *fmt,...)

Shows an alert message dialog box.
- int [fl_ask](#) (const char *fmt,...)

*Shows a dialog displaying the *fmt* message, this dialog features 2 yes/no buttons.*
- void [fl_beep](#) (int type)

Emits a system beep message.
- int [fl_choice](#) (const char *fmt, const char *b0, const char *b1, const char *b2,...)

*Shows a dialog displaying the printf style *fmt* message, this dialog features up to 3 customizable choice buttons.*
- int [fl_choice_n](#) (const char *fmt, const char *b0, const char *b1, const char *b2,...)

Like [fl_choice\(\)](#) but with extended (negative) return values.
- const char * [fl_input](#) (const char *fmt, const char *defstr,...)

*Shows an input dialog displaying the *fmt* message.*
- void [fl_message](#) (const char *fmt,...)

Shows an information message dialog box.
- void [fl_message_hotspot](#) (int enable)

Sets whether or not to move the common message box used in many common dialogs like [fl_message\(\)](#), [fl_alert\(\)](#), [fl_ask\(\)](#), [fl_choice\(\)](#), [fl_input\(\)](#), [fl_password\(\)](#) to follow the mouse pointer.
- int [fl_message_hotspot](#) (void)

Gets whether or not to move the common message box used in many common dialogs like [fl_message\(\)](#), [fl_alert\(\)](#), [fl_ask\(\)](#), [fl_choice\(\)](#), [fl_input\(\)](#), [fl_password\(\)](#) to follow the mouse pointer.
- [Fl_Widget](#) * [fl_message_icon](#) ()

Gets the [Fl_Box](#) icon container of the current default dialog used in many common dialogs like [fl_message\(\)](#), [fl_alert\(\)](#), [fl_ask\(\)](#), [fl_choice\(\)](#), [fl_input\(\)](#), [fl_password\(\)](#)
- void [fl_message_title](#) (const char *title)

Sets the title of the dialog window used in many common dialogs.
- void [fl_message_title_default](#) (const char *title)

Sets the default title of the dialog window used in many common dialogs.
- const char * [fl_password](#) (const char *fmt, const char *defstr,...)

*Shows an input dialog displaying the *fmt* message.*

Variables

- const char * [fl_cancel](#) = "Cancel"

string pointer used in common dialogs, you can change it to another language
- const char * [fl_close](#) = "Close"

string pointer used in common dialogs, you can change it to another language
- [Fl_Font](#) [fl_message_font_](#) = [FL_HELVETICA](#)

- `Fl_Fontsize fl_message_size_ = -1`
- `const char * fl_no = "No"`
string pointer used in common dialogs, you can change it to another language
- `const char * fl_ok = "OK"`
string pointer used in common dialogs, you can change it to another language
- `const char * fl_yes = "Yes"`
string pointer used in common dialogs, you can change it to another language

32.175.1 Detailed Description

Utility Functions for Common Dialogs.

32.176 fl_boxtype.cxx File Reference

drawing code for common box types.

```
#include <FL/Fl.H>
#include <FL/Fl_Widget.H>
#include <FL/fl_draw.H>
#include <config.h>
```

Macros

- `#define D1 BORDER_WIDTH`
- `#define D2 (BORDER_WIDTH+BORDER_WIDTH)`
- `#define fl_border_box fl_rectbound`
allow consistent naming

Functions

- `void fl_border_frame (int x, int y, int w, int h, Fl_Color c)`
Draws a frame of type FL_BORDER_FRAME.
- `void fl_down_box (int x, int y, int w, int h, Fl_Color c)`
Draws a box of type FL_DOWN_BOX.
- `void fl_down_frame (int x, int y, int w, int h, Fl_Color c)`
Draws a frame of type FL_DOWN_FRAME.
- `void fl_draw_box (Fl_Boxtype t, int x, int y, int w, int h, Fl_Color c)`
Draws a box using given type, position, size and color.
- `void fl_embossed_box (int x, int y, int w, int h, Fl_Color c)`
Draws a box of type FL_EMBOSSED_BOX.
- `void fl_embossed_frame (int x, int y, int w, int h, Fl_Color c)`
Draws a frame of type FL_EMBOSSED_FRAME.
- `void fl_engraved_box (int x, int y, int w, int h, Fl_Color c)`
Draws a box of type FL_ENGRAVED_BOX.
- `void fl_engraved_frame (int x, int y, int w, int h, Fl_Color c)`
Draws a frame of type FL_ENGRAVED_FRAME.
- `void fl_flat_box (int x, int y, int w, int h, Fl_Color c)`
Draws a box of type FL_FLAT_BOX.
- `void fl_frame (const char *s, int x, int y, int w, int h)`
Draws a series of line segments around the given box.
- `void fl_frame2 (const char *s, int x, int y, int w, int h)`
Draws a series of line segments around the given box.
- `const uchar * fl_gray_ramp ()`

- void `fl_internal_boxtype` (`Fl_Boxtype` t, `Fl_Box_Draw_F` *f)
Sets the drawing function for a given box type.
- void `fl_no_box` (int, int, int, int, `Fl_Color`)
Draws a box of type `FL_NO_BOX`.
- void `fl_rectbound` (int x, int y, int w, int h, `Fl_Color` bgcolor)
Draws a bounded rectangle with a given position, size and color.
- void `fl_thin_down_box` (int x, int y, int w, int h, `Fl_Color` c)
Draws a box of type `FL_THIN_DOWN_BOX`.
- void `fl_thin_down_frame` (int x, int y, int w, int h, `Fl_Color`)
Draws a frame of type `FL_THIN_DOWN_FRAME`.
- void `fl_thin_up_box` (int x, int y, int w, int h, `Fl_Color` c)
Draws a box of type `FL_THIN_UP_BOX`.
- void `fl_thin_up_frame` (int x, int y, int w, int h, `Fl_Color`)
Draws a frame of type `FL_THIN_UP_FRAME`.
- void `fl_up_box` (int x, int y, int w, int h, `Fl_Color` c)
Draws a box of type `FL_UP_BOX`.
- void `fl_up_frame` (int x, int y, int w, int h, `Fl_Color`)
Draws a frame of type `FL_UP_FRAME`.

32.176.1 Detailed Description

drawing code for common box types.

32.176.2 Function Documentation

32.176.2.1 `fl_internal_boxtype()`

```
void fl_internal_boxtype (
    Fl_Boxtype t,
    Fl_Box_Draw_F * f )
```

Sets the drawing function for a given box type.

Parameters

in	t	box type
in	f	box drawing function

32.176.2.2 `fl_rectbound()`

```
void fl_rectbound (
    int x,
    int y,
    int w,
    int h,
    Fl_Color bgcolor )
```

Draws a bounded rectangle with a given position, size and color.
Equivalent to drawing a box of type `FL_BORDER_BOX`.

32.177 `fl_cmap.h`

```
1          0x00000000,
```

```
2      0xff000000,
3      0x00ff0000,
4      0xfffff000,
5      0x0000ff00,
6      0xff00ff00,
7      0x0fffff00,
8      0xffffffff00,
9      0x55555500,
10     0xc6717100,
11     0x71c67100,
12     0x8e8e3800,
13     0x7171c600,
14     0x8e388e00,
15     0x388e8e00,
16     0x00008000,
17     0xa8a89800,
18     0xe8e8d800,
19     0x68685800,
20     0x98a8a800,
21     0xd8e8e800,
22     0x58686800,
23     0x9c9ca800,
24     0xdc8dc800,
25     0x5c5c6800,
26     0x9ca89c00,
27     0xdc8dc00,
28     0x5c685c00,
29     0x90909000,
30     0xc0c0c000,
31     0x50505000,
32     0xa0a0a000,
33     0x00000000,
34     0x0d0d0d00,
35     0x1a1a1a00,
36     0x26262600,
37     0x31313100,
38     0x3d3d3d00,
39     0x48484800,
40     0x55555500,
41     0x5f5f5f00,
42     0x6a6a6a00,
43     0x75757500,
44     0x80808000,
45     0x8a8a8a00,
46     0x95959500,
47     0xa0a0a000,
48     0xaaaaaaaa00,
49     0xb5b5b500,
50     0xc0c0c000,
51     0xcbcbcb00,
52     0xd5d5d500,
53     0xe0e0e000,
54     0xeaeaea00,
55     0xf5f5f500,
56     0xffffffff00,
57     0x00000000,
58     0x00240000,
59     0x00480000,
60     0x006d0000,
61     0x00910000,
62     0x00b60000,
63     0x00da0000,
64     0x00ff0000,
65     0x3f000000,
66     0x3f240000,
67     0x3f480000,
68     0x3f6d0000,
69     0x3f910000,
70     0x3fb60000,
71     0x3fda0000,
72     0x3fff0000,
73     0x7f000000,
74     0x7f240000,
75     0x7f480000,
76     0x7f6d0000,
77     0x7f910000,
78     0x7fb60000,
79     0x7fda0000,
80     0x7fff0000,
81     0xbf000000,
82     0xbf240000,
83     0xbf480000,
84     0xbf6d0000,
85     0xbf910000,
86     0xbf6d0000,
87     0xbfda0000,
88     0xbfff0000,
```

```
89      0xff000000,
90      0xff240000,
91      0xff480000,
92      0xff6d0000,
93      0xff910000,
94      0xffb60000,
95      0xffda0000,
96      0xffff0000,
97      0x00003f00,
98      0x00243f00,
99      0x00483f00,
100     0x006d3f00,
101     0x00913f00,
102     0x00b63f00,
103     0x00da3f00,
104     0x00ff3f00,
105     0x3f003f00,
106     0x3f243f00,
107     0x3f483f00,
108     0x3f6d3f00,
109     0x3f913f00,
110     0x3fb63f00,
111     0x3fda3f00,
112     0x3fff3f00,
113     0x7f003f00,
114     0x7f243f00,
115     0x7f483f00,
116     0x7f6d3f00,
117     0x7f913f00,
118     0x7fb63f00,
119     0x7fda3f00,
120     0x7fff3f00,
121     0xbf003f00,
122     0xbf243f00,
123     0xbf483f00,
124     0xbf6d3f00,
125     0xbf913f00,
126     0xbf b63f00,
127     0xbfda3f00,
128     0xbfff3f00,
129     0xff003f00,
130     0xff243f00,
131     0xff483f00,
132     0xff6d3f00,
133     0xff913f00,
134     0xffb63f00,
135     0xffda3f00,
136     0xffff3f00,
137     0x00007f00,
138     0x00247f00,
139     0x00487f00,
140     0x006d7f00,
141     0x00917f00,
142     0x00b67f00,
143     0x00da7f00,
144     0x00ff7f00,
145     0x3f007f00,
146     0x3f247f00,
147     0x3f487f00,
148     0x3f6d7f00,
149     0x3f917f00,
150     0x3fb67f00,
151     0x3fda7f00,
152     0x3fff7f00,
153     0x7f007f00,
154     0x7f247f00,
155     0x7f487f00,
156     0x7f6d7f00,
157     0x7f917f00,
158     0x7fb67f00,
159     0x7fda7f00,
160     0x7fff7f00,
161     0xbf007f00,
162     0xbf247f00,
163     0xbf487f00,
164     0xbf6d7f00,
165     0xbf917f00,
166     0xbf b67f00,
167     0xbfda7f00,
168     0xbfff7f00,
169     0xff007f00,
170     0xff247f00,
171     0xff487f00,
172     0xff6d7f00,
173     0xff917f00,
174     0xffb67f00,
175     0xffda7f00,
```

```
176     0xffff7f00,
177     0x0000bf00,
178     0x0024bf00,
179     0x0048bf00,
180     0x006dbf00,
181     0x0091bf00,
182     0x00b6bf00,
183     0x00dabf00,
184     0x00ffbf00,
185     0x3f00bf00,
186     0x3f24bf00,
187     0x3f48bf00,
188     0x3f6dbf00,
189     0x3f91bf00,
190     0x3fb6bf00,
191     0x3fdabf00,
192     0x3fffbf00,
193     0x7f00bf00,
194     0x7f24bf00,
195     0x7f48bf00,
196     0x7f6dbf00,
197     0x7f91bf00,
198     0x7fb6bf00,
199     0x7fdabf00,
200     0x7fffbf00,
201     0xbf00bf00,
202     0xbf24bf00,
203     0xbf48bf00,
204     0xbf6dbf00,
205     0xbf91bf00,
206     0xbfb6bf00,
207     0xbfdabf00,
208     0xbfffbf00,
209     0xff00bf00,
210     0xff24bf00,
211     0xff48bf00,
212     0xff6dbf00,
213     0xff91bf00,
214     0xffb6bf00,
215     0xffdabf00,
216     0xffffbf00,
217     0x0000ff00,
218     0x0024ff00,
219     0x0048ff00,
220     0x006dff00,
221     0x0091ff00,
222     0x00b6ff00,
223     0x00daff00,
224     0x00ffff00,
225     0x3f00ff00,
226     0x3f24ff00,
227     0x3f48ff00,
228     0x3f6dff00,
229     0x3f91ff00,
230     0x3fb6ff00,
231     0x3fdaff00,
232     0x3fffff00,
233     0x7f00ff00,
234     0x7f24ff00,
235     0x7f48ff00,
236     0x7f6dff00,
237     0x7f91ff00,
238     0x7fb6ff00,
239     0x7fdaff00,
240     0x7fffff00,
241     0xbf00ff00,
242     0xbf24ff00,
243     0xbf48ff00,
244     0xbf6dff00,
245     0xbf91ff00,
246     0xbfb6ff00,
247     0xbfdaff00,
248     0xbfffff00,
249     0xff00ff00,
250     0xff24ff00,
251     0xff48ff00,
252     0xff6dff00,
253     0xff91ff00,
254     0xffb6ff00,
255     0xffdaff00,
256     0xffffff00
```

32.178 fl_color.cxx File Reference

Color handling.

```
#include "Fl_XColor.H"  
#include <FL/Fl.H>  
#include <FL/x.H>  
#include <FL/fl_draw.H>  
#include "fl_cmap.h"
```

Macros

- `#define fl_overlay 0`
HAVE_OVERLAY determines whether fl_overlay is variable or defined as 0.

Functions

- `Fl_Color fl_color_average (Fl_Color color1, Fl_Color color2, float weight)`
Returns the weighted average color between the two given colors.
- `Fl_Color fl_contrast (Fl_Color fg, Fl_Color bg)`
Returns a color that contrasts with the background color.
- `Fl_Color fl_inactive (Fl_Color c)`
Returns the inactive, dimmed version of the given color.
- `ulong fl_xpixel (Fl_Color i)`
Returns the X pixel number used to draw the given FLTK color index.
- `ulong fl_xpixel (uchar r, uchar g, uchar b)`
Returns the X pixel number used to draw the given rgb color.

Variables

- `uchar fl_bluemask`
color mask used in current color map handling
- `int fl_blueshift`
color shift used in current color map handling
- `int fl_extrashift`
color shift used in current color map handling
- `uchar fl_greenmask`
color mask used in current color map handling
- `int fl_greenshift`
color shift used in current color map handling
- `uchar fl_redmask`
color mask used in current color map handling
- `int fl_redshift`
color shift used in current color map handling
- `Fl_XColor fl_xmap [1][256]`
HAVE_OVERLAY determines whether fl_xmap is one or two planes.

32.178.1 Detailed Description

Color handling.

32.179 FL_compose.cxx File Reference

Utility functions to support text input.

```
#include <FL/Fl.H>
#include <FL/x.H>
```

Variables

- XIC `fl_xim_ic`

32.179.1 Detailed Description

Utility functions to support text input.

32.180 fl_curve.cxx File Reference

Utility for drawing Bezier curves, adding the points to the current `fl_begin/fl_vertex/fl_end` path.

```
#include <FL/fl_draw.H>
#include <math.h>
```

32.180.1 Detailed Description

Utility for drawing Bezier curves, adding the points to the current `fl_begin/fl_vertex/fl_end` path.

Incremental math implementation: I very much doubt this is optimal! From Foley/vanDam page 511. If anybody has a better algorithm, please send it!

32.181 FL_Double_Window.cxx File Reference

`FL_Double_Window` implementation.

```
#include <config.h>
#include <FL/Fl.H>
#include <FL/Fl_Double_Window.H>
#include <FL/Fl_Overlay_Window.H>
#include <FL/Fl_Printer.H>
#include <FL/x.H>
#include <FL/fl_draw.H>
```

Functions

- void `fl_begin_offscreen` (FL_Offscreen ctx)
Send all subsequent drawing commands to this offscreen buffer.
- char `fl_can_do_alpha_blending` ()
Checks whether platform supports true alpha blending for RGBA images.
- void `fl_copy_offscreen` (int x, int y, int w, int h, FL_Offscreen pixmap, int srcx, int srcy)
Copy a rectangular area of the given offscreen buffer into the current drawing destination.
- FL_Offscreen `fl_create_offscreen` (int w, int h)
Creation of an offscreen graphics buffer.
- void `fl_delete_offscreen` (FL_Offscreen ctx)
Deletion of an offscreen graphics buffer.
- void `fl_end_offscreen` ()
Quit sending drawing commands to the current offscreen buffer.

Variables

- const int **stack_max** = 16

32.181.1 Detailed Description

[Fl_Double_Window](#) implementation.

32.182 Fl_Font.H

```

1 //
2 // "$Id$"
3 //
4 // Font definitions for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2011 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 // Two internal fltk data structures:
20 //
21 // Fl_Fontdesc: an entry into the fl_font() table. There is one of these
22 // for each fltk font number.
23 //
24 #ifndef FL_FONT_
25 #define FL_FONT_
26
27 #include <config.h>
28
29 # if USE_XFT
30 typedef struct _XftFont XftFont;
31 # elif !defined(WIN32) && !defined(__APPLE__)
32 #   include "Xutf8.h"
33 # endif // USE_XFT
34
35 class Fl_Font_Descriptor {
36 public:
37     Fl_Font_Descriptor *next;
38     Fl_Fontsize size;
39 #ifndef FL_DOXYGEN // don't bother with platform dependant details in the doc.
40 #   ifdef WIN32
41         HFONT fid;
42         int *width[64];
43         TEXTMETRIC metr;
44         int angle;
45     FL_EXPORT Fl_Font_Descriptor(const char* fontname, Fl_Fontsize size);
46 #   elif defined(__APPLE_QUARTZ__)
47         FL_EXPORT Fl_Font_Descriptor(const char* fontname, Fl_Fontsize size);
48         ATSTextLayout layout;
49 #   elif MAC_OS_X_VERSION_MAX_ALLOWED >= MAC_OS_X_VERSION_10_5
50         CTFontRef fontref;
51         // the unicode span is divided in 512 blocks of 128 characters
52         float *width[512]; // array of arrays of character widths
53     #   endif
54         ATSTextLayout style;
55         short ascent, descent, q_width;
56 #   elif USE_XFT
57         XftFont* font;
58         //const char* encoding;
59         int angle;
60     FL_EXPORT Fl_Font_Descriptor(const char* xfontname, Fl_Fontsize size, int angle);
61 #   else
62         XUtf8FontStruct* font; // X UTF-8 font information
63     FL_EXPORT Fl_Font_Descriptor(const char* xfontname);
64 #   endif
65 #   if HAVE_GL
66         unsigned int listbase; // base of display list, 0 = none
67 #   endif
68 #   ifndef __APPLE_QUARTZ__
69         char glok[64];
70 #   endif // __APPLE_QUARTZ__
71 #   endif // HAVE_GL
72
73     FL_EXPORT ~Fl_Font_Descriptor();

```



```

80
81 #endif // FL_DOXYGEN
82 };
83
84 //extern FL_EXPORT Fl_Font_Descriptor *fl_fontsize; // the currently selected one
85
86 struct Fl_Fontdesc {
87     const char *name;
88     char fontname[128]; // "Pretty" font name
89     Fl_Font_Descriptor *first; // linked list of sizes of this style
90 #   ifdef WIN32
91     char **xlist; // matched X font names
92     int n; // size of xlist, negative = don't free xlist!
93 #   endif
94 };
95
96 extern FL_EXPORT Fl_Fontdesc *fl_fonts; // the table
97
98 #   ifdef WIN32
99 // functions for parsing X font names:
100 FL_EXPORT const char* fl_font_word(const char *p, int n);
101 FL_EXPORT char *fl_find_fontsize(char *name);
102 #   endif
103
104 #endif
105
106 //
107 // End of "$Id$".
108 //

```

32.183 Fl_Gl_Choice.H

```

1 //
2 // "$Id$"
3 //
4 // OpenGL definitions for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2018 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 // Internal interface to set up OpenGL.
20 //
21 // A "Fl_Gl_Choice" is created from an OpenGL mode and holds information
22 // necessary to create a window (on X) and to create an OpenGL "context"
23 // (on both X and Win32).
24 //
25 // fl_create_gl_context takes a window (necessary only on Win32) and an
26 // Fl_Gl_Choice and returns a new OpenGL context. All contexts share
27 // display lists with each other.
28 //
29 // On X another fl_create_gl_context is provided to create it for any
30 // X visual.
31 //
32 // fl_set_gl_context makes the given OpenGL context current and makes
33 // it draw into the passed window. It tracks the current one context
34 // to avoid calling the context switching code when the same context
35 // is used, though it is a mystery to me why the GLX/WGL libraries
36 // don't do this themselves...
37 //
38 // fl_no_gl_context clears that cache so the next fl_set_gl_context is
39 // guaranteed to work.
40 //
41 // fl_delete_gl_context destroys the context.
42 //
43 // This code is used by Fl_Gl_Window, gl_start(), and gl_visual()
44
45 #ifndef Fl_Gl_Choice_H
46 #define Fl_Gl_Choice_H
47
48 // Warning: whatever GLContext is defined to must take exactly the same
49 // space in a structure as a void*!!!
50 #ifdef WIN32
51 #   include <FL/gl.h>
52 #   define GLContext HGLRC
53 #elif defined(__APPLE_QUARTZ__)

```

```

54 # include <OpenGL/gl.h>
55 #ifdef __OBJC__
56 @class NSOpenGLPixelFormat;
57 @class NSOpenGLContext;
58 #else
59 class NSOpenGLPixelFormat;
60 class NSOpenGLContext;
61 #endif // __OBJC__
62 typedef NSOpenGLContext* FLOpenGLContextPtr;
63 # define GLContext FLOpenGLContextPtr
64 #else
65 # include <GL/glx.h>
66 # define GLContext GLXContext
67 # if ! defined(GLX_VERSION_1_3)
68 #   typedef void *GLXFBConfig;
69 # endif
70 #endif
71
72 // Describes crap needed to create a GLContext.
73 class Fl_Gl_Choice {
74   int mode;
75   const int *alist;
76   Fl_Gl_Choice *next;
77 public:
78 #ifdef WIN32
79   int pixelformat; // the visual to use
80   PIXELFORMATDESCRIPTOR pfd; // some wgl calls need this thing
81 #elif defined(__APPLE_QUARTZ__)
82   NSOpenGLPixelFormat* pixelformat;
83 #else
84   XVisualInfo *vis; // the visual to use
85   Colormap colormap; // a colormap for that visual
86   GLXFBConfig best_fb;
87 #endif
88   // Return one of these structures for a given gl mode.
89   // The second argument is a glX attribute list, and is used if mode is
90   // zero. This is not supported on Win32:
91   static Fl_Gl_Choice *find(int mode, const int *);
92 };
93
94 class Fl_Window;
95
96 #ifdef WIN32
97
98 GLContext fl_create_gl_context(Fl_Window*, const Fl_Gl_Choice*, int layer=0);
99
100 #elif defined(__APPLE_QUARTZ__)
101
102 GLContext fl_create_gl_context(Fl_Window*, const Fl_Gl_Choice*, int layer=0);
103
104 #else
105
106 GLContext fl_create_gl_context(XVisualInfo* vis);
107
108 //static inline
109 GLContext fl_create_gl_context(Fl_Window*, const Fl_Gl_Choice* g);/* {
110   return fl_create_gl_context(g->vis);
111 }*/
112
113 #endif
114
115 void fl_set_gl_context(Fl_Window*, GLContext);
116 void fl_no_gl_context();
117 void fl_delete_gl_context(GLContext);
118
119 #endif
120
121 //
122 // End of "$Id$".
123 //

```

32.184 fl_line_style.cxx File Reference

Line style drawing utility hiding different platforms.

```

#include <FL/Fl.H>
#include <FL/fl_draw.H>
#include <FL/x.H>
#include <FL/Fl_Printer.H>
#include "flstring.h"
#include <stdio.h>

```

Variables

- int `fl_line_width_` = 0

32.184.1 Detailed Description

Line style drawing utility hiding different platforms.

32.185 FL_Paged_Device.cxx File Reference

implementation of class [FL_Paged_Device](#).

```
#include <FL/Fl_Paged_Device.H>
#include <FL/Fl.H>
#include <FL/fl_draw.H>
```

32.185.1 Detailed Description

implementation of class [FL_Paged_Device](#).

32.186 fl_rect.cxx File Reference

Drawing and clipping routines for rectangles.

```
#include <config.h>
#include <FL/Fl.H>
#include <FL/Fl_Widget.H>
#include <FL/Fl_Printer.H>
#include <FL/fl_draw.H>
#include <FL/x.H>
```

Functions

- `FL_Region XRectangleRegion` (int x, int y, int w, int h)

Variables

- int `fl_line_width_`

32.186.1 Detailed Description

Drawing and clipping routines for rectangles.

32.187 fl_vertex.cxx File Reference

Portable drawing code for drawing arbitrary shapes with simple 2D transformations.

```
#include <config.h>
#include <FL/fl_draw.H>
#include <FL/x.H>
#include <FL/Fl.H>
#include <FL/math.h>
#include <stdlib.h>
```

32.187.1 Detailed Description

Portable drawing code for drawing arbitrary shapes with simple 2D transformations.

32.188 Fl_XColor.H

```

1 //
2 // "$Id$"
3 //
4 // X-specific color definitions for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18
19 #include <config.h>
20 #include <FL/Enumerations.H>
21
22 // one of these for each color in fltk's "colormap":
23 // if overlays are enabled, another one for the overlay
24 struct Fl_XColor {
25     unsigned char r,g,b; // actual color used by X
26     unsigned char mapped; // true when XAllocColor done
27     unsigned long pixel; // the X pixel to use
28 };
29 extern Fl_XColor fl_xmap[/*overlay*/][256];
30
31 // mask & shifts to produce xcolor for truecolor visuals:
32 extern unsigned char fl_redmask, fl_greenmask, fl_bluemask;
33 extern int fl_redshift, fl_greenshift, fl_blueshift, fl_extrashift;
34
35 //
36 // End of "$Id$".
37 //

```

32.189 flstring.h

```

1 /*
2 * "$Id$"
3 *
4 * Common string header file for the Fast Light Tool Kit (FLTK).
5 *
6 * Copyright 1998-2016 by Bill Spitzak and others.
7 *
8 * This library is free software. Distribution and use rights are outlined in
9 * the file "COPYING" which should have been included with this file. If this
10 * file is missing or damaged, see the license at:
11 *
12 *     http://www.fltk.org/COPYING.php
13 *
14 * Please report all bugs and problems on the following page:
15 *
16 *     http://www.fltk.org/str.php
17 */
18
19 #ifndef flstring_h
20 # define flstring_h
21
22 # include <FL/Fl_Export.H>
23 # include <config.h>
24 # include <stdio.h>
25 # include <stdarg.h>
26 # include <string.h>
27 # ifdef HAVE_STRINGS_H
28 # include <strings.h>
29 # endif /* HAVE_STRINGS_H */
30 # include <ctype.h>
31
32 /*
33 * Apparently Unixware defines "index" to strchr (!) rather than
34 * providing a proper entry point or not providing the (obsolete)
35 * BSD function. Make sure index is not defined...
36 */
37
38 # ifdef index
39 # undef index
40 # endif /* index */
41
42 # if defined(WIN32) && !defined(__CYGWIN__) && !defined(__MINGW32__)

```

```

43 #   define strcasecmp(s,t)      _stricmp((s), (t))
44 #   define strncasecmp(s,t,n)   _strnicmp((s), (t), (n))
45 /* Visual C++ 2005 incorrectly displays a warning about the use of POSIX APIs
46 * on Windows, which is supposed to be POSIX compliant...   Some of these
47 * functions are also defined in ISO C99...
48 */
49 #   ifndef __WATCOMC__
50 #       define strdup _strdup
51 #       define unlink _unlink
52 #   endif /* !__WATCOMC__ */
53 #   elif defined(__EMX__)
54 #       define strcasecmp(s,t)      stricmp((s), (t))
55 #       define strncasecmp(s,t,n)   strnicmp((s), (t), (n))
56 #   endif /* WIN32 */
57
58 #   ifdef __cplusplus
59 extern "C" {
60 #   endif /* __cplusplus */
61
62 FL_EXPORT extern int fl_snprintf(char *, size_t, const char *, ...);
63 #   ifndef HAVE_SNPRINTF
64 #       define snprintf fl_snprintf
65 #   endif /* !HAVE_SNPRINTF */
66
67 FL_EXPORT extern int fl_vsnprintf(char *, size_t, const char *, va_list ap);
68 #   ifndef HAVE_VSNPRINTF
69 #       define vsnprintf fl_vsnprintf
70 #   endif /* !HAVE_VSNPRINTF */
71
72 /*
73 * strlcpy() and strlcat() are some really useful BSD string functions
74 * that work the way strncpy() and strncat() *should* have worked.
75 */
76
77 FL_EXPORT extern size_t fl_strlcat(char *, const char *, size_t);
78 #   ifndef HAVE_STRLCAT
79 #       define strlcat fl_strlcat
80 #   endif /* !HAVE_STRLCAT */
81
82 FL_EXPORT extern size_t fl_strlcpy(char *, const char *, size_t);
83 #   ifndef HAVE_STRLCPY
84 #       define strlcpy fl_strlcpy
85 #   endif /* !HAVE_STRLCPY */
86
87 /*
88 * locale independent ascii compare, does not introduce locale
89 * pbs as w/ case cmp
90 */
91 FL_EXPORT extern int fl_ascii_strcasecmp(const char *s, const char *t);
92
93 #   ifdef __cplusplus
94 }
95 #   endif /* __cplusplus */
96
97 #endif /* !flstring_h */
98
99 /*
100 * End of "$Id$".
101 */

```

32.190 freeglut_teapot_data.h

```

1 /*
2 * freeglut_teapot_data.h
3 *
4 * The freeglut library teapot data include file.
5 *
6 * Permission is hereby granted, free of charge, to any person obtaining a
7 * copy of this software and associated documentation files (the "Software"),
8 * to deal in the Software without restriction, including without limitation
9 * the rights to use, copy, modify, merge, publish, distribute, sublicense,
10 * and/or sell copies of the Software, and to permit persons to whom the
11 * Software is furnished to do so, subject to the following conditions:
12 *
13 * The above copyright notice and this permission notice shall be included
14 * in all copies or substantial portions of the Software.
15 *
16 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS
17 * OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
18 * FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL
19 * PAWEL W. OLSZTA BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
20 * IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
21 * CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
22 */
23

```

```

24 #ifndef FREEGLUT_TEAPOT_DATA_H
25 #define FREEGLUT_TEAPOT_DATA_H
26
27 /*
28 * Original teapot code copyright follows:
29 */
30
31 /*
32 * (c) Copyright 1993, Silicon Graphics, Inc.
33 *
34 * ALL RIGHTS RESERVED
35 *
36 * Permission to use, copy, modify, and distribute this software
37 * for any purpose and without fee is hereby granted, provided
38 * that the above copyright notice appear in all copies and that
39 * both the copyright notice and this permission notice appear in
40 * supporting documentation, and that the name of Silicon
41 * Graphics, Inc. not be used in advertising or publicity
42 * pertaining to distribution of the software without specific,
43 * written prior permission.
44 *
45 * THE MATERIAL EMBODIED ON THIS SOFTWARE IS PROVIDED TO YOU
46 * "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EXPRESS, IMPLIED OR
47 * OTHERWISE, INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF
48 * MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO
49 * EVENT SHALL SILICON GRAPHICS, INC. BE LIABLE TO YOU OR ANYONE
50 * ELSE FOR ANY DIRECT, SPECIAL, INCIDENTAL, INDIRECT OR
51 * CONSEQUENTIAL DAMAGES OF ANY KIND, OR ANY DAMAGES WHATSOEVER,
52 * INCLUDING WITHOUT LIMITATION, LOSS OF PROFIT, LOSS OF USE,
53 * SAVINGS OR REVENUE, OR THE CLAIMS OF THIRD PARTIES, WHETHER OR
54 * NOT SILICON GRAPHICS, INC. HAS BEEN ADVISED OF THE POSSIBILITY
55 * OF SUCH LOSS, HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
56 * ARISING OUT OF OR IN CONNECTION WITH THE POSSESSION, USE OR
57 * PERFORMANCE OF THIS SOFTWARE.
58 *
59 * US Government Users Restricted Rights
60 *
61 * Use, duplication, or disclosure by the Government is subject to
62 * restrictions set forth in FAR 52.227.19(c)(2) or subparagraph
63 * (c)(1)(ii) of the Rights in Technical Data and Computer
64 * Software clause at DFARS 252.227-7013 and/or in similar or
65 * successor clauses in the FAR or the DOD or NASA FAR
66 * Supplement. Unpublished-- rights reserved under the copyright
67 * laws of the United States. Contractor/manufacturer is Silicon
68 * Graphics, Inc., 2011 N. Shoreline Blvd., Mountain View, CA
69 * 94039-7311.
70 *
71 * OpenGL(TM) is a trademark of Silicon Graphics, Inc.
72 */
73
74 /*
75 * Rim, body, lid, and bottom data must be reflected in x and y;
76 * handle and spout data across the y axis only.
77 */
78 static int patchdata[][16] =
79 {
80   { 102, 103, 104, 105, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 }, /* rim */
81   { 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27 }, /* body */
82   { 24, 25, 26, 27, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40 },
83   { 96, 96, 96, 96, 96, 97, 98, 99, 100, 101, 101, 101, 101, 0, 1, 2, 3 }, /* lid */
84   { 0, 1, 2, 3, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117 },
85   { 118, 118, 118, 118, 124, 122, 119, 121, 123, 126, 125, 120, 40, 39, 38, 37 }, /* bottom */
86   { 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56 }, /* handle */
87   { 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 28, 65, 66, 67 },
88   { 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83 }, /* spout */
89   { 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95 }
90 };
91
92 static double cpdata[][3] =
93 {
94   { 0.2, 0, 2.7 }, { 0.2, -0.112, 2.7 }, { 0.112, -0.2, 2.7 }, { 0,
95   -0.2, 2.7 }, { 1.3375, 0, 2.53125 }, { 1.3375, -0.749, 2.53125 },
96   { 0.749, -1.3375, 2.53125 }, { 0, -1.3375, 2.53125 }, { 1.4375,
97   0, 2.53125 }, { 1.4375, -0.805, 2.53125 }, { 0.805, -1.4375,
98   2.53125 }, { 0, -1.4375, 2.53125 }, { 1.5, 0, 2.4 }, { 1.5, -0.84,
99   2.4 }, { 0.84, -1.5, 2.4 }, { 0, -1.5, 2.4 }, { 1.75, 0, 1.875 },
100  { 1.75, -0.98, 1.875 }, { 0.98, -1.75, 1.875 }, { 0, -1.75,
101  1.875 }, { 2, 0, 1.35 }, { 2, -1.12, 1.35 }, { 1.12, -2, 1.35 },
102  { 0, -2, 1.35 }, { 2, 0, 0.9 }, { 2, -1.12, 0.9 }, { 1.12, -2,
103  0.9 }, { 0, -2, 0.9 }, { -2, 0, 0.9 }, { 2, 0, 0.45 }, { 2, -1.12,
104  0.45 }, { 1.12, -2, 0.45 }, { 0, -2, 0.45 }, { 1.5, 0, 0.225 },
105  { 1.5, -0.84, 0.225 }, { 0.84, -1.5, 0.225 }, { 0, -1.5, 0.225 },
106  { 1.5, 0, 0.15 }, { 1.5, -0.84, 0.15 }, { 0.84, -1.5, 0.15 }, { 0,
107  -1.5, 0.15 }, { -1.6, 0, 2.025 }, { -1.6, -0.3, 2.025 }, { -1.5,
108  -0.3, 2.25 }, { -1.5, 0, 2.25 }, { -2.3, 0, 2.025 }, { -2.3, -0.3,
109  2.025 }, { -2.5, -0.3, 2.25 }, { -2.5, 0, 2.25 }, { -2.7, 0,
110  2.025 }, { -2.7, -0.3, 2.025 }, { -3, -0.3, 2.25 }, { -3, 0,

```

```

111 2.25}, {-2.7, 0, 1.8}, {-2.7, -0.3, 1.8}, {-3, -0.3, 1.8},
112 {-3, 0, 1.8}, {-2.7, 0, 1.575}, {-2.7, -0.3, 1.575}, {-3,
113 -0.3, 1.35}, {-3, 0, 1.35}, {-2.5, 0, 1.125}, {-2.5, -0.3,
114 1.125}, {-2.65, -0.3, 0.9375}, {-2.65, 0, 0.9375}, {-2,
115 -0.3, 0.9}, {-1.9, -0.3, 0.6}, {-1.9, 0, 0.6}, {1.7, 0,
116 1.425}, {1.7, -0.66, 1.425}, {1.7, -0.66, 0.6}, {1.7, 0,
117 0.6}, {2.6, 0, 1.425}, {2.6, -0.66, 1.425}, {3.1, -0.66,
118 0.825}, {3.1, 0, 0.825}, {2.3, 0, 2.1}, {2.3, -0.25, 2.1},
119 {2.4, -0.25, 2.025}, {2.4, 0, 2.025}, {2.7, 0, 2.4}, {2.7,
120 -0.25, 2.4}, {3.3, -0.25, 2.4}, {3.3, 0, 2.4}, {2.8, 0,
121 2.475}, {2.8, -0.25, 2.475}, {3.525, -0.25, 2.49375},
122 {3.525, 0, 2.49375}, {2.9, 0, 2.475}, {2.9, -0.15, 2.475},
123 {3.45, -0.15, 2.5125}, {3.45, 0, 2.5125}, {2.8, 0, 2.4},
124 {2.8, -0.15, 2.4}, {3.2, -0.15, 2.4}, {3.2, 0, 2.4}, {0, 0,
125 3.15}, {0.8, 0, 3.15}, {0.8, -0.45, 3.15}, {0.45, -0.8,
126 3.15}, {0, -0.8, 3.15}, {0, 0, 2.85}, {1.4, 0, 2.4}, {1.4,
127 -0.784, 2.4}, {0.784, -1.4, 2.4}, {0, -1.4, 2.4}, {0.4, 0,
128 2.55}, {0.4, -0.224, 2.55}, {0.224, -0.4, 2.55}, {0, -0.4,
129 2.55}, {1.3, 0, 2.55}, {1.3, -0.728, 2.55}, {0.728, -1.3,
130 2.55}, {0, -1.3, 2.55}, {1.3, 0, 2.4}, {1.3, -0.728, 2.4},
131 {0.728, -1.3, 2.4}, {0, -1.3, 2.4}, {0, 0, 0}, {1.425,
132 -0.798, 0}, {1.5, 0, 0.075}, {1.425, 0, 0}, {0.798, -1.425,
133 0}, {0, -1.5, 0.075}, {0, -1.425, 0}, {1.5, -0.84, 0.075},
134 {0.84, -1.5, 0.075}
135 };
136
137 static double tex[2][2][2] =
138 {
139     { {0.0, 0.0}, {1.0, 0.0} },
140     { {0.0, 1.0}, {1.0, 1.0} }
141 };
142
143
144 #endif /* FREEGLUT_TEAPOT_DATA_H */
145

```

32.191 mediuarrow.h

```

1 #define mediuarrow_width 16
2 #define mediuarrow_height 16
3 static const unsigned char mediuarrow_bits[] = {
4     0x40, 0x00, 0x60, 0x00, 0x70, 0x00, 0x78, 0x00, 0xfc, 0x3f, 0x78, 0x00,
5     0x70, 0x00, 0x60, 0x02, 0x40, 0x06, 0x00, 0x0e, 0x00, 0x1e, 0xfc, 0x3f,
6     0x00, 0x1e, 0x00, 0x0e, 0x00, 0x06, 0x00, 0x02};

```

32.192 print_panel.h

```

1 //
2 // "$Id$"
3 //
4 // Print panel for the Fast Light Tool Kit (FLTK).
5 //
6 // Copyright 1998-2010 by Bill Spitzak and others.
7 //
8 // This library is free software. Distribution and use rights are outlined in
9 // the file "COPYING" which should have been included with this file. If this
10 // file is missing or damaged, see the license at:
11 //
12 //     http://www.fltk.org/COPYING.php
13 //
14 // Please report all bugs and problems on the following page:
15 //
16 //     http://www.fltk.org/str.php
17 //
18 //
19 //
20 // This is a temporary file. It is only for development and will
21 // probably be removed later.
22 //
23
24 #ifndef print_panel_h
25 #define print_panel_h
26 #include <FL/Fl.H>
27 #include <FL/Fl_Double_Window.H>
28 #include <FL/Fl_Group.H>
29 #include <FL/Fl_Choice.H>
30 #include <FL/Fl_Button.H>
31 #include <FL/Fl_Box.H>
32 #include <FL/Fl_Round_Button.H>
33 #include <FL/Fl_Input.H>
34 #include <FL/Fl_Spinner.H>
35 #include <FL/Fl_Check_Button.H>

```

```

36 #include <FL/Fl_Return_Button.H>
37 #include <FL/Fl_Progress.H>
38 enum printing_style {SystemV, BSD};
39 static Fl_Double_Window* make_print_panel();
40 static void print_cb(Fl_Return_Button *, void *);
41 static printing_style print_load();
42 static void print_update_status();
43 #endif
44
45 //
46 // End of "$Id$".
47 //

```

32.193 slowarrow.h

```

1 #define slowarrow_width 16
2 #define slowarrow_height 16
3 static const unsigned char slowarrow_bits[] = {
4     0x40, 0x00, 0x40, 0x00, 0x60, 0x00, 0x60, 0x00, 0xf0, 0x0f, 0x60, 0x00,
5     0x60, 0x00, 0x40, 0x02, 0x40, 0x02, 0x00, 0x06, 0x00, 0x06, 0xf0, 0x0f,
6     0x00, 0x06, 0x00, 0x06, 0x00, 0x02, 0x00, 0x02};

```

32.194 Xutf8.h

```

1 /* "$Id$"
2 *
3 * Author: Jean-Marc Lienher ( http://oksid.ch )
4 * Copyright 2000-2010 by O'ksi'D.
5 *
6 * This library is free software. Distribution and use rights are outlined in
7 * the file "COPYING" which should have been included with this file. If this
8 * file is missing or damaged, see the license at:
9 *
10 *     http://www.fltk.org/COPYING.php
11 *
12 * Please report all bugs and problems on the following page:
13 *
14 *     http://www.fltk.org/str.php
15 */
16
17 #if ! ( defined(_Xutf8_h) || defined(FL_DOXYGEN) )
18 #define _Xutf8_h
19
20 # ifdef __cplusplus
21 extern "C" {
22 # endif
23
24 #include <X11/X.h>
25 #include <X11/Xlib.h>
26 #include <X11/Xlocale.h>
27 #include <X11/Xutil.h>
28 #include <FL/Fl_Export.H>
29
30 typedef struct {
31     int nb_font;
32     char **font_name_list;
33     int *encodings;
34     XFontStruct **fonts;
35     Font fid;
36     int ascent;
37     int descent;
38     int *ranges;
39 } XUtf8FontStruct;
40
41 XUtf8FontStruct *
42 XCreateUtf8FontStruct (
43     Display *dpy,
44     const char *base_font_name_list);
45
46 void
47 XUtf8DrawString(
48     Display *display,
49     Drawable d,
50     XUtf8FontStruct *font_set,
51     GC gc,
52     int x,
53     int y,
54     const char *string,
55     int num_bytes);
56
57 void
58 XUtf8_measure_extents(

```



```

59     Display          *display,
60     Drawable        d,
61     XUtf8FontStruct *font_set,
62     GC              gc,
63     int             *xx,
64     int             *yy,
65     int             *ww,
66     int             *hh,
67     const char      *string,
68     int             num_bytes);
69
70 void
71 XUtf8DrawRtlString(
72     Display          *display,
73     Drawable        d,
74     XUtf8FontStruct *font_set,
75     GC              gc,
76     int             x,
77     int             y,
78     const char      *string,
79     int             num_bytes);
80
81 void
82 XUtf8DrawImageString(
83     Display          *display,
84     Drawable        d,
85     XUtf8FontStruct *font_set,
86     GC              gc,
87     int             x,
88     int             y,
89     const char      *string,
90     int             num_bytes);
91
92 int
93 XUtf8TextWidth(
94     XUtf8FontStruct *font_set,
95     const char      *string,
96     int             num_bytes);
97 int
98 XUtf8UcsWidth(
99     XUtf8FontStruct *font_set,
100    unsigned int    ucs);
101
102 FL_EXPORT int
103 fl_XGetUtf8FontAndGlyph(
104     XUtf8FontStruct *font_set,
105     unsigned int    ucs,
106     XFontStruct     **fnt,
107     unsigned short  *id);
108
109 void
110 XFreeUtf8FontStruct(
111     Display          *dpy,
112     XUtf8FontStruct *font_set);
113
114
115 int
116 XConvertUtf8ToUcs(
117     const unsigned char *buf,
118     int                 len,
119     unsigned int         *ucs);
120
121 int
122 XConvertUcsToUtf8(
123     unsigned int    ucs,
124     char            *buf);
125
126 int
127 XUtf8CharByteLen(
128     const unsigned char *buf,
129     int                 len);
130
131 int
132 XCountUtf8Char(
133     const unsigned char *buf,
134     int len);
135
136 int
137 XFastConvertUtf8ToUcs(
138     const unsigned char *buf,
139     int                 len,
140     unsigned int         *ucs);
141
142 long
143 XKeysymToUcs(
144     KeySym keysym);
145

```

```

146 #ifdef X_HAVE_UTF8_STRING
147 #define XUtf8LookupString Xutf8LookupString
148 #else
149 int
150 XUtf8LookupString(
151     XIC          ic,
152     XKeyPressedEvent* event,
153     char*        buffer_return,
154     int          bytes_buffer,
155     KeySym*      keysym,
156     Status*      status_return);
157 #endif
158
159 unsigned short
160 XUtf8IsNonSpacing(
161     unsigned int ucs);
162
163 unsigned short
164 XUtf8IsRightToLeft(
165     unsigned int ucs);
166
167
168 int
169 XUtf8ToLower(
170     int ucs);
171
172 int
173 XUtf8ToUpper(
174     int ucs);
175
176
177 # ifdef __cplusplus
178 }
179 # endif
180
181 #endif
182
183 /*
184 * End of "$Id$".
185 */

```

32.195 case.h

```

1 /* spacing */
2
3 static const unsigned short ucs_table_0041[] = {
4 /* U+0041 */ 0x0061,
5 /* U+0042 */ 0x0062,
6 /* U+0043 */ 0x0063,
7 /* U+0044 */ 0x0064,
8 /* U+0045 */ 0x0065,
9 /* U+0046 */ 0x0066,
10 /* U+0047 */ 0x0067,
11 /* U+0048 */ 0x0068,
12 /* U+0049 */ 0x0069,
13 /* U+004A */ 0x006A,
14 /* U+004B */ 0x006B,
15 /* U+004C */ 0x006C,
16 /* U+004D */ 0x006D,
17 /* U+004E */ 0x006E,
18 /* U+004F */ 0x006F,
19 /* U+0050 */ 0x0070,
20 /* U+0051 */ 0x0071,
21 /* U+0052 */ 0x0072,
22 /* U+0053 */ 0x0073,
23 /* U+0054 */ 0x0074,
24 /* U+0055 */ 0x0075,
25 /* U+0056 */ 0x0076,
26 /* U+0057 */ 0x0077,
27 /* U+0058 */ 0x0078,
28 /* U+0059 */ 0x0079,
29 /* U+005A */ 0x007A,
30 0x00,
31 0x00,
32 0x00,
33 0x00,
34 0x00,
35 0x00,
36 0x00,
37 0x00,
38 0x00,
39 0x00,
40 0x00,
41 0x00,
42 0x00,

```

```
43 0x00,  
44 0x00,  
45 0x00,  
46 0x00,  
47 0x00,  
48 0x00,  
49 0x00,  
50 0x00,  
51 0x00,  
52 0x00,  
53 0x00,  
54 0x00,  
55 0x00,  
56 0x00,  
57 0x00,  
58 0x00,  
59 0x00,  
60 0x00,  
61 0x00,  
62 0x00,  
63 0x00,  
64 0x00,  
65 0x00,  
66 0x00,  
67 0x00,  
68 0x00,  
69 0x00,  
70 0x00,  
71 0x00,  
72 0x00,  
73 0x00,  
74 0x00,  
75 0x00,  
76 0x00,  
77 0x00,  
78 0x00,  
79 0x00,  
80 0x00,  
81 0x00,  
82 0x00,  
83 0x00,  
84 0x00,  
85 0x00,  
86 0x00,  
87 0x00,  
88 0x00,  
89 0x00,  
90 0x00,  
91 0x00,  
92 0x00,  
93 0x00,  
94 0x00,  
95 0x00,  
96 0x00,  
97 0x00,  
98 0x00,  
99 0x00,  
100 0x00,  
101 0x00,  
102 0x00,  
103 0x00,  
104 0x00,  
105 0x00,  
106 0x00,  
107 0x00,  
108 0x00,  
109 0x00,  
110 0x00,  
111 0x00,  
112 0x00,  
113 0x00,  
114 0x00,  
115 0x00,  
116 0x00,  
117 0x00,  
118 0x00,  
119 0x00,  
120 0x00,  
121 0x00,  
122 0x00,  
123 0x00,  
124 0x00,  
125 0x00,  
126 0x00,  
127 0x00,  
128 0x00,  
129 0x00,
```

```
130 0x00,
131 /* U+00C0 */ 0x00E0,
132 /* U+00C1 */ 0x00E1,
133 /* U+00C2 */ 0x00E2,
134 /* U+00C3 */ 0x00E3,
135 /* U+00C4 */ 0x00E4,
136 /* U+00C5 */ 0x00E5,
137 /* U+00C6 */ 0x00E6,
138 /* U+00C7 */ 0x00E7,
139 /* U+00C8 */ 0x00E8,
140 /* U+00C9 */ 0x00E9,
141 /* U+00CA */ 0x00EA,
142 /* U+00CB */ 0x00EB,
143 /* U+00CC */ 0x00EC,
144 /* U+00CD */ 0x00ED,
145 /* U+00CE */ 0x00EE,
146 /* U+00CF */ 0x00EF,
147 /* U+00D0 */ 0x00F0,
148 /* U+00D1 */ 0x00F1,
149 /* U+00D2 */ 0x00F2,
150 /* U+00D3 */ 0x00F3,
151 /* U+00D4 */ 0x00F4,
152 /* U+00D5 */ 0x00F5,
153 /* U+00D6 */ 0x00F6,
154 0x00,
155 /* U+00D8 */ 0x00F8,
156 /* U+00D9 */ 0x00F9,
157 /* U+00DA */ 0x00FA,
158 /* U+00DB */ 0x00FB,
159 /* U+00DC */ 0x00FC,
160 /* U+00DD */ 0x00FD,
161 /* U+00DE */ 0x00FE,
162 0x00,
163 0x00,
164 0x00,
165 0x00,
166 0x00,
167 0x00,
168 0x00,
169 0x00,
170 0x00,
171 0x00,
172 0x00,
173 0x00,
174 0x00,
175 0x00,
176 0x00,
177 0x00,
178 0x00,
179 0x00,
180 0x00,
181 0x00,
182 0x00,
183 0x00,
184 0x00,
185 0x00,
186 0x00,
187 0x00,
188 0x00,
189 0x00,
190 0x00,
191 0x00,
192 0x00,
193 0x00,
194 0x00,
195 /* U+0100 */ 0x0101,
196 0x00,
197 /* U+0102 */ 0x0103,
198 0x00,
199 /* U+0104 */ 0x0105,
200 0x00,
201 /* U+0106 */ 0x0107,
202 0x00,
203 /* U+0108 */ 0x0109,
204 0x00,
205 /* U+010A */ 0x010B,
206 0x00,
207 /* U+010C */ 0x010D,
208 0x00,
209 /* U+010E */ 0x010F,
210 0x00,
211 /* U+0110 */ 0x0111,
212 0x00,
213 /* U+0112 */ 0x0113,
214 0x00,
215 /* U+0114 */ 0x0115,
216 0x00,
```

```
217 /* U+0116 */ 0x0117,
218 0x00,
219 /* U+0118 */ 0x0119,
220 0x00,
221 /* U+011A */ 0x011B,
222 0x00,
223 /* U+011C */ 0x011D,
224 0x00,
225 /* U+011E */ 0x011F,
226 0x00,
227 /* U+0120 */ 0x0121,
228 0x00,
229 /* U+0122 */ 0x0123,
230 0x00,
231 /* U+0124 */ 0x0125,
232 0x00,
233 /* U+0126 */ 0x0127,
234 0x00,
235 /* U+0128 */ 0x0129,
236 0x00,
237 /* U+012A */ 0x012B,
238 0x00,
239 /* U+012C */ 0x012D,
240 0x00,
241 /* U+012E */ 0x012F,
242 0x00,
243 /* U+0130 */ 0x0,
244 0x00,
245 /* U+0132 */ 0x0133,
246 0x00,
247 /* U+0134 */ 0x0135,
248 0x00,
249 /* U+0136 */ 0x0137,
250 0x00,
251 0x00,
252 /* U+0139 */ 0x013A,
253 0x00,
254 /* U+013B */ 0x013C,
255 0x00,
256 /* U+013D */ 0x013E,
257 0x00,
258 /* U+013F */ 0x0140,
259 0x00,
260 /* U+0141 */ 0x0142,
261 0x00,
262 /* U+0143 */ 0x0144,
263 0x00,
264 /* U+0145 */ 0x0146,
265 0x00,
266 /* U+0147 */ 0x0148,
267 0x00,
268 0x00,
269 /* U+014A */ 0x014B,
270 0x00,
271 /* U+014C */ 0x014D,
272 0x00,
273 /* U+014E */ 0x014F,
274 0x00,
275 /* U+0150 */ 0x0151,
276 0x00,
277 /* U+0152 */ 0x0153,
278 0x00,
279 /* U+0154 */ 0x0155,
280 0x00,
281 /* U+0156 */ 0x0157,
282 0x00,
283 /* U+0158 */ 0x0159,
284 0x00,
285 /* U+015A */ 0x015B,
286 0x00,
287 /* U+015C */ 0x015D,
288 0x00,
289 /* U+015E */ 0x015F,
290 0x00,
291 /* U+0160 */ 0x0161,
292 0x00,
293 /* U+0162 */ 0x0163,
294 0x00,
295 /* U+0164 */ 0x0165,
296 0x00,
297 /* U+0166 */ 0x0167,
298 0x00,
299 /* U+0168 */ 0x0169,
300 0x00,
301 /* U+016A */ 0x016B,
302 0x00,
303 /* U+016C */ 0x016D,
```

```
304 0x00,
305 /* U+016E */ 0x016F,
306 0x00,
307 /* U+0170 */ 0x0171,
308 0x00,
309 /* U+0172 */ 0x0173,
310 0x00,
311 /* U+0174 */ 0x0175,
312 0x00,
313 /* U+0176 */ 0x0177,
314 0x00,
315 /* U+0178 */ 0x00FF,
316 /* U+0179 */ 0x017A,
317 0x00,
318 /* U+017B */ 0x017C,
319 0x00,
320 /* U+017D */ 0x017E,
321 0x00,
322 0x00,
323 0x00,
324 /* U+0181 */ 0x0253,
325 /* U+0182 */ 0x0183,
326 0x00,
327 /* U+0184 */ 0x0185,
328 0x00,
329 /* U+0186 */ 0x0254,
330 /* U+0187 */ 0x0188,
331 0x00,
332 /* U+0189 */ 0x0,
333 /* U+018A */ 0x0257,
334 /* U+018B */ 0x018C,
335 0x00,
336 0x00,
337 /* U+018E */ 0x0258,
338 /* U+018F */ 0x0259,
339 /* U+0190 */ 0x025B,
340 /* U+0191 */ 0x0192,
341 0x00,
342 /* U+0193 */ 0x0260,
343 /* U+0194 */ 0x0263,
344 0x00,
345 /* U+0196 */ 0x0269,
346 /* U+0197 */ 0x0268,
347 /* U+0198 */ 0x0199,
348 0x00,
349 0x00,
350 0x00,
351 /* U+019C */ 0x026F,
352 /* U+019D */ 0x0272,
353 0x00,
354 /* U+019F */ 0x0,
355 /* U+01A0 */ 0x01A1,
356 0x00,
357 /* U+01A2 */ 0x01A3,
358 0x00,
359 /* U+01A4 */ 0x01A5,
360 0x00,
361 0x00,
362 /* U+01A7 */ 0x01A8,
363 0x00,
364 /* U+01A9 */ 0x0283,
365 0x00,
366 0x00,
367 /* U+01AC */ 0x01AD,
368 0x00,
369 /* U+01AE */ 0x0288,
370 /* U+01AF */ 0x01B0,
371 0x00,
372 /* U+01B1 */ 0x028A,
373 /* U+01B2 */ 0x028B,
374 /* U+01B3 */ 0x01B4,
375 0x00,
376 /* U+01B5 */ 0x01B6,
377 0x00,
378 /* U+01B7 */ 0x0292,
379 /* U+01B8 */ 0x01B9,
380 0x00,
381 0x00,
382 0x00,
383 /* U+01BC */ 0x01BD,
384 0x00,
385 0x00,
386 0x00,
387 0x00,
388 0x00,
389 0x00,
390 0x00,
```

```
391 /* U+01C4 */ 0x01C6,
392 /* U+01C5 */ 0x0,
393 0x00,
394 /* U+01C7 */ 0x01C9,
395 /* U+01C8 */ 0x0,
396 0x00,
397 /* U+01CA */ 0x01CC,
398 /* U+01CB */ 0x0,
399 0x00,
400 /* U+01CD */ 0x01CE,
401 0x00,
402 /* U+01CF */ 0x01D0,
403 0x00,
404 /* U+01D1 */ 0x01D2,
405 0x00,
406 /* U+01D3 */ 0x01D4,
407 0x00,
408 /* U+01D5 */ 0x01D6,
409 0x00,
410 /* U+01D7 */ 0x01D8,
411 0x00,
412 /* U+01D9 */ 0x01DA,
413 0x00,
414 /* U+01DB */ 0x01DC,
415 0x00,
416 0x00,
417 /* U+01DE */ 0x01DF,
418 0x00,
419 /* U+01E0 */ 0x01E1,
420 0x00,
421 /* U+01E2 */ 0x01E3,
422 0x00,
423 /* U+01E4 */ 0x01E5,
424 0x00,
425 /* U+01E6 */ 0x01E7,
426 0x00,
427 /* U+01E8 */ 0x01E9,
428 0x00,
429 /* U+01EA */ 0x01EB,
430 0x00,
431 /* U+01EC */ 0x01ED,
432 0x00,
433 /* U+01EE */ 0x01EF,
434 0x00,
435 0x00,
436 /* U+01F1 */ 0x01F3,
437 /* U+01F2 */ 0x0,
438 0x00,
439 /* U+01F4 */ 0x01F5,
440 0x00,
441 0x00,
442 0x00,
443 0x00,
444 0x00,
445 /* U+01FA */ 0x01FB,
446 0x00,
447 /* U+01FC */ 0x01FD,
448 0x00,
449 /* U+01FE */ 0x01FF,
450 0x00,
451 /* U+0200 */ 0x0201,
452 0x00,
453 /* U+0202 */ 0x0203,
454 0x00,
455 /* U+0204 */ 0x0205,
456 0x00,
457 /* U+0206 */ 0x0207,
458 0x00,
459 /* U+0208 */ 0x0209,
460 0x00,
461 /* U+020A */ 0x020B,
462 0x00,
463 /* U+020C */ 0x020D,
464 0x00,
465 /* U+020E */ 0x020F,
466 0x00,
467 /* U+0210 */ 0x0211,
468 0x00,
469 /* U+0212 */ 0x0213,
470 0x00,
471 /* U+0214 */ 0x0215,
472 0x00,
473 /* U+0216 */ 0x0217,
474 0x00,
475 0x00,
476 0x00,
477 0x00,
```

```
478 0x00,  
479 0x00,  
480 0x00,  
481 0x00,  
482 0x00,  
483 0x00,  
484 0x00,  
485 0x00,  
486 0x00,  
487 0x00,  
488 0x00,  
489 0x00,  
490 0x00,  
491 0x00,  
492 0x00,  
493 0x00,  
494 0x00,  
495 0x00,  
496 0x00,  
497 0x00,  
498 0x00,  
499 0x00,  
500 0x00,  
501 0x00,  
502 0x00,  
503 0x00,  
504 0x00,  
505 0x00,  
506 0x00,  
507 0x00,  
508 0x00,  
509 0x00,  
510 0x00,  
511 0x00,  
512 0x00,  
513 0x00,  
514 0x00,  
515 0x00,  
516 0x00,  
517 0x00,  
518 0x00,  
519 0x00,  
520 0x00,  
521 0x00,  
522 0x00,  
523 0x00,  
524 0x00,  
525 0x00,  
526 0x00,  
527 0x00,  
528 0x00,  
529 0x00,  
530 0x00,  
531 0x00,  
532 0x00,  
533 0x00,  
534 0x00,  
535 0x00,  
536 0x00,  
537 0x00,  
538 0x00,  
539 0x00,  
540 0x00,  
541 0x00,  
542 0x00,  
543 0x00,  
544 0x00,  
545 0x00,  
546 0x00,  
547 0x00,  
548 0x00,  
549 /* U+0262 */ 0x0,  
550 0x00,  
551 0x00,  
552 0x00,  
553 0x00,  
554 0x00,  
555 0x00,  
556 0x00,  
557 /* U+026A */ 0x0,  
558 0x00,  
559 0x00,  
560 0x00,  
561 0x00,  
562 0x00,  
563 0x00,  
564 0x00,
```



```
565 0x00,
566 0x00,
567 /* U+0274 */ 0x0,
568 0x00,
569 /* U+0276 */ 0x0,
570 0x00,
571 0x00,
572 0x00,
573 0x00,
574 0x00,
575 0x00,
576 0x00,
577 0x00,
578 0x00,
579 /* U+0280 */ 0x0,
580 /* U+0281 */ 0x0,
581 0x00,
582 0x00,
583 0x00,
584 0x00,
585 0x00,
586 0x00,
587 0x00,
588 0x00,
589 0x00,
590 0x00,
591 0x00,
592 0x00,
593 0x00,
594 /* U+028F */ 0x0,
595 0x00,
596 0x00,
597 0x00,
598 0x00,
599 0x00,
600 0x00,
601 0x00,
602 0x00,
603 0x00,
604 /* U+0299 */ 0x0,
605 0x00,
606 /* U+029B */ 0x0,
607 /* U+029C */ 0x0,
608 0x00,
609 0x00,
610 /* U+029F */ 0x0,
611 0x00,
612 0x00,
613 0x00,
614 0x00,
615 0x00,
616 0x00,
617 0x00,
618 0x00,
619 0x00,
620 0x00,
621 0x00,
622 0x00,
623 0x00,
624 0x00,
625 0x00,
626 0x00,
627 0x00,
628 0x00,
629 0x00,
630 0x00,
631 0x00,
632 0x00,
633 /* U+02B6 */ 0x0,
634 };
635
636 static const unsigned short ucs_table_0386[] = {
637 /* U+0386 */ 0x03AC,
638 0x00,
639 /* U+0388 */ 0x03AD,
640 /* U+0389 */ 0x03AE,
641 /* U+038A */ 0x03AF,
642 0x00,
643 /* U+038C */ 0x03CC,
644 0x00,
645 /* U+038E */ 0x03CD,
646 /* U+038F */ 0x03CE,
647 0x00,
648 /* U+0391 */ 0x03B1,
649 /* U+0392 */ 0x03B2,
650 /* U+0393 */ 0x03B3,
651 /* U+0394 */ 0x03B4,
```

```
652 /* U+0395 */ 0x03B5,
653 /* U+0396 */ 0x03B6,
654 /* U+0397 */ 0x03B7,
655 /* U+0398 */ 0x03B8,
656 /* U+0399 */ 0x03B9,
657 /* U+039A */ 0x03BA,
658 /* U+039B */ 0x03BB,
659 /* U+039C */ 0x03BC,
660 /* U+039D */ 0x03BD,
661 /* U+039E */ 0x03BE,
662 /* U+039F */ 0x03BF,
663 /* U+03A0 */ 0x03C0,
664 /* U+03A1 */ 0x03C1,
665 0x00,
666 /* U+03A3 */ 0x03C3,
667 /* U+03A4 */ 0x03C4,
668 /* U+03A5 */ 0x03C5,
669 /* U+03A6 */ 0x03C6,
670 /* U+03A7 */ 0x03C7,
671 /* U+03A8 */ 0x03C8,
672 /* U+03A9 */ 0x03C9,
673 /* U+03AA */ 0x03CA,
674 /* U+03AB */ 0x03CB,
675 0x00,
676 0x00,
677 0x00,
678 0x00,
679 0x00,
680 0x00,
681 0x00,
682 0x00,
683 0x00,
684 0x00,
685 0x00,
686 0x00,
687 0x00,
688 0x00,
689 0x00,
690 0x00,
691 0x00,
692 0x00,
693 0x00,
694 0x00,
695 0x00,
696 0x00,
697 0x00,
698 0x00,
699 0x00,
700 0x00,
701 0x00,
702 0x00,
703 0x00,
704 0x00,
705 0x00,
706 0x00,
707 0x00,
708 0x00,
709 0x00,
710 0x00,
711 0x00,
712 0x00,
713 /* U+03D2 */ 0x03D2,
714 /* U+03D3 */ 0x03D3,
715 /* U+03D4 */ 0x03D4,
716 0x00,
717 0x00,
718 0x00,
719 0x00,
720 0x00,
721 /* U+03DA */ 0x03DA,
722 0x00,
723 /* U+03DC */ 0x03DC,
724 0x00,
725 /* U+03DE */ 0x03DE,
726 0x00,
727 /* U+03E0 */ 0x03E0,
728 0x00,
729 /* U+03E2 */ 0x03E3,
730 0x00,
731 /* U+03E4 */ 0x03E5,
732 0x00,
733 /* U+03E6 */ 0x03E7,
734 0x00,
735 /* U+03E8 */ 0x03E9,
736 0x00,
737 /* U+03EA */ 0x03EB,
738 0x00,
```

```
739 /* U+03EC */ 0x03ED,
740 0x00,
741 /* U+03EE */ 0x03EF,
742 0x00,
743 0x00,
744 0x00,
745 0x00,
746 0x00,
747 0x00,
748 0x00,
749 0x00,
750 0x00,
751 0x00,
752 0x00,
753 0x00,
754 0x00,
755 0x00,
756 0x00,
757 0x00,
758 0x00,
759 0x00,
760 /* U+0401 */ 0x0451,
761 /* U+0402 */ 0x0452,
762 /* U+0403 */ 0x0453,
763 /* U+0404 */ 0x0454,
764 /* U+0405 */ 0x0455,
765 /* U+0406 */ 0x0456,
766 /* U+0407 */ 0x0457,
767 /* U+0408 */ 0x0458,
768 /* U+0409 */ 0x0459,
769 /* U+040A */ 0x045A,
770 /* U+040B */ 0x045B,
771 /* U+040C */ 0x045C,
772 0x00,
773 /* U+040E */ 0x045E,
774 /* U+040F */ 0x045F,
775 /* U+0410 */ 0x0430,
776 /* U+0411 */ 0x0431,
777 /* U+0412 */ 0x0432,
778 /* U+0413 */ 0x0433,
779 /* U+0414 */ 0x0434,
780 /* U+0415 */ 0x0435,
781 /* U+0416 */ 0x0436,
782 /* U+0417 */ 0x0437,
783 /* U+0418 */ 0x0438,
784 /* U+0419 */ 0x0439,
785 /* U+041A */ 0x043A,
786 /* U+041B */ 0x043B,
787 /* U+041C */ 0x043C,
788 /* U+041D */ 0x043D,
789 /* U+041E */ 0x043E,
790 /* U+041F */ 0x043F,
791 /* U+0420 */ 0x0440,
792 /* U+0421 */ 0x0441,
793 /* U+0422 */ 0x0442,
794 /* U+0423 */ 0x0443,
795 /* U+0424 */ 0x0444,
796 /* U+0425 */ 0x0445,
797 /* U+0426 */ 0x0446,
798 /* U+0427 */ 0x0447,
799 /* U+0428 */ 0x0448,
800 /* U+0429 */ 0x0449,
801 /* U+042A */ 0x044A,
802 /* U+042B */ 0x044B,
803 /* U+042C */ 0x044C,
804 /* U+042D */ 0x044D,
805 /* U+042E */ 0x044E,
806 /* U+042F */ 0x044F,
807 0x00,
808 0x00,
809 0x00,
810 0x00,
811 0x00,
812 0x00,
813 0x00,
814 0x00,
815 0x00,
816 0x00,
817 0x00,
818 0x00,
819 0x00,
820 0x00,
821 0x00,
822 0x00,
823 0x00,
824 0x00,
825 0x00,
```

```
826 0x00,
827 0x00,
828 0x00,
829 0x00,
830 0x00,
831 0x00,
832 0x00,
833 0x00,
834 0x00,
835 0x00,
836 0x00,
837 0x00,
838 0x00,
839 0x00,
840 0x00,
841 0x00,
842 0x00,
843 0x00,
844 0x00,
845 0x00,
846 0x00,
847 0x00,
848 0x00,
849 0x00,
850 0x00,
851 0x00,
852 0x00,
853 0x00,
854 0x00,
855 /* U+0460 */ 0x0461,
856 0x00,
857 /* U+0462 */ 0x0463,
858 0x00,
859 /* U+0464 */ 0x0465,
860 0x00,
861 /* U+0466 */ 0x0467,
862 0x00,
863 /* U+0468 */ 0x0469,
864 0x00,
865 /* U+046A */ 0x046B,
866 0x00,
867 /* U+046C */ 0x046D,
868 0x00,
869 /* U+046E */ 0x046F,
870 0x00,
871 /* U+0470 */ 0x0471,
872 0x00,
873 /* U+0472 */ 0x0473,
874 0x00,
875 /* U+0474 */ 0x0475,
876 0x00,
877 /* U+0476 */ 0x0477,
878 0x00,
879 /* U+0478 */ 0x0479,
880 0x00,
881 /* U+047A */ 0x047B,
882 0x00,
883 /* U+047C */ 0x047D,
884 0x00,
885 /* U+047E */ 0x047F,
886 0x00,
887 /* U+0480 */ 0x0481,
888 0x00,
889 0x00,
890 0x00,
891 0x00,
892 0x00,
893 0x00,
894 0x00,
895 0x00,
896 0x00,
897 0x00,
898 0x00,
899 0x00,
900 0x00,
901 0x00,
902 0x00,
903 /* U+0490 */ 0x0491,
904 0x00,
905 /* U+0492 */ 0x0493,
906 0x00,
907 /* U+0494 */ 0x0495,
908 0x00,
909 /* U+0496 */ 0x0497,
910 0x00,
911 /* U+0498 */ 0x0499,
912 0x00,
```

```
913 /* U+049A */ 0x049B,
914 0x00,
915 /* U+049C */ 0x049D,
916 0x00,
917 /* U+049E */ 0x049F,
918 0x00,
919 /* U+04A0 */ 0x04A1,
920 0x00,
921 /* U+04A2 */ 0x04A3,
922 0x00,
923 /* U+04A4 */ 0x04A5,
924 0x00,
925 /* U+04A6 */ 0x04A7,
926 0x00,
927 /* U+04A8 */ 0x04A9,
928 0x00,
929 /* U+04AA */ 0x04AB,
930 0x00,
931 /* U+04AC */ 0x04AD,
932 0x00,
933 /* U+04AE */ 0x04AF,
934 0x00,
935 /* U+04B0 */ 0x04B1,
936 0x00,
937 /* U+04B2 */ 0x04B3,
938 0x00,
939 /* U+04B4 */ 0x04B5,
940 0x00,
941 /* U+04B6 */ 0x04B7,
942 0x00,
943 /* U+04B8 */ 0x04B9,
944 0x00,
945 /* U+04BA */ 0x04BB,
946 0x00,
947 /* U+04BC */ 0x04BD,
948 0x00,
949 /* U+04BE */ 0x04BF,
950 0x00,
951 0x00,
952 /* U+04C1 */ 0x04C2,
953 0x00,
954 /* U+04C3 */ 0x04C4,
955 0x00,
956 0x00,
957 0x00,
958 /* U+04C7 */ 0x04C8,
959 0x00,
960 0x00,
961 0x00,
962 /* U+04CB */ 0x04CC,
963 0x00,
964 0x00,
965 0x00,
966 0x00,
967 /* U+04D0 */ 0x04D1,
968 0x00,
969 /* U+04D2 */ 0x04D3,
970 0x00,
971 /* U+04D4 */ 0x04D5,
972 0x00,
973 /* U+04D6 */ 0x04D7,
974 0x00,
975 /* U+04D8 */ 0x04D9,
976 0x00,
977 /* U+04DA */ 0x04DB,
978 0x00,
979 /* U+04DC */ 0x04DD,
980 0x00,
981 /* U+04DE */ 0x04DF,
982 0x00,
983 /* U+04E0 */ 0x04E1,
984 0x00,
985 /* U+04E2 */ 0x04E3,
986 0x00,
987 /* U+04E4 */ 0x04E5,
988 0x00,
989 /* U+04E6 */ 0x04E7,
990 0x00,
991 /* U+04E8 */ 0x04E9,
992 0x00,
993 /* U+04EA */ 0x04EB,
994 0x00,
995 0x00,
996 0x00,
997 /* U+04EE */ 0x04EF,
998 0x00,
999 /* U+04F0 */ 0x04F1,
```

```
1000 0x00,
1001 /* U+04F2 */ 0x04F3,
1002 0x00,
1003 /* U+04F4 */ 0x04F5,
1004 0x00,
1005 0x00,
1006 0x00,
1007 /* U+04F8 */ 0x04F9,
1008 0x00,
1009 0x00,
1010 0x00,
1011 0x00,
1012 0x00,
1013 0x00,
1014 0x00,
1015 0x00,
1016 0x00,
1017 0x00,
1018 0x00,
1019 0x00,
1020 0x00,
1021 0x00,
1022 0x00,
1023 0x00,
1024 0x00,
1025 0x00,
1026 0x00,
1027 0x00,
1028 0x00,
1029 0x00,
1030 0x00,
1031 0x00,
1032 0x00,
1033 0x00,
1034 0x00,
1035 0x00,
1036 0x00,
1037 0x00,
1038 0x00,
1039 0x00,
1040 0x00,
1041 0x00,
1042 0x00,
1043 0x00,
1044 0x00,
1045 0x00,
1046 0x00,
1047 0x00,
1048 0x00,
1049 0x00,
1050 0x00,
1051 0x00,
1052 0x00,
1053 0x00,
1054 0x00,
1055 0x00,
1056 0x00,
1057 0x00,
1058 0x00,
1059 0x00,
1060 0x00,
1061 0x00,
1062 0x00,
1063 0x00,
1064 /* U+0531 */ 0x0561,
1065 /* U+0532 */ 0x0562,
1066 /* U+0533 */ 0x0563,
1067 /* U+0534 */ 0x0564,
1068 /* U+0535 */ 0x0565,
1069 /* U+0536 */ 0x0566,
1070 /* U+0537 */ 0x0567,
1071 /* U+0538 */ 0x0568,
1072 /* U+0539 */ 0x0569,
1073 /* U+053A */ 0x056A,
1074 /* U+053B */ 0x056B,
1075 /* U+053C */ 0x056C,
1076 /* U+053D */ 0x056D,
1077 /* U+053E */ 0x056E,
1078 /* U+053F */ 0x056F,
1079 /* U+0540 */ 0x0570,
1080 /* U+0541 */ 0x0571,
1081 /* U+0542 */ 0x0572,
1082 /* U+0543 */ 0x0573,
1083 /* U+0544 */ 0x0574,
1084 /* U+0545 */ 0x0575,
1085 /* U+0546 */ 0x0576,
1086 /* U+0547 */ 0x0577,
```

```
1087 /* U+0548 */ 0x0578,
1088 /* U+0549 */ 0x0579,
1089 /* U+054A */ 0x057A,
1090 /* U+054B */ 0x057B,
1091 /* U+054C */ 0x057C,
1092 /* U+054D */ 0x057D,
1093 /* U+054E */ 0x057E,
1094 /* U+054F */ 0x057F,
1095 /* U+0550 */ 0x0580,
1096 /* U+0551 */ 0x0581,
1097 /* U+0552 */ 0x0582,
1098 /* U+0553 */ 0x0583,
1099 /* U+0554 */ 0x0584,
1100 /* U+0555 */ 0x0585,
1101 /* U+0556 */ 0x0586,
1102 };
1103
1104 static const unsigned short ucs_table_10A0[] = {
1105 /* U+10A0 */ 0x10D0,
1106 /* U+10A1 */ 0x10D1,
1107 /* U+10A2 */ 0x10D2,
1108 /* U+10A3 */ 0x10D3,
1109 /* U+10A4 */ 0x10D4,
1110 /* U+10A5 */ 0x10D5,
1111 /* U+10A6 */ 0x10D6,
1112 /* U+10A7 */ 0x10D7,
1113 /* U+10A8 */ 0x10D8,
1114 /* U+10A9 */ 0x10D9,
1115 /* U+10AA */ 0x10DA,
1116 /* U+10AB */ 0x10DB,
1117 /* U+10AC */ 0x10DC,
1118 /* U+10AD */ 0x10DD,
1119 /* U+10AE */ 0x10DE,
1120 /* U+10AF */ 0x10DF,
1121 /* U+10B0 */ 0x10E0,
1122 /* U+10B1 */ 0x10E1,
1123 /* U+10B2 */ 0x10E2,
1124 /* U+10B3 */ 0x10E3,
1125 /* U+10B4 */ 0x10E4,
1126 /* U+10B5 */ 0x10E5,
1127 /* U+10B6 */ 0x10E6,
1128 /* U+10B7 */ 0x10E7,
1129 /* U+10B8 */ 0x10E8,
1130 /* U+10B9 */ 0x10E9,
1131 /* U+10BA */ 0x10EA,
1132 /* U+10BB */ 0x10EB,
1133 /* U+10BC */ 0x10EC,
1134 /* U+10BD */ 0x10ED,
1135 /* U+10BE */ 0x10EE,
1136 /* U+10BF */ 0x10EF,
1137 /* U+10C0 */ 0x10F0,
1138 /* U+10C1 */ 0x10F1,
1139 /* U+10C2 */ 0x10F2,
1140 /* U+10C3 */ 0x10F3,
1141 /* U+10C4 */ 0x10F4,
1142 /* U+10C5 */ 0x10F5,
1143 };
1144
1145 static const unsigned short ucs_table_1E00[] = {
1146 /* U+1E00 */ 0x1E01,
1147 0x00,
1148 /* U+1E02 */ 0x1E03,
1149 0x00,
1150 /* U+1E04 */ 0x1E05,
1151 0x00,
1152 /* U+1E06 */ 0x1E07,
1153 0x00,
1154 /* U+1E08 */ 0x1E09,
1155 0x00,
1156 /* U+1E0A */ 0x1E0B,
1157 0x00,
1158 /* U+1E0C */ 0x1E0D,
1159 0x00,
1160 /* U+1E0E */ 0x1E0F,
1161 0x00,
1162 /* U+1E10 */ 0x1E11,
1163 0x00,
1164 /* U+1E12 */ 0x1E13,
1165 0x00,
1166 /* U+1E14 */ 0x1E15,
1167 0x00,
1168 /* U+1E16 */ 0x1E17,
1169 0x00,
1170 /* U+1E18 */ 0x1E19,
1171 0x00,
1172 /* U+1E1A */ 0x1E1B,
1173 0x00,
```

```
1174 /* U+1E1C */ 0x1E1D,
1175 0x00,
1176 /* U+1E1E */ 0x1E1F,
1177 0x00,
1178 /* U+1E20 */ 0x1E21,
1179 0x00,
1180 /* U+1E22 */ 0x1E23,
1181 0x00,
1182 /* U+1E24 */ 0x1E25,
1183 0x00,
1184 /* U+1E26 */ 0x1E27,
1185 0x00,
1186 /* U+1E28 */ 0x1E29,
1187 0x00,
1188 /* U+1E2A */ 0x1E2B,
1189 0x00,
1190 /* U+1E2C */ 0x1E2D,
1191 0x00,
1192 /* U+1E2E */ 0x1E2F,
1193 0x00,
1194 /* U+1E30 */ 0x1E31,
1195 0x00,
1196 /* U+1E32 */ 0x1E33,
1197 0x00,
1198 /* U+1E34 */ 0x1E35,
1199 0x00,
1200 /* U+1E36 */ 0x1E37,
1201 0x00,
1202 /* U+1E38 */ 0x1E39,
1203 0x00,
1204 /* U+1E3A */ 0x1E3B,
1205 0x00,
1206 /* U+1E3C */ 0x1E3D,
1207 0x00,
1208 /* U+1E3E */ 0x1E3F,
1209 0x00,
1210 /* U+1E40 */ 0x1E41,
1211 0x00,
1212 /* U+1E42 */ 0x1E43,
1213 0x00,
1214 /* U+1E44 */ 0x1E45,
1215 0x00,
1216 /* U+1E46 */ 0x1E47,
1217 0x00,
1218 /* U+1E48 */ 0x1E49,
1219 0x00,
1220 /* U+1E4A */ 0x1E4B,
1221 0x00,
1222 /* U+1E4C */ 0x1E4D,
1223 0x00,
1224 /* U+1E4E */ 0x1E4F,
1225 0x00,
1226 /* U+1E50 */ 0x1E51,
1227 0x00,
1228 /* U+1E52 */ 0x1E53,
1229 0x00,
1230 /* U+1E54 */ 0x1E55,
1231 0x00,
1232 /* U+1E56 */ 0x1E57,
1233 0x00,
1234 /* U+1E58 */ 0x1E59,
1235 0x00,
1236 /* U+1E5A */ 0x1E5B,
1237 0x00,
1238 /* U+1E5C */ 0x1E5D,
1239 0x00,
1240 /* U+1E5E */ 0x1E5F,
1241 0x00,
1242 /* U+1E60 */ 0x1E61,
1243 0x00,
1244 /* U+1E62 */ 0x1E63,
1245 0x00,
1246 /* U+1E64 */ 0x1E65,
1247 0x00,
1248 /* U+1E66 */ 0x1E67,
1249 0x00,
1250 /* U+1E68 */ 0x1E69,
1251 0x00,
1252 /* U+1E6A */ 0x1E6B,
1253 0x00,
1254 /* U+1E6C */ 0x1E6D,
1255 0x00,
1256 /* U+1E6E */ 0x1E6F,
1257 0x00,
1258 /* U+1E70 */ 0x1E71,
1259 0x00,
1260 /* U+1E72 */ 0x1E73,
```



```
1261 0x00,
1262 /* U+1E74 */ 0x1E75,
1263 0x00,
1264 /* U+1E76 */ 0x1E77,
1265 0x00,
1266 /* U+1E78 */ 0x1E79,
1267 0x00,
1268 /* U+1E7A */ 0x1E7B,
1269 0x00,
1270 /* U+1E7C */ 0x1E7D,
1271 0x00,
1272 /* U+1E7E */ 0x1E7F,
1273 0x00,
1274 /* U+1E80 */ 0x1E81,
1275 0x00,
1276 /* U+1E82 */ 0x1E83,
1277 0x00,
1278 /* U+1E84 */ 0x1E85,
1279 0x00,
1280 /* U+1E86 */ 0x1E87,
1281 0x00,
1282 /* U+1E88 */ 0x1E89,
1283 0x00,
1284 /* U+1E8A */ 0x1E8B,
1285 0x00,
1286 /* U+1E8C */ 0x1E8D,
1287 0x00,
1288 /* U+1E8E */ 0x1E8F,
1289 0x00,
1290 /* U+1E90 */ 0x1E91,
1291 0x00,
1292 /* U+1E92 */ 0x1E93,
1293 0x00,
1294 /* U+1E94 */ 0x1E95,
1295 0x00,
1296 0x00,
1297 0x00,
1298 0x00,
1299 0x00,
1300 0x00,
1301 0x00,
1302 0x00,
1303 0x00,
1304 0x00,
1305 0x00,
1306 /* U+1EA0 */ 0x1EA1,
1307 0x00,
1308 /* U+1EA2 */ 0x1EA3,
1309 0x00,
1310 /* U+1EA4 */ 0x1EA5,
1311 0x00,
1312 /* U+1EA6 */ 0x1EA7,
1313 0x00,
1314 /* U+1EA8 */ 0x1EA9,
1315 0x00,
1316 /* U+1EAA */ 0x1EAB,
1317 0x00,
1318 /* U+1EAC */ 0x1EAD,
1319 0x00,
1320 /* U+1EAE */ 0x1EAF,
1321 0x00,
1322 /* U+1EB0 */ 0x1EB1,
1323 0x00,
1324 /* U+1EB2 */ 0x1EB3,
1325 0x00,
1326 /* U+1EB4 */ 0x1EB5,
1327 0x00,
1328 /* U+1EB6 */ 0x1EB7,
1329 0x00,
1330 /* U+1EB8 */ 0x1EB9,
1331 0x00,
1332 /* U+1EBA */ 0x1EBB,
1333 0x00,
1334 /* U+1EBC */ 0x1EBD,
1335 0x00,
1336 /* U+1EBE */ 0x1EBF,
1337 0x00,
1338 /* U+1EC0 */ 0x1EC1,
1339 0x00,
1340 /* U+1EC2 */ 0x1EC3,
1341 0x00,
1342 /* U+1EC4 */ 0x1EC5,
1343 0x00,
1344 /* U+1EC6 */ 0x1EC7,
1345 0x00,
1346 /* U+1EC8 */ 0x1EC9,
1347 0x00,
```

```
1348 /* U+1ECA */ 0x1ECB,
1349 0x00,
1350 /* U+1ECC */ 0x1ECD,
1351 0x00,
1352 /* U+1ECE */ 0x1ECF,
1353 0x00,
1354 /* U+1ED0 */ 0x1ED1,
1355 0x00,
1356 /* U+1ED2 */ 0x1ED3,
1357 0x00,
1358 /* U+1ED4 */ 0x1ED5,
1359 0x00,
1360 /* U+1ED6 */ 0x1ED7,
1361 0x00,
1362 /* U+1ED8 */ 0x1ED9,
1363 0x00,
1364 /* U+1EDA */ 0x1EDB,
1365 0x00,
1366 /* U+1EDC */ 0x1EDD,
1367 0x00,
1368 /* U+1EDE */ 0x1EDF,
1369 0x00,
1370 /* U+1EE0 */ 0x1EE1,
1371 0x00,
1372 /* U+1EE2 */ 0x1EE3,
1373 0x00,
1374 /* U+1EE4 */ 0x1EE5,
1375 0x00,
1376 /* U+1EE6 */ 0x1EE7,
1377 0x00,
1378 /* U+1EE8 */ 0x1EE9,
1379 0x00,
1380 /* U+1EEA */ 0x1EEB,
1381 0x00,
1382 /* U+1EEC */ 0x1EED,
1383 0x00,
1384 /* U+1EEE */ 0x1EEF,
1385 0x00,
1386 /* U+1EF0 */ 0x1EF1,
1387 0x00,
1388 /* U+1EF2 */ 0x1EF3,
1389 0x00,
1390 /* U+1EF4 */ 0x1EF5,
1391 0x00,
1392 /* U+1EF6 */ 0x1EF7,
1393 0x00,
1394 /* U+1EF8 */ 0x1EF9,
1395 0x00,
1396 0x00,
1397 0x00,
1398 0x00,
1399 0x00,
1400 0x00,
1401 0x00,
1402 0x00,
1403 0x00,
1404 0x00,
1405 0x00,
1406 0x00,
1407 0x00,
1408 0x00,
1409 0x00,
1410 /* U+1F08 */ 0x1F00,
1411 /* U+1F09 */ 0x1F01,
1412 /* U+1F0A */ 0x1F02,
1413 /* U+1F0B */ 0x1F03,
1414 /* U+1F0C */ 0x1F04,
1415 /* U+1F0D */ 0x1F05,
1416 /* U+1F0E */ 0x1F06,
1417 /* U+1F0F */ 0x1F07,
1418 0x00,
1419 0x00,
1420 0x00,
1421 0x00,
1422 0x00,
1423 0x00,
1424 0x00,
1425 0x00,
1426 /* U+1F18 */ 0x1F10,
1427 /* U+1F19 */ 0x1F11,
1428 /* U+1F1A */ 0x1F12,
1429 /* U+1F1B */ 0x1F13,
1430 /* U+1F1C */ 0x1F14,
1431 /* U+1F1D */ 0x1F15,
1432 0x00,
1433 0x00,
1434 0x00,
```

```
1435 0x00,
1436 0x00,
1437 0x00,
1438 0x00,
1439 0x00,
1440 0x00,
1441 0x00,
1442 /* U+1F28 */ 0x1F20,
1443 /* U+1F29 */ 0x1F21,
1444 /* U+1F2A */ 0x1F22,
1445 /* U+1F2B */ 0x1F23,
1446 /* U+1F2C */ 0x1F24,
1447 /* U+1F2D */ 0x1F25,
1448 /* U+1F2E */ 0x1F26,
1449 /* U+1F2F */ 0x1F27,
1450 0x00,
1451 0x00,
1452 0x00,
1453 0x00,
1454 0x00,
1455 0x00,
1456 0x00,
1457 0x00,
1458 /* U+1F38 */ 0x1F30,
1459 /* U+1F39 */ 0x1F31,
1460 /* U+1F3A */ 0x1F32,
1461 /* U+1F3B */ 0x1F33,
1462 /* U+1F3C */ 0x1F34,
1463 /* U+1F3D */ 0x1F35,
1464 /* U+1F3E */ 0x1F36,
1465 /* U+1F3F */ 0x1F37,
1466 0x00,
1467 0x00,
1468 0x00,
1469 0x00,
1470 0x00,
1471 0x00,
1472 0x00,
1473 0x00,
1474 /* U+1F48 */ 0x1F40,
1475 /* U+1F49 */ 0x1F41,
1476 /* U+1F4A */ 0x1F42,
1477 /* U+1F4B */ 0x1F43,
1478 /* U+1F4C */ 0x1F44,
1479 /* U+1F4D */ 0x1F45,
1480 0x00,
1481 0x00,
1482 0x00,
1483 0x00,
1484 0x00,
1485 0x00,
1486 0x00,
1487 0x00,
1488 0x00,
1489 0x00,
1490 0x00,
1491 /* U+1F59 */ 0x1F51,
1492 0x00,
1493 /* U+1F5B */ 0x1F53,
1494 0x00,
1495 /* U+1F5D */ 0x1F55,
1496 0x00,
1497 /* U+1F5F */ 0x1F57,
1498 0x00,
1499 0x00,
1500 0x00,
1501 0x00,
1502 0x00,
1503 0x00,
1504 0x00,
1505 0x00,
1506 /* U+1F68 */ 0x1F60,
1507 /* U+1F69 */ 0x1F61,
1508 /* U+1F6A */ 0x1F62,
1509 /* U+1F6B */ 0x1F63,
1510 /* U+1F6C */ 0x1F64,
1511 /* U+1F6D */ 0x1F65,
1512 /* U+1F6E */ 0x1F66,
1513 /* U+1F6F */ 0x1F67,
1514 0x00,
1515 0x00,
1516 0x00,
1517 0x00,
1518 0x00,
1519 0x00,
1520 0x00,
1521 0x00,
```

```
1522 0x00,
1523 0x00,
1524 0x00,
1525 0x00,
1526 0x00,
1527 0x00,
1528 0x00,
1529 0x00,
1530 0x00,
1531 0x00,
1532 0x00,
1533 0x00,
1534 0x00,
1535 0x00,
1536 0x00,
1537 0x00,
1538 /* U+1F88 */ 0x0,
1539 /* U+1F89 */ 0x0,
1540 /* U+1F8A */ 0x0,
1541 /* U+1F8B */ 0x0,
1542 /* U+1F8C */ 0x0,
1543 /* U+1F8D */ 0x0,
1544 /* U+1F8E */ 0x0,
1545 /* U+1F8F */ 0x0,
1546 0x00,
1547 0x00,
1548 0x00,
1549 0x00,
1550 0x00,
1551 0x00,
1552 0x00,
1553 0x00,
1554 /* U+1F98 */ 0x0,
1555 /* U+1F99 */ 0x0,
1556 /* U+1F9A */ 0x0,
1557 /* U+1F9B */ 0x0,
1558 /* U+1F9C */ 0x0,
1559 /* U+1F9D */ 0x0,
1560 /* U+1F9E */ 0x0,
1561 /* U+1F9F */ 0x0,
1562 0x00,
1563 0x00,
1564 0x00,
1565 0x00,
1566 0x00,
1567 0x00,
1568 0x00,
1569 0x00,
1570 /* U+1FA8 */ 0x0,
1571 /* U+1FA9 */ 0x0,
1572 /* U+1FAA */ 0x0,
1573 /* U+1FAB */ 0x0,
1574 /* U+1FAC */ 0x0,
1575 /* U+1FAD */ 0x0,
1576 /* U+1FAE */ 0x0,
1577 /* U+1FAF */ 0x0,
1578 0x00,
1579 0x00,
1580 0x00,
1581 0x00,
1582 0x00,
1583 0x00,
1584 0x00,
1585 0x00,
1586 /* U+1FB8 */ 0x1FB0,
1587 /* U+1FB9 */ 0x1FB1,
1588 /* U+1FBA */ 0x1F70,
1589 /* U+1FBB */ 0x1F71,
1590 /* U+1FBC */ 0x0,
1591 0x00,
1592 0x00,
1593 0x00,
1594 0x00,
1595 0x00,
1596 0x00,
1597 0x00,
1598 0x00,
1599 0x00,
1600 0x00,
1601 0x00,
1602 /* U+1FC8 */ 0x1F72,
1603 /* U+1FC9 */ 0x1F73,
1604 /* U+1FCA */ 0x1F74,
1605 /* U+1FCB */ 0x1F75,
1606 /* U+1FCC */ 0x0,
1607 0x00,
1608 0x00,
```

```
1609 0x00,
1610 0x00,
1611 0x00,
1612 0x00,
1613 0x00,
1614 0x00,
1615 0x00,
1616 0x00,
1617 0x00,
1618 /* U+1FD8 */ 0x1FD0,
1619 /* U+1FD9 */ 0x1FD1,
1620 /* U+1FDA */ 0x1F76,
1621 /* U+1FDB */ 0x1F77,
1622 0x00,
1623 0x00,
1624 0x00,
1625 0x00,
1626 0x00,
1627 0x00,
1628 0x00,
1629 0x00,
1630 0x00,
1631 0x00,
1632 0x00,
1633 0x00,
1634 /* U+1FE8 */ 0x1FE0,
1635 /* U+1FE9 */ 0x1FE1,
1636 /* U+1FEA */ 0x1F7A,
1637 /* U+1FEB */ 0x1F7B,
1638 /* U+1FEC */ 0x1FE5,
1639 0x00,
1640 0x00,
1641 0x00,
1642 0x00,
1643 0x00,
1644 0x00,
1645 0x00,
1646 0x00,
1647 0x00,
1648 0x00,
1649 0x00,
1650 /* U+1FF8 */ 0x1F78,
1651 /* U+1FF9 */ 0x1F79,
1652 /* U+1FFA */ 0x1F7C,
1653 /* U+1FFB */ 0x1F7D,
1654 /* U+1FFC */ 0x0,
1655 };
1656
1657 static const unsigned short ucs_table_2102[] = {
1658 /* U+2102 */ 0x0,
1659 0x00,
1660 0x00,
1661 0x00,
1662 0x00,
1663 0x00,
1664 0x00,
1665 0x00,
1666 0x00,
1667 /* U+210B */ 0x0,
1668 /* U+210C */ 0x0,
1669 /* U+210D */ 0x0,
1670 0x00,
1671 0x00,
1672 /* U+2110 */ 0x0,
1673 /* U+2111 */ 0x0,
1674 /* U+2112 */ 0x2113,
1675 0x00,
1676 0x00,
1677 /* U+2115 */ 0x0,
1678 0x00,
1679 0x00,
1680 /* U+2118 */ 0x0,
1681 /* U+2119 */ 0x0,
1682 /* U+211A */ 0x0,
1683 /* U+211B */ 0x0,
1684 /* U+211C */ 0x0,
1685 /* U+211D */ 0x0,
1686 0x00,
1687 0x00,
1688 0x00,
1689 0x00,
1690 0x00,
1691 0x00,
1692 /* U+2124 */ 0x0,
1693 0x00,
1694 0x00,
1695 0x00,
```

```
1696 /* U+2128 */ 0x0,
1697 0x00,
1698 0x00,
1699 0x00,
1700 /* U+212C */ 0x0,
1701 /* U+212D */ 0x0,
1702 0x00,
1703 0x00,
1704 /* U+2130 */ 0x212F,
1705 /* U+2131 */ 0x0,
1706 /* U+2132 */ 0x0,
1707 /* U+2133 */ 0x0,
1708 };
1709
1710 static const unsigned short ucs_table_24B6[] = {
1711 /* U+24B6 */ 0x24D0,
1712 /* U+24B7 */ 0x24D1,
1713 /* U+24B8 */ 0x24D2,
1714 /* U+24B9 */ 0x24D3,
1715 /* U+24BA */ 0x24D4,
1716 /* U+24BB */ 0x24D5,
1717 /* U+24BC */ 0x24D6,
1718 /* U+24BD */ 0x24D7,
1719 /* U+24BE */ 0x24D8,
1720 /* U+24BF */ 0x24D9,
1721 /* U+24C0 */ 0x24DA,
1722 /* U+24C1 */ 0x24DB,
1723 /* U+24C2 */ 0x24DC,
1724 /* U+24C3 */ 0x24DD,
1725 /* U+24C4 */ 0x24DE,
1726 /* U+24C5 */ 0x24DF,
1727 /* U+24C6 */ 0x24E0,
1728 /* U+24C7 */ 0x24E1,
1729 /* U+24C8 */ 0x24E2,
1730 /* U+24C9 */ 0x24E3,
1731 /* U+24CA */ 0x24E4,
1732 /* U+24CB */ 0x24E5,
1733 /* U+24CC */ 0x24E6,
1734 /* U+24CD */ 0x24E7,
1735 /* U+24CE */ 0x24E8,
1736 /* U+24CF */ 0x24E9,
1737 };
1738
1739 static const unsigned short ucs_table_33CE[] = {
1740 /* U+33CE */ 0x0,
1741 };
1742
1743 static const unsigned short ucs_table_FF21[] = {
1744 /* U+FF21 */ 0xFF41,
1745 /* U+FF22 */ 0xFF42,
1746 /* U+FF23 */ 0xFF43,
1747 /* U+FF24 */ 0xFF44,
1748 /* U+FF25 */ 0xFF45,
1749 /* U+FF26 */ 0xFF46,
1750 /* U+FF27 */ 0xFF47,
1751 /* U+FF28 */ 0xFF48,
1752 /* U+FF29 */ 0xFF49,
1753 /* U+FF2A */ 0xFF4A,
1754 /* U+FF2B */ 0xFF4B,
1755 /* U+FF2C */ 0xFF4C,
1756 /* U+FF2D */ 0xFF4D,
1757 /* U+FF2E */ 0xFF4E,
1758 /* U+FF2F */ 0xFF4F,
1759 /* U+FF30 */ 0xFF50,
1760 /* U+FF31 */ 0xFF51,
1761 /* U+FF32 */ 0xFF52,
1762 /* U+FF33 */ 0xFF53,
1763 /* U+FF34 */ 0xFF54,
1764 /* U+FF35 */ 0xFF55,
1765 /* U+FF36 */ 0xFF56,
1766 /* U+FF37 */ 0xFF57,
1767 /* U+FF38 */ 0xFF58,
1768 /* U+FF39 */ 0xFF59,
1769 /* U+FF3A */ 0xFF5A,
1770 };
```

32.196 dingbats_.h

```
1 /* dingbats */
2
3 static const char unicode_to_dingbats_1b_0020[] = {
4 /* U+0020 */ 0x20,
5 0x00,
6 0x00,
7 0x00,
```

```
8 0x00,  
9 0x00,  
10 0x00,  
11 0x00,  
12 0x00,  
13 0x00,  
14 0x00,  
15 0x00,  
16 0x00,  
17 0x00,  
18 0x00,  
19 0x00,  
20 0x00,  
21 0x00,  
22 0x00,  
23 0x00,  
24 0x00,  
25 0x00,  
26 0x00,  
27 0x00,  
28 0x00,  
29 0x00,  
30 0x00,  
31 0x00,  
32 0x00,  
33 0x00,  
34 0x00,  
35 0x00,  
36 0x00,  
37 0x00,  
38 0x00,  
39 0x00,  
40 0x00,  
41 0x00,  
42 0x00,  
43 0x00,  
44 0x00,  
45 0x00,  
46 0x00,  
47 0x00,  
48 0x00,  
49 0x00,  
50 0x00,  
51 0x00,  
52 0x00,  
53 0x00,  
54 0x00,  
55 0x00,  
56 0x00,  
57 0x00,  
58 0x00,  
59 0x00,  
60 0x00,  
61 0x00,  
62 0x00,  
63 0x00,  
64 0x00,  
65 0x00,  
66 0x00,  
67 0x00,  
68 0x00,  
69 0x00,  
70 0x00,  
71 0x00,  
72 0x00,  
73 0x00,  
74 0x00,  
75 0x00,  
76 0x00,  
77 0x00,  
78 0x00,  
79 0x00,  
80 0x00,  
81 0x00,  
82 0x00,  
83 0x00,  
84 0x00,  
85 0x00,  
86 0x00,  
87 0x00,  
88 0x00,  
89 0x00,  
90 0x00,  
91 0x00,  
92 0x00,  
93 0x00,  
94 0x00,
```

```
95 0x00,
96 0x00,
97 0x00,
98 0x00,
99 0x00,
100 0x00,
101 0x00,
102 0x00,
103 0x00,
104 0x00,
105 0x00,
106 0x00,
107 0x00,
108 0x00,
109 0x00,
110 0x00,
111 0x00,
112 0x00,
113 0x00,
114 0x00,
115 0x00,
116 0x00,
117 0x00,
118 0x00,
119 0x00,
120 0x00,
121 0x00,
122 0x00,
123 0x00,
124 0x00,
125 0x00,
126 0x00,
127 0x00,
128 0x00,
129 0x00,
130 0x00,
131 0x00,
132 /* U+00A0 */ 0x20,
133 };
134
135 static const char unicode_to_dingbats_lb_2192[] = {
136 /* U+2192 */ (char) 0xD5,
137 0x00,
138 /* U+2194 */ (char) 0xD6,
139 /* U+2195 */ (char) 0xD7,
140 };
141
142 static const char unicode_to_dingbats_lb_2460[] = {
143 /* U+2460 */ (char) 0xAC,
144 /* U+2461 */ (char) 0xAD,
145 /* U+2462 */ (char) 0xAE,
146 /* U+2463 */ (char) 0xAF,
147 /* U+2464 */ (char) 0xB0,
148 /* U+2465 */ (char) 0xB1,
149 /* U+2466 */ (char) 0xB2,
150 /* U+2467 */ (char) 0xB3,
151 /* U+2468 */ (char) 0xB4,
152 /* U+2469 */ (char) 0xB5,
153 };
154
155 static const char unicode_to_dingbats_lb_25A0[] = {
156 /* U+25A0 */ 0x6E,
157 0x00,
158 0x00,
159 0x00,
160 0x00,
161 0x00,
162 0x00,
163 0x00,
164 0x00,
165 0x00,
166 0x00,
167 0x00,
168 0x00,
169 0x00,
170 0x00,
171 0x00,
172 0x00,
173 0x00,
174 /* U+25B2 */ 0x73,
175 0x00,
176 0x00,
177 0x00,
178 0x00,
179 0x00,
180 0x00,
181 0x00,
```



```
182 0x00,
183 0x00,
184 /* U+25BC */ 0x74,
185 0x00,
186 0x00,
187 0x00,
188 0x00,
189 0x00,
190 0x00,
191 0x00,
192 0x00,
193 0x00,
194 /* U+25C6 */ 0x75,
195 0x00,
196 0x00,
197 0x00,
198 0x00,
199 0x00,
200 0x00,
201 0x00,
202 0x00,
203 /* U+25CF */ 0x6C,
204 0x00,
205 0x00,
206 0x00,
207 0x00,
208 0x00,
209 0x00,
210 0x00,
211 /* U+25D7 */ 0x77,
212 0x00,
213 0x00,
214 0x00,
215 0x00,
216 0x00,
217 0x00,
218 0x00,
219 0x00,
220 0x00,
221 0x00,
222 0x00,
223 0x00,
224 0x00,
225 0x00,
226 0x00,
227 0x00,
228 0x00,
229 0x00,
230 0x00,
231 0x00,
232 0x00,
233 0x00,
234 0x00,
235 0x00,
236 0x00,
237 0x00,
238 0x00,
239 0x00,
240 0x00,
241 0x00,
242 0x00,
243 0x00,
244 0x00,
245 0x00,
246 0x00,
247 0x00,
248 0x00,
249 0x00,
250 0x00,
251 0x00,
252 0x00,
253 0x00,
254 0x00,
255 0x00,
256 0x00,
257 /* U+2605 */ 0x48,
258 0x00,
259 0x00,
260 0x00,
261 0x00,
262 0x00,
263 0x00,
264 0x00,
265 0x00,
266 /* U+260E */ 0x25,
267 0x00,
268 0x00,
```

```
269 0x00,
270 0x00,
271 0x00,
272 0x00,
273 0x00,
274 0x00,
275 0x00,
276 0x00,
277 0x00,
278 0x00,
279 /* U+261B */ 0x2A,
280 0x00,
281 0x00,
282 /* U+261E */ 0x2B,
283 0x00,
284 0x00,
285 0x00,
286 0x00,
287 0x00,
288 0x00,
289 0x00,
290 0x00,
291 0x00,
292 0x00,
293 0x00,
294 0x00,
295 0x00,
296 0x00,
297 0x00,
298 0x00,
299 0x00,
300 0x00,
301 0x00,
302 0x00,
303 0x00,
304 0x00,
305 0x00,
306 0x00,
307 0x00,
308 0x00,
309 0x00,
310 0x00,
311 0x00,
312 0x00,
313 0x00,
314 0x00,
315 0x00,
316 0x00,
317 0x00,
318 0x00,
319 0x00,
320 0x00,
321 0x00,
322 0x00,
323 0x00,
324 0x00,
325 0x00,
326 0x00,
327 0x00,
328 0x00,
329 0x00,
330 0x00,
331 0x00,
332 0x00,
333 0x00,
334 0x00,
335 0x00,
336 0x00,
337 0x00,
338 0x00,
339 0x00,
340 0x00,
341 0x00,
342 0x00,
343 0x00,
344 0x00,
345 0x00,
346 0x00,
347 0x00,
348 /* U+2660 */ (char) 0xAB,
349 0x00,
350 0x00,
351 /* U+2663 */ (char) 0xA8,
352 0x00,
353 /* U+2665 */ (char) 0xAA,
354 /* U+2666 */ (char) 0xA9,
355 };
```

```
356
357 static const char unicode_to_dingbats_1b_2701[] = {
358 /* U+2701 */ 0x21,
359 /* U+2702 */ 0x22,
360 /* U+2703 */ 0x23,
361 /* U+2704 */ 0x24,
362 0x00,
363 /* U+2706 */ 0x26,
364 /* U+2707 */ 0x27,
365 /* U+2708 */ 0x28,
366 /* U+2709 */ 0x29,
367 0x00,
368 0x00,
369 /* U+270C */ 0x2C,
370 /* U+270D */ 0x2D,
371 /* U+270E */ 0x2E,
372 /* U+270F */ 0x2F,
373 /* U+2710 */ 0x30,
374 /* U+2711 */ 0x31,
375 /* U+2712 */ 0x32,
376 /* U+2713 */ 0x33,
377 /* U+2714 */ 0x34,
378 /* U+2715 */ 0x35,
379 /* U+2716 */ 0x36,
380 /* U+2717 */ 0x37,
381 /* U+2718 */ 0x38,
382 /* U+2719 */ 0x39,
383 /* U+271A */ 0x3A,
384 /* U+271B */ 0x3B,
385 /* U+271C */ 0x3C,
386 /* U+271D */ 0x3D,
387 /* U+271E */ 0x3E,
388 /* U+271F */ 0x3F,
389 /* U+2720 */ 0x40,
390 /* U+2721 */ 0x41,
391 /* U+2722 */ 0x42,
392 /* U+2723 */ 0x43,
393 /* U+2724 */ 0x44,
394 /* U+2725 */ 0x45,
395 /* U+2726 */ 0x46,
396 /* U+2727 */ 0x47,
397 0x00,
398 /* U+2729 */ 0x49,
399 /* U+272A */ 0x4A,
400 /* U+272B */ 0x4B,
401 /* U+272C */ 0x4C,
402 /* U+272D */ 0x4D,
403 /* U+272E */ 0x4E,
404 /* U+272F */ 0x4F,
405 /* U+2730 */ 0x50,
406 /* U+2731 */ 0x51,
407 /* U+2732 */ 0x52,
408 /* U+2733 */ 0x53,
409 /* U+2734 */ 0x54,
410 /* U+2735 */ 0x55,
411 /* U+2736 */ 0x56,
412 /* U+2737 */ 0x57,
413 /* U+2738 */ 0x58,
414 /* U+2739 */ 0x59,
415 /* U+273A */ 0x5A,
416 /* U+273B */ 0x5B,
417 /* U+273C */ 0x5C,
418 /* U+273D */ 0x5D,
419 /* U+273E */ 0x5E,
420 /* U+273F */ 0x5F,
421 /* U+2740 */ 0x60,
422 /* U+2741 */ 0x61,
423 /* U+2742 */ 0x62,
424 /* U+2743 */ 0x63,
425 /* U+2744 */ 0x64,
426 /* U+2745 */ 0x65,
427 /* U+2746 */ 0x66,
428 /* U+2747 */ 0x67,
429 /* U+2748 */ 0x68,
430 /* U+2749 */ 0x69,
431 /* U+274A */ 0x6A,
432 /* U+274B */ 0x6B,
433 0x00,
434 /* U+274D */ 0x6D,
435 0x00,
436 /* U+274F */ 0x6F,
437 /* U+2750 */ 0x70,
438 /* U+2751 */ 0x71,
439 /* U+2752 */ 0x72,
440 0x00,
441 0x00,
442 0x00,
```

```
443 /* U+2756 */ 0x76,
444 0x00,
445 /* U+2758 */ 0x78,
446 /* U+2759 */ 0x79,
447 /* U+275A */ 0x7A,
448 /* U+275B */ 0x7B,
449 /* U+275C */ 0x7C,
450 /* U+275D */ 0x7D,
451 /* U+275E */ 0x7E,
452 0x00,
453 0x00,
454 /* U+2761 */ (char) 0xA1,
455 /* U+2762 */ (char) 0xA2,
456 /* U+2763 */ (char) 0xA3,
457 /* U+2764 */ (char) 0xA4,
458 /* U+2765 */ (char) 0xA5,
459 /* U+2766 */ (char) 0xA6,
460 /* U+2767 */ (char) 0xA7,
461 0x00,
462 0x00,
463 0x00,
464 0x00,
465 0x00,
466 0x00,
467 0x00,
468 0x00,
469 0x00,
470 0x00,
471 0x00,
472 0x00,
473 0x00,
474 0x00,
475 /* U+2776 */ (char) 0xB6,
476 /* U+2777 */ (char) 0xB7,
477 /* U+2778 */ (char) 0xB8,
478 /* U+2779 */ (char) 0xB9,
479 /* U+277A */ (char) 0xBA,
480 /* U+277B */ (char) 0xBB,
481 /* U+277C */ (char) 0xBC,
482 /* U+277D */ (char) 0xBD,
483 /* U+277E */ (char) 0xBE,
484 /* U+277F */ (char) 0xBF,
485 /* U+2780 */ (char) 0xC0,
486 /* U+2781 */ (char) 0xC1,
487 /* U+2782 */ (char) 0xC2,
488 /* U+2783 */ (char) 0xC3,
489 /* U+2784 */ (char) 0xC4,
490 /* U+2785 */ (char) 0xC5,
491 /* U+2786 */ (char) 0xC6,
492 /* U+2787 */ (char) 0xC7,
493 /* U+2788 */ (char) 0xC8,
494 /* U+2789 */ (char) 0xC9,
495 /* U+278A */ (char) 0xCA,
496 /* U+278B */ (char) 0xCB,
497 /* U+278C */ (char) 0xCC,
498 /* U+278D */ (char) 0xCD,
499 /* U+278E */ (char) 0xCE,
500 /* U+278F */ (char) 0xCF,
501 /* U+2790 */ (char) 0xD0,
502 /* U+2791 */ (char) 0xD1,
503 /* U+2792 */ (char) 0xD2,
504 /* U+2793 */ (char) 0xD3,
505 /* U+2794 */ (char) 0xD4,
506 0x00,
507 0x00,
508 0x00,
509 /* U+2798 */ (char) 0xD8,
510 /* U+2799 */ (char) 0xD9,
511 /* U+279A */ (char) 0xDA,
512 /* U+279B */ (char) 0xDB,
513 /* U+279C */ (char) 0xDC,
514 /* U+279D */ (char) 0xDD,
515 /* U+279E */ (char) 0xDE,
516 /* U+279F */ (char) 0xDF,
517 /* U+27A0 */ (char) 0xE0,
518 /* U+27A1 */ (char) 0xE1,
519 /* U+27A2 */ (char) 0xE2,
520 /* U+27A3 */ (char) 0xE3,
521 /* U+27A4 */ (char) 0xE4,
522 /* U+27A5 */ (char) 0xE5,
523 /* U+27A6 */ (char) 0xE6,
524 /* U+27A7 */ (char) 0xE7,
525 /* U+27A8 */ (char) 0xE8,
526 /* U+27A9 */ (char) 0xE9,
527 /* U+27AA */ (char) 0xEA,
528 /* U+27AB */ (char) 0xEB,
529 /* U+27AC */ (char) 0xEC,
```

```

530 /* U+27AD */ (char) 0xED,
531 /* U+27AE */ (char) 0xEE,
532 /* U+27AF */ (char) 0xEF,
533 0x00,
534 /* U+27B1 */ (char) 0xF1,
535 /* U+27B2 */ (char) 0xF2,
536 /* U+27B3 */ (char) 0xF3,
537 /* U+27B4 */ (char) 0xF4,
538 /* U+27B5 */ (char) 0xF5,
539 /* U+27B6 */ (char) 0xF6,
540 /* U+27B7 */ (char) 0xF7,
541 /* U+27B8 */ (char) 0xF8,
542 /* U+27B9 */ (char) 0xF9,
543 /* U+27BA */ (char) 0xFA,
544 /* U+27BB */ (char) 0xFB,
545 /* U+27BC */ (char) 0xFC,
546 /* U+27BD */ (char) 0xFD,
547 /* U+27BE */ (char) 0xFE,
548 };
549
550 static const char unicode_to_dingbats_lb_F8D7[] = {
551 /* U+F8D7 */ (char) 0x80,
552 /* U+F8D8 */ (char) 0x81,
553 /* U+F8D9 */ (char) 0x82,
554 /* U+F8DA */ (char) 0x83,
555 /* U+F8DB */ (char) 0x84,
556 /* U+F8DC */ (char) 0x85,
557 /* U+F8DD */ (char) 0x86,
558 /* U+F8DE */ (char) 0x87,
559 /* U+F8DF */ (char) 0x88,
560 /* U+F8E0 */ (char) 0x89,
561 /* U+F8E1 */ (char) 0x8A,
562 /* U+F8E2 */ (char) 0x8B,
563 /* U+F8E3 */ (char) 0x8C,
564 /* U+F8E4 */ (char) 0x8D,
565 };

```

32.197 spacing.h

```

1 /* spacing */
2
3 static const unsigned short ucs_table_0300[] = {
4 /* U+0300 */ 0x0060,
5 /* U+0301 */ 0x00B4,
6 /* U+0302 */ 0x005E,
7 /* U+0303 */ 0x02DC,
8 /* U+0304 */ 0x00AF,
9 /* U+0305 */ 0x203E,
10 /* U+0306 */ 0x02D8,
11 /* U+0307 */ 0x02D9,
12 /* U+0308 */ 0x00A8,
13 /* U+0309 */ 0x0309,
14 /* U+030A */ 0x02DA,
15 /* U+030B */ 0x02DD,
16 /* U+030C */ 0x030C,
17 /* U+030D */ 0x030D,
18 /* U+030E */ 0x030E,
19 /* U+030F */ 0x030F,
20 /* U+0310 */ 0x0310,
21 /* U+0311 */ 0x0311,
22 /* U+0312 */ 0x0312,
23 /* U+0313 */ 0x1FBD,
24 /* U+0314 */ 0x1FFE,
25 /* U+0315 */ 0x0315,
26 /* U+0316 */ 0x0316,
27 /* U+0317 */ 0x0317,
28 /* U+0318 */ 0x0318,
29 /* U+0319 */ 0x0319,
30 /* U+031A */ 0x031A,
31 /* U+031B */ 0x031B,
32 /* U+031C */ 0x031C,
33 /* U+031D */ 0x031D,
34 /* U+031E */ 0x031E,
35 /* U+031F */ 0x031F,
36 /* U+0320 */ 0x0320,
37 /* U+0321 */ 0x0321,
38 /* U+0322 */ 0x0322,
39 /* U+0323 */ 0x0323,
40 /* U+0324 */ 0x0324,
41 /* U+0325 */ 0x0325,
42 /* U+0326 */ 0x0326,
43 /* U+0327 */ 0x00B8,
44 /* U+0328 */ 0x02DB,
45 /* U+0329 */ 0x0329,
46 /* U+032A */ 0x032A,

```

```
47 /* U+032B */ 0x032B,
48 /* U+032C */ 0x032C,
49 /* U+032D */ 0x032D,
50 /* U+032E */ 0x032E,
51 /* U+032F */ 0x032F,
52 /* U+0330 */ 0x0330,
53 /* U+0331 */ 0x0331,
54 /* U+0332 */ 0x005F,
55 /* U+0333 */ 0x2017,
56 /* U+0334 */ 0x0334,
57 /* U+0335 */ 0x0335,
58 /* U+0336 */ 0x0336,
59 /* U+0337 */ 0x0337,
60 /* U+0338 */ 0x0338,
61 /* U+0339 */ 0x0339,
62 /* U+033A */ 0x033A,
63 /* U+033B */ 0x033B,
64 /* U+033C */ 0x033C,
65 /* U+033D */ 0x033D,
66 /* U+033E */ 0x033E,
67 /* U+033F */ 0x033F,
68 /* U+0340 */ 0x0340,
69 /* U+0341 */ 0x0341,
70 /* U+0342 */ 0x1FC0,
71 /* U+0343 */ 0x0343,
72 /* U+0344 */ 0x0344,
73 /* U+0345 */ 0x037A,
74 0x00,
75 0x00,
76 0x00,
77 0x00,
78 0x00,
79 0x00,
80 0x00,
81 0x00,
82 0x00,
83 0x00,
84 0x00,
85 0x00,
86 0x00,
87 0x00,
88 0x00,
89 0x00,
90 0x00,
91 0x00,
92 0x00,
93 0x00,
94 0x00,
95 0x00,
96 0x00,
97 0x00,
98 0x00,
99 0x00,
100 /* U+0360 */ 0x0360,
101 /* U+0361 */ 0x0361,
102 };
103
104 static const unsigned short ucs_table_0483[] = {
105 /* U+0483 */ 0x0483,
106 /* U+0484 */ 0x0484,
107 /* U+0485 */ 0x0485,
108 /* U+0486 */ 0x0486,
109 };
110
111 static const unsigned short ucs_table_0591[] = {
112 /* U+0591 */ 0x0591,
113 /* U+0592 */ 0x0592,
114 /* U+0593 */ 0x0593,
115 /* U+0594 */ 0x0594,
116 /* U+0595 */ 0x0595,
117 /* U+0596 */ 0x0596,
118 /* U+0597 */ 0x0597,
119 /* U+0598 */ 0x0598,
120 /* U+0599 */ 0x0599,
121 /* U+059A */ 0x059A,
122 /* U+059B */ 0x059B,
123 /* U+059C */ 0x059C,
124 /* U+059D */ 0x059D,
125 /* U+059E */ 0x059E,
126 /* U+059F */ 0x059F,
127 /* U+05A0 */ 0x05A0,
128 /* U+05A1 */ 0x05A1,
129 0x00,
130 /* U+05A3 */ 0x05A3,
131 /* U+05A4 */ 0x05A4,
132 /* U+05A5 */ 0x05A5,
133 /* U+05A6 */ 0x05A6,
```

```
134 /* U+05A7 */ 0x05A7,
135 /* U+05A8 */ 0x05A8,
136 /* U+05A9 */ 0x05A9,
137 /* U+05AA */ 0x05AA,
138 /* U+05AB */ 0x05AB,
139 /* U+05AC */ 0x05AC,
140 /* U+05AD */ 0x05AD,
141 /* U+05AE */ 0x05AE,
142 /* U+05AF */ 0x05AF,
143 /* U+05B0 */ 0x05B0,
144 /* U+05B1 */ 0x05B1,
145 /* U+05B2 */ 0x05B2,
146 /* U+05B3 */ 0x05B3,
147 /* U+05B4 */ 0x05B4,
148 /* U+05B5 */ 0x05B5,
149 /* U+05B6 */ 0x05B6,
150 /* U+05B7 */ 0x05B7,
151 /* U+05B8 */ 0x05B8,
152 /* U+05B9 */ 0x05B9,
153 0x00,
154 /* U+05BB */ 0x05BB,
155 /* U+05BC */ 0x05BC,
156 /* U+05BD */ 0x05BD,
157 0x00,
158 /* U+05BF */ 0x05BF,
159 0x00,
160 /* U+05C1 */ 0x05C1,
161 /* U+05C2 */ 0x05C2,
162 0x00,
163 /* U+05C4 */ 0x05C4,
164 };
165
166 static const unsigned short ucs_table_064B[] = {
167 /* U+064B */ 0xFE70,
168 /* U+064C */ 0xFE72,
169 /* U+064D */ 0xFE74,
170 /* U+064E */ 0xFE76,
171 /* U+064F */ 0xFE78,
172 /* U+0650 */ 0xFE7A,
173 /* U+0651 */ 0xFE7C,
174 /* U+0652 */ 0xFE7E,
175 0x00,
176 0x00,
177 0x00,
178 0x00,
179 0x00,
180 0x00,
181 0x00,
182 0x00,
183 0x00,
184 0x00,
185 0x00,
186 0x00,
187 0x00,
188 0x00,
189 0x00,
190 0x00,
191 0x00,
192 0x00,
193 0x00,
194 0x00,
195 0x00,
196 0x00,
197 0x00,
198 0x00,
199 0x00,
200 0x00,
201 0x00,
202 0x00,
203 0x00,
204 /* U+0670 */ 0x0670,
205 0x00,
206 0x00,
207 0x00,
208 0x00,
209 0x00,
210 0x00,
211 0x00,
212 0x00,
213 0x00,
214 0x00,
215 0x00,
216 0x00,
217 0x00,
218 0x00,
219 0x00,
220 0x00,
```

```
221 0x00,  
222 0x00,  
223 0x00,  
224 0x00,  
225 0x00,  
226 0x00,  
227 0x00,  
228 0x00,  
229 0x00,  
230 0x00,  
231 0x00,  
232 0x00,  
233 0x00,  
234 0x00,  
235 0x00,  
236 0x00,  
237 0x00,  
238 0x00,  
239 0x00,  
240 0x00,  
241 0x00,  
242 0x00,  
243 0x00,  
244 0x00,  
245 0x00,  
246 0x00,  
247 0x00,  
248 0x00,  
249 0x00,  
250 0x00,  
251 0x00,  
252 0x00,  
253 0x00,  
254 0x00,  
255 0x00,  
256 0x00,  
257 0x00,  
258 0x00,  
259 0x00,  
260 0x00,  
261 0x00,  
262 0x00,  
263 0x00,  
264 0x00,  
265 0x00,  
266 0x00,  
267 0x00,  
268 0x00,  
269 0x00,  
270 0x00,  
271 0x00,  
272 0x00,  
273 0x00,  
274 0x00,  
275 0x00,  
276 0x00,  
277 0x00,  
278 0x00,  
279 0x00,  
280 0x00,  
281 0x00,  
282 0x00,  
283 0x00,  
284 0x00,  
285 0x00,  
286 0x00,  
287 0x00,  
288 0x00,  
289 0x00,  
290 0x00,  
291 0x00,  
292 0x00,  
293 0x00,  
294 0x00,  
295 0x00,  
296 0x00,  
297 0x00,  
298 0x00,  
299 0x00,  
300 0x00,  
301 0x00,  
302 0x00,  
303 0x00,  
304 0x00,  
305 0x00,  
306 /* U+06D6 */ 0x06D6,  
307 /* U+06D7 */ 0x06D7,
```



```
308 /* U+06D8 */ 0x06D8,
309 /* U+06D9 */ 0x06D9,
310 /* U+06DA */ 0x06DA,
311 /* U+06DB */ 0x06DB,
312 /* U+06DC */ 0x06DC,
313 0x00,
314 0x00,
315 /* U+06DF */ 0x06DF,
316 /* U+06E0 */ 0x06E0,
317 /* U+06E1 */ 0x06E1,
318 /* U+06E2 */ 0x06E2,
319 /* U+06E3 */ 0x06E3,
320 /* U+06E4 */ 0x06E4,
321 0x00,
322 0x00,
323 /* U+06E7 */ 0x06E7,
324 /* U+06E8 */ 0x06E8,
325 0x00,
326 /* U+06EA */ 0x06EA,
327 /* U+06EB */ 0x06EB,
328 /* U+06EC */ 0x06EC,
329 /* U+06ED */ 0x06ED,
330 };
331
332 static const unsigned short ucs_table_0901[] = {
333 /* U+0901 */ 0x0901,
334 /* U+0902 */ 0x0902,
335 0x00,
336 0x00,
337 0x00,
338 0x00,
339 0x00,
340 0x00,
341 0x00,
342 0x00,
343 0x00,
344 0x00,
345 0x00,
346 0x00,
347 0x00,
348 0x00,
349 0x00,
350 0x00,
351 0x00,
352 0x00,
353 0x00,
354 0x00,
355 0x00,
356 0x00,
357 0x00,
358 0x00,
359 0x00,
360 0x00,
361 0x00,
362 0x00,
363 0x00,
364 0x00,
365 0x00,
366 0x00,
367 0x00,
368 0x00,
369 0x00,
370 0x00,
371 0x00,
372 0x00,
373 0x00,
374 0x00,
375 0x00,
376 0x00,
377 0x00,
378 0x00,
379 0x00,
380 0x00,
381 0x00,
382 0x00,
383 0x00,
384 0x00,
385 0x00,
386 0x00,
387 0x00,
388 0x00,
389 0x00,
390 0x00,
391 0x00,
392 /* U+093C */ 0x093C,
393 0x00,
394 0x00,
```

```
395 0x00,
396 0x00,
397 /* U+0941 */ 0x0941,
398 /* U+0942 */ 0x0942,
399 /* U+0943 */ 0x0943,
400 /* U+0944 */ 0x0944,
401 /* U+0945 */ 0x0945,
402 /* U+0946 */ 0x0946,
403 /* U+0947 */ 0x0947,
404 /* U+0948 */ 0x0948,
405 0x00,
406 0x00,
407 0x00,
408 0x00,
409 /* U+094D */ 0x094D,
410 0x00,
411 0x00,
412 0x00,
413 /* U+0951 */ 0x0951,
414 /* U+0952 */ 0x0952,
415 /* U+0953 */ 0x0953,
416 /* U+0954 */ 0x0954,
417 0x00,
418 0x00,
419 0x00,
420 0x00,
421 0x00,
422 0x00,
423 0x00,
424 0x00,
425 0x00,
426 0x00,
427 0x00,
428 0x00,
429 0x00,
430 /* U+0962 */ 0x0962,
431 /* U+0963 */ 0x0963,
432 0x00,
433 0x00,
434 0x00,
435 0x00,
436 0x00,
437 0x00,
438 0x00,
439 0x00,
440 0x00,
441 0x00,
442 0x00,
443 0x00,
444 0x00,
445 0x00,
446 0x00,
447 0x00,
448 0x00,
449 0x00,
450 0x00,
451 0x00,
452 0x00,
453 0x00,
454 0x00,
455 0x00,
456 0x00,
457 0x00,
458 0x00,
459 0x00,
460 0x00,
461 /* U+0981 */ 0x0981,
462 0x00,
463 0x00,
464 0x00,
465 0x00,
466 0x00,
467 0x00,
468 0x00,
469 0x00,
470 0x00,
471 0x00,
472 0x00,
473 0x00,
474 0x00,
475 0x00,
476 0x00,
477 0x00,
478 0x00,
479 0x00,
480 0x00,
481 0x00,
```

```
482 0x00,
483 0x00,
484 0x00,
485 0x00,
486 0x00,
487 0x00,
488 0x00,
489 0x00,
490 0x00,
491 0x00,
492 0x00,
493 0x00,
494 0x00,
495 0x00,
496 0x00,
497 0x00,
498 0x00,
499 0x00,
500 0x00,
501 0x00,
502 0x00,
503 0x00,
504 0x00,
505 0x00,
506 0x00,
507 0x00,
508 0x00,
509 0x00,
510 0x00,
511 0x00,
512 0x00,
513 0x00,
514 0x00,
515 0x00,
516 0x00,
517 0x00,
518 0x00,
519 0x00,
520 /* U+09BC */ 0x09BC,
521 0x00,
522 0x00,
523 0x00,
524 0x00,
525 /* U+09C1 */ 0x09C1,
526 /* U+09C2 */ 0x09C2,
527 /* U+09C3 */ 0x09C3,
528 /* U+09C4 */ 0x09C4,
529 0x00,
530 0x00,
531 0x00,
532 0x00,
533 0x00,
534 0x00,
535 0x00,
536 0x00,
537 /* U+09CD */ 0x09CD,
538 0x00,
539 0x00,
540 0x00,
541 0x00,
542 0x00,
543 0x00,
544 0x00,
545 0x00,
546 0x00,
547 0x00,
548 0x00,
549 0x00,
550 0x00,
551 0x00,
552 0x00,
553 0x00,
554 0x00,
555 0x00,
556 0x00,
557 0x00,
558 /* U+09E2 */ 0x09E2,
559 /* U+09E3 */ 0x09E3,
560 0x00,
561 0x00,
562 0x00,
563 0x00,
564 0x00,
565 0x00,
566 0x00,
567 0x00,
568 0x00,
```

```
569 0x00,  
570 0x00,  
571 0x00,  
572 0x00,  
573 0x00,  
574 0x00,  
575 0x00,  
576 0x00,  
577 0x00,  
578 0x00,  
579 0x00,  
580 0x00,  
581 0x00,  
582 0x00,  
583 0x00,  
584 0x00,  
585 0x00,  
586 0x00,  
587 0x00,  
588 0x00,  
589 0x00,  
590 /* U+0A02 */ 0x0A02,  
591 0x00,  
592 0x00,  
593 0x00,  
594 0x00,  
595 0x00,  
596 0x00,  
597 0x00,  
598 0x00,  
599 0x00,  
600 0x00,  
601 0x00,  
602 0x00,  
603 0x00,  
604 0x00,  
605 0x00,  
606 0x00,  
607 0x00,  
608 0x00,  
609 0x00,  
610 0x00,  
611 0x00,  
612 0x00,  
613 0x00,  
614 0x00,  
615 0x00,  
616 0x00,  
617 0x00,  
618 0x00,  
619 0x00,  
620 0x00,  
621 0x00,  
622 0x00,  
623 0x00,  
624 0x00,  
625 0x00,  
626 0x00,  
627 0x00,  
628 0x00,  
629 0x00,  
630 0x00,  
631 0x00,  
632 0x00,  
633 0x00,  
634 0x00,  
635 0x00,  
636 0x00,  
637 0x00,  
638 0x00,  
639 0x00,  
640 0x00,  
641 0x00,  
642 0x00,  
643 0x00,  
644 0x00,  
645 0x00,  
646 0x00,  
647 0x00,  
648 /* U+0A3C */ 0x0A3C,  
649 0x00,  
650 0x00,  
651 0x00,  
652 0x00,  
653 /* U+0A41 */ 0x0A41,  
654 /* U+0A42 */ 0x0A42,  
655 0x00,
```

```
656 0x00,
657 0x00,
658 0x00,
659 /* U+0A47 */ 0x0A47,
660 /* U+0A48 */ 0x0A48,
661 0x00,
662 0x00,
663 /* U+0A4B */ 0x0A4B,
664 /* U+0A4C */ 0x0A4C,
665 /* U+0A4D */ 0x0A4D,
666 0x00,
667 0x00,
668 0x00,
669 0x00,
670 0x00,
671 0x00,
672 0x00,
673 0x00,
674 0x00,
675 0x00,
676 0x00,
677 0x00,
678 0x00,
679 0x00,
680 0x00,
681 0x00,
682 0x00,
683 0x00,
684 0x00,
685 0x00,
686 0x00,
687 0x00,
688 0x00,
689 0x00,
690 0x00,
691 0x00,
692 0x00,
693 0x00,
694 0x00,
695 0x00,
696 0x00,
697 0x00,
698 0x00,
699 0x00,
700 /* U+0A70 */ 0x0A70,
701 /* U+0A71 */ 0x0A71,
702 0x00,
703 0x00,
704 0x00,
705 0x00,
706 0x00,
707 0x00,
708 0x00,
709 0x00,
710 0x00,
711 0x00,
712 0x00,
713 0x00,
714 0x00,
715 0x00,
716 0x00,
717 /* U+0A81 */ 0x0A81,
718 /* U+0A82 */ 0x0A82,
719 0x00,
720 0x00,
721 0x00,
722 0x00,
723 0x00,
724 0x00,
725 0x00,
726 0x00,
727 0x00,
728 0x00,
729 0x00,
730 0x00,
731 0x00,
732 0x00,
733 0x00,
734 0x00,
735 0x00,
736 0x00,
737 0x00,
738 0x00,
739 0x00,
740 0x00,
741 0x00,
742 0x00,
```

```
743 0x00,  
744 0x00,  
745 0x00,  
746 0x00,  
747 0x00,  
748 0x00,  
749 0x00,  
750 0x00,  
751 0x00,  
752 0x00,  
753 0x00,  
754 0x00,  
755 0x00,  
756 0x00,  
757 0x00,  
758 0x00,  
759 0x00,  
760 0x00,  
761 0x00,  
762 0x00,  
763 0x00,  
764 0x00,  
765 0x00,  
766 0x00,  
767 0x00,  
768 0x00,  
769 0x00,  
770 0x00,  
771 0x00,  
772 0x00,  
773 0x00,  
774 0x00,  
775 0x00,  
776 /* U+0ABC */ 0x0ABC,  
777 0x00,  
778 0x00,  
779 0x00,  
780 0x00,  
781 /* U+0AC1 */ 0x0AC1,  
782 /* U+0AC2 */ 0x0AC2,  
783 /* U+0AC3 */ 0x0AC3,  
784 /* U+0AC4 */ 0x0AC4,  
785 /* U+0AC5 */ 0x0AC5,  
786 0x00,  
787 /* U+0AC7 */ 0x0AC7,  
788 /* U+0AC8 */ 0x0AC8,  
789 0x00,  
790 0x00,  
791 0x00,  
792 0x00,  
793 /* U+0ACD */ 0x0ACD,  
794 0x00,  
795 0x00,  
796 0x00,  
797 0x00,  
798 0x00,  
799 0x00,  
800 0x00,  
801 0x00,  
802 0x00,  
803 0x00,  
804 0x00,  
805 0x00,  
806 0x00,  
807 0x00,  
808 0x00,  
809 0x00,  
810 0x00,  
811 0x00,  
812 0x00,  
813 0x00,  
814 0x00,  
815 0x00,  
816 0x00,  
817 0x00,  
818 0x00,  
819 0x00,  
820 0x00,  
821 0x00,  
822 0x00,  
823 0x00,  
824 0x00,  
825 0x00,  
826 0x00,  
827 0x00,  
828 0x00,  
829 0x00,
```

```
830 0x00,
831 0x00,
832 0x00,
833 0x00,
834 0x00,
835 0x00,
836 0x00,
837 0x00,
838 0x00,
839 0x00,
840 0x00,
841 0x00,
842 0x00,
843 0x00,
844 0x00,
845 /* U+0B01 */ 0x0B01,
846 0x00,
847 0x00,
848 0x00,
849 0x00,
850 0x00,
851 0x00,
852 0x00,
853 0x00,
854 0x00,
855 0x00,
856 0x00,
857 0x00,
858 0x00,
859 0x00,
860 0x00,
861 0x00,
862 0x00,
863 0x00,
864 0x00,
865 0x00,
866 0x00,
867 0x00,
868 0x00,
869 0x00,
870 0x00,
871 0x00,
872 0x00,
873 0x00,
874 0x00,
875 0x00,
876 0x00,
877 0x00,
878 0x00,
879 0x00,
880 0x00,
881 0x00,
882 0x00,
883 0x00,
884 0x00,
885 0x00,
886 0x00,
887 0x00,
888 0x00,
889 0x00,
890 0x00,
891 0x00,
892 0x00,
893 0x00,
894 0x00,
895 0x00,
896 0x00,
897 0x00,
898 0x00,
899 0x00,
900 0x00,
901 0x00,
902 0x00,
903 0x00,
904 /* U+0B3C */ 0x0B3C,
905 0x00,
906 0x00,
907 /* U+0B3F */ 0x0B3F,
908 0x00,
909 /* U+0B41 */ 0x0B41,
910 /* U+0B42 */ 0x0B42,
911 /* U+0B43 */ 0x0B43,
912 0x00,
913 0x00,
914 0x00,
915 0x00,
916 0x00,
```

```
917 0x00,  
918 0x00,  
919 0x00,  
920 0x00,  
921 /* U+0B4D */ 0x0B4D,  
922 0x00,  
923 0x00,  
924 0x00,  
925 0x00,  
926 0x00,  
927 0x00,  
928 0x00,  
929 0x00,  
930 /* U+0B56 */ 0x0B56,  
931 0x00,  
932 0x00,  
933 0x00,  
934 0x00,  
935 0x00,  
936 0x00,  
937 0x00,  
938 0x00,  
939 0x00,  
940 0x00,  
941 0x00,  
942 0x00,  
943 0x00,  
944 0x00,  
945 0x00,  
946 0x00,  
947 0x00,  
948 0x00,  
949 0x00,  
950 0x00,  
951 0x00,  
952 0x00,  
953 0x00,  
954 0x00,  
955 0x00,  
956 0x00,  
957 0x00,  
958 0x00,  
959 0x00,  
960 0x00,  
961 0x00,  
962 0x00,  
963 0x00,  
964 0x00,  
965 0x00,  
966 0x00,  
967 0x00,  
968 0x00,  
969 0x00,  
970 0x00,  
971 0x00,  
972 0x00,  
973 0x00,  
974 /* U+0B82 */ 0x0B82,  
975 0x00,  
976 0x00,  
977 0x00,  
978 0x00,  
979 0x00,  
980 0x00,  
981 0x00,  
982 0x00,  
983 0x00,  
984 0x00,  
985 0x00,  
986 0x00,  
987 0x00,  
988 0x00,  
989 0x00,  
990 0x00,  
991 0x00,  
992 0x00,  
993 0x00,  
994 0x00,  
995 0x00,  
996 0x00,  
997 0x00,  
998 0x00,  
999 0x00,  
1000 0x00,  
1001 0x00,  
1002 0x00,  
1003 0x00,
```



```
1004 0x00,
1005 0x00,
1006 0x00,
1007 0x00,
1008 0x00,
1009 0x00,
1010 0x00,
1011 0x00,
1012 0x00,
1013 0x00,
1014 0x00,
1015 0x00,
1016 0x00,
1017 0x00,
1018 0x00,
1019 0x00,
1020 0x00,
1021 0x00,
1022 0x00,
1023 0x00,
1024 0x00,
1025 0x00,
1026 0x00,
1027 0x00,
1028 0x00,
1029 0x00,
1030 0x00,
1031 0x00,
1032 0x00,
1033 0x00,
1034 0x00,
1035 0x00,
1036 /* U+0BC0 */ 0x0BC0,
1037 0x00,
1038 0x00,
1039 0x00,
1040 0x00,
1041 0x00,
1042 0x00,
1043 0x00,
1044 0x00,
1045 0x00,
1046 0x00,
1047 0x00,
1048 0x00,
1049 /* U+0BCD */ 0x0BCD,
1050 0x00,
1051 0x00,
1052 0x00,
1053 0x00,
1054 0x00,
1055 0x00,
1056 0x00,
1057 0x00,
1058 0x00,
1059 0x00,
1060 0x00,
1061 0x00,
1062 0x00,
1063 0x00,
1064 0x00,
1065 0x00,
1066 0x00,
1067 0x00,
1068 0x00,
1069 0x00,
1070 0x00,
1071 0x00,
1072 0x00,
1073 0x00,
1074 0x00,
1075 0x00,
1076 0x00,
1077 0x00,
1078 0x00,
1079 0x00,
1080 0x00,
1081 0x00,
1082 0x00,
1083 0x00,
1084 0x00,
1085 0x00,
1086 0x00,
1087 0x00,
1088 0x00,
1089 0x00,
1090 0x00,
```

```
1091 0x00,
1092 0x00,
1093 0x00,
1094 0x00,
1095 0x00,
1096 0x00,
1097 0x00,
1098 0x00,
1099 0x00,
1100 0x00,
1101 0x00,
1102 0x00,
1103 0x00,
1104 0x00,
1105 0x00,
1106 0x00,
1107 0x00,
1108 0x00,
1109 0x00,
1110 0x00,
1111 0x00,
1112 0x00,
1113 0x00,
1114 0x00,
1115 0x00,
1116 0x00,
1117 0x00,
1118 0x00,
1119 0x00,
1120 0x00,
1121 0x00,
1122 0x00,
1123 0x00,
1124 0x00,
1125 0x00,
1126 0x00,
1127 0x00,
1128 0x00,
1129 0x00,
1130 0x00,
1131 0x00,
1132 0x00,
1133 0x00,
1134 0x00,
1135 0x00,
1136 0x00,
1137 0x00,
1138 0x00,
1139 0x00,
1140 0x00,
1141 0x00,
1142 0x00,
1143 0x00,
1144 0x00,
1145 0x00,
1146 0x00,
1147 0x00,
1148 0x00,
1149 0x00,
1150 0x00,
1151 0x00,
1152 0x00,
1153 0x00,
1154 0x00,
1155 0x00,
1156 0x00,
1157 0x00,
1158 0x00,
1159 0x00,
1160 0x00,
1161 0x00,
1162 /* U+0C3E */ 0x0C3E,
1163 /* U+0C3F */ 0x0C3F,
1164 /* U+0C40 */ 0x0C40,
1165 0x00,
1166 0x00,
1167 0x00,
1168 0x00,
1169 0x00,
1170 /* U+0C46 */ 0x0C46,
1171 /* U+0C47 */ 0x0C47,
1172 /* U+0C48 */ 0x0C48,
1173 0x00,
1174 /* U+0C4A */ 0x0C4A,
1175 /* U+0C4B */ 0x0C4B,
1176 /* U+0C4C */ 0x0C4C,
1177 /* U+0C4D */ 0x0C4D,
```

```
1178 0x00,
1179 0x00,
1180 0x00,
1181 0x00,
1182 0x00,
1183 0x00,
1184 0x00,
1185 /* U+0C55 */ 0x0C55,
1186 /* U+0C56 */ 0x0C56,
1187 0x00,
1188 0x00,
1189 0x00,
1190 0x00,
1191 0x00,
1192 0x00,
1193 0x00,
1194 0x00,
1195 0x00,
1196 0x00,
1197 0x00,
1198 0x00,
1199 0x00,
1200 0x00,
1201 0x00,
1202 0x00,
1203 0x00,
1204 0x00,
1205 0x00,
1206 0x00,
1207 0x00,
1208 0x00,
1209 0x00,
1210 0x00,
1211 0x00,
1212 0x00,
1213 0x00,
1214 0x00,
1215 0x00,
1216 0x00,
1217 0x00,
1218 0x00,
1219 0x00,
1220 0x00,
1221 0x00,
1222 0x00,
1223 0x00,
1224 0x00,
1225 0x00,
1226 0x00,
1227 0x00,
1228 0x00,
1229 0x00,
1230 0x00,
1231 0x00,
1232 0x00,
1233 0x00,
1234 0x00,
1235 0x00,
1236 0x00,
1237 0x00,
1238 0x00,
1239 0x00,
1240 0x00,
1241 0x00,
1242 0x00,
1243 0x00,
1244 0x00,
1245 0x00,
1246 0x00,
1247 0x00,
1248 0x00,
1249 0x00,
1250 0x00,
1251 0x00,
1252 0x00,
1253 0x00,
1254 0x00,
1255 0x00,
1256 0x00,
1257 0x00,
1258 0x00,
1259 0x00,
1260 0x00,
1261 0x00,
1262 0x00,
1263 0x00,
1264 0x00,
```

```
1265 0x00,
1266 0x00,
1267 0x00,
1268 0x00,
1269 0x00,
1270 0x00,
1271 0x00,
1272 0x00,
1273 0x00,
1274 0x00,
1275 0x00,
1276 0x00,
1277 0x00,
1278 0x00,
1279 0x00,
1280 0x00,
1281 0x00,
1282 0x00,
1283 0x00,
1284 0x00,
1285 0x00,
1286 0x00,
1287 0x00,
1288 0x00,
1289 0x00,
1290 0x00,
1291 /* U+0CBF */ 0x0CBF,
1292 0x00,
1293 0x00,
1294 0x00,
1295 0x00,
1296 0x00,
1297 0x00,
1298 /* U+0CC6 */ 0x0CC6,
1299 0x00,
1300 0x00,
1301 0x00,
1302 0x00,
1303 0x00,
1304 /* U+0CCC */ 0x0CCC,
1305 /* U+0CCD */ 0x0CCD,
1306 0x00,
1307 0x00,
1308 0x00,
1309 0x00,
1310 0x00,
1311 0x00,
1312 0x00,
1313 0x00,
1314 0x00,
1315 0x00,
1316 0x00,
1317 0x00,
1318 0x00,
1319 0x00,
1320 0x00,
1321 0x00,
1322 0x00,
1323 0x00,
1324 0x00,
1325 0x00,
1326 0x00,
1327 0x00,
1328 0x00,
1329 0x00,
1330 0x00,
1331 0x00,
1332 0x00,
1333 0x00,
1334 0x00,
1335 0x00,
1336 0x00,
1337 0x00,
1338 0x00,
1339 0x00,
1340 0x00,
1341 0x00,
1342 0x00,
1343 0x00,
1344 0x00,
1345 0x00,
1346 0x00,
1347 0x00,
1348 0x00,
1349 0x00,
1350 0x00,
1351 0x00,
```

```
1352 0x00,
1353 0x00,
1354 0x00,
1355 0x00,
1356 0x00,
1357 0x00,
1358 0x00,
1359 0x00,
1360 0x00,
1361 0x00,
1362 0x00,
1363 0x00,
1364 0x00,
1365 0x00,
1366 0x00,
1367 0x00,
1368 0x00,
1369 0x00,
1370 0x00,
1371 0x00,
1372 0x00,
1373 0x00,
1374 0x00,
1375 0x00,
1376 0x00,
1377 0x00,
1378 0x00,
1379 0x00,
1380 0x00,
1381 0x00,
1382 0x00,
1383 0x00,
1384 0x00,
1385 0x00,
1386 0x00,
1387 0x00,
1388 0x00,
1389 0x00,
1390 0x00,
1391 0x00,
1392 0x00,
1393 0x00,
1394 0x00,
1395 0x00,
1396 0x00,
1397 0x00,
1398 0x00,
1399 0x00,
1400 0x00,
1401 0x00,
1402 0x00,
1403 0x00,
1404 0x00,
1405 0x00,
1406 0x00,
1407 0x00,
1408 0x00,
1409 0x00,
1410 0x00,
1411 0x00,
1412 0x00,
1413 0x00,
1414 0x00,
1415 0x00,
1416 0x00,
1417 0x00,
1418 0x00,
1419 0x00,
1420 0x00,
1421 /* U+0D41 */ 0x0D41,
1422 /* U+0D42 */ 0x0D42,
1423 /* U+0D43 */ 0x0D43,
1424 0x00,
1425 0x00,
1426 0x00,
1427 0x00,
1428 0x00,
1429 0x00,
1430 0x00,
1431 0x00,
1432 0x00,
1433 /* U+0D4D */ 0x0D4D,
1434 };
1435
1436 static const unsigned short ucs_table_0E31[] = {
1437 /* U+0E31 */ 0x0E31,
1438 0x00,
```

```
1439 0x00,  
1440 /* U+0E34 */ 0x0E34,  
1441 /* U+0E35 */ 0x0E35,  
1442 /* U+0E36 */ 0x0E36,  
1443 /* U+0E37 */ 0x0E37,  
1444 /* U+0E38 */ 0x0E38,  
1445 /* U+0E39 */ 0x0E39,  
1446 /* U+0E3A */ 0x0E3A,  
1447 0x00,  
1448 0x00,  
1449 0x00,  
1450 0x00,  
1451 0x00,  
1452 0x00,  
1453 0x00,  
1454 0x00,  
1455 0x00,  
1456 0x00,  
1457 0x00,  
1458 0x00,  
1459 /* U+0E47 */ 0x0E47,  
1460 /* U+0E48 */ 0x0E48,  
1461 /* U+0E49 */ 0x0E49,  
1462 /* U+0E4A */ 0x0E4A,  
1463 /* U+0E4B */ 0x0E4B,  
1464 /* U+0E4C */ 0x0E4C,  
1465 /* U+0E4D */ 0x0E4D,  
1466 /* U+0E4E */ 0x0E4E,  
1467 0x00,  
1468 0x00,  
1469 0x00,  
1470 0x00,  
1471 0x00,  
1472 0x00,  
1473 0x00,  
1474 0x00,  
1475 0x00,  
1476 0x00,  
1477 0x00,  
1478 0x00,  
1479 0x00,  
1480 0x00,  
1481 0x00,  
1482 0x00,  
1483 0x00,  
1484 0x00,  
1485 0x00,  
1486 0x00,  
1487 0x00,  
1488 0x00,  
1489 0x00,  
1490 0x00,  
1491 0x00,  
1492 0x00,  
1493 0x00,  
1494 0x00,  
1495 0x00,  
1496 0x00,  
1497 0x00,  
1498 0x00,  
1499 0x00,  
1500 0x00,  
1501 0x00,  
1502 0x00,  
1503 0x00,  
1504 0x00,  
1505 0x00,  
1506 0x00,  
1507 0x00,  
1508 0x00,  
1509 0x00,  
1510 0x00,  
1511 0x00,  
1512 0x00,  
1513 0x00,  
1514 0x00,  
1515 0x00,  
1516 0x00,  
1517 0x00,  
1518 0x00,  
1519 0x00,  
1520 0x00,  
1521 0x00,  
1522 0x00,  
1523 0x00,  
1524 0x00,  
1525 0x00,
```

```
1526 0x00,
1527 0x00,
1528 0x00,
1529 0x00,
1530 0x00,
1531 0x00,
1532 0x00,
1533 0x00,
1534 0x00,
1535 0x00,
1536 0x00,
1537 0x00,
1538 0x00,
1539 0x00,
1540 0x00,
1541 0x00,
1542 0x00,
1543 0x00,
1544 0x00,
1545 0x00,
1546 0x00,
1547 0x00,
1548 0x00,
1549 0x00,
1550 0x00,
1551 0x00,
1552 0x00,
1553 0x00,
1554 0x00,
1555 0x00,
1556 0x00,
1557 0x00,
1558 0x00,
1559 0x00,
1560 0x00,
1561 0x00,
1562 0x00,
1563 0x00,
1564 0x00,
1565 /* U+0EB1 */ 0x0EB1,
1566 0x00,
1567 0x00,
1568 /* U+0EB4 */ 0x0EB4,
1569 /* U+0EB5 */ 0x0EB5,
1570 /* U+0EB6 */ 0x0EB6,
1571 /* U+0EB7 */ 0x0EB7,
1572 /* U+0EB8 */ 0x0EB8,
1573 /* U+0EB9 */ 0x0EB9,
1574 0x00,
1575 /* U+0EBB */ 0x0EBB,
1576 /* U+0EBC */ 0x0EBC,
1577 0x00,
1578 0x00,
1579 0x00,
1580 0x00,
1581 0x00,
1582 0x00,
1583 0x00,
1584 0x00,
1585 0x00,
1586 0x00,
1587 0x00,
1588 /* U+0EC8 */ 0x0EC8,
1589 /* U+0EC9 */ 0x0EC9,
1590 /* U+0ECA */ 0x0ECA,
1591 /* U+0ECB */ 0x0ECB,
1592 /* U+0ECC */ 0x0ECC,
1593 /* U+0ECD */ 0x0ECD,
1594 0x00,
1595 0x00,
1596 0x00,
1597 0x00,
1598 0x00,
1599 0x00,
1600 0x00,
1601 0x00,
1602 0x00,
1603 0x00,
1604 0x00,
1605 0x00,
1606 0x00,
1607 0x00,
1608 0x00,
1609 0x00,
1610 0x00,
1611 0x00,
1612 0x00,
```

```
1613 0x00,
1614 0x00,
1615 0x00,
1616 0x00,
1617 0x00,
1618 0x00,
1619 0x00,
1620 0x00,
1621 0x00,
1622 0x00,
1623 0x00,
1624 0x00,
1625 0x00,
1626 0x00,
1627 0x00,
1628 0x00,
1629 0x00,
1630 0x00,
1631 0x00,
1632 0x00,
1633 0x00,
1634 0x00,
1635 0x00,
1636 0x00,
1637 0x00,
1638 0x00,
1639 0x00,
1640 0x00,
1641 0x00,
1642 0x00,
1643 0x00,
1644 0x00,
1645 0x00,
1646 0x00,
1647 0x00,
1648 0x00,
1649 0x00,
1650 0x00,
1651 0x00,
1652 0x00,
1653 0x00,
1654 0x00,
1655 0x00,
1656 0x00,
1657 0x00,
1658 0x00,
1659 0x00,
1660 0x00,
1661 0x00,
1662 0x00,
1663 0x00,
1664 0x00,
1665 0x00,
1666 0x00,
1667 0x00,
1668 /* U+0F18 */ 0x0F18,
1669 /* U+0F19 */ 0x0F19,
1670 0x00,
1671 0x00,
1672 0x00,
1673 0x00,
1674 0x00,
1675 0x00,
1676 0x00,
1677 0x00,
1678 0x00,
1679 0x00,
1680 0x00,
1681 0x00,
1682 0x00,
1683 0x00,
1684 0x00,
1685 0x00,
1686 0x00,
1687 0x00,
1688 0x00,
1689 0x00,
1690 0x00,
1691 0x00,
1692 0x00,
1693 0x00,
1694 0x00,
1695 0x00,
1696 0x00,
1697 /* U+0F35 */ 0x0F35,
1698 0x00,
1699 /* U+0F37 */ 0x0F37,
```



```
1700 0x00,
1701 /* U+0F39 */ 0x0F39,
1702 0x00,
1703 0x00,
1704 0x00,
1705 0x00,
1706 0x00,
1707 0x00,
1708 0x00,
1709 0x00,
1710 0x00,
1711 0x00,
1712 0x00,
1713 0x00,
1714 0x00,
1715 0x00,
1716 0x00,
1717 0x00,
1718 0x00,
1719 0x00,
1720 0x00,
1721 0x00,
1722 0x00,
1723 0x00,
1724 0x00,
1725 0x00,
1726 0x00,
1727 0x00,
1728 0x00,
1729 0x00,
1730 0x00,
1731 0x00,
1732 0x00,
1733 0x00,
1734 0x00,
1735 0x00,
1736 0x00,
1737 0x00,
1738 0x00,
1739 0x00,
1740 0x00,
1741 0x00,
1742 0x00,
1743 0x00,
1744 0x00,
1745 0x00,
1746 0x00,
1747 0x00,
1748 0x00,
1749 0x00,
1750 0x00,
1751 0x00,
1752 0x00,
1753 0x00,
1754 0x00,
1755 0x00,
1756 0x00,
1757 /* U+0F71 */ 0x0F71,
1758 /* U+0F72 */ 0x0F72,
1759 /* U+0F73 */ 0x0F73,
1760 /* U+0F74 */ 0x0F74,
1761 /* U+0F75 */ 0x0F75,
1762 /* U+0F76 */ 0x0F76,
1763 /* U+0F77 */ 0x0F77,
1764 /* U+0F78 */ 0x0F78,
1765 /* U+0F79 */ 0x0F79,
1766 /* U+0F7A */ 0x0F7A,
1767 /* U+0F7B */ 0x0F7B,
1768 /* U+0F7C */ 0x0F7C,
1769 /* U+0F7D */ 0x0F7D,
1770 /* U+0F7E */ 0x0F7E,
1771 0x00,
1772 /* U+0F80 */ 0x0F80,
1773 /* U+0F81 */ 0x0F81,
1774 /* U+0F82 */ 0x0F82,
1775 /* U+0F83 */ 0x0F83,
1776 /* U+0F84 */ 0x0F84,
1777 0x00,
1778 /* U+0F86 */ 0x0F86,
1779 /* U+0F87 */ 0x0F87,
1780 0x00,
1781 0x00,
1782 0x00,
1783 0x00,
1784 0x00,
1785 0x00,
1786 0x00,
```

```
1787 0x00,
1788 /* U+0F90 */ 0x0F90,
1789 /* U+0F91 */ 0x0F91,
1790 /* U+0F92 */ 0x0F92,
1791 /* U+0F93 */ 0x0F93,
1792 /* U+0F94 */ 0x0F94,
1793 /* U+0F95 */ 0x0F95,
1794 0x00,
1795 /* U+0F97 */ 0x0F97,
1796 0x00,
1797 /* U+0F99 */ 0x0F99,
1798 /* U+0F9A */ 0x0F9A,
1799 /* U+0F9B */ 0x0F9B,
1800 /* U+0F9C */ 0x0F9C,
1801 /* U+0F9D */ 0x0F9D,
1802 /* U+0F9E */ 0x0F9E,
1803 /* U+0F9F */ 0x0F9F,
1804 /* U+0FA0 */ 0x0FA0,
1805 /* U+0FA1 */ 0x0FA1,
1806 /* U+0FA2 */ 0x0FA2,
1807 /* U+0FA3 */ 0x0FA3,
1808 /* U+0FA4 */ 0x0FA4,
1809 /* U+0FA5 */ 0x0FA5,
1810 /* U+0FA6 */ 0x0FA6,
1811 /* U+0FA7 */ 0x0FA7,
1812 /* U+0FA8 */ 0x0FA8,
1813 /* U+0FA9 */ 0x0FA9,
1814 /* U+0FAA */ 0x0FAA,
1815 /* U+0FAB */ 0x0FAB,
1816 /* U+0FAC */ 0x0FAC,
1817 /* U+0FAD */ 0x0FAD,
1818 0x00,
1819 0x00,
1820 0x00,
1821 /* U+0FB1 */ 0x0FB1,
1822 /* U+0FB2 */ 0x0FB2,
1823 /* U+0FB3 */ 0x0FB3,
1824 /* U+0FB4 */ 0x0FB4,
1825 /* U+0FB5 */ 0x0FB5,
1826 /* U+0FB6 */ 0x0FB6,
1827 /* U+0FB7 */ 0x0FB7,
1828 0x00,
1829 /* U+0FB9 */ 0x0FB9,
1830 };
1831
1832 static const unsigned short ucs_table_20D0[] = {
1833 /* U+20D0 */ 0x20D0,
1834 /* U+20D1 */ 0x20D1,
1835 /* U+20D2 */ 0x20D2,
1836 /* U+20D3 */ 0x20D3,
1837 /* U+20D4 */ 0x20D4,
1838 /* U+20D5 */ 0x20D5,
1839 /* U+20D6 */ 0x20D6,
1840 /* U+20D7 */ 0x20D7,
1841 /* U+20D8 */ 0x20D8,
1842 /* U+20D9 */ 0x20D9,
1843 /* U+20DA */ 0x20DA,
1844 /* U+20DB */ 0x20DB,
1845 /* U+20DC */ 0x20DC,
1846 0x00,
1847 0x00,
1848 0x00,
1849 0x00,
1850 /* U+20E1 */ 0x20E1,
1851 };
1852
1853 static const unsigned short ucs_table_302A[] = {
1854 /* U+302A */ 0x302A,
1855 /* U+302B */ 0x302B,
1856 /* U+302C */ 0x302C,
1857 /* U+302D */ 0x302D,
1858 /* U+302E */ 0x302E,
1859 /* U+302F */ 0x302F,
1860 0x00,
1861 0x00,
1862 0x00,
1863 0x00,
1864 0x00,
1865 0x00,
1866 0x00,
1867 0x00,
1868 0x00,
1869 0x00,
1870 0x00,
1871 0x00,
1872 0x00,
1873 0x00,
```

```
1874 0x00,  
1875 0x00,  
1876 0x00,  
1877 0x00,  
1878 0x00,  
1879 0x00,  
1880 0x00,  
1881 0x00,  
1882 0x00,  
1883 0x00,  
1884 0x00,  
1885 0x00,  
1886 0x00,  
1887 0x00,  
1888 0x00,  
1889 0x00,  
1890 0x00,  
1891 0x00,  
1892 0x00,  
1893 0x00,  
1894 0x00,  
1895 0x00,  
1896 0x00,  
1897 0x00,  
1898 0x00,  
1899 0x00,  
1900 0x00,  
1901 0x00,  
1902 0x00,  
1903 0x00,  
1904 0x00,  
1905 0x00,  
1906 0x00,  
1907 0x00,  
1908 0x00,  
1909 0x00,  
1910 0x00,  
1911 0x00,  
1912 0x00,  
1913 0x00,  
1914 0x00,  
1915 0x00,  
1916 0x00,  
1917 0x00,  
1918 0x00,  
1919 0x00,  
1920 0x00,  
1921 0x00,  
1922 0x00,  
1923 0x00,  
1924 0x00,  
1925 0x00,  
1926 0x00,  
1927 0x00,  
1928 0x00,  
1929 0x00,  
1930 0x00,  
1931 0x00,  
1932 0x00,  
1933 0x00,  
1934 0x00,  
1935 0x00,  
1936 0x00,  
1937 0x00,  
1938 0x00,  
1939 0x00,  
1940 0x00,  
1941 0x00,  
1942 0x00,  
1943 0x00,  
1944 0x00,  
1945 0x00,  
1946 0x00,  
1947 0x00,  
1948 0x00,  
1949 0x00,  
1950 0x00,  
1951 0x00,  
1952 0x00,  
1953 0x00,  
1954 0x00,  
1955 0x00,  
1956 0x00,  
1957 0x00,  
1958 0x00,  
1959 0x00,  
1960 0x00,
```

```
1961 0x00,
1962 0x00,
1963 0x00,
1964 0x00,
1965 /* U+3099 */ 0x309B,
1966 /* U+309A */ 0x309C,
1967 };
1968
1969 static const unsigned short ucs_table_FB1E[] = {
1970 /* U+FB1E */ 0xFB1E,
1971 };
1972
1973 static const unsigned short ucs_table_FE20[] = {
1974 /* U+FE20 */ 0xFE20,
1975 /* U+FE21 */ 0xFE21,
1976 /* U+FE22 */ 0xFE22,
1977 /* U+FE23 */ 0xFE23,
1978 };
```

32.198 symbol_h

```
1 /* symbol */
2
3 static const char unicode_to_symbol_1b_0020[] = {
4 /* U+0020 */ 0x20,
5 /* U+0021 */ 0x21,
6 0x00,
7 /* U+0023 */ 0x23,
8 0x00,
9 /* U+0025 */ 0x25,
10 /* U+0026 */ 0x26,
11 0x00,
12 /* U+0028 */ 0x28,
13 /* U+0029 */ 0x29,
14 0x00,
15 /* U+002B */ 0x2B,
16 /* U+002C */ 0x2C,
17 0x00,
18 /* U+002E */ 0x2E,
19 /* U+002F */ 0x2F,
20 /* U+0030 */ 0x30,
21 /* U+0031 */ 0x31,
22 /* U+0032 */ 0x32,
23 /* U+0033 */ 0x33,
24 /* U+0034 */ 0x34,
25 /* U+0035 */ 0x35,
26 /* U+0036 */ 0x36,
27 /* U+0037 */ 0x37,
28 /* U+0038 */ 0x38,
29 /* U+0039 */ 0x39,
30 /* U+003A */ 0x3A,
31 /* U+003B */ 0x3B,
32 /* U+003C */ 0x3C,
33 /* U+003D */ 0x3D,
34 /* U+003E */ 0x3E,
35 /* U+003F */ 0x3F,
36 0x00,
37 0x00,
38 0x00,
39 0x00,
40 0x00,
41 0x00,
42 0x00,
43 0x00,
44 0x00,
45 0x00,
46 0x00,
47 0x00,
48 0x00,
49 0x00,
50 0x00,
51 0x00,
52 0x00,
53 0x00,
54 0x00,
55 0x00,
56 0x00,
57 0x00,
58 0x00,
59 0x00,
60 0x00,
61 0x00,
62 0x00,
63 /* U+005B */ 0x5B,
64 0x00,
```

```
65 /* U+005D */ 0x5D,
66 0x00,
67 /* U+005F */ 0x5F,
68 0x00,
69 0x00,
70 0x00,
71 0x00,
72 0x00,
73 0x00,
74 0x00,
75 0x00,
76 0x00,
77 0x00,
78 0x00,
79 0x00,
80 0x00,
81 0x00,
82 0x00,
83 0x00,
84 0x00,
85 0x00,
86 0x00,
87 0x00,
88 0x00,
89 0x00,
90 0x00,
91 0x00,
92 0x00,
93 0x00,
94 0x00,
95 /* U+007B */ 0x7B,
96 /* U+007C */ 0x7C,
97 /* U+007D */ 0x7D,
98 0x00,
99 0x00,
100 0x00,
101 0x00,
102 0x00,
103 0x00,
104 0x00,
105 0x00,
106 0x00,
107 0x00,
108 0x00,
109 0x00,
110 0x00,
111 0x00,
112 0x00,
113 0x00,
114 0x00,
115 0x00,
116 0x00,
117 0x00,
118 0x00,
119 0x00,
120 0x00,
121 0x00,
122 0x00,
123 0x00,
124 0x00,
125 0x00,
126 0x00,
127 0x00,
128 0x00,
129 0x00,
130 0x00,
131 0x00,
132 /* U+00A0 */ 0x20,
133 0x00,
134 0x00,
135 0x00,
136 0x00,
137 0x00,
138 0x00,
139 0x00,
140 0x00,
141 0x00,
142 0x00,
143 0x00,
144 /* U+00AC */ (char) 0xD8,
145 0x00,
146 0x00,
147 0x00,
148 /* U+00B0 */ (char) 0xB0,
149 /* U+00B1 */ (char) 0xB1,
150 0x00,
151 0x00,
```

```
152 0x00,
153 /* U+00B5 */ 0x6D,
154 0x00,
155 0x00,
156 0x00,
157 0x00,
158 0x00,
159 0x00,
160 0x00,
161 0x00,
162 0x00,
163 0x00,
164 0x00,
165 0x00,
166 0x00,
167 0x00,
168 0x00,
169 0x00,
170 0x00,
171 0x00,
172 0x00,
173 0x00,
174 0x00,
175 0x00,
176 0x00,
177 0x00,
178 0x00,
179 0x00,
180 0x00,
181 0x00,
182 0x00,
183 0x00,
184 0x00,
185 0x00,
186 0x00,
187 /* U+00D7 */ (char)0xB4,
188 0x00,
189 0x00,
190 0x00,
191 0x00,
192 0x00,
193 0x00,
194 0x00,
195 0x00,
196 0x00,
197 0x00,
198 0x00,
199 0x00,
200 0x00,
201 0x00,
202 0x00,
203 0x00,
204 0x00,
205 0x00,
206 0x00,
207 0x00,
208 0x00,
209 0x00,
210 0x00,
211 0x00,
212 0x00,
213 0x00,
214 0x00,
215 0x00,
216 0x00,
217 0x00,
218 0x00,
219 /* U+00F7 */ (char)0xB8,
220 };
221
222 static const char unicode_to_symbol_1b_0192[] = {
223 /* U+0192 */ (char)0xA6,
224 };
225
226 static const char unicode_to_symbol_1b_0391[] = {
227 /* U+0391 */ 0x41,
228 /* U+0392 */ 0x42,
229 /* U+0393 */ 0x47,
230 /* U+0394 */ 0x44,
231 /* U+0395 */ 0x45,
232 /* U+0396 */ 0x5A,
233 /* U+0397 */ 0x48,
234 /* U+0398 */ 0x51,
235 /* U+0399 */ 0x49,
236 /* U+039A */ 0x4B,
237 /* U+039B */ 0x4C,
238 /* U+039C */ 0x4D,
```

```
239 /* U+039D */ 0x4E,
240 /* U+039E */ 0x58,
241 /* U+039F */ 0x4F,
242 /* U+03A0 */ 0x50,
243 /* U+03A1 */ 0x52,
244 0x00,
245 /* U+03A3 */ 0x53,
246 /* U+03A4 */ 0x54,
247 /* U+03A5 */ 0x55,
248 /* U+03A6 */ 0x46,
249 /* U+03A7 */ 0x43,
250 /* U+03A8 */ 0x59,
251 /* U+03A9 */ 0x57,
252 0x00,
253 0x00,
254 0x00,
255 0x00,
256 0x00,
257 0x00,
258 0x00,
259 /* U+03B1 */ 0x61,
260 /* U+03B2 */ 0x62,
261 /* U+03B3 */ 0x67,
262 /* U+03B4 */ 0x64,
263 /* U+03B5 */ 0x65,
264 /* U+03B6 */ 0x7A,
265 /* U+03B7 */ 0x68,
266 /* U+03B8 */ 0x71,
267 /* U+03B9 */ 0x69,
268 /* U+03BA */ 0x6B,
269 /* U+03BB */ 0x6C,
270 /* U+03BC */ 0x6D,
271 /* U+03BD */ 0x6E,
272 /* U+03BE */ 0x78,
273 /* U+03BF */ 0x6F,
274 /* U+03C0 */ 0x70,
275 /* U+03C1 */ 0x72,
276 /* U+03C2 */ 0x56,
277 /* U+03C3 */ 0x73,
278 /* U+03C4 */ 0x74,
279 /* U+03C5 */ 0x75,
280 /* U+03C6 */ 0x66,
281 /* U+03C7 */ 0x63,
282 /* U+03C8 */ 0x79,
283 /* U+03C9 */ 0x77,
284 0x00,
285 0x00,
286 0x00,
287 0x00,
288 0x00,
289 0x00,
290 0x00,
291 /* U+03D1 */ 0x4A,
292 /* U+03D2 */ (char) 0xA1,
293 0x00,
294 0x00,
295 /* U+03D5 */ 0x6A,
296 /* U+03D6 */ 0x76,
297 };
298
299 static const char unicode_to_symbol_1b_2022[] = {
300 /* U+2022 */ (char) 0xB7,
301 0x00,
302 0x00,
303 0x00,
304 /* U+2026 */ (char) 0xBC,
305 0x00,
306 0x00,
307 0x00,
308 0x00,
309 0x00,
310 0x00,
311 0x00,
312 0x00,
313 0x00,
314 0x00,
315 0x00,
316 /* U+2032 */ (char) 0xA2,
317 /* U+2033 */ (char) 0xB2,
318 0x00,
319 0x00,
320 0x00,
321 0x00,
322 0x00,
323 0x00,
324 0x00,
325 0x00,
```

```
326 0x00,  
327 0x00,  
328 0x00,  
329 0x00,  
330 0x00,  
331 0x00,  
332 0x00,  
333 0x00,  
334 /* U+2044 */ (char) 0xA4,  
335 0x00,  
336 0x00,  
337 0x00,  
338 0x00,  
339 0x00,  
340 0x00,  
341 0x00,  
342 0x00,  
343 0x00,  
344 0x00,  
345 0x00,  
346 0x00,  
347 0x00,  
348 0x00,  
349 0x00,  
350 0x00,  
351 0x00,  
352 0x00,  
353 0x00,  
354 0x00,  
355 0x00,  
356 0x00,  
357 0x00,  
358 0x00,  
359 0x00,  
360 0x00,  
361 0x00,  
362 0x00,  
363 0x00,  
364 0x00,  
365 0x00,  
366 0x00,  
367 0x00,  
368 0x00,  
369 0x00,  
370 0x00,  
371 0x00,  
372 0x00,  
373 0x00,  
374 0x00,  
375 0x00,  
376 0x00,  
377 0x00,  
378 0x00,  
379 0x00,  
380 0x00,  
381 0x00,  
382 0x00,  
383 0x00,  
384 0x00,  
385 0x00,  
386 0x00,  
387 0x00,  
388 0x00,  
389 0x00,  
390 0x00,  
391 0x00,  
392 0x00,  
393 0x00,  
394 0x00,  
395 0x00,  
396 0x00,  
397 0x00,  
398 0x00,  
399 0x00,  
400 0x00,  
401 0x00,  
402 0x00,  
403 0x00,  
404 0x00,  
405 0x00,  
406 0x00,  
407 0x00,  
408 0x00,  
409 0x00,  
410 0x00,  
411 0x00,  
412 0x00,
```



```
413 0x00,  
414 0x00,  
415 0x00,  
416 0x00,  
417 0x00,  
418 0x00,  
419 0x00,  
420 0x00,  
421 0x00,  
422 0x00,  
423 0x00,  
424 0x00,  
425 0x00,  
426 0x00,  
427 0x00,  
428 0x00,  
429 0x00,  
430 0x00,  
431 0x00,  
432 0x00,  
433 0x00,  
434 0x00,  
435 0x00,  
436 0x00,  
437 0x00,  
438 /* U+20AC */ (char) 0xA0,  
439 0x00,  
440 0x00,  
441 0x00,  
442 0x00,  
443 0x00,  
444 0x00,  
445 0x00,  
446 0x00,  
447 0x00,  
448 0x00,  
449 0x00,  
450 0x00,  
451 0x00,  
452 0x00,  
453 0x00,  
454 0x00,  
455 0x00,  
456 0x00,  
457 0x00,  
458 0x00,  
459 0x00,  
460 0x00,  
461 0x00,  
462 0x00,  
463 0x00,  
464 0x00,  
465 0x00,  
466 0x00,  
467 0x00,  
468 0x00,  
469 0x00,  
470 0x00,  
471 0x00,  
472 0x00,  
473 0x00,  
474 0x00,  
475 0x00,  
476 0x00,  
477 0x00,  
478 0x00,  
479 0x00,  
480 0x00,  
481 0x00,  
482 0x00,  
483 0x00,  
484 0x00,  
485 0x00,  
486 0x00,  
487 0x00,  
488 0x00,  
489 0x00,  
490 0x00,  
491 0x00,  
492 0x00,  
493 0x00,  
494 0x00,  
495 0x00,  
496 0x00,  
497 0x00,  
498 0x00,  
499 0x00,
```

```
500 0x00,
501 0x00,
502 0x00,
503 0x00,
504 0x00,
505 0x00,
506 0x00,
507 0x00,
508 0x00,
509 0x00,
510 0x00,
511 0x00,
512 0x00,
513 0x00,
514 0x00,
515 0x00,
516 0x00,
517 0x00,
518 0x00,
519 0x00,
520 0x00,
521 0x00,
522 0x00,
523 0x00,
524 0x00,
525 0x00,
526 0x00,
527 0x00,
528 0x00,
529 0x00,
530 0x00,
531 0x00,
532 0x00,
533 0x00,
534 0x00,
535 0x00,
536 0x00,
537 0x00,
538 0x00,
539 /* U+2111 */ (char)0xC1,
540 0x00,
541 0x00,
542 0x00,
543 0x00,
544 0x00,
545 0x00,
546 /* U+2118 */ (char)0xC3,
547 0x00,
548 0x00,
549 0x00,
550 /* U+211C */ (char)0xC2,
551 0x00,
552 0x00,
553 0x00,
554 0x00,
555 0x00,
556 0x00,
557 0x00,
558 0x00,
559 0x00,
560 /* U+2126 */ 0x57,
561 0x00,
562 0x00,
563 0x00,
564 0x00,
565 0x00,
566 0x00,
567 0x00,
568 0x00,
569 0x00,
570 0x00,
571 0x00,
572 0x00,
573 0x00,
574 0x00,
575 /* U+2135 */ (char)0xC0,
576 0x00,
577 0x00,
578 0x00,
579 0x00,
580 0x00,
581 0x00,
582 0x00,
583 0x00,
584 0x00,
585 0x00,
586 0x00,
```

```
587 0x00,  
588 0x00,  
589 0x00,  
590 0x00,  
591 0x00,  
592 0x00,  
593 0x00,  
594 0x00,  
595 0x00,  
596 0x00,  
597 0x00,  
598 0x00,  
599 0x00,  
600 0x00,  
601 0x00,  
602 0x00,  
603 0x00,  
604 0x00,  
605 0x00,  
606 0x00,  
607 0x00,  
608 0x00,  
609 0x00,  
610 0x00,  
611 0x00,  
612 0x00,  
613 0x00,  
614 0x00,  
615 0x00,  
616 0x00,  
617 0x00,  
618 0x00,  
619 0x00,  
620 0x00,  
621 0x00,  
622 0x00,  
623 0x00,  
624 0x00,  
625 0x00,  
626 0x00,  
627 0x00,  
628 0x00,  
629 0x00,  
630 0x00,  
631 0x00,  
632 0x00,  
633 0x00,  
634 0x00,  
635 0x00,  
636 0x00,  
637 0x00,  
638 0x00,  
639 0x00,  
640 0x00,  
641 0x00,  
642 0x00,  
643 0x00,  
644 0x00,  
645 0x00,  
646 0x00,  
647 0x00,  
648 0x00,  
649 0x00,  
650 0x00,  
651 0x00,  
652 0x00,  
653 0x00,  
654 0x00,  
655 0x00,  
656 0x00,  
657 0x00,  
658 0x00,  
659 0x00,  
660 0x00,  
661 0x00,  
662 0x00,  
663 0x00,  
664 0x00,  
665 0x00,  
666 /* U+2190 */ (char) 0xAC,  
667 /* U+2191 */ (char) 0xAD,  
668 /* U+2192 */ (char) 0xAE,  
669 /* U+2193 */ (char) 0xAF,  
670 /* U+2194 */ (char) 0xAB,  
671 0x00,  
672 0x00,  
673 0x00,
```

```
674 0x00,  
675 0x00,  
676 0x00,  
677 0x00,  
678 0x00,  
679 0x00,  
680 0x00,  
681 0x00,  
682 0x00,  
683 0x00,  
684 0x00,  
685 0x00,  
686 0x00,  
687 0x00,  
688 0x00,  
689 0x00,  
690 0x00,  
691 0x00,  
692 0x00,  
693 0x00,  
694 0x00,  
695 0x00,  
696 0x00,  
697 0x00,  
698 0x00,  
699 0x00,  
700 0x00,  
701 0x00,  
702 0x00,  
703 /* U+21B5 */ (char) 0xBF,  
704 0x00,  
705 0x00,  
706 0x00,  
707 0x00,  
708 0x00,  
709 0x00,  
710 0x00,  
711 0x00,  
712 0x00,  
713 0x00,  
714 0x00,  
715 0x00,  
716 0x00,  
717 0x00,  
718 0x00,  
719 0x00,  
720 0x00,  
721 0x00,  
722 0x00,  
723 0x00,  
724 0x00,  
725 0x00,  
726 0x00,  
727 0x00,  
728 0x00,  
729 0x00,  
730 /* U+21D0 */ (char) 0xDC,  
731 /* U+21D1 */ (char) 0xDD,  
732 /* U+21D2 */ (char) 0xDE,  
733 /* U+21D3 */ (char) 0xDF,  
734 /* U+21D4 */ (char) 0xDB,  
735 0x00,  
736 0x00,  
737 0x00,  
738 0x00,  
739 0x00,  
740 0x00,  
741 0x00,  
742 0x00,  
743 0x00,  
744 0x00,  
745 0x00,  
746 0x00,  
747 0x00,  
748 0x00,  
749 0x00,  
750 0x00,  
751 0x00,  
752 0x00,  
753 0x00,  
754 0x00,  
755 0x00,  
756 0x00,  
757 0x00,  
758 0x00,  
759 0x00,  
760 0x00,
```

```
761 0x00,
762 0x00,
763 0x00,
764 0x00,
765 0x00,
766 0x00,
767 0x00,
768 0x00,
769 0x00,
770 0x00,
771 0x00,
772 0x00,
773 0x00,
774 0x00,
775 0x00,
776 0x00,
777 0x00,
778 /* U+2200 */ 0x22,
779 0x00,
780 /* U+2202 */ (char) 0xB6,
781 /* U+2203 */ 0x24,
782 0x00,
783 /* U+2205 */ (char) 0xC6,
784 /* U+2206 */ 0x44,
785 /* U+2207 */ (char) 0xD1,
786 /* U+2208 */ (char) 0xCE,
787 /* U+2209 */ (char) 0xCF,
788 0x00,
789 /* U+220B */ 0x27,
790 0x00,
791 0x00,
792 0x00,
793 /* U+220F */ (char) 0xD5,
794 0x00,
795 /* U+2211 */ (char) 0xE5,
796 /* U+2212 */ 0x2D,
797 0x00,
798 0x00,
799 /* U+2215 */ (char) 0xA4,
800 0x00,
801 /* U+2217 */ 0x2A,
802 0x00,
803 0x00,
804 /* U+221A */ (char) 0xD6,
805 0x00,
806 0x00,
807 /* U+221D */ (char) 0xB5,
808 /* U+221E */ (char) 0xA5,
809 0x00,
810 /* U+2220 */ (char) 0xD0,
811 0x00,
812 0x00,
813 0x00,
814 0x00,
815 0x00,
816 0x00,
817 /* U+2227 */ (char) 0xD9,
818 /* U+2228 */ (char) 0xDA,
819 /* U+2229 */ (char) 0xC7,
820 /* U+222A */ (char) 0xC8,
821 /* U+222B */ (char) 0xF2,
822 0x00,
823 0x00,
824 0x00,
825 0x00,
826 0x00,
827 0x00,
828 0x00,
829 0x00,
830 /* U+2234 */ 0x5C,
831 0x00,
832 0x00,
833 0x00,
834 0x00,
835 0x00,
836 0x00,
837 0x00,
838 /* U+223C */ 0x7E,
839 0x00,
840 0x00,
841 0x00,
842 0x00,
843 0x00,
844 0x00,
845 0x00,
846 0x00,
847 /* U+2245 */ 0x40,
```

```
848 0x00,  
849 0x00,  
850 /* U+2248 */ (char) 0xBB,  
851 0x00,  
852 0x00,  
853 0x00,  
854 0x00,  
855 0x00,  
856 0x00,  
857 0x00,  
858 0x00,  
859 0x00,  
860 0x00,  
861 0x00,  
862 0x00,  
863 0x00,  
864 0x00,  
865 0x00,  
866 0x00,  
867 0x00,  
868 0x00,  
869 0x00,  
870 0x00,  
871 0x00,  
872 0x00,  
873 0x00,  
874 /* U+2260 */ (char) 0xB9,  
875 /* U+2261 */ (char) 0xBA,  
876 0x00,  
877 0x00,  
878 /* U+2264 */ (char) 0xA3,  
879 /* U+2265 */ (char) 0xB3,  
880 0x00,  
881 0x00,  
882 0x00,  
883 0x00,  
884 0x00,  
885 0x00,  
886 0x00,  
887 0x00,  
888 0x00,  
889 0x00,  
890 0x00,  
891 0x00,  
892 0x00,  
893 0x00,  
894 0x00,  
895 0x00,  
896 0x00,  
897 0x00,  
898 0x00,  
899 0x00,  
900 0x00,  
901 0x00,  
902 0x00,  
903 0x00,  
904 0x00,  
905 0x00,  
906 0x00,  
907 0x00,  
908 /* U+2282 */ (char) 0xCC,  
909 /* U+2283 */ (char) 0xC9,  
910 /* U+2284 */ (char) 0xCB,  
911 0x00,  
912 /* U+2286 */ (char) 0xCD,  
913 /* U+2287 */ (char) 0xCA,  
914 0x00,  
915 0x00,  
916 0x00,  
917 0x00,  
918 0x00,  
919 0x00,  
920 0x00,  
921 0x00,  
922 0x00,  
923 0x00,  
924 0x00,  
925 0x00,  
926 0x00,  
927 /* U+2295 */ (char) 0xC5,  
928 0x00,  
929 /* U+2297 */ (char) 0xC4,  
930 0x00,  
931 0x00,  
932 0x00,  
933 0x00,  
934 0x00,
```

```
935 0x00,
936 0x00,
937 0x00,
938 0x00,
939 0x00,
940 0x00,
941 0x00,
942 0x00,
943 /* U+22A5 */ 0x5E,
944 0x00,
945 0x00,
946 0x00,
947 0x00,
948 0x00,
949 0x00,
950 0x00,
951 0x00,
952 0x00,
953 0x00,
954 0x00,
955 0x00,
956 0x00,
957 0x00,
958 0x00,
959 0x00,
960 0x00,
961 0x00,
962 0x00,
963 0x00,
964 0x00,
965 0x00,
966 0x00,
967 0x00,
968 0x00,
969 0x00,
970 0x00,
971 0x00,
972 0x00,
973 0x00,
974 0x00,
975 /* U+22C5 */ (char)0xD7,
976 0x00,
977 0x00,
978 0x00,
979 0x00,
980 0x00,
981 0x00,
982 0x00,
983 0x00,
984 0x00,
985 0x00,
986 0x00,
987 0x00,
988 0x00,
989 0x00,
990 0x00,
991 0x00,
992 0x00,
993 0x00,
994 0x00,
995 0x00,
996 0x00,
997 0x00,
998 0x00,
999 0x00,
1000 0x00,
1001 0x00,
1002 0x00,
1003 0x00,
1004 0x00,
1005 0x00,
1006 0x00,
1007 0x00,
1008 0x00,
1009 0x00,
1010 0x00,
1011 0x00,
1012 0x00,
1013 0x00,
1014 0x00,
1015 0x00,
1016 0x00,
1017 0x00,
1018 0x00,
1019 0x00,
1020 0x00,
1021 0x00,
```

```
1022 0x00,
1023 0x00,
1024 0x00,
1025 0x00,
1026 0x00,
1027 0x00,
1028 0x00,
1029 0x00,
1030 0x00,
1031 0x00,
1032 0x00,
1033 0x00,
1034 0x00,
1035 0x00,
1036 0x00,
1037 0x00,
1038 0x00,
1039 0x00,
1040 0x00,
1041 0x00,
1042 0x00,
1043 0x00,
1044 0x00,
1045 0x00,
1046 0x00,
1047 0x00,
1048 0x00,
1049 0x00,
1050 0x00,
1051 0x00,
1052 0x00,
1053 0x00,
1054 0x00,
1055 0x00,
1056 0x00,
1057 0x00,
1058 0x00,
1059 0x00,
1060 0x00,
1061 0x00,
1062 0x00,
1063 0x00,
1064 0x00,
1065 0x00,
1066 /* U+2320 */ (char)0xF3,
1067 /* U+2321 */ (char)0xF5,
1068 0x00,
1069 0x00,
1070 0x00,
1071 0x00,
1072 0x00,
1073 0x00,
1074 0x00,
1075 /* U+2329 */ (char)0xE1,
1076 /* U+232A */ (char)0xF1,
1077 };
1078
1079 static const char unicode_to_symbol_lb_25CA[] = {
1080 /* U+25CA */ (char)0xE0,
1081 };
1082
1083 static const char unicode_to_symbol_lb_2660[] = {
1084 /* U+2660 */ (char)0xAA,
1085 0x00,
1086 0x00,
1087 /* U+2663 */ (char)0xA7,
1088 0x00,
1089 /* U+2665 */ (char)0xA9,
1090 /* U+2666 */ (char)0xA8,
1091 };
1092
1093 static const char unicode_to_symbol_lb_F6D9[] = {
1094 /* U+F6D9 */ (char)0xD3,
1095 /* U+F6DA */ (char)0xD2,
1096 /* U+F6DB */ (char)0xD4,
1097 };
1098
1099 static const char unicode_to_symbol_lb_F8E5[] = {
1100 /* U+F8E5 */ 0x60,
1101 /* U+F8E6 */ (char)0xBD,
1102 /* U+F8E7 */ (char)0xBE,
1103 /* U+F8E8 */ (char)0xE2,
1104 /* U+F8E9 */ (char)0xE3,
1105 /* U+F8EA */ (char)0xE4,
1106 /* U+F8EB */ (char)0xE6,
1107 /* U+F8EC */ (char)0xE7,
1108 /* U+F8ED */ (char)0xE8,
```



```

1109 /* U+F8EE */ (char)0xE9,
1110 /* U+F8EF */ (char)0xEA,
1111 /* U+F8F0 */ (char)0xEB,
1112 /* U+F8F1 */ (char)0xEC,
1113 /* U+F8F2 */ (char)0xED,
1114 /* U+F8F3 */ (char)0xEE,
1115 /* U+F8F4 */ (char)0xEF,
1116 /* U+F8F5 */ (char)0xF4,
1117 /* U+F8F6 */ (char)0xF6,
1118 /* U+F8F7 */ (char)0xF7,
1119 /* U+F8F8 */ (char)0xF8,
1120 /* U+F8F9 */ (char)0xF9,
1121 /* U+F8FA */ (char)0xFA,
1122 /* U+F8FB */ (char)0xFB,
1123 /* U+F8FC */ (char)0xFC,
1124 /* U+F8FD */ (char)0xFD,
1125 /* U+F8FE */ (char)0xFE,
1126 };

```

32.199 armSCII_8.h

```

1 /* $XFree86: xc/lib/X11/lcUniConv/armSCII_8.h,v 1.4 2003/07/15 17:33:45 pascal Exp $ */
2
3 /*
4  * ARMSCII-8
5  */
6
7 static const unsigned short armSCII_8_2uni[96] = {
8     /* 0xa0 */
9     0x00a0, 0xffff, 0x0587, 0x0589, 0x0029, 0x0028, 0x00bb, 0x00ab,
10    0x2014, 0x002e, 0x055d, 0x002c, 0x002d, 0x058a, 0x2026, 0x055c,
11    /* 0xb0 */
12    0x055b, 0x055e, 0x0531, 0x0561, 0x0532, 0x0562, 0x0533, 0x0563,
13    0x0534, 0x0564, 0x0535, 0x0565, 0x0536, 0x0566, 0x0537, 0x0567,
14    /* 0xc0 */
15    0x0538, 0x0568, 0x0539, 0x0569, 0x053a, 0x056a, 0x053b, 0x056b,
16    0x053c, 0x056c, 0x053d, 0x056d, 0x053e, 0x056e, 0x053f, 0x056f,
17    /* 0xd0 */
18    0x0540, 0x0570, 0x0541, 0x0571, 0x0542, 0x0572, 0x0543, 0x0573,
19    0x0544, 0x0574, 0x0545, 0x0575, 0x0546, 0x0576, 0x0547, 0x0577,
20    /* 0xe0 */
21    0x0548, 0x0578, 0x0549, 0x0579, 0x054a, 0x057a, 0x054b, 0x057b,
22    0x054c, 0x057c, 0x054d, 0x057d, 0x054e, 0x057e, 0x054f, 0x057f,
23    /* 0xf0 */
24    0x0550, 0x0580, 0x0551, 0x0581, 0x0552, 0x0582, 0x0553, 0x0583,
25    0x0554, 0x0584, 0x0555, 0x0585, 0x0556, 0x0586, 0x055a, 0xffff,
26 };
27
28 static int
29 armSCII_8_mbtowc (conv_t conv, ucs4_t *pwc, const unsigned char *s, int n)
30 {
31     unsigned char c = *s;
32     if (c < 0xa0) {
33         *pwc = (ucs4_t) c;
34         return 1;
35     }
36     else {
37         unsigned short wc = armSCII_8_2uni[c-0xa0];
38         if (wc != 0xffff) {
39             *pwc = (ucs4_t) wc;
40             return 1;
41         }
42     }
43     return RET_ILSEQ;
44 }
45
46 static const unsigned char armSCII_8_page00[8] = {
47     0xa5, 0xa4, 0x2a, 0x2b, 0xab, 0xac, 0xa9, 0x2f, /* 0x28-0x2f */
48 };
49 static const unsigned char armSCII_8_page00_1[32] = {
50     0xa0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xa0-0xa7 */
51     0x00, 0x00, 0x00, 0xa7, 0x00, 0x00, 0x00, 0x00, /* 0xa8-0xaf */
52     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xb0-0xb7 */
53     0x00, 0x00, 0x00, 0xa6, 0x00, 0x00, 0x00, 0x00, /* 0xb8-0xbf */
54 };
55 static const unsigned char armSCII_8_page05[96] = {
56     0x00, 0xb2, 0xb4, 0xb6, 0xb8, 0xba, 0xbc, 0xbe, /* 0x30-0x37 */
57     0xc0, 0xc2, 0xc4, 0xc6, 0xc8, 0xca, 0xcc, 0xce, /* 0x38-0x3f */
58     0xd0, 0xd2, 0xd4, 0xd6, 0xd8, 0xda, 0xdc, 0xde, /* 0x40-0x47 */
59     0xe0, 0xe2, 0xe4, 0xe6, 0xe8, 0xea, 0xec, 0xee, /* 0x48-0x4f */
60     0xf0, 0xf2, 0xf4, 0xf6, 0xf8, 0xfa, 0xfc, 0x00, /* 0x50-0x57 */
61     0x00, 0x00, 0xfe, 0xb0, 0xaf, 0xaa, 0xb1, 0x00, /* 0x58-0x5f */
62     0x00, 0xb3, 0xb5, 0xb7, 0xb9, 0xbb, 0xbd, 0xbf, /* 0x60-0x67 */
63     0xc1, 0xc3, 0xc5, 0xc7, 0xc9, 0xcb, 0xcd, 0xcf, /* 0x68-0x6f */
64     0xd1, 0xd3, 0xd5, 0xd7, 0xd9, 0xdb, 0xdd, 0xdf, /* 0x70-0x77 */

```

```

65 0xe1, 0xe3, 0xe5, 0xe7, 0xe9, 0xeb, 0xed, 0xef, /* 0x78-0x7f */
66 0xf1, 0xf3, 0xf5, 0xf7, 0xf9, 0xfb, 0xfd, 0xa2, /* 0x80-0x87 */
67 0x00, 0xa3, 0xad, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x88-0x8f */
68 };
69 static const unsigned char armSCII_8_page20[24] = {
70 0x00, 0x00, 0x00, 0x00, 0xa8, 0x00, 0x00, 0x00, /* 0x10-0x17 */
71 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x18-0x1f */
72 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xae, 0x00, /* 0x20-0x27 */
73 };
74
75 static int
76 armSCII_8_wctomb (conv_t conv, unsigned char *r, ucs4_t wc, int n)
77 {
78     unsigned char c = 0;
79     if (wc < 0x0028) {
80         *r = wc;
81         return 1;
82     }
83     else if (wc >= 0x0028 && wc < 0x0030)
84         c = armSCII_8_page00[wc-0x0028];
85     else if (wc >= 0x0030 && wc < 0x00a0)
86         c = wc;
87     else if (wc >= 0x00a0 && wc < 0x00c0)
88         c = armSCII_8_page00_1[wc-0x00a0];
89     else if (wc >= 0x0530 && wc < 0x0590)
90         c = armSCII_8_page05[wc-0x0530];
91     else if (wc >= 0x2010 && wc < 0x2028)
92         c = armSCII_8_page20[wc-0x2010];
93     if (c != 0) {
94         *r = c;
95         return 1;
96     }
97     return RET_ILSEQ;
98 }

```

32.200 ascii.h

```

1 /* $XFree86: xc/lib/X11/lcUniConv/ascii.h,v 1.3 2000/11/29 17:40:28 dawes Exp $ */
2
3 /*
4 * ASCII
5 */
6
7 static int
8 ascii_mbtowc (conv_t conv, ucs4_t *pwc, const unsigned char *s, int n)
9 {
10     unsigned char c = *s;
11     if (c < 0x80) {
12         *pwc = (ucs4_t) c;
13         return 1;
14     }
15     return RET_ILSEQ;
16 }
17
18 static int
19 ascii_wctomb (conv_t conv, unsigned char *r, ucs4_t wc, int n)
20 {
21     if (wc < 0x0080) {
22         *r = wc;
23         return 1;
24     }
25     return RET_ILSEQ;
26 }

```

32.201 big5.h

```

1 /* $XFree86: xc/lib/X11/lcUniConv/big5.h,v 1.2 2003/05/27 22:26:28 tsi Exp $ */
2
3 /*
4 * BIG5
5 */
6 #ifdef NEED_TOWC
7 static const unsigned short big5_2uni_page1[6121] = {
8     /* 0xa1 */
9     0x3000, 0xff0c, 0x3001, 0x3002, 0xff0e, 0x2022, 0xff1b, 0xff1a,
10     0xff1f, 0xff01, 0xfe30, 0x2026, 0x2025, 0xfe50, 0xff64, 0xfe52,
11     0x00b7, 0xfe54, 0xfe55, 0xfe56, 0xfe57, 0xff5c, 0x2013, 0xfe31,
12     0x2014, 0xfe33, 0xfffd, 0xfe34, 0xfe4f, 0xff08, 0xff09, 0xfe35,
13     0xfe36, 0xff5b, 0xff5d, 0xfe37, 0xfe38, 0x3014, 0x3015, 0xfe39,
14     0xfe3a, 0x3010, 0x3011, 0xfe3b, 0xfe3c, 0x300a, 0x300b, 0xfe3d,
15     0xfe3e, 0x3008, 0x3009, 0xfe3f, 0xfe40, 0x300c, 0x300d, 0xfe41,
16     0xfe42, 0x300e, 0x300f, 0xfe43, 0xfe44, 0xfe59, 0xfe5a, 0xfe5b,

```

```
17 0xfe5c, 0xfe5d, 0xfe5e, 0x2018, 0x2019, 0x201c, 0x201d, 0x301d,
18 0x301e, 0x2035, 0x2032, 0xff03, 0xff06, 0xff0a, 0x203b, 0x00a7,
19 0x3003, 0x25cb, 0x25cf, 0x25b3, 0x25b2, 0x25ce, 0x2606, 0x2605,
20 0x25c7, 0x25c6, 0x25a1, 0x25a0, 0x25bd, 0x25bc, 0x32a3, 0x2105,
21 0x203e, 0xffff, 0xff3f, 0xffffd, 0xfe49, 0xfe4a, 0xfe4d, 0xfe4e,
22 0xfe4b, 0xfe4c, 0xfe5f, 0xfe60, 0xfe61, 0xff0b, 0xff0d, 0x00d7,
23 0x00f7, 0x00b1, 0x221a, 0xff1c, 0xff1e, 0xff1d, 0x2266, 0x2267,
24 0x2260, 0x221e, 0x2252, 0x2261, 0xfe62, 0xfe63, 0xfe64, 0xfe65,
25 0xfe66, 0x223c, 0x2229, 0x222a, 0x22a5, 0x2220, 0x221f, 0x22bf,
26 0x33d2, 0x33d1, 0x222b, 0x222e, 0x2235, 0x2234, 0x2640, 0x2642,
27 0x2641, 0x2609, 0x2191, 0x2193, 0x2190, 0x2192, 0x2196, 0x2197,
28 0x2199, 0x2198, 0x2225, 0x2223, 0xffff,
29 /* 0xa2 */
30 0xffff, 0xff0f, 0xff3c, 0xff04, 0x00a5, 0x3012, 0x00a2, 0x00a3,
31 0xff05, 0xff20, 0x2103, 0x2109, 0xfe69, 0xfe6a, 0xfe6b, 0x33d5,
32 0x339c, 0x339d, 0x339e, 0x33ce, 0x33a1, 0x338e, 0x338f, 0x33c4,
33 0x00b0, 0x5159, 0x515b, 0x515e, 0x515d, 0x5161, 0x5163, 0x55e7,
34 0x74e9, 0x7cce, 0x2581, 0x2582, 0x2583, 0x2584, 0x2585, 0x2586,
35 0x2587, 0x2588, 0x258f, 0x258e, 0x258d, 0x258c, 0x258b, 0x258a,
36 0x2589, 0x253c, 0x2534, 0x252c, 0x2524, 0x251c, 0x2594, 0x2500,
37 0x2502, 0x2595, 0x259c, 0x250c, 0x2510, 0x2514, 0x2518, 0x256d, 0x256e,
38 0x2570, 0x256f, 0x2550, 0x255e, 0x256a, 0x2561, 0x25e2, 0x25e3,
39 0x25e5, 0x25e4, 0x2571, 0x2572, 0x2573, 0xff10, 0xff11, 0xff12,
40 0xff13, 0xff14, 0xff15, 0xff16, 0xff17, 0xff18, 0xff19, 0x2160,
41 0x2161, 0x2162, 0x2163, 0x2164, 0x2165, 0x2166, 0x2167, 0x2168,
42 0x2169, 0x3021, 0x3022, 0x3023, 0x3024, 0x3025, 0x3026, 0x3027,
43 0x3028, 0x3029, 0xffff, 0x5344, 0xffffd, 0xff21, 0xff22, 0xff23,
44 0xff24, 0xff25, 0xff26, 0xff27, 0xff28, 0xff29, 0xff2a, 0xff2b,
45 0xff2c, 0xff2d, 0xff2e, 0xff2f, 0xff30, 0xff31, 0xff32, 0xff33,
46 0xff34, 0xff35, 0xff36, 0xff37, 0xff38, 0xff39, 0xff3a, 0xff3b,
47 0xff3c, 0xff3d, 0xff3e, 0xff3f, 0xff40, 0xff41, 0xff42, 0xff43,
48 0xff44, 0xff45, 0xff46, 0xff47, 0xff48, 0xff49, 0xff4a,
49 0xff4b, 0xff4c, 0xff4d, 0xff4e, 0xff4f, 0xff50, 0xff51,
50 0xff52, 0xff53, 0xff54, 0xff55, 0xff56,
51 /* 0xa3 */
52 0xff57, 0xff58, 0xff59, 0xff5a, 0x0391, 0x0392, 0x0393, 0x0394,
53 0x0395, 0x0396, 0x0397, 0x0398, 0x0399, 0x039a, 0x039b, 0x039c,
54 0x039d, 0x039e, 0x039f, 0x03a0, 0x03a1, 0x03a3, 0x03a4, 0x03a5,
55 0x03a6, 0x03a7, 0x03a8, 0x03a9, 0x03b1, 0x03b2, 0x03b3, 0x03b4,
56 0x03b5, 0x03b6, 0x03b7, 0x03b8, 0x03b9, 0x03ba, 0x03bb, 0x03bc,
57 0x03bd, 0x03be, 0x03bf, 0x03c0, 0x03c1, 0x03c3, 0x03c4, 0x03c5,
58 0x03c6, 0x03c7, 0x03c8, 0x03c9, 0x3105, 0x3106, 0x3107, 0x3108,
59 0x3109, 0x310a, 0x310b, 0x310c, 0x310d, 0x310e, 0x310f, 0x3110,
60 0x3111, 0x3112, 0x3113, 0x3114, 0x3115, 0x3116, 0x3117, 0x3118,
61 0x3119, 0x311a, 0x311b, 0x311c, 0x311d, 0x311e, 0x311f, 0x3120,
62 0x3121, 0x3122, 0x3123, 0x3124, 0x3125, 0x3126, 0x3127, 0x3128,
63 0x3129, 0x02d9, 0x02c9, 0x02ca, 0x02c7, 0x02cb, 0xffffd, 0xffffd,
64 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
65 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
66 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
67 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
68 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
69 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
70 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
71 /* 0xa4 */
72 0x4e00, 0x4e59, 0x4e01, 0x4e03, 0x4e43, 0x4e5d, 0x4e86, 0x4e8c,
73 0x4eba, 0x513f, 0x5165, 0x516b, 0x51e0, 0x5200, 0x5201, 0x529b,
74 0x5315, 0x5341, 0x535c, 0x53c8, 0x4e09, 0x4e0b, 0x4e08, 0x4e0a,
75 0x4e2b, 0x4e38, 0x51e1, 0x4e45, 0x4e48, 0x4e5f, 0x4e5e, 0x4e8e,
76 0x4ea1, 0x5140, 0x5203, 0x52fa, 0x5343, 0x53c9, 0x53ce, 0x571f,
77 0x58eb, 0x5915, 0x5927, 0x5973, 0x5b50, 0x5b51, 0x5b53, 0x5b58,
78 0x5c0f, 0x5c22, 0x5c38, 0x5c71, 0x5ddd, 0x5de5, 0x5df1, 0x5df2,
79 0x5df3, 0x5dfe, 0x5e72, 0x5efe, 0x5f0b, 0x5f13, 0x624d, 0x4e11,
80 0x4e10, 0x4e0d, 0x4e2d, 0x4e30, 0x4e39, 0x4e4b, 0x5c39, 0x4e88,
81 0x4e91, 0x4e95, 0x4e92, 0x4e94, 0x4ea2, 0x4ec1, 0x4ec0, 0x4ec3,
82 0x4ec6, 0x4ec7, 0x4ecd, 0x4eca, 0x4ecb, 0x4ec4, 0x5143, 0x5141,
83 0x5167, 0x516d, 0x516e, 0x516c, 0x5197, 0x51f6, 0x5206, 0x5207,
84 0x5208, 0x52fb, 0x52fe, 0x52ff, 0x5316, 0x5339, 0x5348, 0x5347,
85 0x5345, 0x535e, 0x5384, 0x53cb, 0x53ca, 0x53cd, 0x58ec, 0x5929,
86 0x592b, 0x592a, 0x592d, 0x5b54, 0x5c11, 0x5c24, 0x5c3a, 0x5c6f,
87 0x5df4, 0x5e7b, 0x5eff, 0x5f14, 0x5f15, 0x5fc3, 0x6208, 0x6236,
88 0x624b, 0x624e, 0x652f, 0x6587, 0x6597, 0x65a4, 0x65b9, 0x65e5,
89 0x66f0, 0x6708, 0x6728, 0x6b20, 0x6b62, 0x6b79, 0x6bcb, 0x6bd4,
90 0x6bdb, 0x6c0f, 0x6c34, 0x706b, 0x722a, 0x7236, 0x723b, 0x7247,
91 0x7259, 0x725b, 0x72ac, 0x738b, 0x4e19,
92 /* 0xa5 */
93 0x4e16, 0x4e15, 0x4e14, 0x4e18, 0x4e3b, 0x4e4d, 0x4e4f, 0x4e4e,
94 0x4ee5, 0x4ed8, 0x4ed4, 0x4ed5, 0x4ed6, 0x4ed7, 0x4ee3, 0x4ee4,
95 0x4ed9, 0x4ede, 0x5145, 0x5144, 0x5189, 0x518a, 0x51ac, 0x51f9,
96 0x51fa, 0x51f8, 0x520a, 0x52a0, 0x529f, 0x5305, 0x5306, 0x5317,
97 0x531d, 0x4edf, 0x534a, 0x5349, 0x5361, 0x5360, 0x536f, 0x536e,
98 0x53bb, 0x53be, 0x53e4, 0x53f3, 0x53ec, 0x53ee, 0x53e9, 0x53e8,
99 0x53fc, 0x53f8, 0x53f5, 0x53eb, 0x53e6, 0x53ea, 0x53f2, 0x53f1,
100 0x53f0, 0x53e5, 0x53ed, 0x53fb, 0x56db, 0x56da, 0x5916, 0x592e,
101 0x5931, 0x5974, 0x5976, 0x5b55, 0x5b83, 0x5c3c, 0x5de8, 0x5de7,
102 0x5de6, 0x5e02, 0x5e03, 0x5e73, 0x5e7c, 0x5f01, 0x5f18, 0x5f17,
103 0x5fc5, 0x620a, 0x6253, 0x6254, 0x6252, 0x6251, 0x65a5, 0x65e6,
```

```
104 0x672e, 0x672c, 0x672a, 0x672b, 0x672d, 0x6b63, 0x6bcd, 0x6c11,
105 0x6c10, 0x6c38, 0x6c41, 0x6c40, 0x6c3e, 0x72af, 0x7384, 0x7389,
106 0x74dc, 0x74e6, 0x7518, 0x7518, 0x751f, 0x7528, 0x7529, 0x7530, 0x7531,
107 0x7532, 0x7533, 0x758b, 0x767d, 0x76ae, 0x76bf, 0x76ee, 0x77db,
108 0x77e2, 0x77f3, 0x793a, 0x79be, 0x7a74, 0x7acb, 0x4e1e, 0x4e1f,
109 0x4e52, 0x4e53, 0x4e69, 0x4e69, 0x4e99, 0x4ea4, 0x4ea6, 0x4ea5, 0x4eff,
110 0x4f09, 0x4f19, 0x4f0a, 0x4f15, 0x4f0d, 0x4f10, 0x4f11, 0x4f0f,
111 0x4ef2, 0x4ef6, 0x4efb, 0x4ef0, 0x4ef3, 0x4efd, 0x4f01, 0x4f0b,
112 0x5149, 0x5147, 0x5146, 0x5148, 0x5168,
113 /* 0xa6 */
114 0x5171, 0x518d, 0x51b0, 0x5217, 0x5211, 0x5212, 0x520e, 0x5216,
115 0x52a3, 0x5308, 0x5321, 0x5320, 0x5370, 0x5371, 0x5409, 0x540f,
116 0x540c, 0x540a, 0x5410, 0x5401, 0x540b, 0x5404, 0x5411, 0x540d,
117 0x5408, 0x5403, 0x540e, 0x5406, 0x5412, 0x56e0, 0x56de, 0x56dd,
118 0x5733, 0x5730, 0x5728, 0x5728, 0x572d, 0x572c, 0x572f, 0x5729, 0x5919,
119 0x591a, 0x5937, 0x5938, 0x5984, 0x5978, 0x5983, 0x597d, 0x5979,
120 0x5982, 0x5981, 0x5b57, 0x5b58, 0x5b87, 0x5b88, 0x5b85, 0x5b89,
121 0x5bfa, 0x5c16, 0x5c79, 0x5dde, 0x5e06, 0x5e76, 0x5e74, 0x5f0f,
122 0x5f1b, 0x5fd9, 0x5fd6, 0x620e, 0x620c, 0x620d, 0x6210, 0x6263,
123 0x625b, 0x6258, 0x6536, 0x65e9, 0x65e8, 0x65ec, 0x65ed, 0x66f2,
124 0x66f3, 0x6709, 0x673d, 0x673d, 0x6734, 0x6731, 0x6735, 0x6b21, 0x6b64,
125 0x6b7b, 0x6c16, 0x6c5d, 0x6c57, 0x6c59, 0x6c5f, 0x6c60, 0x6c50,
126 0x6c55, 0x6c61, 0x6c5b, 0x6c4d, 0x6c4e, 0x7070, 0x725f, 0x725d,
127 0x767e, 0x7af9, 0x7c73, 0x7c73, 0x7f36, 0x7f36, 0x7f8a, 0x7fbd, 0x8001,
128 0x8003, 0x800c, 0x8012, 0x8033, 0x807f, 0x8089, 0x808b, 0x808c,
129 0x81e3, 0x81ea, 0x81f3, 0x81fc, 0x820c, 0x821b, 0x821f, 0x826e,
130 0x8272, 0x827e, 0x866b, 0x8840, 0x884c, 0x8863, 0x897f, 0x9621,
131 0x4e32, 0x4ea8, 0x4f4d, 0x4f4f, 0x4f47, 0x4f57, 0x4f5e, 0x4f34,
132 0x4f5b, 0x4f55, 0x4f30, 0x4f50, 0x4f51, 0x4f3d, 0x4f3a, 0x4f38,
133 0x4f43, 0x4f54, 0x4f3c, 0x4f46, 0x4f63,
134 /* 0xa7 */
135 0x4f5c, 0x4f60, 0x4f2f, 0x4f4e, 0x4f36, 0x4f59, 0x4f5d, 0x4f48,
136 0x4f5a, 0x514c, 0x514b, 0x514d, 0x5175, 0x51b6, 0x51b7, 0x5225,
137 0x5224, 0x5229, 0x522a, 0x5228, 0x52ab, 0x52a9, 0x52aa, 0x52ac,
138 0x5323, 0x5373, 0x5375, 0x541d, 0x542d, 0x541e, 0x543e, 0x5426,
139 0x544e, 0x5427, 0x5446, 0x5446, 0x5443, 0x5433, 0x5448, 0x5442, 0x541b,
140 0x5429, 0x544a, 0x5439, 0x543b, 0x5438, 0x542e, 0x5435, 0x5436,
141 0x5420, 0x543c, 0x5440, 0x5431, 0x542b, 0x541f, 0x542c, 0x56ea,
142 0x56f0, 0x56e4, 0x56eb, 0x574a, 0x5751, 0x5740, 0x574d, 0x5747,
143 0x574e, 0x573e, 0x5750, 0x574f, 0x573b, 0x58ef, 0x593e, 0x599d,
144 0x5992, 0x59a8, 0x599e, 0x59a3, 0x5999, 0x5996, 0x598d, 0x59a4,
145 0x5993, 0x598a, 0x59a5, 0x5b5d, 0x5b5c, 0x5b5a, 0x5b5b, 0x5b8c,
146 0x5b8b, 0x5b8f, 0x5c2c, 0x5c40, 0x5c41, 0x5c3f, 0x5c3e, 0x5c90,
147 0x5c91, 0x5c94, 0x5c8c, 0x5deb, 0x5e0c, 0x5e8f, 0x5e87, 0x5e8a,
148 0x5ef7, 0x5f04, 0x5f1f, 0x5f64, 0x5f62, 0x5f77, 0x5f79, 0x5fd8,
149 0x5fcc, 0x5fd7, 0x5fcd, 0x5ff1, 0x5feb, 0x5ff8, 0x5fea, 0x6212,
150 0x6211, 0x6284, 0x6297, 0x6296, 0x6280, 0x6276, 0x6289, 0x626d,
151 0x628a, 0x627c, 0x627e, 0x6279, 0x6273, 0x6292, 0x626f, 0x6298,
152 0x626e, 0x6295, 0x6293, 0x6291, 0x6286, 0x6539, 0x653b, 0x6538,
153 0x65f1, 0x66f4, 0x675f, 0x674e, 0x674f, 0x6750, 0x6751, 0x675c,
154 0x6756, 0x675e, 0x6749, 0x6746, 0x6760,
155 /* 0xa8 */
156 0x6753, 0x6757, 0x6b65, 0x6bcf, 0x6c42, 0x6c5e, 0x6c99, 0x6c81,
157 0x6c88, 0x6c89, 0x6c85, 0x6c9b, 0x6c6a, 0x6c7a, 0x6c90, 0x6c70,
158 0x6c8c, 0x6c68, 0x6c96, 0x6c92, 0x6c7d, 0x6c83, 0x6c72, 0x6c7e,
159 0x6c74, 0x6c86, 0x6c76, 0x6c8d, 0x6c94, 0x6c98, 0x6c82, 0x7076,
160 0x707c, 0x707d, 0x7078, 0x7262, 0x7261, 0x7260, 0x72c4, 0x72c2,
161 0x7396, 0x752c, 0x752b, 0x7537, 0x7538, 0x7682, 0x76ef, 0x77e3,
162 0x79c1, 0x79c0, 0x79bf, 0x7a76, 0x7cfc, 0x7f55, 0x8096, 0x8093,
163 0x809d, 0x8098, 0x809b, 0x809a, 0x80b2, 0x826f, 0x8292, 0x828b,
164 0x828d, 0x898b, 0x89d2, 0x8a00, 0x8c37, 0x8c46, 0x8c55, 0x8c9d,
165 0x8d64, 0x8d70, 0x8db3, 0x8eab, 0x8eca, 0x8f9b, 0x8fb0, 0x8fc2,
166 0x8fc6, 0x8fc5, 0x8fc4, 0x5de1, 0x9091, 0x90a2, 0x90aa, 0x90a6,
167 0x90a3, 0x9149, 0x91c6, 0x91cc, 0x9632, 0x962e, 0x9631, 0x962a,
168 0x962c, 0x4e26, 0x4e56, 0x4e73, 0x4e8b, 0x4e9b, 0x4e9e, 0x4eab,
169 0x4eac, 0x4f6f, 0x4f9d, 0x4f8d, 0x4f73, 0x4f7f, 0x4f6c, 0x4f9b,
170 0x4f8b, 0x4f86, 0x4f83, 0x4f70, 0x4f75, 0x4f88, 0x4f69, 0x4f7b,
171 0x4f96, 0x4f7e, 0x4f8f, 0x4f91, 0x4f7a, 0x5154, 0x5152, 0x5155,
172 0x5169, 0x5177, 0x5176, 0x5178, 0x51bd, 0x51fd, 0x523b, 0x5238,
173 0x5237, 0x523a, 0x5230, 0x522e, 0x5236, 0x5241, 0x52be, 0x52bb,
174 0x5352, 0x5354, 0x5353, 0x5353, 0x5351, 0x5366, 0x5377, 0x5378, 0x5379,
175 0x53d6, 0x53d4, 0x53d7, 0x5473, 0x5475,
176 /* 0xa9 */
177 0x5496, 0x5478, 0x5495, 0x5480, 0x547b, 0x5477, 0x5484, 0x5492,
178 0x5486, 0x547c, 0x5490, 0x5471, 0x5476, 0x548c, 0x549a, 0x5462,
179 0x5468, 0x548b, 0x547d, 0x548e, 0x56fa, 0x5783, 0x5777, 0x576a,
180 0x5769, 0x5761, 0x5766, 0x5764, 0x577c, 0x591c, 0x5949, 0x5947,
181 0x5948, 0x5944, 0x5954, 0x59be, 0x59bb, 0x59d4, 0x59b9, 0x59ae,
182 0x59d1, 0x59c6, 0x59d0, 0x59cd, 0x59cb, 0x59d3, 0x59ca, 0x59af,
183 0x59b3, 0x59d2, 0x59c5, 0x5b5f, 0x5b64, 0x5b63, 0x5b97, 0x5b9a,
184 0x5b98, 0x5b9c, 0x5b99, 0x5b9b, 0x5c1a, 0x5c48, 0x5c45, 0x5c46,
185 0x5cb7, 0x5ca1, 0x5cb8, 0x5ca9, 0x5cab, 0x5cb1, 0x5cb3, 0x5e18,
186 0x5e1a, 0x5e16, 0x5e15, 0x5e1b, 0x5e11, 0x5e78, 0x5e9a, 0x5e97,
187 0x5e9c, 0x5e95, 0x5e96, 0x5ef6, 0x5f26, 0x5f27, 0x5f29, 0x5f80,
188 0x5f81, 0x5f7f, 0x5f7c, 0x5fdd, 0x5fe0, 0x5ffd, 0x5ff5, 0x5fff,
189 0x600f, 0x6014, 0x602f, 0x6035, 0x6016, 0x602a, 0x6015, 0x6021,
190 0x6027, 0x6029, 0x602b, 0x601b, 0x6216, 0x6215, 0x623f, 0x623e,
```

```
191 0x6240, 0x627f, 0x62c9, 0x62cc, 0x62c4, 0x62bf, 0x62c2, 0x62b9,
192 0x62d2, 0x62db, 0x62ab, 0x62d3, 0x62d4, 0x62cb, 0x62c8, 0x62a8,
193 0x62bd, 0x62bc, 0x62d0, 0x62d9, 0x62c7, 0x62cd, 0x62b5, 0x62da,
194 0x62b1, 0x62d8, 0x62d6, 0x62d7, 0x62c6, 0x62ac, 0x62ce, 0x653e,
195 0x65a7, 0x65bc, 0x65fa, 0x6614, 0x6613, 0x660c, 0x6606, 0x6602,
196 0x660e, 0x6600, 0x660f, 0x6615, 0x660a,
197 /* 0xaa */
198 0x6607, 0x670d, 0x670b, 0x676d, 0x678b, 0x6795, 0x6771, 0x679c,
199 0x6773, 0x6777, 0x6787, 0x679d, 0x6797, 0x676f, 0x6770, 0x677f,
200 0x6789, 0x677e, 0x6790, 0x6775, 0x679a, 0x6793, 0x677c, 0x676a,
201 0x6772, 0x6b23, 0x6b66, 0x6b67, 0x6b7f, 0x6c13, 0x6c1b, 0x6ce3,
202 0x6ce8, 0x6cf3, 0x6cb1, 0x6ccc, 0x6ce5, 0x6cb3, 0x6cbd, 0x6cbe,
203 0x6cbc, 0x6ce2, 0x6cab, 0x6cd5, 0x6cd3, 0x6cb8, 0x6cc4, 0x6cb9,
204 0x6cc1, 0x6cae, 0x6cd7, 0x6cc5, 0x6cf1, 0x6cbf, 0x6cbb, 0x6ce1,
205 0x6cdb, 0x6cca, 0x6cac, 0x6cef, 0x6cdc, 0x6cd6, 0x6ce0, 0x7095,
206 0x708e, 0x7092, 0x708a, 0x7099, 0x722c, 0x722d, 0x7238, 0x7248,
207 0x7267, 0x7269, 0x72c0, 0x72ce, 0x72d9, 0x72d7, 0x72d0, 0x73a9,
208 0x73a8, 0x739f, 0x73ab, 0x73ab, 0x73a5, 0x753d, 0x759d, 0x7599, 0x759a,
209 0x7684, 0x76c2, 0x76f2, 0x76f4, 0x77e5, 0x77fd, 0x793e, 0x7940,
210 0x7941, 0x79c9, 0x79c8, 0x7a7a, 0x7a79, 0x7afa, 0x7cfe, 0x7f54,
211 0x7f8c, 0x7f8b, 0x8005, 0x8005, 0x80ba, 0x80a5, 0x80a2, 0x80b1, 0x80a1,
212 0x80ab, 0x80a9, 0x80b4, 0x80aa, 0x80af, 0x81e5, 0x81fe, 0x820d,
213 0x82b3, 0x829d, 0x8299, 0x82ad, 0x82bd, 0x829f, 0x82b9, 0x82b1,
214 0x82ac, 0x82a5, 0x82af, 0x82b8, 0x82a3, 0x82b0, 0x82be, 0x82b7,
215 0x864e, 0x8671, 0x521d, 0x8868, 0x8ecb, 0x8fce, 0x8fd4, 0x8fd1,
216 0x90b5, 0x90b8, 0x90b1, 0x90b6, 0x91c7, 0x91d1, 0x9577, 0x9580,
217 0x961c, 0x9640, 0x963f, 0x963b, 0x9644,
218 /* 0xab */
219 0x9642, 0x96b9, 0x96e8, 0x9752, 0x975e, 0x4e9f, 0x4ead, 0x4eae,
220 0x4fe1, 0x4fb5, 0x4faf, 0x4fbf, 0x4fe0, 0x4fd1, 0x4fcf, 0x4fdd,
221 0x4fc3, 0x4fb6, 0x4fd8, 0x4fdf, 0x4fca, 0x4fd7, 0x4fae, 0x4fd0,
222 0x4fc4, 0x4fc2, 0x4fda, 0x4fce, 0x4fde, 0x4fb7, 0x5157, 0x5192,
223 0x5191, 0x51a0, 0x524e, 0x524e, 0x5243, 0x524a, 0x524d, 0x524c, 0x524b,
224 0x5247, 0x52c7, 0x52c9, 0x52c3, 0x52c1, 0x530d, 0x5357, 0x537b,
225 0x539a, 0x53db, 0x54ac, 0x54c0, 0x54a8, 0x54ce, 0x54c9, 0x54b8,
226 0x54a6, 0x54b3, 0x54c7, 0x54c2, 0x54bd, 0x54aa, 0x54c1, 0x54c4,
227 0x54c8, 0x54af, 0x54ab, 0x54b1, 0x54bb, 0x54a9, 0x54a7, 0x54bf,
228 0x56ff, 0x5782, 0x578b, 0x57a0, 0x57a3, 0x57a2, 0x57ce, 0x57ae,
229 0x5793, 0x5955, 0x5951, 0x594f, 0x594e, 0x5950, 0x59dc, 0x59d8,
230 0x59ff, 0x59e3, 0x59e8, 0x5a03, 0x59e5, 0x59ea, 0x59da, 0x59e6,
231 0x5a01, 0x59fb, 0x5b69, 0x5ba3, 0x5ba6, 0x5ba4, 0x5ba2, 0x5ba5,
232 0x5c01, 0x5c4e, 0x5c4f, 0x5c4d, 0x5c4b, 0x5cd9, 0x5cd2, 0x5df7,
233 0x5e1d, 0x5e25, 0x5e1f, 0x5e7d, 0x5ea0, 0x5ea6, 0x5efa, 0x5f08,
234 0x5f2d, 0x5f65, 0x5f88, 0x5f85, 0x5f8a, 0x5f8b, 0x5f87, 0x5f8c,
235 0x5f89, 0x6012, 0x601d, 0x6020, 0x6025, 0x600e, 0x6028, 0x604d,
236 0x6070, 0x6068, 0x6062, 0x6046, 0x6043, 0x606c, 0x606b, 0x606a,
237 0x6064, 0x6241, 0x62dc, 0x6316, 0x6309, 0x62fc, 0x62ed, 0x6301,
238 0x62ee, 0x62fd, 0x6307, 0x62f1, 0x62f7,
239 /* 0xac */
240 0x62ef, 0x62ec, 0x62fe, 0x62f4, 0x6311, 0x6302, 0x653f, 0x6545,
241 0x65ab, 0x65bd, 0x65e2, 0x6625, 0x662d, 0x6620, 0x6627, 0x662f,
242 0x661f, 0x6628, 0x6631, 0x6624, 0x66f7, 0x67ff, 0x67d3, 0x67f1,
243 0x67d4, 0x67d0, 0x67ec, 0x67b6, 0x67af, 0x67f5, 0x67e9, 0x67ef,
244 0x67c4, 0x67d1, 0x67b4, 0x67da, 0x67e5, 0x67b8, 0x67cf, 0x67de,
245 0x67f3, 0x67b0, 0x67d9, 0x67e2, 0x67dd, 0x67d2, 0x6b6a, 0x6b83,
246 0x6b86, 0x6bb5, 0x6bd2, 0x6bd7, 0x6c1f, 0x6cc9, 0x6d0b, 0x6d32,
247 0x6d2a, 0x6d41, 0x6d25, 0x6d0c, 0x6d31, 0x6d1e, 0x6d17, 0x6d3b,
248 0x6d3d, 0x6d3e, 0x6d36, 0x6d1b, 0x6cf5, 0x6d39, 0x6d27, 0x6d38,
249 0x6d29, 0x6d2e, 0x6d35, 0x6d0e, 0x6d2b, 0x70ab, 0x70ba, 0x70b3,
250 0x70ac, 0x70af, 0x70ad, 0x70b8, 0x70ae, 0x70a4, 0x7230, 0x7272,
251 0x726f, 0x7274, 0x72e9, 0x72e0, 0x72e1, 0x73b7, 0x73ca, 0x73bb,
252 0x73b2, 0x73cd, 0x73c0, 0x73b3, 0x751a, 0x752d, 0x754f, 0x754c,
253 0x754e, 0x754b, 0x75ab, 0x75a5, 0x75a5, 0x75a2, 0x75a3, 0x7678,
254 0x7686, 0x7687, 0x7688, 0x76c8, 0x76c6, 0x76c3, 0x76c5, 0x7701,
255 0x76f9, 0x76f8, 0x7709, 0x770b, 0x76fe, 0x76fc, 0x7707, 0x77dc,
256 0x7802, 0x7814, 0x780c, 0x780d, 0x7946, 0x7946, 0x7949, 0x7948, 0x7947,
257 0x79b9, 0x79ba, 0x79d1, 0x79d2, 0x79cb, 0x7a7f, 0x7a81, 0x7aff,
258 0x7afd, 0x7c7d, 0x7d02, 0x7d05, 0x7d00, 0x7d09, 0x7d07, 0x7d04,
259 0x7d06, 0x7f38, 0x7f8e, 0x7fbf, 0x8004,
260 /* 0xad */
261 0x8010, 0x800d, 0x8011, 0x8036, 0x80d6, 0x80e5, 0x80da, 0x80c3,
262 0x80c4, 0x80cc, 0x80e1, 0x80db, 0x80ce, 0x80de, 0x80e4, 0x80dd,
263 0x81f4, 0x8222, 0x82e7, 0x8303, 0x8305, 0x82e3, 0x82db, 0x82e6,
264 0x8304, 0x82e5, 0x8302, 0x8309, 0x82d2, 0x82d7, 0x82f1, 0x8301,
265 0x82dc, 0x82d4, 0x82d1, 0x82de, 0x82d3, 0x82df, 0x82ef, 0x8306,
266 0x8650, 0x8679, 0x867b, 0x867a, 0x884d, 0x886b, 0x8981, 0x89d4,
267 0x8a08, 0x8a02, 0x8a03, 0x8c9e, 0x8ca0, 0x8d74, 0x8d73, 0x8db4,
268 0x8ecd, 0x8ecc, 0x8ff0, 0x8fe6, 0x8fe2, 0x8fea, 0x8fe5, 0x8fed,
269 0x8feb, 0x8fe4, 0x8fe8, 0x90ca, 0x90ce, 0x90c1, 0x90c3, 0x914b,
270 0x914a, 0x91cd, 0x9582, 0x9650, 0x964b, 0x964c, 0x964d, 0x9762,
271 0x9769, 0x97cb, 0x97ed, 0x97f3, 0x9801, 0x98a8, 0x98db, 0x98df,
272 0x9996, 0x9999, 0x4e58, 0x4eb3, 0x500c, 0x500d, 0x5023, 0x4fef,
273 0x5026, 0x5025, 0x4ff8, 0x5029, 0x5016, 0x5006, 0x503c, 0x501f,
274 0x501a, 0x5012, 0x5011, 0x4ffa, 0x5000, 0x5014, 0x5028, 0x4ff1,
275 0x5021, 0x500b, 0x5019, 0x5018, 0x4ff3, 0x4fee, 0x502d, 0x502a,
276 0x4ffe, 0x502b, 0x5009, 0x517c, 0x51a4, 0x51a5, 0x51a2, 0x51cd,
277 0x51cc, 0x51c6, 0x51cb, 0x5256, 0x525c, 0x5254, 0x525b, 0x525d,
```

```
278 0x532a, 0x537f, 0x539f, 0x539d, 0x53df, 0x54e8, 0x5510, 0x5501,
279 0x5537, 0x54fc, 0x54e5, 0x54f2, 0x5506, 0x54fa, 0x5514, 0x54e9,
280 0x54ed, 0x54e1, 0x5509, 0x54ee, 0x54ea,
281 /* 0xae */
282 0x54e6, 0x5527, 0x5507, 0x54fd, 0x550f, 0x5703, 0x5704, 0x57c2,
283 0x57d4, 0x57cb, 0x57c3, 0x5809, 0x590f, 0x5957, 0x5958, 0x595a,
284 0x5a11, 0x5a18, 0x5a1c, 0x5a1f, 0x5a1b, 0x5a13, 0x59ec, 0x5a20,
285 0x5a23, 0x5a29, 0x5a25, 0x5a0c, 0x5a09, 0x5b6b, 0x5c58, 0x5bb0,
286 0x5bb3, 0x5bb6, 0x5bb4, 0x5bae, 0x5bb5, 0x5bb9, 0x5bb8, 0x5c04,
287 0x5c51, 0x5c55, 0x5c50, 0x5ced, 0x5cfd, 0x5cfb, 0x5cea, 0x5ce8,
288 0x5cf0, 0x5cf6, 0x5d01, 0x5cf4, 0x5dee, 0x5e2d, 0x5e2b, 0x5eab,
289 0x5ead, 0x5ea7, 0x5f31, 0x5f92, 0x5f91, 0x5f90, 0x6059, 0x6063,
290 0x6065, 0x6050, 0x6055, 0x606d, 0x6069, 0x606f, 0x6084, 0x609f,
291 0x609a, 0x608d, 0x6094, 0x608c, 0x6085, 0x6096, 0x6247, 0x62f3,
292 0x6308, 0x62ff, 0x634e, 0x633e, 0x632f, 0x6355, 0x6342, 0x6346,
293 0x634f, 0x6349, 0x633a, 0x6350, 0x633d, 0x632a, 0x632b, 0x6328,
294 0x634d, 0x634c, 0x6548, 0x6549, 0x6599, 0x65c1, 0x65c5, 0x6642,
295 0x6649, 0x664f, 0x6643, 0x6652, 0x664c, 0x6645, 0x6641, 0x66f8,
296 0x6714, 0x6715, 0x6717, 0x6821, 0x6838, 0x6848, 0x6846, 0x6853,
297 0x6839, 0x6842, 0x6854, 0x6829, 0x68b3, 0x6817, 0x684c, 0x6851,
298 0x683d, 0x67f4, 0x6850, 0x6840, 0x683c, 0x6843, 0x682a, 0x6845,
299 0x6813, 0x6818, 0x6841, 0x6b8a, 0x6b89, 0x6bb7, 0x6c23, 0x6c27,
300 0x6c28, 0x6c26, 0x6c24, 0x6cf0, 0x6d6a, 0x6d95, 0x6d88, 0x6d87,
301 0x6d66, 0x6d78, 0x6d77, 0x6d59, 0x6d93,
302 /* 0xaf */
303 0x6d6c, 0x6d89, 0x6d6e, 0x6d5a, 0x6d74, 0x6d69, 0x6d8c, 0x6d8a,
304 0x6d79, 0x6d85, 0x6d65, 0x6d94, 0x70ca, 0x70d8, 0x70e4, 0x70d9,
305 0x70c8, 0x70cf, 0x7239, 0x7279, 0x72fc, 0x72f9, 0x72fd, 0x72f8,
306 0x72f7, 0x7386, 0x73ed, 0x7409, 0x73ee, 0x73e0, 0x73ea, 0x73de,
307 0x7554, 0x755d, 0x755c, 0x755a, 0x7559, 0x755e, 0x75c5, 0x75c7,
308 0x75b2, 0x75b3, 0x75bd, 0x75bc, 0x75b9, 0x75c2, 0x75b8, 0x768b,
309 0x76b0, 0x76ca, 0x76cd, 0x76ce, 0x7729, 0x771f, 0x7720, 0x7728,
310 0x77e9, 0x7830, 0x783d, 0x7827, 0x7838, 0x781d, 0x7834, 0x7837, 0x7825,
311 0x782d, 0x7820, 0x781f, 0x7832, 0x7955, 0x7950, 0x7960, 0x795f,
312 0x7956, 0x795e, 0x795d, 0x7957, 0x795a, 0x79e4, 0x79e3, 0x79e7,
313 0x79df, 0x79e6, 0x79e9, 0x79d8, 0x7a84, 0x7a88, 0x7ad9, 0x7b06,
314 0x7b11, 0x7c89, 0x7d21, 0x7d17, 0x7d0b, 0x7d0a, 0x7d20, 0x7d22,
315 0x7d14, 0x7d10, 0x7d15, 0x7d1a, 0x7d1c, 0x7d0d, 0x7d19, 0x7d1b,
316 0x7f3a, 0x7f5f, 0x7f94, 0x7fc5, 0x7fc1, 0x8006, 0x8018, 0x8015,
317 0x8019, 0x8017, 0x803d, 0x803f, 0x80f1, 0x8102, 0x80f0, 0x8105,
318 0x80ed, 0x80f4, 0x8106, 0x80f8, 0x80f3, 0x8108, 0x80fd, 0x810a,
319 0x80fc, 0x80ef, 0x81ed, 0x81ec, 0x8200, 0x8210, 0x822a, 0x822b,
320 0x8228, 0x822c, 0x82bb, 0x832b, 0x8352, 0x8354, 0x834a, 0x8338,
321 0x8350, 0x8349, 0x8335, 0x8334, 0x834f, 0x8332, 0x8339, 0x8336,
322 0x8317, 0x8340, 0x8331, 0x8328, 0x8343,
323 /* 0xb0 */
324 0x8654, 0x868a, 0x86aa, 0x8693, 0x86a4, 0x86a9, 0x868c, 0x86a3,
325 0x869c, 0x8870, 0x8877, 0x8881, 0x8882, 0x887d, 0x8879, 0x8a18,
326 0x8a10, 0x8a0e, 0x8a0c, 0x8a15, 0x8a0a, 0x8a17, 0x8a13, 0x8a16,
327 0x8a0f, 0x8a11, 0x8c48, 0x8c7a, 0x8c79, 0x8ca1, 0x8ca2, 0x8d77,
328 0x8eac, 0x8ed2, 0x8ed4, 0x8ecf, 0x8fb1, 0x9001, 0x9006, 0x8ff7,
329 0x9000, 0x8ffa, 0x8ff4, 0x9003, 0x8ffd, 0x9005, 0x8ff8, 0x9095,
330 0x90e1, 0x90dd, 0x90e2, 0x9152, 0x914d, 0x914c, 0x91d8, 0x91dd,
331 0x91d7, 0x91dc, 0x91d9, 0x9583, 0x9662, 0x9663, 0x9661, 0x965b,
332 0x965d, 0x9664, 0x9658, 0x965e, 0x96bb, 0x98e2, 0x99ac, 0x9aa8,
333 0x9ad8, 0x9b25, 0x9b32, 0x9b3c, 0x4e7e, 0x507a, 0x507d, 0x505c,
334 0x5047, 0x5043, 0x504c, 0x505a, 0x5049, 0x5065, 0x5076, 0x504e,
335 0x5055, 0x5075, 0x5074, 0x5077, 0x504f, 0x500f, 0x506f, 0x506d,
336 0x515c, 0x5195, 0x51f0, 0x526a, 0x526f, 0x52d2, 0x52d9, 0x52d8,
337 0x52d5, 0x5310, 0x530f, 0x5319, 0x533f, 0x5340, 0x533e, 0x53c3,
338 0x66fc, 0x5546, 0x556a, 0x5566, 0x5544, 0x555e, 0x5561, 0x5543,
339 0x554a, 0x5531, 0x5556, 0x554f, 0x5555, 0x552f, 0x5564, 0x5538,
340 0x552e, 0x555c, 0x552c, 0x5563, 0x5533, 0x5541, 0x5557, 0x5708,
341 0x570b, 0x5709, 0x57df, 0x5805, 0x580a, 0x5806, 0x57e0, 0x57e4,
342 0x57fa, 0x5802, 0x5835, 0x57f7, 0x57f9, 0x5920, 0x5962, 0x5a36,
343 0x5a41, 0x5a49, 0x5a66, 0x5a6a, 0x5a40,
344 /* 0xb1 */
345 0x5a3c, 0x5a62, 0x5a5a, 0x5a46, 0x5a4a, 0x5b70, 0x5bc7, 0x5bc5,
346 0x5bc4, 0x5bc2, 0x5bbf, 0x5bc6, 0x5c09, 0x5c08, 0x5c07, 0x5c60,
347 0x5c5c, 0x5c5d, 0x5d07, 0x5d06, 0x5d0e, 0x5d1b, 0x5d16, 0x5d22,
348 0x5d11, 0x5d29, 0x5d14, 0x5d19, 0x5d24, 0x5d27, 0x5d17, 0x5de2,
349 0x5e38, 0x5e36, 0x5e33, 0x5e37, 0x5eb7, 0x5eb8, 0x5eb6, 0x5eb5,
350 0x5ebe, 0x5f35, 0x5f37, 0x5f57, 0x5f6c, 0x5f69, 0x5f6b, 0x5f97,
351 0x5f99, 0x5f9e, 0x5f98, 0x5fa1, 0x5fa0, 0x5f9c, 0x607f, 0x60a3,
352 0x6089, 0x60a0, 0x60a8, 0x60cb, 0x60b4, 0x60e6, 0x60bd, 0x60c5,
353 0x60bb, 0x60b5, 0x60dc, 0x60bc, 0x60d8, 0x60d5, 0x60c6, 0x60df,
354 0x60b8, 0x60da, 0x60c7, 0x621a, 0x621b, 0x6248, 0x63a0, 0x63a7,
355 0x6372, 0x6396, 0x63a2, 0x63a5, 0x6377, 0x6367, 0x6398, 0x63aa,
356 0x6371, 0x63a9, 0x6389, 0x6383, 0x639b, 0x636b, 0x63a8, 0x6384,
357 0x6388, 0x6399, 0x63a1, 0x63ac, 0x6392, 0x638f, 0x6380, 0x637b,
358 0x6369, 0x6368, 0x637a, 0x655d, 0x6556, 0x6551, 0x6559, 0x6557,
359 0x555f, 0x654f, 0x6558, 0x6555, 0x6554, 0x659c, 0x659b, 0x65ac,
360 0x65cf, 0x65cb, 0x65cc, 0x65ce, 0x665d, 0x665a, 0x6664, 0x6668,
361 0x6666, 0x666e, 0x66f9, 0x52d7, 0x671b, 0x6881, 0x68af, 0x68a2,
362 0x6893, 0x68b5, 0x687f, 0x6876, 0x68b1, 0x68a7, 0x6897, 0x68b0,
363 0x6883, 0x68c4, 0x68ad, 0x6886, 0x6885, 0x6894, 0x689d, 0x68a8,
364 0x689f, 0x68a1, 0x6882, 0x6b32, 0x6bba,
```

```

365 /* 0xb2 */
366 0x6beb, 0x6bec, 0x6c2b, 0x6d8e, 0x6dbc, 0x6df3, 0x6dd9, 0x6db2,
367 0x6de1, 0x6dcc, 0x6de4, 0x6dfb, 0x6dfa, 0x6e05, 0x6dc7, 0x6ccb,
368 0x6daf, 0x6dd1, 0x6dae, 0x6dde, 0x6df9, 0x6db8, 0x6df7, 0x6df5,
369 0x6dc5, 0x6dd2, 0x6e1a, 0x6db5, 0x6dda, 0x6deb, 0x6dd8, 0x6dea,
370 0x6df1, 0x6dee, 0x6de8, 0x6dc6, 0x6dc4, 0x6daa, 0x6dec, 0x6dbf,
371 0x6de6, 0x70f9, 0x7109, 0x710a, 0x70fd, 0x70ef, 0x723d, 0x727d,
372 0x7281, 0x731c, 0x731b, 0x7316, 0x7313, 0x7319, 0x7387, 0x7405,
373 0x740a, 0x7403, 0x7406, 0x73fe, 0x740d, 0x74e0, 0x74f6, 0x74f7,
374 0x751c, 0x7522, 0x7565, 0x7566, 0x7562, 0x7570, 0x758f, 0x75d4,
375 0x75d5, 0x75b5, 0x75ca, 0x75cd, 0x768e, 0x76d4, 0x76d2, 0x76db,
376 0x7737, 0x773e, 0x773c, 0x7736, 0x7738, 0x773a, 0x786b, 0x7843,
377 0x784e, 0x7965, 0x7968, 0x796d, 0x79fb, 0x7a92, 0x7a95, 0x7b20,
378 0x7b28, 0x7b1b, 0x7b2c, 0x7b26, 0x7b19, 0x7b1e, 0x7b2e, 0x7c92,
379 0x7c97, 0x7c95, 0x7d46, 0x7d43, 0x7d43, 0x7d71, 0x7d2e, 0x7d39, 0x7d3c,
380 0x7d40, 0x7d30, 0x7d33, 0x7d44, 0x7d2f, 0x7d42, 0x7d32, 0x7d31,
381 0x7f3d, 0x7f9e, 0x7f9a, 0x7fcc, 0x7fce, 0x7fd2, 0x801c, 0x804a,
382 0x8046, 0x812f, 0x8116, 0x8123, 0x812b, 0x8129, 0x8130, 0x8124,
383 0x8202, 0x8235, 0x8237, 0x8236, 0x8239, 0x838e, 0x839e, 0x8398,
384 0x8378, 0x83a2, 0x8396, 0x83bd, 0x83ab, 0x8392, 0x838a, 0x8393,
385 0x8389, 0x83a0, 0x8377, 0x837b, 0x837c,
386 /* 0xb3 */
387 0x8386, 0x83a7, 0x8655, 0x5f6a, 0x86c7, 0x86c0, 0x86b6, 0x86c4,
388 0x86b5, 0x86c6, 0x86cb, 0x86b1, 0x86af, 0x86c9, 0x8853, 0x889e,
389 0x8888, 0x88ab, 0x8892, 0x8896, 0x888d, 0x888b, 0x8993, 0x898f,
390 0x8a2a, 0x8a1d, 0x8a23, 0x8a25, 0x8a31, 0x8a2d, 0x8a1f, 0x8a1b,
391 0x8a22, 0x8c49, 0x8c5a, 0x8ca9, 0x8cac, 0x8cab, 0x8ca8, 0x8caa,
392 0x8ca7, 0x8d67, 0x8d66, 0x8dbe, 0x8dba, 0x8edb, 0x8edf, 0x9019,
393 0x900d, 0x901a, 0x9017, 0x9023, 0x901f, 0x901d, 0x9010, 0x9015,
394 0x901e, 0x9020, 0x900f, 0x9022, 0x9016, 0x901b, 0x9014, 0x90e8,
395 0x90ed, 0x90fd, 0x9157, 0x91ce, 0x91f5, 0x91e6, 0x91e3, 0x91e7,
396 0x91ed, 0x91e9, 0x9589, 0x966a, 0x9675, 0x9673, 0x9678, 0x9670,
397 0x9674, 0x967f, 0x9677, 0x966c, 0x96c0, 0x96ea, 0x96e9, 0x7ae0,
398 0x7adf, 0x9802, 0x9803, 0x9b5a, 0x9ce5, 0x9e75, 0x9e7f, 0x9ea5,
399 0x9ebb, 0x50a2, 0x508d, 0x5085, 0x5099, 0x5091, 0x5080, 0x5096,
400 0x5098, 0x509a, 0x6700, 0x51f1, 0x5272, 0x5274, 0x5275, 0x5269,
401 0x52de, 0x52dd, 0x52db, 0x535a, 0x53a5, 0x557b, 0x5580, 0x55a7,
402 0x557c, 0x558a, 0x559d, 0x5598, 0x5582, 0x559c, 0x55aa, 0x5594,
403 0x5587, 0x558b, 0x5583, 0x55b3, 0x55ae, 0x559f, 0x553e, 0x55b2,
404 0x559a, 0x55bb, 0x55ac, 0x55b1, 0x557e, 0x5589, 0x55ab, 0x5599,
405 0x570d, 0x582f, 0x582a, 0x5834, 0x5824, 0x5830, 0x5831, 0x5821,
406 0x581d, 0x5820, 0x58f9, 0x58fa, 0x5960,
407 /* 0xb4 */
408 0x5a77, 0x5a9a, 0x5a7f, 0x5a92, 0x5a9b, 0x5aa7, 0x5b73, 0x5b71,
409 0x5bd2, 0x5bcc, 0x5bd3, 0x5bd0, 0x5c0a, 0x5c0b, 0x5c31, 0x5d4c,
410 0x5d50, 0x5d34, 0x5d47, 0x5dfd, 0x5e45, 0x5e3d, 0x5e40, 0x5e43,
411 0x5e7e, 0x5eca, 0x5ec1, 0x5ec2, 0x5ec4, 0x5f3c, 0x5f6d, 0x5fa9,
412 0x5faa, 0x5fa8, 0x60d1, 0x60e1, 0x60b2, 0x60b6, 0x60e0, 0x611c,
413 0x6123, 0x60fa, 0x6115, 0x60f0, 0x60fb, 0x60f4, 0x6168, 0x60f1,
414 0x610e, 0x60f6, 0x6109, 0x6100, 0x6112, 0x621f, 0x6249, 0x63a3,
415 0x638c, 0x63cf, 0x63c0, 0x63e9, 0x63c9, 0x63c6, 0x63cd, 0x63d2,
416 0x63e3, 0x63d0, 0x63e1, 0x63d6, 0x63ed, 0x63ee, 0x6376, 0x63f4,
417 0x63ea, 0x63db, 0x6452, 0x63da, 0x63f9, 0x655e, 0x6566, 0x6562,
418 0x6563, 0x6591, 0x6590, 0x65af, 0x666e, 0x6670, 0x6674, 0x6676,
419 0x666f, 0x6691, 0x667a, 0x667e, 0x6677, 0x66fe, 0x66ff, 0x671f,
420 0x671d, 0x68fa, 0x68d5, 0x68e0, 0x68d8, 0x68d7, 0x6905, 0x68df,
421 0x68f5, 0x68ee, 0x68e7, 0x68f9, 0x68d2, 0x68f2, 0x68e3, 0x68cb,
422 0x68cd, 0x690d, 0x6912, 0x690e, 0x68c9, 0x68da, 0x696e, 0x68fb,
423 0x6b3e, 0x6b3a, 0x6b3d, 0x6b98, 0x6b96, 0x6bbc, 0x6bef, 0x6c2e,
424 0x6c2f, 0x6c2c, 0x6e2f, 0x6e38, 0x6e54, 0x6e21, 0x6e32, 0x6e67,
425 0x6e4a, 0x6e20, 0x6e25, 0x6e23, 0x6e1b, 0x6e5b, 0x6e58, 0x6e24,
426 0x6e56, 0x6e6e, 0x6e2d, 0x6e26, 0x6e6f, 0x6e34, 0x6e4d, 0x6e3a,
427 0x6e2c, 0x6e43, 0x6e1d, 0x6e3e, 0x6ecb,
428 /* 0xb5 */
429 0x6e89, 0x6e19, 0x6e4e, 0x6e63, 0x6e44, 0x6e72, 0x6e69, 0x6e5f,
430 0x7119, 0x711a, 0x7126, 0x7130, 0x7121, 0x7136, 0x716e, 0x711c,
431 0x724c, 0x7284, 0x7280, 0x7336, 0x7325, 0x7334, 0x7329, 0x743a,
432 0x742a, 0x7433, 0x7422, 0x7425, 0x7435, 0x7436, 0x7434, 0x742f,
433 0x741b, 0x7426, 0x7428, 0x7525, 0x7526, 0x756b, 0x756a, 0x75e2,
434 0x75db, 0x75e3, 0x75d9, 0x75d8, 0x75de, 0x75e0, 0x767b, 0x767c,
435 0x7696, 0x7693, 0x76b4, 0x76dc, 0x774f, 0x77ed, 0x785d, 0x786c,
436 0x786f, 0x7a0d, 0x7a08, 0x7a0b, 0x7a05, 0x7a00, 0x7a98, 0x7a97,
437 0x7a96, 0x7ae5, 0x7ae3, 0x7b49, 0x7b56, 0x7b46, 0x7b50, 0x7b52,
438 0x7b54, 0x7b4d, 0x7b4b, 0x7b4f, 0x7b51, 0x7c9f, 0x7ca5, 0x7d5e,
439 0x7d50, 0x7d68, 0x7d55, 0x7d2b, 0x7d6e, 0x7d72, 0x7d61, 0x7d66,
440 0x7d62, 0x7d70, 0x7d73, 0x5584, 0x7fd4, 0x7fd5, 0x800b, 0x8052,
441 0x8085, 0x8155, 0x8154, 0x814b, 0x814e, 0x8151, 0x814e, 0x8139, 0x8146,
442 0x813e, 0x814c, 0x8153, 0x8174, 0x8212, 0x821c, 0x83e9, 0x8403,
443 0x83f8, 0x840d, 0x83e0, 0x83c5, 0x840b, 0x83c1, 0x83ef, 0x83f1,
444 0x83f4, 0x8457, 0x840a, 0x83f0, 0x840c, 0x83cc, 0x83fd, 0x83f2,
445 0x83ca, 0x8438, 0x840e, 0x8404, 0x83dc, 0x8407, 0x83d4, 0x83df,
446 0x865b, 0x86df, 0x86d9, 0x86ed, 0x86d4, 0x86db, 0x86e4, 0x86d0,
447 0x86de, 0x8857, 0x88c1, 0x88c2, 0x88b1, 0x8983, 0x8996, 0x8a3b,
448 0x8a60, 0x8a55, 0x8a5e, 0x8a3c, 0x8a41,
449 /* 0xb6 */
450 0x8a54, 0x8a5b, 0x8a50, 0x8a46, 0x8a34, 0x8a3a, 0x8a36, 0x8a56,
451 0x8c61, 0x8c82, 0x8caf, 0x8cbc, 0x8cb3, 0x8cbd, 0x8cc1, 0x8ccb,

```

```
452 0x8cc0, 0x8cb4, 0x8cb7, 0x8cb6, 0x8cbf, 0x8cb8, 0x8d8a, 0x8d85,
453 0x8d81, 0x8dce, 0x8ddd, 0x8dcb, 0x8dda, 0x8dd1, 0x8dcc, 0x8ddb,
454 0x8dc6, 0x8efb, 0x8ef8, 0x8efc, 0x8f9c, 0x902e, 0x9035, 0x9031,
455 0x9038, 0x9032, 0x9036, 0x9102, 0x90f5, 0x9109, 0x90fe, 0x9163,
456 0x9165, 0x91cf, 0x9214, 0x9215, 0x9223, 0x9209, 0x921e, 0x920d,
457 0x9210, 0x9207, 0x9211, 0x9211, 0x9594, 0x958f, 0x958b, 0x9591, 0x9593,
458 0x9592, 0x958e, 0x968a, 0x968e, 0x968b, 0x967d, 0x9685, 0x9686,
459 0x968d, 0x9672, 0x9684, 0x96c1, 0x96c5, 0x96c4, 0x96c6, 0x96c7,
460 0x96ef, 0x96f2, 0x97cc, 0x9805, 0x9806, 0x9808, 0x98e7, 0x98ea,
461 0x98ef, 0x98e9, 0x98f2, 0x98ed, 0x99ae, 0x99ad, 0x99ec, 0x99cd,
462 0x99ed1, 0x4e82, 0x50ad, 0x50b5, 0x50b2, 0x50b3, 0x50c5, 0x50be,
463 0x50ac, 0x50b7, 0x50bb, 0x50af, 0x50c7, 0x527f, 0x5277, 0x527d,
464 0x52df, 0x52e6, 0x52e4, 0x52e2, 0x52e3, 0x532f, 0x55df, 0x55e8,
465 0x55d3, 0x55e6, 0x55ce, 0x55dc, 0x55c7, 0x55d1, 0x55e3, 0x55e4,
466 0x55ef, 0x55da, 0x55e1, 0x55c5, 0x55c6, 0x55e5, 0x55c9, 0x5712,
467 0x5713, 0x585e, 0x5851, 0x5858, 0x5857, 0x585a, 0x5854, 0x586b,
468 0x584c, 0x586d, 0x584a, 0x5862, 0x5852, 0x584b, 0x5967, 0x5ac1,
469 0x5ac9, 0x5acc, 0x5abe, 0x5abd, 0x5abc,
470 /* 0xb7 */
471 0x5ab3, 0x5ac2, 0x5ab2, 0x5d69, 0x5d6f, 0x5e4c, 0x5e79, 0x5ec9,
472 0x5ec8, 0x5f12, 0x5f59, 0x5fac, 0x5fae, 0x611a, 0x610f, 0x6148,
473 0x611f, 0x60f3, 0x611b, 0x60f9, 0x6101, 0x6108, 0x614e, 0x614c,
474 0x6144, 0x614d, 0x613e, 0x6134, 0x6127, 0x610d, 0x6106, 0x6137,
475 0x6221, 0x6222, 0x6413, 0x643e, 0x641e, 0x642a, 0x642d, 0x643d,
476 0x642c, 0x640f, 0x641c, 0x6414, 0x640d, 0x6436, 0x6416, 0x6417,
477 0x6406, 0x656c, 0x659f, 0x65b0, 0x6697, 0x6689, 0x6687, 0x6688,
478 0x6696, 0x6684, 0x6698, 0x668d, 0x6703, 0x6994, 0x696d, 0x695a,
479 0x6977, 0x6960, 0x6954, 0x6975, 0x6930, 0x6982, 0x694a, 0x6968,
480 0x696b, 0x695e, 0x6953, 0x6979, 0x6986, 0x695d, 0x6963, 0x695b,
481 0x6b47, 0x6b72, 0x6bfc, 0x6bbf, 0x6bd3, 0x6bfd, 0x6ea2, 0x6eaf,
482 0x6ed3, 0x6eb6, 0x6ec2, 0x6e90, 0x6e9d, 0x6ec7, 0x6ec5, 0x6ea5,
483 0x6e98, 0x6ebc, 0x6eba, 0x6eab, 0x6ed1, 0x6e96, 0x6e9c, 0x6ec4,
484 0x6ed4, 0x6eaa, 0x6ea7, 0x6eb4, 0x714e, 0x7159, 0x7169, 0x7164,
485 0x7149, 0x7167, 0x715c, 0x716c, 0x7166, 0x714c, 0x7165, 0x715e,
486 0x7146, 0x7168, 0x7156, 0x723a, 0x7252, 0x7337, 0x7345, 0x733f,
487 0x733e, 0x746f, 0x745a, 0x7455, 0x745f, 0x745e, 0x7441, 0x743f,
488 0x7459, 0x745b, 0x745c, 0x7576, 0x7578, 0x7600, 0x75f0, 0x7601,
489 0x75f2, 0x75f1, 0x75fa, 0x75ff, 0x75f4, 0x75f3, 0x76de, 0x76df,
490 0x775b, 0x776b, 0x7766, 0x775e, 0x7763,
491 /* 0xb8 */
492 0x7779, 0x776a, 0x776c, 0x775c, 0x7765, 0x7768, 0x7762, 0x77ee,
493 0x788e, 0x78b0, 0x7897, 0x7898, 0x788c, 0x7889, 0x787c, 0x7891,
494 0x7893, 0x787f, 0x797a, 0x797f, 0x7981, 0x842c, 0x79bd, 0x7a1c,
495 0x7a1a, 0x7a20, 0x7a14, 0x7a1f, 0x7a1e, 0x7a9f, 0x7aa0, 0x7b77,
496 0x7bc0, 0x7b60, 0x7b6e, 0x7cb7, 0x7cb1, 0x7cb3, 0x7cb5, 0x7d93,
497 0x7d79, 0x7d91, 0x7d81, 0x7d8f, 0x7d5b, 0x7f6e, 0x7f69, 0x7f6a,
498 0x7f72, 0x7fa9, 0x7fa8, 0x7fa4, 0x8056, 0x8058, 0x8086, 0x8084,
499 0x8171, 0x8170, 0x8178, 0x8165, 0x816e, 0x8173, 0x816b, 0x8179,
500 0x817a, 0x8166, 0x8205, 0x8247, 0x8482, 0x8477, 0x843d, 0x8431,
501 0x8475, 0x8466, 0x846b, 0x8449, 0x846c, 0x845b, 0x843c, 0x8435,
502 0x8461, 0x8463, 0x8469, 0x846d, 0x8446, 0x865e, 0x865c, 0x865f,
503 0x86f9, 0x8713, 0x8708, 0x8707, 0x8700, 0x86fe, 0x86fb, 0x8702,
504 0x8703, 0x8706, 0x870a, 0x8859, 0x88df, 0x88d4, 0x88d9, 0x88dc,
505 0x88d8, 0x88dd, 0x88e1, 0x88ca, 0x88d5, 0x88d2, 0x899c, 0x89e3,
506 0x8a6b, 0x8a72, 0x8a73, 0x8a66, 0x8a69, 0x8a70, 0x8a87, 0x8a7c,
507 0x8a63, 0x8aa0, 0x8a71, 0x8a85, 0x8a6d, 0x8a62, 0x8a6e, 0x8a6c,
508 0x8a79, 0x8a7b, 0x8a3e, 0x8a68, 0x8c62, 0x8c8a, 0x8c89, 0x8cca,
509 0x8cc7, 0x8cc8, 0x8cc4, 0x8cb2, 0x8cc3, 0x8cc2, 0x8cc5, 0x8de1,
510 0x8ddf, 0x8de8, 0x8def, 0x8df3, 0x8dfa, 0x8dea, 0x8de4, 0x8de6,
511 0x8eb2, 0x8f03, 0x8f09, 0x8efe, 0x8f0a,
512 /* 0xb9 */
513 0x8f9f, 0x8fb2, 0x904b, 0x904a, 0x9053, 0x9042, 0x9054, 0x903c,
514 0x9055, 0x9050, 0x9047, 0x904f, 0x904e, 0x904d, 0x9051, 0x903e,
515 0x9041, 0x9112, 0x9117, 0x916c, 0x916a, 0x9169, 0x91c9, 0x9237,
516 0x9257, 0x9238, 0x923d, 0x9240, 0x923e, 0x925b, 0x924b, 0x9264,
517 0x9251, 0x9234, 0x9249, 0x924d, 0x9245, 0x9239, 0x923f, 0x925a,
518 0x9598, 0x9698, 0x9694, 0x9695, 0x96cd, 0x96cb, 0x96c9, 0x96ca,
519 0x96f7, 0x96fb, 0x96f9, 0x96f6, 0x9756, 0x9774, 0x9776, 0x9810,
520 0x9811, 0x9813, 0x980a, 0x9812, 0x980c, 0x98fc, 0x98f4, 0x98fd,
521 0x98fe, 0x99b3, 0x99b1, 0x99b4, 0x99ae, 0x99ce, 0x99e2, 0x99fe,
522 0x9f13, 0x9f20, 0x50e7, 0x50ee, 0x50e5, 0x50d6, 0x50ed, 0x50da,
523 0x50d5, 0x50cf, 0x50d1, 0x50f1, 0x50ce, 0x50e9, 0x5162, 0x51f3,
524 0x5283, 0x5282, 0x5331, 0x53ad, 0x55fe, 0x5600, 0x561b, 0x5617,
525 0x55fd, 0x5614, 0x5606, 0x5609, 0x560d, 0x560e, 0x55f7, 0x5616,
526 0x561f, 0x5608, 0x5610, 0x55f6, 0x5718, 0x5716, 0x5875, 0x587e,
527 0x5883, 0x5893, 0x588a, 0x5879, 0x5885, 0x587d, 0x58fd, 0x5925,
528 0x5922, 0x5924, 0x596a, 0x5969, 0x5ae1, 0x5ae6, 0x5ae9, 0x5ad7,
529 0x5ad6, 0x5ad8, 0x5ae3, 0x5b75, 0x5bde, 0x5be7, 0x5be1, 0x5be5,
530 0x5be6, 0x5be8, 0x5be2, 0x5be4, 0x5bdf, 0x5c0d, 0x5c62, 0x5d84,
531 0x5d87, 0x5e5b, 0x5e63, 0x5e55, 0x5e57, 0x5e54, 0x5ed3, 0x5ed6,
532 0x5f0a, 0x5f46, 0x5f70, 0x5fb9, 0x6147,
533 /* 0xba */
534 0x613f, 0x614b, 0x6177, 0x6162, 0x6163, 0x615f, 0x615a, 0x6158,
535 0x6175, 0x622a, 0x6487, 0x6458, 0x6454, 0x64a4, 0x6478, 0x645f,
536 0x647a, 0x6451, 0x6467, 0x6434, 0x646d, 0x647b, 0x6572, 0x65a1,
537 0x65d7, 0x65d6, 0x66a2, 0x66a8, 0x669d, 0x669c, 0x69a8, 0x6995,
538 0x69c1, 0x69ae, 0x69d3, 0x69cb, 0x699b, 0x69b7, 0x69bb, 0x69ab,
```



```
539 0x69b4, 0x69d0, 0x69cd, 0x69ad, 0x69cc, 0x69a6, 0x69c3, 0x69a3,
540 0x6b49, 0x6b4c, 0x6c33, 0x6f33, 0x6f14, 0x6efe, 0x6f13, 0x6ef4,
541 0x6f29, 0x6f3e, 0x6f20, 0x6f2c, 0x6f0f, 0x6f02, 0x6f22, 0x6eff,
542 0x6eef, 0x6f06, 0x6f31, 0x6f38, 0x6f32, 0x6f23, 0x6f15, 0x6f2b,
543 0x6f2f, 0x6f88, 0x6f2a, 0x6eec, 0x6f01, 0x6ef2, 0x6ecc, 0x6ef7,
544 0x7194, 0x7199, 0x719d, 0x717d, 0x718a, 0x7184, 0x7192, 0x723e, 0x7292,
545 0x7296, 0x7344, 0x7350, 0x7464, 0x7463, 0x746a, 0x7470, 0x746d,
546 0x7504, 0x7591, 0x7627, 0x760d, 0x760b, 0x7609, 0x7613, 0x76e1,
547 0x76e3, 0x7784, 0x777d, 0x777f, 0x7761, 0x78c1, 0x789f, 0x78a7,
548 0x78b3, 0x78a9, 0x78a3, 0x798e, 0x798f, 0x798d, 0x7a2e, 0x7a31,
549 0x7aaa, 0x7aa9, 0x7aed, 0x7aef, 0x7ba1, 0x7b95, 0x7b8b, 0x7b75,
550 0x7b97, 0x7b9d, 0x7b9c, 0x7b94, 0x7b8f, 0x7bb8, 0x7b87, 0x7b84, 0x7cb9,
551 0x7cbd, 0x7cbe, 0x7dbb, 0x7db0, 0x7d9c, 0x7dbd, 0x7dbe, 0x7da0,
552 0x7dca, 0x7db4, 0x7db2, 0x7db1, 0x7dba, 0x7da2, 0x7dbf, 0x7db5,
553 0x7db8, 0x7dad, 0x7dad, 0x7dc7, 0x7dac,
554 /* 0xbb */
555 0x7f70, 0x7fe0, 0x7fe1, 0x7fdf, 0x805e, 0x805a, 0x8087, 0x8150,
556 0x8180, 0x818f, 0x8188, 0x818a, 0x817f, 0x8182, 0x81e7, 0x81fa,
557 0x8207, 0x8214, 0x821e, 0x824b, 0x84c9, 0x84bf, 0x84c6, 0x84c4,
558 0x8499, 0x849e, 0x84b2, 0x849c, 0x84cb, 0x84b8, 0x84c0, 0x84d3,
559 0x8490, 0x84bc, 0x84bc, 0x84d1, 0x84ca, 0x873f, 0x871c, 0x873b, 0x8722,
560 0x8725, 0x8734, 0x8718, 0x8755, 0x8737, 0x8729, 0x88f3, 0x8902,
561 0x88f4, 0x88f9, 0x88f8, 0x88fd, 0x88e8, 0x891a, 0x88ef, 0x8aa6,
562 0x8a8c, 0x8a9e, 0x8aa3, 0x8a8d, 0x8aa1, 0x8a93, 0x8aa4, 0x8aaa,
563 0x8aa5, 0x8aa8, 0x8a98, 0x8a91, 0x8a9a, 0x8aa7, 0x8c6a, 0x8c8d,
564 0x8c8c, 0x8cd3, 0x8cd1, 0x8cd2, 0x8d6b, 0x8d99, 0x8d95, 0x8dfc,
565 0x8f14, 0x8f12, 0x8f15, 0x8f13, 0x8fa3, 0x9060, 0x9058, 0x905c,
566 0x9063, 0x9059, 0x905e, 0x9062, 0x905d, 0x905b, 0x9119, 0x9118,
567 0x911e, 0x9175, 0x9178, 0x9177, 0x9174, 0x9278, 0x9280, 0x9285,
568 0x9298, 0x9296, 0x927b, 0x9293, 0x929c, 0x92a8, 0x927c, 0x9291,
569 0x95a1, 0x95a8, 0x95a9, 0x95a3, 0x95a5, 0x95a4, 0x9699, 0x969c,
570 0x969b, 0x96cc, 0x96d2, 0x9700, 0x977c, 0x9785, 0x97f6, 0x9817,
571 0x9818, 0x98af, 0x98b1, 0x9903, 0x9905, 0x990c, 0x9909, 0x99c1,
572 0x9aaf, 0x9ab0, 0x9ae6, 0x9b41, 0x9b42, 0x9cf4, 0x9cf6, 0x9cf3,
573 0x9ebc, 0x9f3b, 0x9f4a, 0x5104, 0x5100, 0x50fb, 0x50f5, 0x50f9,
574 0x5102, 0x5108, 0x5109, 0x5105,
575 /* 0xbc */
576 0x5287, 0x5288, 0x5289, 0x528d, 0x528a, 0x52f0, 0x53b2, 0x562e,
577 0x563b, 0x5639, 0x5632, 0x563f, 0x5634, 0x5629, 0x5653, 0x564e,
578 0x5657, 0x5674, 0x5636, 0x562f, 0x5630, 0x5880, 0x589f, 0x589e,
579 0x58b3, 0x589c, 0x58ae, 0x58a9, 0x58a6, 0x596d, 0x5b09, 0x5afb,
580 0x5b0b, 0x5af5, 0x5b0c, 0x5b08, 0x5bee, 0x5bec, 0x5be9, 0x5beb,
581 0x5c64, 0x5c65, 0x5d9d, 0x5d94, 0x5e62, 0x5e5f, 0x5e61, 0x5ee2,
582 0x5eda, 0x5edf, 0x5edd, 0x5ee3, 0x5ee0, 0x5f48, 0x5f71, 0x5fb7,
583 0x5fb5, 0x6176, 0x6167, 0x616e, 0x615d, 0x6155, 0x6182, 0x617c,
584 0x6170, 0x616b, 0x617e, 0x61a7, 0x6190, 0x61ab, 0x618e, 0x61ac,
585 0x619a, 0x61a4, 0x6194, 0x61ae, 0x622e, 0x6469, 0x646f, 0x6479,
586 0x649e, 0x64b2, 0x6488, 0x6490, 0x64b0, 0x64a5, 0x6493, 0x6495,
587 0x64a9, 0x6492, 0x64ae, 0x64ad, 0x64ab, 0x649a, 0x64ac, 0x6499,
588 0x64a2, 0x64b3, 0x6575, 0x6577, 0x6578, 0x66ae, 0x66ab, 0x66b4,
589 0x66b1, 0x6a23, 0x6a1f, 0x69e8, 0x6a01, 0x6a1e, 0x6a19, 0x69fd,
590 0x6a21, 0x6a13, 0x6a0a, 0x69f3, 0x6a02, 0x6a05, 0x69ed, 0x6a11,
591 0x6b50, 0x6b4e, 0x6ba4, 0x6bc5, 0x6bc6, 0x6f3f, 0x6f7c, 0x6f84,
592 0x6f51, 0x6f66, 0x6f54, 0x6f86, 0x6f6d, 0x6f5b, 0x6f78, 0x6f6e,
593 0x6f8e, 0x6f7a, 0x6f70, 0x6f64, 0x6f97, 0x6f58, 0x6ed5, 0x6f6f,
594 0x6f60, 0x6f5f, 0x719f, 0x71ac, 0x71b1, 0x71a8, 0x7256, 0x729b,
595 0x734e, 0x7357, 0x7469, 0x748b, 0x7483,
596 /* 0xbd */
597 0x747e, 0x7480, 0x757f, 0x7620, 0x7629, 0x761f, 0x7624, 0x7626,
598 0x7621, 0x7622, 0x769a, 0x76ba, 0x76e4, 0x778e, 0x7787, 0x778c,
599 0x7791, 0x778b, 0x78cb, 0x78c5, 0x78ba, 0x78ca, 0x78be, 0x78d5,
600 0x78bc, 0x78d0, 0x7a3f, 0x7a3c, 0x7a40, 0x7a3d, 0x7a37, 0x7a3b,
601 0x7aaf, 0x7aae, 0x7bad, 0x7bb1, 0x7bc4, 0x7bb4, 0x7bc6, 0x7bc7,
602 0x7bc1, 0x7ba0, 0x7bcc, 0x7cca, 0x7de0, 0x7df4, 0x7def, 0x7dfb,
603 0x7dd8, 0x7dec, 0x7ddd, 0x7de8, 0x7de3, 0x7dda, 0x7dde, 0x7de9,
604 0x7d9e, 0x7dd9, 0x7df2, 0x7df9, 0x7f75, 0x7f77, 0x7faf, 0x7fe9,
605 0x8026, 0x819b, 0x819c, 0x819d, 0x81a0, 0x819a, 0x8198, 0x8517,
606 0x853d, 0x851a, 0x84ee, 0x852c, 0x852d, 0x8513, 0x8511, 0x8523,
607 0x8521, 0x8514, 0x84ec, 0x8525, 0x84ff, 0x8506, 0x8782, 0x8774,
608 0x8776, 0x8760, 0x8766, 0x8778, 0x8768, 0x8759, 0x8757, 0x874c,
609 0x8753, 0x885b, 0x885d, 0x8910, 0x8907, 0x8912, 0x8913, 0x8915,
610 0x890a, 0x8abc, 0x8ad2, 0x8ac7, 0x8ac4, 0x8a95, 0x8acb, 0x8af8,
611 0x8ab2, 0x8ac9, 0x8ac2, 0x8abf, 0x8ab0, 0x8ad6, 0x8acd, 0x8ab6,
612 0x8ab9, 0x8adb, 0x8c4c, 0x8c4e, 0x8c6c, 0x8ce0, 0x8cde, 0x8ce6,
613 0x8ce4, 0x8cec, 0x8ced, 0x8ce2, 0x8ce3, 0x8cdc, 0x8cea, 0x8ce1,
614 0x8d6d, 0x8d9f, 0x8da3, 0x8e2b, 0x8e10, 0x8e1d, 0x8e22, 0x8e0f,
615 0x8e29, 0x8e1f, 0x8e21, 0x8e1e, 0x8eba, 0x8f1d, 0x8f1b, 0x8f1f,
616 0x8f29, 0x8f26, 0x8f2a, 0x8f1c, 0x8f1e,
617 /* 0xbe */
618 0x8f25, 0x9069, 0x906e, 0x9068, 0x906d, 0x9077, 0x9130, 0x912d,
619 0x9127, 0x9131, 0x9187, 0x9189, 0x918b, 0x9183, 0x92c5, 0x92bb,
620 0x92b7, 0x92ea, 0x92ac, 0x92e4, 0x92c1, 0x92b3, 0x92bc, 0x92d2,
621 0x92c7, 0x92f0, 0x92b2, 0x95ad, 0x95b1, 0x9704, 0x9706, 0x9707,
622 0x9709, 0x9760, 0x978d, 0x978b, 0x978f, 0x9821, 0x982b, 0x981c,
623 0x98b3, 0x990a, 0x9913, 0x9912, 0x9918, 0x99dd, 0x99d0, 0x99df,
624 0x99db, 0x99d1, 0x99d5, 0x99d2, 0x99d9, 0x99ab7, 0x9aee, 0x9aef,
625 0x9b27, 0x9b45, 0x9b44, 0x9b77, 0x9b6f, 0x9d06, 0x9d09, 0x9d03,
```

```

626 0x9ea9, 0x9ebe, 0x9ece, 0x58a8, 0x9f52, 0x5112, 0x5118, 0x5114,
627 0x5110, 0x5115, 0x5180, 0x51aa, 0x51dd, 0x5291, 0x5293, 0x52f3,
628 0x5659, 0x566b, 0x5679, 0x5669, 0x5664, 0x5678, 0x566a, 0x5668,
629 0x5665, 0x5671, 0x566f, 0x566c, 0x5662, 0x5676, 0x58c1, 0x58be,
630 0x58c7, 0x58c5, 0x596e, 0x5b1d, 0x5b34, 0x5b78, 0x5bf0, 0x5c0e,
631 0x5f4a, 0x61b2, 0x6191, 0x61a9, 0x618a, 0x61cd, 0x61b6, 0x61be,
632 0x61ca, 0x61c8, 0x6230, 0x64c5, 0x64c1, 0x64cb, 0x64bb, 0x64bc,
633 0x64da, 0x64c4, 0x64c7, 0x64c2, 0x64cd, 0x64bf, 0x64d2, 0x64d4,
634 0x64be, 0x6574, 0x66c6, 0x66c9, 0x66b9, 0x66c4, 0x66c7, 0x66b8,
635 0x6a3d, 0x6a38, 0x6a3a, 0x6a59, 0x6a6b, 0x6a58, 0x6a39, 0x6a44,
636 0x6a62, 0x6a61, 0x6a4b, 0x6a47, 0x6a35, 0x6a5f, 0x6a48, 0x6b59,
637 0x6b77, 0x6c05, 0x6fc2, 0x6fb1, 0x6fa1,
638 /* 0xbf */
639 0x6fc3, 0x6fa4, 0x6fc1, 0x6fa7, 0x6fb3, 0x6fc0, 0x6fb9, 0x6fb6,
640 0x6fa6, 0x6fa0, 0x6fb4, 0x71be, 0x71c9, 0x71d0, 0x71d2, 0x71c8,
641 0x71d5, 0x71b9, 0x71ce, 0x71d9, 0x71dc, 0x71c3, 0x71c4, 0x7368,
642 0x749c, 0x74a3, 0x7498, 0x749f, 0x749e, 0x74e2, 0x750c, 0x750d,
643 0x7634, 0x7638, 0x7638, 0x763a, 0x76e7, 0x76e5, 0x77a0, 0x779e, 0x779f,
644 0x77a5, 0x78e8, 0x78da, 0x78ec, 0x78e7, 0x79a6, 0x7a4d, 0x7a4e,
645 0x7a46, 0x7a4c, 0x7a4b, 0x7aba, 0x7bd9, 0x7c11, 0x7bc9, 0x7be4,
646 0x7bdb, 0x7be1, 0x7be9, 0x7be6, 0x7cd5, 0x7cd6, 0x7e0a, 0x7e11,
647 0x7e08, 0x7e1b, 0x7e23, 0x7e1e, 0x7e1d, 0x7e09, 0x7e10, 0x7f79,
648 0x7fb2, 0x7ff0, 0x7ff1, 0x7fee, 0x8028, 0x81b3, 0x81a9, 0x81a8,
649 0x81fb, 0x8208, 0x8258, 0x8259, 0x854a, 0x8559, 0x8548, 0x8568,
650 0x8569, 0x8543, 0x8549, 0x856d, 0x856a, 0x855e, 0x8783, 0x879f,
651 0x879e, 0x87a2, 0x878d, 0x8861, 0x892a, 0x8932, 0x8925, 0x892b,
652 0x8921, 0x89aa, 0x89a6, 0x8ae6, 0x8aef, 0x8aeb, 0x8af1, 0x8b00,
653 0x8adc, 0x8ae7, 0x8aee, 0x8afe, 0x8b01, 0x8b02, 0x8af7, 0x8aed,
654 0x8af3, 0x8af6, 0x8afc, 0x8cb6, 0x8c6d, 0x8c93, 0x8cf4, 0x8e44,
655 0x8e31, 0x8e34, 0x8e42, 0x8e39, 0x8e35, 0x8f3b, 0x8f2f, 0x8f38,
656 0x8f33, 0x8fa8, 0x8fa6, 0x9075, 0x9074, 0x9078, 0x9072, 0x907c,
657 0x907a, 0x9134, 0x9192, 0x9320, 0x9336, 0x92f8, 0x9333, 0x932f,
658 0x9322, 0x92fc, 0x932b, 0x9304, 0x931a,
659 /* 0xc0 */
660 0x9310, 0x9326, 0x9321, 0x9315, 0x932e, 0x9319, 0x95bb, 0x96a7,
661 0x96a8, 0x96aa, 0x96d5, 0x970e, 0x9711, 0x9716, 0x970d, 0x9713,
662 0x970f, 0x975b, 0x975c, 0x9766, 0x9798, 0x9830, 0x9838, 0x983b,
663 0x9837, 0x982d, 0x9839, 0x9824, 0x9910, 0x9928, 0x991e, 0x991b,
664 0x9921, 0x991a, 0x99ed, 0x99e2, 0x99f1, 0x99ab, 0x99ab, 0x99af,
665 0x9aed, 0x9b28, 0x9b91, 0x9d15, 0x9d23, 0x9d26, 0x9d28, 0x9d12,
666 0x9d1b, 0x9ed8, 0x9ed4, 0x9f8d, 0x9f9c, 0x512a, 0x511f, 0x5121,
667 0x5132, 0x52f5, 0x568e, 0x5680, 0x5690, 0x5685, 0x5687, 0x568f,
668 0x58d5, 0x58d3, 0x58d1, 0x58ce, 0x5b30, 0x5b2a, 0x5b24, 0x5b7a,
669 0x5c37, 0x5c68, 0x5dbc, 0x5dba, 0x5dbd, 0x5db8, 0x5e6b, 0x5f4c,
670 0x5fbd, 0x61c9, 0x61c2, 0x61c7, 0x61e6, 0x61cb, 0x6232, 0x6234,
671 0x64ce, 0x64ca, 0x64d8, 0x64e0, 0x64f0, 0x64e6, 0x64ec, 0x64f1,
672 0x64e2, 0x64ed, 0x6582, 0x6583, 0x66d9, 0x66d6, 0x6a80, 0x6a94,
673 0x6a84, 0x6aa2, 0x6aa9, 0x6adb, 0x6aa3, 0x6a7e, 0x6a97, 0x6a90,
674 0x6aa0, 0x6b5c, 0x6bae, 0x6bda, 0x6c08, 0x6fd8, 0x6ff1, 0x6fdf,
675 0x6fe0, 0x6fdb, 0x6fe4, 0x6feb, 0x6fef, 0x6f80, 0x6fec, 0x6fe1,
676 0x6fe9, 0x6fd5, 0x6fee, 0x6ff0, 0x71e7, 0x71d7, 0x71ee, 0x71e6,
677 0x71e5, 0x71ed, 0x71ec, 0x71f4, 0x71e0, 0x7235, 0x7246, 0x7370,
678 0x7372, 0x74a9, 0x74b0, 0x74a6, 0x74a8, 0x7646, 0x7642, 0x764c,
679 0x76ea, 0x77b3, 0x77aa, 0x77b0, 0x77ac,
680 /* 0xc1 */
681 0x77a7, 0x77ad, 0x77ef, 0x78f7, 0x78fa, 0x78f4, 0x78ef, 0x7901,
682 0x79a7, 0x79aa, 0x7a57, 0x7abf, 0x7c07, 0x7c0d, 0x7bfe, 0x7bf7,
683 0x7c0c, 0x7be0, 0x7ce0, 0x7cdc, 0x7cde, 0x7ce2, 0x7cdf, 0x7cd9,
684 0x7cdd, 0x7e2e, 0x7e3e, 0x7e46, 0x7e37, 0x7e32, 0x7e43, 0x7e2b,
685 0x7e3d, 0x7e31, 0x7e45, 0x7e41, 0x7e34, 0x7e39, 0x7e48, 0x7e35,
686 0x7e3f, 0x7e2f, 0x7f44, 0x7ff3, 0x7ffc, 0x8071, 0x8072, 0x8070,
687 0x806f, 0x8073, 0x81c6, 0x81c3, 0x81ba, 0x81c2, 0x81c0, 0x81bf,
688 0x81bd, 0x81c9, 0x81be, 0x81e8, 0x8209, 0x8271, 0x85aa, 0x8584,
689 0x857e, 0x859c, 0x8591, 0x8594, 0x85af, 0x859b, 0x8587, 0x85a8,
690 0x858a, 0x8667, 0x87c0, 0x87d1, 0x87b3, 0x87d2, 0x87c6, 0x87ab,
691 0x87bb, 0x87ba, 0x87c8, 0x87cb, 0x893b, 0x8936, 0x8944, 0x8938,
692 0x893d, 0x89ac, 0x8b0e, 0x8b17, 0x8b19, 0x8b1b, 0x8b0a, 0x8b20,
693 0x8b1d, 0x8b04, 0x8b10, 0x8c41, 0x8c3f, 0x8c73, 0x8cfa, 0x8cfd,
694 0x8cfc, 0x8cf8, 0x8cfb, 0x8da8, 0x8e49, 0x8e4b, 0x8e48, 0x8e4a,
695 0x8f44, 0x8f3e, 0x8f42, 0x8f45, 0x8f3f, 0x907f, 0x907d, 0x9084,
696 0x9081, 0x9082, 0x9080, 0x9139, 0x91a3, 0x919e, 0x919c, 0x934d,
697 0x9382, 0x9328, 0x9375, 0x934a, 0x9365, 0x934b, 0x9318, 0x937e,
698 0x936c, 0x935b, 0x9370, 0x935a, 0x9354, 0x95ca, 0x95cb, 0x95cc,
699 0x95c8, 0x95c6, 0x96b1, 0x96b8, 0x96d6, 0x971c, 0x971e, 0x97a0,
700 0x97d3, 0x9846, 0x98b6, 0x9935, 0x9a01,
701 /* 0xc2 */
702 0x99ff, 0x9bae, 0x9bab, 0x9baa, 0x9bad, 0x9d3b, 0x9d3f, 0x9e8b,
703 0x9ecf, 0x9ede, 0x9edc, 0x9edd, 0x9edb, 0x9f3e, 0x9f4b, 0x53e2,
704 0x5695, 0x56ae, 0x58d9, 0x58d8, 0x5b38, 0x5f5d, 0x61e3, 0x6233,
705 0x64f4, 0x64f2, 0x64fe, 0x6506, 0x64fa, 0x64fb, 0x64f7, 0x65b7,
706 0x66dc, 0x6726, 0x6ab3, 0x6aac, 0x6ac3, 0x6abb, 0x6ab8, 0x6ac2,
707 0x6aae, 0x6aaf, 0x6b5f, 0x6b78, 0x6baf, 0x7009, 0x700b, 0x6ffe,
708 0x7006, 0x6ffa, 0x7011, 0x700f, 0x71fb, 0x71fc, 0x71fe, 0x71f8,
709 0x7377, 0x7375, 0x74a7, 0x74bf, 0x7515, 0x7656, 0x7658, 0x7652,
710 0x77bd, 0x77bf, 0x77bb, 0x77bc, 0x790e, 0x79ae, 0x7a61, 0x7a62,
711 0x7a60, 0x7ac4, 0x7ac5, 0x7c2b, 0x7c27, 0x7c2a, 0x7c1e, 0x7c23,
712 0x7c21, 0x7ce7, 0x7e54, 0x7e55, 0x7e5e, 0x7e5a, 0x7e61, 0x7e52,

```

```
713 0x7e59, 0x7f48, 0x7ff9, 0x7ffb, 0x8077, 0x8076, 0x81cd, 0x81cf,
714 0x820a, 0x85cf, 0x85a9, 0x85cd, 0x85d0, 0x85c9, 0x85b0, 0x85ba,
715 0x85b9, 0x85a6, 0x87ef, 0x87ec, 0x87f2, 0x87e0, 0x8986, 0x89b2,
716 0x89f4, 0x8b28, 0x8b39, 0x8b2c, 0x8b2b, 0x8c50, 0x8d05, 0x8e59,
717 0x8e63, 0x8e66, 0x8e64, 0x8e5f, 0x8e55, 0x8ec0, 0x8f49, 0x8f4d,
718 0x9087, 0x9083, 0x9088, 0x91ab, 0x91ac, 0x91d0, 0x9394, 0x938a,
719 0x9396, 0x93a2, 0x93b3, 0x93ae, 0x93ac, 0x93b0, 0x9398, 0x939a,
720 0x9397, 0x95d4, 0x95d6, 0x95d0, 0x95d5, 0x96e2, 0x96dc, 0x96d9,
721 0x96db, 0x96de, 0x9724, 0x97a3, 0x97a6,
722 /* 0xc3 */
723 0x97ad, 0x97f9, 0x984d, 0x984f, 0x984c, 0x984e, 0x9853, 0x98ba,
724 0x993e, 0x993f, 0x993d, 0x992e, 0x99a5, 0x9a0e, 0x9ac1, 0x9b03,
725 0x9b06, 0x9b4f, 0x9b4e, 0x9b4d, 0x9bca, 0x9bc9, 0x9bfd, 0x9bc8,
726 0x9bc0, 0x9d51, 0x9d5d, 0x9d60, 0x9ee0, 0x9f15, 0x9f2c, 0x5133,
727 0x56a5, 0x58de, 0x58df, 0x58e2, 0x5bf5, 0x9f90, 0x5eec, 0x61f2,
728 0x61f7, 0x61f6, 0x61f5, 0x6500, 0x650f, 0x66e0, 0x66dd, 0x6ae5,
729 0x6add, 0x6ada, 0x6ad3, 0x701b, 0x701f, 0x7028, 0x701a, 0x701d,
730 0x7015, 0x7018, 0x7206, 0x720d, 0x7258, 0x72a2, 0x7378, 0x737a,
731 0x74bd, 0x74ca, 0x74e3, 0x7587, 0x7586, 0x765f, 0x7661, 0x77c7,
732 0x7919, 0x79b1, 0x7a6b, 0x7a69, 0x7c3e, 0x7c3f, 0x7c38, 0x7c3d,
733 0x7c37, 0x7c40, 0x7e6b, 0x7e6d, 0x7e79, 0x7e69, 0x7e6a, 0x7f85,
734 0x7e73, 0x7fb6, 0x7fb9, 0x7fb8, 0x81d8, 0x85e9, 0x85dd, 0x85ea,
735 0x85d5, 0x85e4, 0x85e5, 0x85f7, 0x87fb, 0x8805, 0x880d, 0x87f9,
736 0x87fe, 0x8960, 0x895f, 0x8956, 0x895e, 0x8b41, 0x8b5c, 0x8b58,
737 0x8b49, 0x8b5a, 0x8b4e, 0x8b4f, 0x8b46, 0x8b59, 0x8d08, 0x8d0a,
738 0x8e7c, 0x8e72, 0x8e87, 0x8e76, 0x8e6c, 0x8e7a, 0x8e74, 0x8f54,
739 0x8f4e, 0x8fad, 0x908a, 0x908b, 0x91b1, 0x91ae, 0x931e, 0x93d1,
740 0x93df, 0x93c3, 0x93c8, 0x93dc, 0x93dd, 0x93d6, 0x93e2, 0x93cd,
741 0x93d8, 0x93e4, 0x93d7, 0x93e8, 0x95dc, 0x96b4, 0x96e3, 0x972a,
742 0x9727, 0x9761, 0x97dc, 0x97fb, 0x985e,
743 /* 0xc4 */
744 0x9858, 0x985b, 0x98bc, 0x9945, 0x9949, 0x9a16, 0x9a19, 0x9b0d,
745 0x9be8, 0x9be7, 0x9bd6, 0x9bdb, 0x9d89, 0x9d61, 0x9d72, 0x9d6a,
746 0x9d6c, 0x9e92, 0x9e97, 0x9e93, 0x9e94, 0x52f8, 0x56a8, 0x56b7,
747 0x56b6, 0x56b4, 0x56bc, 0x58e4, 0x5b40, 0x5b43, 0x5b7d, 0x5b7f,
748 0x5dc9, 0x61f8, 0x61fa, 0x61fb, 0x6518, 0x6519, 0x66e6, 0x6727,
749 0x6aec, 0x703e, 0x7030, 0x7032, 0x7210, 0x737b, 0x74fc, 0x7662,
750 0x7665, 0x7926, 0x792a, 0x792c, 0x792b, 0x7ac7, 0x7af6, 0x7c4c,
751 0x7c43, 0x7c4d, 0x7cef, 0x7cf0, 0x8fae, 0x7e7d, 0x7e7c, 0x7e82,
752 0x7f4c, 0x8000, 0x81da, 0x8266, 0x85fb, 0x85f9, 0x8611, 0x85fa,
753 0x8606, 0x860b, 0x8607, 0x860a, 0x8814, 0x8815, 0x8964, 0x89ba,
754 0x89f8, 0x8b70, 0x8b6c, 0x8b66, 0x8b6f, 0x8b5f, 0x8b6b, 0x8d0f,
755 0x8d0d, 0x8e89, 0x8e81, 0x8e85, 0x8e82, 0x91b4, 0x91cb, 0x9418,
756 0x9403, 0x93fd, 0x95e1, 0x9730, 0x98c4, 0x9952, 0x9951, 0x99a8,
757 0x9a2b, 0x9a30, 0x9a37, 0x9a35, 0x9c13, 0x9c0d, 0x9e79, 0x9eb5,
758 0x9ee8, 0x9f2f, 0x9f5f, 0x9f63, 0x9f61, 0x5137, 0x5138, 0x56c1,
759 0x56c0, 0x56c2, 0x5914, 0x5c6c, 0x5dcd, 0x61fc, 0x61fe, 0x651d,
760 0x651c, 0x6595, 0x66e9, 0x6afb, 0x6b04, 0x6afa, 0x6bb2, 0x704c,
761 0x721b, 0x72a7, 0x74d6, 0x74d4, 0x7669, 0x77d3, 0x7c50, 0x7e8f,
762 0x7e8c, 0x7fbc, 0x8617, 0x862d, 0x861a, 0x8823, 0x8822, 0x8821,
763 0x881f, 0x896a, 0x896c, 0x89bd, 0x8b74,
764 /* 0xc5 */
765 0x8b77, 0x8b7d, 0x8d13, 0x8e8a, 0x8e8d, 0x8e8b, 0x8f5f, 0x8faf,
766 0x91ba, 0x942e, 0x9433, 0x9435, 0x943a, 0x9438, 0x9432, 0x942b,
767 0x95e2, 0x9738, 0x9739, 0x9732, 0x97ff, 0x9867, 0x9865, 0x9957,
768 0x9a45, 0x9a43, 0x9a40, 0x9a3e, 0x9acf, 0x9b54, 0x9b51, 0x9c2d,
769 0x9c25, 0x9daf, 0x9db4, 0x9dc2, 0x9db8, 0x9e9d, 0x9eef, 0x9f19,
770 0x9f5c, 0x9f66, 0x9f67, 0x513c, 0x513b, 0x56c8, 0x56ca, 0x56c9,
771 0x5b7f, 0x5dd4, 0x5dd2, 0x5f4e, 0x61ff, 0x6524, 0x6b0a, 0x6b61,
772 0x7051, 0x7058, 0x7380, 0x74e4, 0x758a, 0x766e, 0x766c, 0x79b3,
773 0x7c60, 0x7c5f, 0x807e, 0x807d, 0x81df, 0x8972, 0x896f, 0x89fc,
774 0x8b80, 0x8d16, 0x8d17, 0x8e91, 0x8e93, 0x8f61, 0x9148, 0x9444,
775 0x9451, 0x9452, 0x973d, 0x973e, 0x97c3, 0x97c1, 0x986b, 0x9955,
776 0x9a55, 0x9a4d, 0x9ad2, 0x9b1a, 0x9c49, 0x9c31, 0x9c3e, 0x9c3b,
777 0x9dd3, 0x9dd7, 0x9f34, 0x9f6c, 0x9f6a, 0x9f94, 0x56cc, 0x5dd6,
778 0x6200, 0x6523, 0x652b, 0x652a, 0x66ec, 0x6b10, 0x74da, 0x7aca,
779 0x7c64, 0x7c63, 0x7c65, 0x7e93, 0x7e96, 0x7e94, 0x81e2, 0x8638,
780 0x863f, 0x8831, 0x8b8a, 0x9090, 0x908f, 0x9463, 0x9460, 0x9464,
781 0x9768, 0x986f, 0x995c, 0x9a5a, 0x9a5b, 0x9a57, 0x9ad3, 0x9ad4,
782 0x9ad1, 0x9c54, 0x9c57, 0x9c56, 0x9de5, 0x9e9f, 0x9ef4, 0x56d1,
783 0x58e9, 0x652c, 0x705e, 0x7671, 0x7672, 0x77d7, 0x7f50, 0x7f88,
784 0x8836, 0x8839, 0x8862, 0x8b93, 0x8b92,
785 /* 0xc6 */
786 0x8b96, 0x8277, 0x8d1b, 0x91c0, 0x946a, 0x9742, 0x9748, 0x9744,
787 0x97c6, 0x9870, 0x9a5f, 0x9b22, 0x9b58, 0x9c5f, 0x9df9, 0x9dfa,
788 0x9e7c, 0x9e7d, 0x9f07, 0x9f77, 0x9f72, 0x5ef3, 0x6b16, 0x7063,
789 0x7c6c, 0x7c6e, 0x883b, 0x89c0, 0x8ea1, 0x91c1, 0x9472, 0x9470,
790 0x9871, 0x995e, 0x9ad6, 0x9b23, 0x9ecc, 0x7064, 0x77da, 0x8b9a,
791 0x9477, 0x97c9, 0x9a62, 0x9a65, 0x7e9c, 0x8b9c, 0x8eaa, 0x91c5,
792 0x947d, 0x947e, 0x947c, 0x947c, 0x9c77, 0x9c78, 0x9ef7, 0x8c54, 0x947f,
793 0x9e1a, 0x7228, 0x9a6a, 0x9b31, 0x9e1b, 0x9e1e, 0x7c72, 0x30fe,
794 0x309d, 0x309e, 0x3005, 0x3041, 0x3042, 0x3043, 0x3044, 0x3045,
795 0x3046, 0x3047, 0x3048, 0x3049, 0x304a, 0x304b, 0x304c, 0x304d,
796 0x304e, 0x304f, 0x3050, 0x3051, 0x3052, 0x3053, 0x3054, 0x3055,
797 0x3056, 0x3057, 0x3058, 0x3059, 0x305a, 0x305b, 0x305c, 0x305d,
798 0x305e, 0x305f, 0x3060, 0x3061, 0x3062, 0x3063, 0x3064, 0x3065,
799 0x3066, 0x3067, 0x3068, 0x3069, 0x306a, 0x306b, 0x306c, 0x306d,
```

```

800 0x306e, 0x306f, 0x3070, 0x3071, 0x3072, 0x3073, 0x3074, 0x3075,
801 0x3076, 0x3077, 0x3078, 0x3079, 0x307a, 0x307b, 0x307c, 0x307d,
802 0x307e, 0x307f, 0x3080, 0x3081, 0x3082, 0x3083, 0x3084, 0x3085,
803 0x3086, 0x3087, 0x3088, 0x3089, 0x308a, 0x308b, 0x308c, 0x308d,
804 0x308e, 0x308f, 0x3090, 0x3091, 0x3092, 0x3093, 0x30a1, 0x30a2,
805 0x30a3, 0x30a4, 0x30a5, 0x30a6, 0x30a7,
806 /* 0xc7 */
807 0x30a8, 0x30a9, 0x30aa, 0x30ab, 0x30ac, 0x30ad, 0x30ae, 0x30af,
808 0x30b0, 0x30b1, 0x30b2, 0x30b3, 0x30b4, 0x30b5, 0x30b6, 0x30b7,
809 0x30b8, 0x30b9, 0x30ba, 0x30bb, 0x30bc, 0x30bd, 0x30be, 0x30bf,
810 0x30c0, 0x30c1, 0x30c2, 0x30c3, 0x30c4, 0x30c5, 0x30c6, 0x30c7,
811 0x30c8, 0x30c9, 0x30ca, 0x30cb, 0x30cc, 0x30cd, 0x30ce, 0x30cf,
812 0x30d0, 0x30d1, 0x30d2, 0x30d3, 0x30d4, 0x30d5, 0x30d6, 0x30d7,
813 0x30d8, 0x30d9, 0x30da, 0x30db, 0x30dc, 0x30dd, 0x30de, 0x30df,
814 0x30e0, 0x30e1, 0x30e2, 0x30e3, 0x30e4, 0x30e5, 0x30e6, 0x30e7,
815 0x30e8, 0x30e9, 0x30ea, 0x30eb, 0x30ec, 0x30ed, 0x30ee, 0x30ef,
816 0x30f0, 0x30f1, 0x30f2, 0x30f3, 0x30f4, 0x30f5, 0x30f6, 0x0414,
817 0x0415, 0x0416, 0x0417, 0x0418, 0x0419, 0x041a, 0x041b,
818 0x041c, 0x0423, 0x0424, 0x0425, 0x0426, 0x0427, 0x0428, 0x0429,
819 0x042a, 0x042b, 0x042c, 0x042d, 0x042e, 0x042f, 0x0430, 0x0431,
820 0x0432, 0x0433, 0x0434, 0x0435, 0x0436, 0x0437, 0x0438,
821 0x0439, 0x043a, 0x043b, 0x043c, 0x043d, 0x043e, 0x043f, 0x0440,
822 0x0441, 0x0442, 0x0443, 0x0444, 0x0445, 0x0446, 0x0447, 0x0448,
823 0x0449, 0x044a, 0x044b, 0x044c, 0x044d, 0x044e, 0x044f, 0x2460,
824 0x2461, 0x2462, 0x2463, 0x2464, 0x2465, 0x2466, 0x2467, 0x2468,
825 0x2469, 0x2474, 0x2475, 0x2476, 0x2477, 0x2478, 0x2479, 0x247a,
826 0x247b, 0x247c, 0x247d,
827 };
828 static const unsigned short big5_2uni_pagec9[7652] = {
829 /* 0xc9 */
830 0x4e42, 0x4e5c, 0x51f5, 0x531a, 0x5382, 0x4e07, 0x4e0c, 0x4e47,
831 0x4e8d, 0x56d7, 0xfa0c, 0x5c6e, 0x5f73, 0x4e0f, 0x5187, 0x4e0e,
832 0x4e2e, 0x4e93, 0x4ec2, 0x4ec9, 0x4ec8, 0x5198, 0x52fc, 0x536c,
833 0x53b9, 0x5720, 0x5903, 0x592c, 0x5c10, 0x5dff, 0x65e1, 0x6bb3,
834 0x6bcc, 0x6c14, 0x723f, 0x4e31, 0x4e3c, 0x4ee8, 0x4edc, 0x4ee9,
835 0x4ee1, 0x4edd, 0x4eda, 0x520c, 0x531c, 0x534c, 0x5722, 0x5723,
836 0x5917, 0x592f, 0x5b81, 0x5b84, 0x5c12, 0x5c3b, 0x5c74, 0x5c73,
837 0x5e04, 0x5e80, 0x5e82, 0x5fc9, 0x6209, 0x6250, 0x6c15, 0x6c36,
838 0x6c43, 0x6c3f, 0x6c3b, 0x72ae, 0x72b0, 0x738a, 0x79b8, 0x808a,
839 0x961e, 0x4f0e, 0x4f18, 0x4f2c, 0x4ef5, 0x4f14, 0x4ef1, 0x4f00,
840 0x4ef7, 0x4f08, 0x4f1d, 0x4f02, 0x4f05, 0x4f22, 0x4f13, 0x4f04,
841 0x4ef4, 0x4f12, 0x51b1, 0x5213, 0x5209, 0x5210, 0x52a6, 0x5322,
842 0x531f, 0x534d, 0x538a, 0x5407, 0x56e1, 0x56df, 0x572e, 0x572a,
843 0x5734, 0x593c, 0x5980, 0x597c, 0x5985, 0x597b, 0x597e, 0x5977,
844 0x597f, 0x5b56, 0x5c15, 0x5c25, 0x5c7c, 0x5c7a, 0x5c7b, 0x5c7e,
845 0x5ddf, 0x5e75, 0x5e84, 0x5f02, 0x5f1a, 0x5f74, 0x5fd5, 0x5fd4,
846 0x5fcf, 0x625c, 0x625e, 0x6264, 0x6261, 0x6266, 0x6262, 0x6259,
847 0x6260, 0x625a, 0x6265, 0x65ef, 0x65ee, 0x673e, 0x6739, 0x6738,
848 0x673b, 0x673a, 0x673f, 0x673c, 0x6733, 0x6c18, 0x6c46, 0x6c52,
849 0x6c5c, 0x6c4f, 0x6c4a, 0x6c54, 0x6c4b,
850 /* 0xca */
851 0x6c4c, 0x7071, 0x725e, 0x72b4, 0x72b5, 0x738e, 0x752a, 0x767f,
852 0x7a75, 0x7f51, 0x8278, 0x827c, 0x8280, 0x827d, 0x827f, 0x864d,
853 0x897e, 0x9099, 0x9097, 0x9098, 0x909b, 0x9094, 0x9622, 0x9624,
854 0x9620, 0x9623, 0x4f56, 0x4f3b, 0x4f62, 0x4f49, 0x4f53, 0x4f64,
855 0x4f3e, 0x4f67, 0x4f52, 0x4f5f, 0x4f41, 0x4f58, 0x4f2d, 0x4f33,
856 0x4f3f, 0x4f61, 0x518f, 0x51b9, 0x521c, 0x521e, 0x5221, 0x52ad,
857 0x52ae, 0x5309, 0x5363, 0x5372, 0x538e, 0x538f, 0x5430, 0x5437,
858 0x542a, 0x5454, 0x5445, 0x5419, 0x541c, 0x5425, 0x5418, 0x543d,
859 0x544f, 0x5441, 0x5428, 0x5424, 0x5447, 0x56ee, 0x56e7, 0x56e5,
860 0x5741, 0x5745, 0x574c, 0x5749, 0x574b, 0x5752, 0x5906, 0x5940,
861 0x59a6, 0x5998, 0x59a0, 0x5997, 0x599e, 0x59a2, 0x5990, 0x599f,
862 0x59a7, 0x59a1, 0x5b8e, 0x5b92, 0x5c28, 0x5c2a, 0x5c8d, 0x5c8f,
863 0x5c88, 0x5c8b, 0x5c89, 0x5c92, 0x5c8a, 0x5c86, 0x5c93, 0x5c95,
864 0x5de0, 0x5e0a, 0x5e0e, 0x5e8b, 0x5e89, 0x5e8c, 0x5e88, 0x5e8d,
865 0x5f05, 0x5f1d, 0x5f78, 0x5f76, 0x5fd2, 0x5fd1, 0x5fd0, 0x5fed,
866 0x5fe8, 0x5fee, 0x5ff3, 0x5fe1, 0x5fe4, 0x5fe3, 0x5ffa, 0x5fef,
867 0x5ff7, 0x5ffb, 0x6000, 0x5ff4, 0x623a, 0x6283, 0x628c, 0x628e,
868 0x628f, 0x6294, 0x6287, 0x6271, 0x627b, 0x627a, 0x6270, 0x6281,
869 0x6288, 0x6277, 0x627d, 0x6272, 0x6274, 0x6537, 0x65f0, 0x65f4,
870 0x65f3, 0x65f2, 0x65f5, 0x6745, 0x6747,
871 /* 0xcb */
872 0x6759, 0x6755, 0x674c, 0x6748, 0x675d, 0x674d, 0x675a, 0x674b,
873 0x6bd0, 0x6c19, 0x6c1a, 0x6c78, 0x6c67, 0x6c6b, 0x6c84, 0x6c8b,
874 0x6c8f, 0x6c71, 0x6c6f, 0x6c69, 0x6c9a, 0x6c6d, 0x6c87, 0x6c95,
875 0x6c9c, 0x6c66, 0x6c73, 0x6c65, 0x6c7b, 0x6c8e, 0x7074, 0x707a,
876 0x7263, 0x72bf, 0x72bd, 0x72c3, 0x72c6, 0x72c1, 0x72ba, 0x72c5,
877 0x7395, 0x7397, 0x7393, 0x7394, 0x7392, 0x753a, 0x7539, 0x7594,
878 0x7595, 0x7681, 0x793d, 0x8034, 0x8095, 0x8099, 0x8090, 0x8092,
879 0x809c, 0x8290, 0x828f, 0x828f, 0x8285, 0x828e, 0x8291, 0x8293, 0x828a,
880 0x8283, 0x8284, 0x8c78, 0x8fc9, 0x8fbf, 0x909f, 0x90a1, 0x90a5,
881 0x909e, 0x90a7, 0x90a0, 0x9630, 0x9628, 0x962f, 0x962d, 0x4e33,
882 0x4f98, 0x4f7c, 0x4f85, 0x4f7d, 0x4f80, 0x4f87, 0x4f76, 0x4f74,
883 0x4f89, 0x4f84, 0x4f77, 0x4f4c, 0x4f97, 0x4f6a, 0x4f9a, 0x4f79,
884 0x4f81, 0x4f78, 0x4f90, 0x4f9c, 0x4f94, 0x4f9e, 0x4f92, 0x4f82,
885 0x4f95, 0x4f6b, 0x4f6e, 0x519e, 0x51bc, 0x51be, 0x5235, 0x5232,
886 0x5233, 0x5246, 0x5231, 0x52bc, 0x530a, 0x530b, 0x533c, 0x5392,

```

```
887 0x5394, 0x5487, 0x547f, 0x5481, 0x5491, 0x5482, 0x5488, 0x546b,
888 0x547a, 0x547e, 0x5465, 0x546c, 0x5474, 0x5466, 0x548d, 0x546f,
889 0x5461, 0x5460, 0x5460, 0x5498, 0x5463, 0x5467, 0x5464, 0x56f7, 0x56f9,
890 0x576f, 0x5772, 0x576d, 0x576b, 0x5771, 0x5770, 0x5776, 0x5780,
891 0x5775, 0x577b, 0x5773, 0x5774, 0x5762,
892 /* 0xcc */
893 0x5768, 0x577d, 0x590c, 0x5945, 0x59b5, 0x59ba, 0x59cf, 0x59ce,
894 0x59b2, 0x59cc, 0x59c1, 0x59b6, 0x59bc, 0x59c3, 0x59d6, 0x59b1,
895 0x59bd, 0x59c0, 0x59c8, 0x59b4, 0x59c7, 0x5b62, 0x5b65, 0x5b93,
896 0x5b95, 0x5c44, 0x5c47, 0x5cae, 0x5ca4, 0x5ca0, 0x5cb5, 0x5caf,
897 0x5ca8, 0x5cac, 0x5c9f, 0x5ca3, 0x5cad, 0x5ca2, 0x5caa, 0x5ca7,
898 0x5c9d, 0x5ca5, 0x5cb6, 0x5cb0, 0x5ca6, 0x5e17, 0x5e14, 0x5e19,
899 0x5f28, 0x5f22, 0x5f23, 0x5f24, 0x5f54, 0x5f82, 0x5f7e, 0x5f7d,
900 0x5fde, 0x5fe5, 0x602d, 0x6026, 0x6019, 0x6032, 0x600b, 0x6034,
901 0x600a, 0x6017, 0x6033, 0x601a, 0x601e, 0x602c, 0x6022, 0x600d,
902 0x6010, 0x602e, 0x6013, 0x6011, 0x600c, 0x6009, 0x601c, 0x6214,
903 0x623d, 0x62ad, 0x62b4, 0x62d1, 0x62be, 0x62aa, 0x62b6, 0x62ca,
904 0x62ae, 0x62b3, 0x62af, 0x62bb, 0x62a9, 0x62b0, 0x62b8, 0x653d,
905 0x65a8, 0x65bb, 0x6609, 0x65fc, 0x6604, 0x6612, 0x6608, 0x65fb,
906 0x6603, 0x660b, 0x660d, 0x6605, 0x65fd, 0x6611, 0x6610, 0x66f6,
907 0x670a, 0x6785, 0x676c, 0x678e, 0x6792, 0x6776, 0x677b, 0x6798,
908 0x6786, 0x6784, 0x6774, 0x678d, 0x678c, 0x677a, 0x679f, 0x6791,
909 0x6799, 0x6783, 0x677d, 0x6781, 0x6778, 0x6779, 0x6794, 0x6b25,
910 0x6b80, 0x6b7e, 0x6bde, 0x6c1d, 0x6c93, 0x6cec, 0x6ceb, 0x6cee,
911 0x6cd9, 0x6cb6, 0x6cd4, 0x6cad, 0x6ce7, 0x6cb7, 0x6cd0, 0x6cc2,
912 0x6cba, 0x6cc3, 0x6cc6, 0x6ced, 0x6cf2,
913 /* 0xcd */
914 0x6cd2, 0x6cdd, 0x6cb4, 0x6c8a, 0x6c9d, 0x6c80, 0x6cde, 0x6cc0,
915 0x6d30, 0x6ccd, 0x6cc7, 0x6cb0, 0x6cf9, 0x6ccf, 0x6ce9, 0x6cd1,
916 0x7094, 0x7098, 0x7085, 0x7085, 0x7086, 0x7084, 0x7091, 0x7096,
917 0x7082, 0x709a, 0x7083, 0x726a, 0x72d6, 0x72cb, 0x72d8, 0x72c9,
918 0x72dc, 0x72d2, 0x72d4, 0x72da, 0x72cc, 0x72d1, 0x73a4, 0x73a1,
919 0x73ad, 0x73a6, 0x73a2, 0x73a0, 0x73ac, 0x739d, 0x74dd, 0x74e8,
920 0x753f, 0x7540, 0x753e, 0x758c, 0x7598, 0x76af, 0x76f3, 0x76f1,
921 0x76f0, 0x76f5, 0x77f8, 0x77fc, 0x77f9, 0x77fb, 0x77fa, 0x77f7,
922 0x7942, 0x793f, 0x79c5, 0x79c5, 0x7a78, 0x7afb, 0x7c75, 0x7cfd,
923 0x8035, 0x808f, 0x80ae, 0x80a3, 0x80b8, 0x80b5, 0x80ad, 0x8220,
924 0x82a0, 0x82c0, 0x82ab, 0x829a, 0x8298, 0x829b, 0x82b5, 0x82a7,
925 0x82ae, 0x82bc, 0x829e, 0x829e, 0x82ba, 0x82b4, 0x82a8, 0x82a1, 0x82a9,
926 0x82c2, 0x82a4, 0x82c3, 0x82b6, 0x82a2, 0x8670, 0x866f, 0x866d,
927 0x866e, 0x8c56, 0x8fd2, 0x8fcb, 0x8fd3, 0x8fcd, 0x8fd6, 0x8fd5,
928 0x8fd7, 0x90b2, 0x90b4, 0x90af, 0x90b3, 0x90b0, 0x9639, 0x963d,
929 0x963c, 0x963a, 0x9643, 0x4fcd, 0x4fc5, 0x4fd3, 0x4fb2, 0x4fc9,
930 0x4fcb, 0x4fc1, 0x4fd4, 0x4fdc, 0x4fd9, 0x4fbb, 0x4fb3, 0x4fdb,
931 0x4fc7, 0x4fd6, 0x4fba, 0x4fc0, 0x4fb9, 0x4fec, 0x5244, 0x5249,
932 0x52c0, 0x52c2, 0x533d, 0x537c, 0x5397, 0x5396, 0x5399, 0x5398,
933 0x54ba, 0x54a1, 0x54ad, 0x54a5, 0x54cf,
934 /* 0xce */
935 0x54c3, 0x830d, 0x54b7, 0x54ae, 0x54d6, 0x54b6, 0x54c5, 0x54c6,
936 0x54a0, 0x5470, 0x54bc, 0x54a2, 0x54be, 0x5472, 0x54de, 0x54b0,
937 0x57b5, 0x579e, 0x579f, 0x57a4, 0x578c, 0x5797, 0x579d, 0x579b,
938 0x5794, 0x5798, 0x578f, 0x5799, 0x57a5, 0x579a, 0x5795, 0x58f4,
939 0x590d, 0x5953, 0x59e1, 0x59de, 0x59ee, 0x5a00, 0x59f1, 0x59dd,
940 0x59fa, 0x59fd, 0x59fc, 0x59f6, 0x59e4, 0x59f2, 0x59f7, 0x59db,
941 0x59e9, 0x59f3, 0x59f5, 0x59e0, 0x59fe, 0x59f4, 0x59ed, 0x5ba8,
942 0x5c4c, 0x5cd0, 0x5cd8, 0x5ccc, 0x5cd7, 0x5ccb, 0x5cdb, 0x5cde,
943 0x5cda, 0x5cc9, 0x5cc7, 0x5cca, 0x5cd6, 0x5cd3, 0x5cd4, 0x5ccf,
944 0x5cc8, 0x5cc6, 0x5cce, 0x5cdf, 0x5cf8, 0x5df9, 0x5e21, 0x5e22,
945 0x5e23, 0x5e20, 0x5e24, 0x5eb0, 0x5ea4, 0x5ea2, 0x5e9b, 0x5ea3,
946 0x5ea5, 0x5f07, 0x5f2e, 0x5f56, 0x5f86, 0x6037, 0x6039, 0x6054,
947 0x6072, 0x605e, 0x6045, 0x6053, 0x6047, 0x6049, 0x605b, 0x604c,
948 0x6040, 0x6042, 0x605f, 0x6024, 0x6044, 0x6058, 0x6066, 0x606e,
949 0x6242, 0x6243, 0x62cf, 0x630d, 0x630b, 0x62f5, 0x630e, 0x6303,
950 0x62eb, 0x62f9, 0x630f, 0x630c, 0x62f8, 0x62f6, 0x6300, 0x6313,
951 0x6314, 0x62fa, 0x6315, 0x62fb, 0x62f0, 0x6541, 0x6543, 0x65aa,
952 0x65bf, 0x6636, 0x6621, 0x6632, 0x6635, 0x661c, 0x6626, 0x6622,
953 0x6633, 0x662b, 0x663a, 0x661d, 0x6634, 0x6639, 0x662e, 0x670f,
954 0x6710, 0x67c1, 0x67f2, 0x67c8, 0x67ba,
955 /* 0xcf */
956 0x67dc, 0x67bb, 0x67f8, 0x67d8, 0x67c0, 0x67b7, 0x67c5, 0x67eb,
957 0x67e4, 0x67df, 0x67b5, 0x67cd, 0x67b3, 0x67f7, 0x67f6, 0x67ee,
958 0x67e3, 0x67c2, 0x67b9, 0x67ce, 0x67e7, 0x67f0, 0x67b2, 0x67fc,
959 0x67c6, 0x67ed, 0x67cc, 0x67ae, 0x67e6, 0x67bd, 0x67fa, 0x67c9,
960 0x67ca, 0x67c3, 0x67ea, 0x67cb, 0x6b28, 0x6b82, 0x6b84, 0x6bb6,
961 0x6bd6, 0x6bd8, 0x6be0, 0x6c20, 0x6c21, 0x6d28, 0x6d34, 0x6d2d,
962 0x6d1f, 0x6d3c, 0x6d3f, 0x6d12, 0x6d0a, 0x6cda, 0x6d33, 0x6d04,
963 0x6d19, 0x6d3a, 0x6d1a, 0x6d11, 0x6d00, 0x6d1d, 0x6d42, 0x6d01,
964 0x6d18, 0x6d37, 0x6d03, 0x6d0f, 0x6d40, 0x6d07, 0x6d20, 0x6d2c,
965 0x6d08, 0x6d22, 0x6d09, 0x6d10, 0x70b7, 0x709f, 0x70be, 0x70b1,
966 0x70b0, 0x70a1, 0x70b4, 0x70b5, 0x70a9, 0x7241, 0x7249, 0x724a,
967 0x726c, 0x7270, 0x7273, 0x726e, 0x72ca, 0x72e4, 0x72e8, 0x72eb,
968 0x72df, 0x72ea, 0x72e6, 0x72e3, 0x7385, 0x73cc, 0x73c2, 0x73c8,
969 0x73c5, 0x73b9, 0x73b6, 0x73b5, 0x73b4, 0x73eb, 0x73bf, 0x73c7,
970 0x73be, 0x73c3, 0x73c6, 0x73b8, 0x73cb, 0x74ec, 0x74ee, 0x752e,
971 0x7547, 0x7548, 0x75a7, 0x75aa, 0x7679, 0x76c4, 0x7708, 0x7703,
972 0x7704, 0x7705, 0x770a, 0x76f7, 0x76fb, 0x76fa, 0x77e7, 0x77e8,
973 0x7806, 0x7811, 0x7812, 0x7805, 0x7810, 0x780f, 0x780e, 0x7809,
```

```
974 0x7803, 0x7813, 0x794a, 0x794c, 0x794b, 0x7945, 0x7944, 0x79d5,
975 0x79cd, 0x79cf, 0x79d6, 0x79ce, 0x7a80,
976 /* 0xd0 */
977 0x7a7e, 0x7ad1, 0x7b00, 0x7b01, 0x7c7a, 0x7c78, 0x7c79, 0x7c7f,
978 0x7c80, 0x7c81, 0x7d03, 0x7d08, 0x7d01, 0x7f58, 0x7f91, 0x7f8d,
979 0x7fbe, 0x8007, 0x800e, 0x800e, 0x800f, 0x8014, 0x8037, 0x80d8, 0x80c7,
980 0x80e0, 0x80d1, 0x80c8, 0x80c2, 0x80d0, 0x80c5, 0x80e3, 0x80d9,
981 0x80dc, 0x80ca, 0x80d5, 0x80c9, 0x80cf, 0x80d7, 0x80e6, 0x80cd,
982 0x81ff, 0x8221, 0x8294, 0x8294, 0x82d9, 0x82fe, 0x82f9, 0x8307, 0x82e8,
983 0x8300, 0x82d5, 0x833a, 0x82eb, 0x82d6, 0x82f4, 0x82ec, 0x82e1,
984 0x82f2, 0x82f5, 0x830c, 0x82fb, 0x82f6, 0x82f0, 0x82ea, 0x82e4,
985 0x82e0, 0x82fa, 0x82f3, 0x82ed, 0x8677, 0x8674, 0x867c, 0x8673,
986 0x8841, 0x884e, 0x8867, 0x886a, 0x8869, 0x89d3, 0x8a04, 0x8a07,
987 0x8d72, 0x8fe3, 0x8fe1, 0x8fee, 0x8fe0, 0x90f1, 0x90bd, 0x90bf,
988 0x90d5, 0x90c5, 0x90be, 0x90c7, 0x90cb, 0x90c8, 0x91d4, 0x91d3,
989 0x9654, 0x964f, 0x9651, 0x9653, 0x964a, 0x964e, 0x501e, 0x5005,
990 0x5007, 0x5013, 0x5022, 0x5030, 0x501b, 0x4ff5, 0x4ff4, 0x5033,
991 0x5037, 0x502c, 0x502c, 0x4ff6, 0x4ff7, 0x5017, 0x501c, 0x5020, 0x5027,
992 0x5035, 0x502f, 0x5031, 0x500e, 0x515a, 0x5194, 0x5193, 0x51ca,
993 0x51c4, 0x51c5, 0x51c8, 0x51ce, 0x5261, 0x525a, 0x5252, 0x525e,
994 0x525f, 0x5255, 0x5262, 0x52cd, 0x530e, 0x539e, 0x5526, 0x54e2,
995 0x5517, 0x5512, 0x54e7, 0x54f3, 0x54e4, 0x551a, 0x54ff, 0x5504,
996 0x5508, 0x54eb, 0x5511, 0x5505, 0x54f1,
997 /* 0xd1 */
998 0x550a, 0x54fb, 0x54f7, 0x54f8, 0x54e0, 0x550e, 0x5503, 0x550b,
999 0x5701, 0x5702, 0x57cc, 0x5832, 0x57d5, 0x57d2, 0x57ba, 0x57c6,
1000 0x57bd, 0x57bc, 0x57b8, 0x57b6, 0x57bf, 0x57c7, 0x57d0, 0x57b9,
1001 0x57c1, 0x590e, 0x594a, 0x5a19, 0x5a16, 0x5a2d, 0x5a2e, 0x5a15,
1002 0x5a0f, 0x5a17, 0x5a0a, 0x5a1e, 0x5a33, 0x5b6c, 0x5ba7, 0x5bad,
1003 0x5bac, 0x5c03, 0x5c56, 0x5c54, 0x5cec, 0x5cfc, 0x5cee, 0x5cfl,
1004 0x5cf7, 0x5d00, 0x5cf9, 0x5e29, 0x5e28, 0x5ea8, 0x5eae, 0x5eaa,
1005 0x5eac, 0x5f33, 0x5f30, 0x5f67, 0x605d, 0x605a, 0x6067, 0x6041,
1006 0x60a2, 0x6088, 0x6080, 0x6092, 0x6081, 0x609d, 0x6083, 0x6095,
1007 0x609b, 0x6097, 0x6087, 0x609c, 0x608e, 0x6219, 0x6246, 0x62f2,
1008 0x6310, 0x6356, 0x632c, 0x6344, 0x6345, 0x6336, 0x6343, 0x63e4,
1009 0x6339, 0x634b, 0x634a, 0x634a, 0x633c, 0x6329, 0x6341, 0x6334, 0x6358,
1010 0x6354, 0x6359, 0x632d, 0x6347, 0x6333, 0x635a, 0x6351, 0x6338,
1011 0x6357, 0x6340, 0x6348, 0x654a, 0x6546, 0x65c6, 0x65c3, 0x65c4,
1012 0x65c2, 0x664a, 0x665f, 0x6647, 0x6651, 0x6712, 0x6713, 0x681f,
1013 0x681a, 0x6849, 0x6832, 0x6833, 0x683b, 0x684b, 0x684f, 0x6816,
1014 0x6831, 0x681c, 0x6835, 0x682b, 0x682d, 0x682f, 0x684e, 0x6844,
1015 0x6834, 0x681d, 0x6812, 0x6814, 0x6826, 0x6828, 0x682e, 0x684d,
1016 0x683a, 0x6825, 0x6820, 0x6b2c, 0x6b2f, 0x6b2d, 0x6b31, 0x6b34,
1017 0x6b6d, 0x8082, 0x6b88, 0x6be6, 0x6be4,
1018 /* 0xd2 */
1019 0x6be8, 0x6be3, 0x6be2, 0x6be7, 0x6c25, 0x6d7a, 0x6d63, 0x6d64,
1020 0x6d76, 0x6d0d, 0x6d61, 0x6d92, 0x6d58, 0x6d62, 0x6d6d, 0x6d6f,
1021 0x6d91, 0x6d8d, 0x6def, 0x6d7f, 0x6d86, 0x6d5e, 0x6d67, 0x6d60,
1022 0x6d97, 0x6d70, 0x6d7c, 0x6d5f, 0x6d82, 0x6d98, 0x6d2f, 0x6d68,
1023 0x6d8b, 0x6d7e, 0x6d80, 0x6d84, 0x6d16, 0x6d83, 0x6d7b, 0x6d7d,
1024 0x6d75, 0x6d90, 0x70dc, 0x70d3, 0x70d1, 0x70dd, 0x70cb, 0x7f39,
1025 0x70e2, 0x70d7, 0x70d2, 0x70de, 0x70e0, 0x70d4, 0x70cd, 0x70c5,
1026 0x70c6, 0x70c7, 0x70da, 0x70ce, 0x70e1, 0x7242, 0x7278, 0x7277,
1027 0x7276, 0x7300, 0x72fa, 0x72f4, 0x72fe, 0x72f6, 0x72f3, 0x72fb,
1028 0x7301, 0x73d3, 0x73d9, 0x73e5, 0x73d6, 0x73bc, 0x73e7, 0x73e3,
1029 0x73e9, 0x73dc, 0x73d2, 0x73db, 0x73d4, 0x73dd, 0x73da, 0x73d7,
1030 0x73d8, 0x73e8, 0x74de, 0x74df, 0x74f4, 0x74f5, 0x7521, 0x755b,
1031 0x755f, 0x75b0, 0x75c1, 0x75bb, 0x75c4, 0x75c0, 0x75bf, 0x75b6,
1032 0x75ba, 0x768a, 0x76c9, 0x771d, 0x771b, 0x7710, 0x7713, 0x7712,
1033 0x7723, 0x7711, 0x7715, 0x7719, 0x771a, 0x7722, 0x7727, 0x7823,
1034 0x782c, 0x7822, 0x7835, 0x782f, 0x7828, 0x782e, 0x782b, 0x7821,
1035 0x7829, 0x7833, 0x782a, 0x7831, 0x7954, 0x795b, 0x794f, 0x795c,
1036 0x7953, 0x7952, 0x7951, 0x79eb, 0x79ec, 0x79e0, 0x79ee, 0x79ed,
1037 0x79ea, 0x79dc, 0x79de, 0x79dd, 0x7a86, 0x7a89, 0x7a85, 0x7a8b,
1038 0x7a8c, 0x7a8a, 0x7a87, 0x7ad8, 0x7b10,
1039 /* 0xd3 */
1040 0x7b04, 0x7b13, 0x7b05, 0x7b0f, 0x7b08, 0x7b0a, 0x7b0e, 0x7b09,
1041 0x7b12, 0x7c84, 0x7c91, 0x7c8a, 0x7c8c, 0x7c88, 0x7c8d, 0x7c85,
1042 0x7d1e, 0x7d1d, 0x7d11, 0x7d0e, 0x7d18, 0x7d16, 0x7d13, 0x7d1f,
1043 0x7d12, 0x7d0f, 0x7d0c, 0x7f5c, 0x7f61, 0x7f5e, 0x7f60, 0x7f5d,
1044 0x7f5b, 0x7f96, 0x7f92, 0x7fc3, 0x7fc2, 0x7fc0, 0x8016, 0x803e,
1045 0x8039, 0x80fa, 0x80f2, 0x80f9, 0x80f5, 0x8101, 0x80fb, 0x8100,
1046 0x8201, 0x822f, 0x8225, 0x8333, 0x832d, 0x8344, 0x8319, 0x8351,
1047 0x8325, 0x8356, 0x833f, 0x8341, 0x8326, 0x833c, 0x8322, 0x8342,
1048 0x834e, 0x831b, 0x832a, 0x8308, 0x833c, 0x834d, 0x8316, 0x8324,
1049 0x8320, 0x8337, 0x832f, 0x8329, 0x8347, 0x8345, 0x834c, 0x8353,
1050 0x831e, 0x832c, 0x834b, 0x8327, 0x8348, 0x8653, 0x8652, 0x86a2,
1051 0x86a8, 0x8696, 0x868d, 0x8691, 0x869e, 0x8687, 0x8697, 0x8686,
1052 0x868b, 0x869a, 0x8685, 0x86a5, 0x8699, 0x86a1, 0x86a7, 0x8695,
1053 0x8698, 0x868e, 0x869d, 0x8690, 0x8694, 0x8843, 0x8844, 0x886d,
1054 0x8875, 0x8876, 0x8872, 0x8880, 0x8871, 0x887f, 0x886f, 0x8883,
1055 0x887e, 0x8874, 0x8877, 0x8a12, 0x8c47, 0x8c57, 0x8c7b, 0x8ca4,
1056 0x8ca3, 0x8d76, 0x8d78, 0x8db5, 0x8db7, 0x8db6, 0x8ed1, 0x8ed3,
1057 0x8ffe, 0x8ff5, 0x9002, 0x8fff, 0x8ffb, 0x9004, 0x8ffc, 0x8fff,
1058 0x90d6, 0x90e0, 0x90d9, 0x90da, 0x90e3, 0x90df, 0x90e5, 0x90d8,
1059 0x90db, 0x90d7, 0x90dc, 0x90e4, 0x9150,
1060 /* 0xd4 */
```

```
1061 0x914e, 0x914f, 0x91d5, 0x91e2, 0x91da, 0x965c, 0x965f, 0x96bc,
1062 0x98e3, 0x9adf, 0x9b2f, 0x4e7f, 0x5070, 0x506a, 0x5061, 0x505e,
1063 0x5060, 0x5053, 0x504b, 0x504d, 0x505d, 0x5072, 0x5048, 0x504d, 0x5041,
1064 0x505b, 0x504a, 0x5062, 0x5015, 0x5045, 0x505f, 0x5069, 0x506b,
1065 0x5063, 0x5064, 0x5046, 0x5040, 0x506e, 0x5073, 0x5057, 0x5051,
1066 0x51d0, 0x526b, 0x526d, 0x526c, 0x526e, 0x5266, 0x52d3, 0x532d,
1067 0x539c, 0x5575, 0x5576, 0x553c, 0x554d, 0x5550, 0x5534, 0x552a,
1068 0x5551, 0x5562, 0x5536, 0x5535, 0x5530, 0x5552, 0x5545, 0x550c,
1069 0x5532, 0x5565, 0x554e, 0x5539, 0x5548, 0x552d, 0x553b, 0x5540,
1070 0x554b, 0x570a, 0x5707, 0x57fb, 0x5814, 0x57e2, 0x57f6, 0x57dc,
1071 0x57f4, 0x5800, 0x57ed, 0x57fd, 0x5808, 0x57f8, 0x580b, 0x57f3,
1072 0x57cf, 0x5807, 0x57ee, 0x57e3, 0x57f2, 0x57e5, 0x57ec, 0x57e1,
1073 0x580e, 0x57fc, 0x5810, 0x57e7, 0x5801, 0x580c, 0x57f1, 0x57e9,
1074 0x57f0, 0x580d, 0x5804, 0x595c, 0x5a60, 0x5a58, 0x5a55, 0x5a67,
1075 0x5a5e, 0x5a38, 0x5a35, 0x5a6d, 0x5a50, 0x5a5f, 0x5a65, 0x5a6c,
1076 0x5a53, 0x5a64, 0x5a57, 0x5a43, 0x5a5d, 0x5a52, 0x5a44, 0x5a5b,
1077 0x5a48, 0x5a8e, 0x5a3e, 0x5a4d, 0x5a39, 0x5a4c, 0x5a70, 0x5a69,
1078 0x5a47, 0x5a51, 0x5a56, 0x5a42, 0x5a5c, 0x5b72, 0x5b6e, 0x5bc1,
1079 0x5bc0, 0x5c59, 0x5d1e, 0x5d0b, 0x5d1d, 0x5d1a, 0x5d20, 0x5d0c,
1080 0x5d28, 0x5d0d, 0x5d26, 0x5d25, 0x5d0f,
1081 /* 0xd5 */
1082 0x5d30, 0x5d12, 0x5d23, 0x5d1f, 0x5d2e, 0x5e3e, 0x5e34, 0x5eb1,
1083 0x5eb4, 0x5eb9, 0x5eb2, 0x5eb3, 0x5f36, 0x5f38, 0x5f9b, 0x5f96,
1084 0x5f9f, 0x608a, 0x6090, 0x6086, 0x60be, 0x60b0, 0x60ba, 0x60d3,
1085 0x60d4, 0x60cf, 0x60e4, 0x60d9, 0x60dd, 0x60c8, 0x60b1, 0x60db,
1086 0x60b7, 0x60ca, 0x60bf, 0x60c3, 0x60cd, 0x60c0, 0x6332, 0x6365,
1087 0x638a, 0x6382, 0x637d, 0x63bd, 0x639e, 0x63ad, 0x639d, 0x6397,
1088 0x63ab, 0x638e, 0x636f, 0x6387, 0x6390, 0x636e, 0x63af, 0x6375,
1089 0x639c, 0x636d, 0x63ae, 0x637c, 0x63a4, 0x633b, 0x639f, 0x6378,
1090 0x6385, 0x6381, 0x6391, 0x638d, 0x6370, 0x6553, 0x65cd, 0x6665,
1091 0x6661, 0x665b, 0x6659, 0x665c, 0x6662, 0x6718, 0x6879, 0x6887,
1092 0x6890, 0x689c, 0x686d, 0x686e, 0x68ae, 0x68ab, 0x6956, 0x686f,
1093 0x68a3, 0x68ac, 0x68a9, 0x6875, 0x6874, 0x68b2, 0x688f, 0x6877,
1094 0x6892, 0x687c, 0x686b, 0x6872, 0x68aa, 0x6880, 0x6871, 0x687e,
1095 0x689b, 0x6896, 0x688b, 0x68a0, 0x6889, 0x68a4, 0x6878, 0x687b,
1096 0x6891, 0x688c, 0x688a, 0x687d, 0x6b36, 0x6b33, 0x6b37, 0x6b38,
1097 0x6b91, 0x6b8f, 0x6b8d, 0x6b8e, 0x6b8c, 0x6c2a, 0x6dc0, 0x6dab,
1098 0x6db4, 0x6db3, 0x6e74, 0x6dac, 0x6de9, 0x6de2, 0x6db7, 0x6df6,
1099 0x6dd4, 0x6e00, 0x6dc8, 0x6de0, 0x6ddf, 0x6dd6, 0x6dbe, 0x6de5,
1100 0x6ddc, 0x6ddd, 0x6ddb, 0x6df4, 0x6dca, 0x6dbd, 0x6ded, 0x6df0,
1101 0x6dba, 0x6dd5, 0x6dc2, 0x6dcf, 0x6dc9,
1102 /* 0xd6 */
1103 0x6dd0, 0x6df2, 0x6dd3, 0x6dfd, 0x6dd7, 0x6dcd, 0x6de3, 0x6dbb,
1104 0x70fa, 0x710d, 0x70f7, 0x7117, 0x70f4, 0x710c, 0x70f0, 0x7104,
1105 0x70f3, 0x7110, 0x70fc, 0x70ff, 0x7106, 0x7113, 0x7100, 0x70f8,
1106 0x70f6, 0x710b, 0x7102, 0x710e, 0x727e, 0x727b, 0x727c, 0x727f,
1107 0x731d, 0x7317, 0x7307, 0x7311, 0x7318, 0x730a, 0x7308, 0x72ff,
1108 0x730f, 0x731e, 0x7388, 0x73f6, 0x73f8, 0x73f5, 0x7404, 0x7401,
1109 0x73fd, 0x7407, 0x7400, 0x73fa, 0x73fc, 0x73ff, 0x740c, 0x740b,
1110 0x73f4, 0x7408, 0x7564, 0x7563, 0x75ce, 0x75d2, 0x75cf, 0x75cb,
1111 0x75cc, 0x75d1, 0x75d0, 0x768f, 0x7689, 0x76d3, 0x7739, 0x772e,
1112 0x772d, 0x7731, 0x7732, 0x7734, 0x7733, 0x773d, 0x7725, 0x773b,
1113 0x7735, 0x7848, 0x7852, 0x7849, 0x784d, 0x784a, 0x784c, 0x7826,
1114 0x7845, 0x7850, 0x7964, 0x7967, 0x7969, 0x796a, 0x7963, 0x796b,
1115 0x7961, 0x79bb, 0x79fa, 0x79f8, 0x79f6, 0x79f7, 0x7a8f, 0x7a94,
1116 0x7a90, 0x7b35, 0x7b47, 0x7b34, 0x7b25, 0x7b30, 0x7b22, 0x7b24,
1117 0x7b33, 0x7b18, 0x7b2a, 0x7b1d, 0x7b31, 0x7b2b, 0x7b2d, 0x7b2f,
1118 0x7b32, 0x7b38, 0x7b1a, 0x7b23, 0x7c94, 0x7c98, 0x7c96, 0x7ca3,
1119 0x7d35, 0x7d3d, 0x7d38, 0x7d36, 0x7d3a, 0x7d45, 0x7d2c, 0x7d29,
1120 0x7d41, 0x7d47, 0x7d3e, 0x7d3f, 0x7d4a, 0x7d3b, 0x7d28, 0x7f63,
1121 0x7f95, 0x7f9c, 0x7f9d, 0x7f9b, 0x7fca, 0x7fcb, 0x7fcd, 0x7fd0,
1122 0x7fd1, 0x7fc7, 0x7fcf, 0x7fc9, 0x801f,
1123 /* 0xd7 */
1124 0x801e, 0x801b, 0x8047, 0x8043, 0x8048, 0x8118, 0x8125, 0x8119,
1125 0x811b, 0x812d, 0x811f, 0x812c, 0x811e, 0x8121, 0x8115, 0x8127,
1126 0x811d, 0x8122, 0x8211, 0x8238, 0x8233, 0x823a, 0x8234, 0x8232,
1127 0x8274, 0x8390, 0x83a3, 0x83a8, 0x838d, 0x837a, 0x8373, 0x83a4,
1128 0x8374, 0x838f, 0x8381, 0x8395, 0x8399, 0x8375, 0x8394, 0x83a9,
1129 0x837d, 0x8383, 0x838c, 0x839d, 0x839b, 0x83aa, 0x838b, 0x837e,
1130 0x83a5, 0x83af, 0x8388, 0x8397, 0x83b0, 0x837f, 0x83a6, 0x8387,
1131 0x83ae, 0x8376, 0x839a, 0x8659, 0x8656, 0x86bf, 0x86b7, 0x86c2,
1132 0x86c1, 0x86c5, 0x86ba, 0x86b0, 0x86c8, 0x86b9, 0x86b3, 0x86b8,
1133 0x86cc, 0x86b4, 0x86bb, 0x86bc, 0x86c3, 0x86bd, 0x86be, 0x8852,
1134 0x8889, 0x8895, 0x88a8, 0x88a2, 0x88aa, 0x889a, 0x8891, 0x88a1,
1135 0x889f, 0x8898, 0x88a7, 0x8899, 0x889b, 0x8897, 0x88a4, 0x88ac,
1136 0x888c, 0x8893, 0x888e, 0x8882, 0x89d6, 0x89d9, 0x89d5, 0x8a30,
1137 0x8a27, 0x8a2c, 0x8a1e, 0x8c39, 0x8c3b, 0x8c5c, 0x8c5d, 0x8c7d,
1138 0x8ca5, 0x8d7d, 0x8d7b, 0x8d79, 0x8dbc, 0x8dc2, 0x8db9, 0x8dbf,
1139 0x8dc1, 0x8ed8, 0x8ede, 0x8edd, 0x8edc, 0x8ed7, 0x8ee0, 0x8ee1,
1140 0x9024, 0x900b, 0x9011, 0x901c, 0x900c, 0x9021, 0x90ef, 0x90ea,
1141 0x90f0, 0x90f4, 0x90f2, 0x90f3, 0x90d4, 0x90eb, 0x90ec, 0x90e9,
1142 0x9156, 0x9158, 0x915a, 0x9153, 0x9155, 0x91ec, 0x91f4, 0x91f1,
1143 0x91f3, 0x91f8, 0x91e4, 0x91f9, 0x91ea,
1144 /* 0xd8 */
1145 0x91eb, 0x91f7, 0x91e8, 0x91ee, 0x957a, 0x9586, 0x9588, 0x967c,
1146 0x966d, 0x966b, 0x9671, 0x966f, 0x96bf, 0x976a, 0x9804, 0x98e5,
1147 0x9997, 0x509b, 0x5095, 0x5094, 0x509e, 0x508b, 0x50a3, 0x5083,
```

```
1148 0x508c, 0x508e, 0x509d, 0x5068, 0x509c, 0x5092, 0x5082, 0x5087,
1149 0x515f, 0x51d4, 0x5312, 0x5311, 0x53a4, 0x53a7, 0x5591, 0x55a8,
1150 0x55a5, 0x55ad, 0x5577, 0x5645, 0x55a2, 0x5593, 0x5588, 0x558f,
1151 0x55b5, 0x5581, 0x55a3, 0x5592, 0x55a4, 0x557d, 0x558c, 0x55a6,
1152 0x557f, 0x5595, 0x55a1, 0x558e, 0x570c, 0x5829, 0x5837, 0x5819,
1153 0x581e, 0x5827, 0x5823, 0x5828, 0x57f5, 0x5848, 0x5825, 0x581c,
1154 0x581b, 0x5833, 0x583f, 0x5836, 0x582e, 0x5839, 0x5838, 0x582d,
1155 0x582c, 0x583b, 0x5961, 0x5aaf, 0x5a94, 0x5a9f, 0x5a7a, 0x5aa2,
1156 0x5a9e, 0x5a78, 0x5aa6, 0x5a7c, 0x5aa5, 0x5aac, 0x5a95, 0x5aae,
1157 0x5a37, 0x5a84, 0x5a8a, 0x5a97, 0x5a83, 0x5a8b, 0x5aa9, 0x5a7b,
1158 0x5a7d, 0x5a8c, 0x5a9c, 0x5a8f, 0x5a93, 0x5a9d, 0x5bea, 0x5bcd,
1159 0x5bcb, 0x5bd4, 0x5bd1, 0x5bca, 0x5bce, 0x5c0c, 0x5c30, 0x5d37,
1160 0x5d43, 0x5d6b, 0x5d41, 0x5d4b, 0x5d3f, 0x5d35, 0x5d51, 0x5d4e,
1161 0x5d55, 0x5d33, 0x5d3a, 0x5d52, 0x5d3d, 0x5d31, 0x5d59, 0x5d42,
1162 0x5d39, 0x5d49, 0x5d38, 0x5d3c, 0x5d32, 0x5d36, 0x5d40, 0x5d45,
1163 0x5e44, 0x5e41, 0x5f58, 0x5fa6, 0x5fa5, 0x5fab, 0x60c9, 0x60b9,
1164 0x60cc, 0x60e2, 0x60ce, 0x60c4, 0x6114,
1165 /* 0xd9 */
1166 0x60f2, 0x610a, 0x6116, 0x6105, 0x60f5, 0x6113, 0x60f8, 0x60fc,
1167 0x60fe, 0x60c1, 0x6103, 0x6118, 0x611d, 0x6110, 0x60ff, 0x6104,
1168 0x610b, 0x624a, 0x6394, 0x63b1, 0x63b0, 0x63ce, 0x63e5, 0x63e8,
1169 0x63ef, 0x63c3, 0x649d, 0x63f3, 0x63ca, 0x63e0, 0x63f6, 0x63d5,
1170 0x63f2, 0x63f5, 0x6461, 0x63df, 0x63be, 0x63dd, 0x63dc, 0x63c4,
1171 0x63d8, 0x63d3, 0x63c2, 0x63c7, 0x63cc, 0x63cb, 0x63c8, 0x63f0,
1172 0x63d7, 0x63d9, 0x6532, 0x6567, 0x656a, 0x6564, 0x655c, 0x6568,
1173 0x6565, 0x658c, 0x659d, 0x659e, 0x65ae, 0x65d0, 0x65d2, 0x667c,
1174 0x666c, 0x667b, 0x6680, 0x6671, 0x6679, 0x666a, 0x6672, 0x6701,
1175 0x690c, 0x68d3, 0x6904, 0x68dc, 0x692a, 0x68ec, 0x68ea, 0x68f1,
1176 0x690f, 0x68d6, 0x68f7, 0x68eb, 0x68e4, 0x68f6, 0x6913, 0x6910,
1177 0x68f3, 0x68e1, 0x6907, 0x68cc, 0x6908, 0x6970, 0x68b4, 0x6911,
1178 0x68ef, 0x68c6, 0x6914, 0x68f8, 0x68d0, 0x68fd, 0x68fc, 0x68e8,
1179 0x690b, 0x690a, 0x6917, 0x68ce, 0x68c8, 0x68dd, 0x68de, 0x68e6,
1180 0x68f4, 0x68d1, 0x6906, 0x68d4, 0x68e9, 0x6915, 0x6925, 0x68c7,
1181 0x6b39, 0x6b3b, 0x6b3f, 0x6b3c, 0x6b94, 0x6b97, 0x6b99, 0x6b95,
1182 0x6bbd, 0x6bf0, 0x6bf2, 0x6bf3, 0x6c30, 0x6dfc, 0x6e46, 0x6e47,
1183 0x6e1f, 0x6e49, 0x6e88, 0x6e3c, 0x6e45, 0x6e62, 0x6e2b,
1184 0x6e3f, 0x6e41, 0x6e5d, 0x6e73, 0x6e1c, 0x6e33, 0x6e4b, 0x6e40,
1185 0x6e51, 0x6e3b, 0x6e03, 0x6e2e, 0x6e5e,
1186 /* 0xda */
1187 0x6e68, 0x6e5c, 0x6e61, 0x6e31, 0x6e28, 0x6e60, 0x6e71, 0x6e6b,
1188 0x6e39, 0x6e22, 0x6e30, 0x6e53, 0x6e65, 0x6e27, 0x6e78, 0x6e64,
1189 0x6e77, 0x6e55, 0x6e79, 0x6e52, 0x6e66, 0x6e35, 0x6e36, 0x6e5a,
1190 0x7120, 0x711e, 0x712f, 0x70fb, 0x712e, 0x7131, 0x7123, 0x7125,
1191 0x7122, 0x7132, 0x711f, 0x7128, 0x713a, 0x711b, 0x724b, 0x725a,
1192 0x7288, 0x7289, 0x7286, 0x7285, 0x728b, 0x7312, 0x730b, 0x7330,
1193 0x7322, 0x7331, 0x7333, 0x7327, 0x7332, 0x732d, 0x7326, 0x7323,
1194 0x7335, 0x730c, 0x742e, 0x742c, 0x7430, 0x742b, 0x7416, 0x741a,
1195 0x7421, 0x742d, 0x7431, 0x7424, 0x7423, 0x741d, 0x7429, 0x7420,
1196 0x7432, 0x74fb, 0x752f, 0x756f, 0x756c, 0x75e7, 0x75da, 0x75e1,
1197 0x75e6, 0x75dd, 0x75df, 0x75e4, 0x75d7, 0x7695, 0x7692, 0x76da,
1198 0x7746, 0x7747, 0x7744, 0x774d, 0x7745, 0x774a, 0x774e, 0x774b,
1199 0x774c, 0x77de, 0x77ec, 0x7860, 0x7864, 0x7865, 0x785c, 0x786d,
1200 0x7871, 0x786a, 0x786e, 0x7870, 0x7869, 0x7868, 0x785e, 0x7862,
1201 0x7974, 0x7973, 0x7972, 0x7970, 0x7a02, 0x7a0a, 0x7a03, 0x7a0c,
1202 0x7a04, 0x7a99, 0x7ae6, 0x7ae4, 0x7b4a, 0x7b3b, 0x7b44, 0x7b48,
1203 0x7b4c, 0x7b4e, 0x7b40, 0x7b58, 0x7b45, 0x7ca2, 0x7c9e, 0x7ca8,
1204 0x7ca1, 0x7d58, 0x7d6f, 0x7d63, 0x7d53, 0x7d56, 0x7d67, 0x7d6a,
1205 0x7d4f, 0x7d6d, 0x7d5c, 0x7d6b, 0x7d52, 0x7d54, 0x7d69, 0x7d51,
1206 0x7d5f, 0x7d4e, 0x7f3e, 0x7f3f, 0x7f65,
1207 /* 0xdb */
1208 0x7f66, 0x7fa2, 0x7fa0, 0x7fa1, 0x7fd7, 0x8051, 0x804f, 0x8050,
1209 0x80fe, 0x80d4, 0x8143, 0x814a, 0x8152, 0x814f, 0x8147, 0x813d,
1210 0x814d, 0x813a, 0x81e6, 0x81ee, 0x81f7, 0x81f8, 0x81f9, 0x8204,
1211 0x823c, 0x823d, 0x823f, 0x8275, 0x833b, 0x83cf, 0x83f9, 0x8423,
1212 0x83c0, 0x83e8, 0x8412, 0x83e7, 0x83e4, 0x83fc, 0x83f6, 0x8410,
1213 0x83c6, 0x83c8, 0x83eb, 0x83e3, 0x83bf, 0x8401, 0x83dd, 0x83e5,
1214 0x83d8, 0x83ff, 0x83e1, 0x83cb, 0x83ce, 0x83d6, 0x83f5, 0x83c9,
1215 0x8409, 0x840f, 0x83de, 0x8411, 0x8406, 0x83c2, 0x83f3, 0x83d5,
1216 0x83fa, 0x83c7, 0x83d1, 0x83ea, 0x8413, 0x83c3, 0x83ec, 0x83ee,
1217 0x83c4, 0x83fb, 0x83d7, 0x83e2, 0x841b, 0x83db, 0x83fe, 0x86d8,
1218 0x86e2, 0x86e6, 0x86d3, 0x86e3, 0x86da, 0x86ea, 0x86dd, 0x86eb,
1219 0x86dc, 0x86ec, 0x86e9, 0x86d7, 0x86e8, 0x86d1, 0x8848, 0x8856,
1220 0x8855, 0x88ba, 0x88d7, 0x88b9, 0x88b8, 0x88c0, 0x88be, 0x88b6,
1221 0x88bc, 0x88b7, 0x88bd, 0x88b2, 0x8901, 0x88c9, 0x8955, 0x8998,
1222 0x8997, 0x89dd, 0x89da, 0x89db, 0x8a4e, 0x8a4d, 0x8a39, 0x8a59,
1223 0x8a40, 0x8a57, 0x8a58, 0x8a44, 0x8a45, 0x8a52, 0x8a48, 0x8a51,
1224 0x8a4a, 0x8a4c, 0x8a4f, 0x8c5f, 0x8c81, 0x8c80, 0x8cba, 0x8cbe,
1225 0x8cb0, 0x8cb9, 0x8cb5, 0x8d84, 0x8d80, 0x8d89, 0x8dd8, 0x8dd3,
1226 0x8dcd, 0x8dc7, 0x8dd6, 0x8ddc, 0x8dcf, 0x8dd5, 0x8dd9, 0x8dc8,
1227 0x8dd7, 0x8dc5, 0x8eef, 0x8ef7, 0x8efa,
1228 /* 0xdc */
1229 0x8ef9, 0x8ee6, 0x8eee, 0x8ee5, 0x8ef5, 0x8ee7, 0x8ee8, 0x8ef6,
1230 0x8eeb, 0x8ef1, 0x8eec, 0x8ef4, 0x8ee9, 0x902d, 0x9034, 0x902f,
1231 0x9106, 0x912c, 0x9104, 0x90ff, 0x90fc, 0x9108, 0x90f9, 0x90fb,
1232 0x9101, 0x9100, 0x9107, 0x9105, 0x9103, 0x9161, 0x9164, 0x9155,
1233 0x9162, 0x9160, 0x9201, 0x920a, 0x9225, 0x9203, 0x921a, 0x9226,
1234 0x920f, 0x920c, 0x9200, 0x9212, 0x91ff, 0x91fd, 0x9206, 0x9204,
```



```
1235 0x9227, 0x9202, 0x921c, 0x9224, 0x9219, 0x9217, 0x9205, 0x9216,
1236 0x957b, 0x958d, 0x958c, 0x9590, 0x9687, 0x967e, 0x9688, 0x9689,
1237 0x9683, 0x9680, 0x96c2, 0x96c8, 0x96c3, 0x96f1, 0x96f0, 0x976c,
1238 0x9770, 0x976e, 0x9807, 0x98a9, 0x98eb, 0x99ce6, 0x99ef9, 0x4e83,
1239 0x4e84, 0x4e86, 0x50bd, 0x50bf, 0x50c6, 0x50ae, 0x50c4, 0x50ca,
1240 0x50b4, 0x50c8, 0x50c2, 0x50c2, 0x50b0, 0x50c1, 0x50ba, 0x50b1, 0x50cb,
1241 0x50c9, 0x50b6, 0x50b8, 0x51d7, 0x527a, 0x5278, 0x527b, 0x527c,
1242 0x55c3, 0x55db, 0x55cc, 0x55d0, 0x55cb, 0x55ca, 0x55dd, 0x55c0,
1243 0x55d4, 0x55c4, 0x55e9, 0x55e9, 0x55bf, 0x55d2, 0x558d, 0x55cf, 0x55d5,
1244 0x55e2, 0x55d6, 0x55c8, 0x55f2, 0x55cd, 0x55d9, 0x55c2, 0x5714,
1245 0x5853, 0x5868, 0x5864, 0x584f, 0x584d, 0x5849, 0x586f, 0x5855,
1246 0x584e, 0x585d, 0x5859, 0x5865, 0x585b, 0x583d, 0x5863, 0x5871,
1247 0x58fc, 0x5ac7, 0x5ac4, 0x5acb, 0x5aba, 0x5ab8, 0x5ab1, 0x5ab5,
1248 0x5ab0, 0x5abf, 0x5ac8, 0x5abb, 0x5ac6,
1249 /* 0xdd */
1250 0x5ab7, 0x5ac0, 0x5aca, 0x5ab4, 0x5ab6, 0x5acd, 0x5ab9, 0x5a90,
1251 0x5bd6, 0x5bd8, 0x5bd9, 0x5c1f, 0x5c33, 0x5d71, 0x5d63, 0x5d4a,
1252 0x5d65, 0x5d72, 0x5d6c, 0x5d5e, 0x5d68, 0x5d67, 0x5d62, 0x5df0,
1253 0x5e4f, 0x5e4e, 0x5e4a, 0x5e4d, 0x5e4b, 0x5ec5, 0x5ecc, 0x5ec6,
1254 0x5ecb, 0x5ec7, 0x5f40, 0x5faf, 0x5fad, 0x60f7, 0x6149, 0x614a,
1255 0x612b, 0x6145, 0x6136, 0x6132, 0x612e, 0x6146, 0x612f, 0x614f,
1256 0x6129, 0x6140, 0x6220, 0x9168, 0x6223, 0x6225, 0x6224, 0x63c5,
1257 0x63f1, 0x63eb, 0x6410, 0x6412, 0x6409, 0x6420, 0x6424, 0x6433,
1258 0x6443, 0x641f, 0x6415, 0x6418, 0x6418, 0x6439, 0x6437, 0x6422, 0x6423,
1259 0x640c, 0x6426, 0x6430, 0x6428, 0x6441, 0x6435, 0x642f, 0x640a,
1260 0x641a, 0x6440, 0x6425, 0x6427, 0x640b, 0x63e7, 0x641b, 0x642e,
1261 0x6421, 0x640e, 0x656f, 0x6592, 0x65d3, 0x6686, 0x668c, 0x6695,
1262 0x6690, 0x668b, 0x668a, 0x6699, 0x6694, 0x6678, 0x6720, 0x6966,
1263 0x695f, 0x6938, 0x694e, 0x6962, 0x6971, 0x693f, 0x6945, 0x696a,
1264 0x6939, 0x6942, 0x6957, 0x6957, 0x6959, 0x697a, 0x6948, 0x6949, 0x6935,
1265 0x696c, 0x6933, 0x693d, 0x6965, 0x68f0, 0x6978, 0x6934, 0x6969,
1266 0x6940, 0x696f, 0x6944, 0x6976, 0x6958, 0x6941, 0x6974, 0x694c,
1267 0x693b, 0x694b, 0x6937, 0x695c, 0x694f, 0x6951, 0x6932, 0x6952,
1268 0x692f, 0x697b, 0x693c, 0x6b46, 0x6b45, 0x6b43, 0x6b42, 0x6b48,
1269 0x6b41, 0x6b9b, 0xfa0d, 0x6bfb, 0x6bfc,
1270 /* 0xde */
1271 0x6bf9, 0x6bf7, 0x6bf8, 0x6e9b, 0x6ed6, 0x6ec8, 0x6e8f, 0x6ec0,
1272 0x6e9f, 0x6e93, 0x6e94, 0x6ea0, 0x6eb1, 0x6eb9, 0x6ec6, 0x6ed2,
1273 0x6ebd, 0x6ec1, 0x6e9e, 0x6ec9, 0x6eb7, 0x6eb0, 0x6ecd, 0x6ea6,
1274 0x6ecf, 0x6eb2, 0x6ebe, 0x6ec3, 0x6edc, 0x6ed8, 0x6e99, 0x6e92,
1275 0x6e8e, 0x6e8d, 0x6ea4, 0x6ea1, 0x6ebf, 0x6eb3, 0x6ed0, 0x6eca,
1276 0x6e97, 0x6ea8, 0x6ea3, 0x7147, 0x7154, 0x7152, 0x7163, 0x7160,
1277 0x7141, 0x715d, 0x7162, 0x7172, 0x7178, 0x716a, 0x7161, 0x7142,
1278 0x7158, 0x7143, 0x714b, 0x7170, 0x715f, 0x7150, 0x7153, 0x7144,
1279 0x714d, 0x715a, 0x724f, 0x724f, 0x728d, 0x728c, 0x7291, 0x7290, 0x728e,
1280 0x733c, 0x7342, 0x733b, 0x733a, 0x7340, 0x734a, 0x7349, 0x7444,
1281 0x744a, 0x744b, 0x7452, 0x7451, 0x7457, 0x7440, 0x744f, 0x7450,
1282 0x744e, 0x7442, 0x7446, 0x744d, 0x7454, 0x74e1, 0x74ff, 0x74fe,
1283 0x74fd, 0x751d, 0x7579, 0x7577, 0x6983, 0x75ef, 0x760f, 0x7603,
1284 0x75f7, 0x75fe, 0x75fc, 0x75f9, 0x75f8, 0x7610, 0x75fb, 0x75f6,
1285 0x75ed, 0x75f5, 0x75fd, 0x7699, 0x76b5, 0x76dd, 0x7755, 0x775e,
1286 0x7760, 0x7752, 0x7756, 0x775a, 0x7769, 0x7767, 0x7754, 0x7759,
1287 0x776d, 0x77e0, 0x7887, 0x789a, 0x789a, 0x7899, 0x788f, 0x7884, 0x7895,
1288 0x7885, 0x7886, 0x78a1, 0x7883, 0x7879, 0x7899, 0x7880, 0x7896,
1289 0x787b, 0x797c, 0x7982, 0x797d, 0x7979, 0x7a11, 0x7a18, 0x7a19,
1290 0x7a12, 0x7a17, 0x7a15, 0x7a22, 0x7a13,
1291 /* 0xdf */
1292 0x7a1b, 0x7a10, 0x7aa3, 0x7aa2, 0x7a9e, 0x7aeb, 0x7b66, 0x7b64,
1293 0x7b6d, 0x7b74, 0x7b69, 0x7b72, 0x7b65, 0x7b73, 0x7b71, 0x7b70,
1294 0x7b61, 0x7b78, 0x7b76, 0x7b63, 0x7cb2, 0x7cb4, 0x7caf, 0x7d88,
1295 0x7d86, 0x7d80, 0x7d8d, 0x7d7f, 0x7d85, 0x7d7a, 0x7d8e, 0x7d7b,
1296 0x7d83, 0x7d7c, 0x7d8c, 0x7d94, 0x7d84, 0x7d7d, 0x7d92, 0x7f6d,
1297 0x7f6b, 0x7f67, 0x7f68, 0x7f6c, 0x7fa6, 0x7fa5, 0x7fa7, 0x7fdb,
1298 0x7fdc, 0x8021, 0x8164, 0x8160, 0x8177, 0x815c, 0x8169, 0x815b,
1299 0x8162, 0x8172, 0x6721, 0x815e, 0x8176, 0x8167, 0x816f, 0x8144,
1300 0x8161, 0x821d, 0x8249, 0x8244, 0x8240, 0x8242, 0x8245, 0x84f1,
1301 0x843f, 0x8456, 0x8476, 0x8479, 0x848f, 0x848d, 0x8465, 0x8451,
1302 0x8440, 0x8486, 0x8467, 0x8430, 0x844d, 0x847d, 0x845a, 0x8459,
1303 0x8474, 0x8473, 0x845d, 0x8507, 0x845e, 0x8437, 0x843a, 0x8434,
1304 0x847a, 0x8443, 0x8478, 0x8432, 0x8445, 0x8429, 0x83d9, 0x844b,
1305 0x842f, 0x8442, 0x842d, 0x845f, 0x8470, 0x8439, 0x844e, 0x844c,
1306 0x8452, 0x846f, 0x84c5, 0x848e, 0x843b, 0x8447, 0x8436, 0x8433,
1307 0x8468, 0x847e, 0x8444, 0x842b, 0x8460, 0x8454, 0x846e, 0x8450,
1308 0x870b, 0x8704, 0x86f7, 0x870c, 0x86fa, 0x86d6, 0x86f5, 0x874d,
1309 0x86f8, 0x870e, 0x8709, 0x8701, 0x86f6, 0x870d, 0x8705, 0x88d6,
1310 0x88cb, 0x88cd, 0x88ce, 0x88de, 0x88de, 0x88da, 0x88cc, 0x88d0,
1311 0x8985, 0x899b, 0x89df, 0x89e5, 0x89e4,
1312 /* 0xe0 */
1313 0x89e1, 0x89e0, 0x89e2, 0x89dc, 0x89e6, 0x8a76, 0x8a86, 0x8a7f,
1314 0x8a61, 0x8a3f, 0x8a77, 0x8a82, 0x8a84, 0x8a75, 0x8a83, 0x8a81,
1315 0x8a74, 0x8a7a, 0x8c3c, 0x8c4b, 0x8c4a, 0x8c65, 0x8c64, 0x8c66,
1316 0x8c86, 0x8c84, 0x8c85, 0x8ccc, 0x8d68, 0x8d69, 0x8d91, 0x8d8c,
1317 0x8d8e, 0x8d8f, 0x8d8d, 0x8d93, 0x8d94, 0x8d90, 0x8d92, 0x8df0,
1318 0x8de0, 0x8dec, 0x8df1, 0x8dee, 0x8dd0, 0x8de9, 0x8de3, 0x8de2,
1319 0x8de7, 0x8df2, 0x8deb, 0x8df4, 0x8f06, 0x8eff, 0x8f01, 0x8f00,
1320 0x8f05, 0x8f07, 0x8f08, 0x8f02, 0x8f0b, 0x9052, 0x903f, 0x9044,
1321 0x9049, 0x903d, 0x9110, 0x910d, 0x910f, 0x9111, 0x9116, 0x9114,
```

```
1322 0x910b, 0x910e, 0x916e, 0x916f, 0x9248, 0x9252, 0x9230, 0x923a,
1323 0x9266, 0x9233, 0x9265, 0x925e, 0x9283, 0x922e, 0x924a, 0x9246,
1324 0x926d, 0x926c, 0x924f, 0x924f, 0x9260, 0x9267, 0x926f, 0x9236, 0x9261,
1325 0x9270, 0x9231, 0x9254, 0x9263, 0x9250, 0x9272, 0x924e, 0x9253,
1326 0x924c, 0x9256, 0x9232, 0x959f, 0x959c, 0x959e, 0x959b, 0x9692,
1327 0x9693, 0x9691, 0x9697, 0x9697, 0x96ce, 0x96fa, 0x96fd, 0x96f8, 0x96f5,
1328 0x9773, 0x9777, 0x9778, 0x9772, 0x980f, 0x980d, 0x980e, 0x98ac,
1329 0x98f6, 0x98f9, 0x99af, 0x99b2, 0x99b0, 0x99b5, 0x9aad, 0x9aab,
1330 0x9b5b, 0x9cea, 0x9ced, 0x9ce7, 0x9e80, 0x9efd, 0x50e6, 0x50d4,
1331 0x50d7, 0x50e8, 0x50f3, 0x50db, 0x50ea, 0x50dd, 0x50e4, 0x50d3,
1332 0x50ec, 0x50f0, 0x50ef, 0x50e3, 0x50e0,
1333 /* 0xe1 */
1334 0x51d8, 0x5280, 0x5281, 0x52e9, 0x52eb, 0x5330, 0x53ac, 0x5627,
1335 0x5615, 0x560c, 0x5612, 0x55fc, 0x560f, 0x561c, 0x5601, 0x5613,
1336 0x5602, 0x55fa, 0x561d, 0x5604, 0x55ff, 0x55f9, 0x5889, 0x587c,
1337 0x5890, 0x5898, 0x5886, 0x5881, 0x587f, 0x5874, 0x588b, 0x587a,
1338 0x5887, 0x5891, 0x588e, 0x5876, 0x5882, 0x5888, 0x587b, 0x5894,
1339 0x588f, 0x586a, 0x596b, 0x596b, 0x5adc, 0x5aee, 0x5ae5, 0x5ad5, 0x5aea,
1340 0x5ada, 0x5aed, 0x5aeb, 0x5af3, 0x5ae2, 0x5ae0, 0x5adb, 0x5aec,
1341 0x5ade, 0x5add, 0x5ad9, 0x5ae8, 0x5adf, 0x5b77, 0x5be0, 0x5be3,
1342 0x5c63, 0x5d82, 0x5d80, 0x5d80, 0x5d7d, 0x5d86, 0x5d7a, 0x5d81, 0x5d77,
1343 0x5d8a, 0x5d89, 0x5d88, 0x5d7e, 0x5d7c, 0x5d8d, 0x5d79, 0x5d7f,
1344 0x5e58, 0x5e59, 0x5e53, 0x5ed8, 0x5ed1, 0x5ed7, 0x5ece, 0x5edc,
1345 0x5ed5, 0x5ed9, 0x5ed2, 0x5ed4, 0x5f44, 0x5f43, 0x5f6f, 0x5fb6,
1346 0x612c, 0x6128, 0x6141, 0x615e, 0x6171, 0x6173, 0x6152, 0x6153,
1347 0x6172, 0x616c, 0x6180, 0x6174, 0x6154, 0x617a, 0x615b, 0x6165,
1348 0x613b, 0x616a, 0x6161, 0x6156, 0x6229, 0x6227, 0x622b, 0x642b,
1349 0x644d, 0x645b, 0x645d, 0x6474, 0x6476, 0x6472, 0x6473, 0x647d,
1350 0x6475, 0x6466, 0x64a6, 0x644e, 0x6482, 0x645e, 0x645c, 0x644b,
1351 0x6453, 0x6460, 0x6450, 0x6450, 0x647f, 0x646c, 0x646b, 0x6459,
1352 0x6465, 0x6477, 0x6573, 0x65a0, 0x66a1, 0x66a0, 0x669f, 0x6705,
1353 0x6704, 0x6722, 0x69b1, 0x69b6, 0x69c9,
1354 /* 0xe2 */
1355 0x69a0, 0x69ce, 0x6996, 0x69b0, 0x69ac, 0x69bc, 0x6991, 0x6999,
1356 0x698e, 0x69a7, 0x698d, 0x69a9, 0x69be, 0x69af, 0x69bf, 0x69c4,
1357 0x69bd, 0x69a4, 0x69d4, 0x69b9, 0x69ca, 0x699a, 0x69cf, 0x69b3,
1358 0x6993, 0x69aa, 0x69a1, 0x699e, 0x69d9, 0x6997, 0x6990, 0x69c2,
1359 0x69b5, 0x69a5, 0x69c6, 0x6b4a, 0x6b4d, 0x6b4b, 0x6b9e, 0x6b9f,
1360 0x6ba0, 0x6bc3, 0x6bc4, 0x6bfe, 0x6ece, 0x6ef5, 0x6ef1, 0x6f03,
1361 0x6f25, 0x6ef8, 0x6f37, 0x6efb, 0x6f2e, 0x6f09, 0x6f4e, 0x6f19,
1362 0x6f1a, 0x6f27, 0x6f18, 0x6f3b, 0x6f12, 0x6eed, 0x6f0a, 0x6f36,
1363 0x6f73, 0x6ef9, 0x6eee, 0x6f2d, 0x6f40, 0x6f30, 0x6f3c, 0x6f35,
1364 0x6eeb, 0x6f07, 0x6f0e, 0x6f43, 0x6f05, 0x6efd, 0x6ef6, 0x6f39,
1365 0x6f1c, 0x6efc, 0x6f3a, 0x6f1f, 0x6f0d, 0x6f1e, 0x6f08, 0x6f21,
1366 0x7187, 0x7190, 0x7189, 0x7189, 0x7180, 0x7185, 0x7182, 0x718f, 0x717b,
1367 0x7186, 0x7181, 0x7197, 0x7244, 0x7253, 0x7297, 0x7295, 0x7293,
1368 0x7343, 0x734d, 0x7351, 0x734c, 0x7462, 0x7473, 0x7471, 0x7475,
1369 0x7472, 0x7467, 0x746e, 0x7500, 0x7502, 0x7503, 0x757d, 0x7590,
1370 0x7616, 0x7608, 0x760c, 0x7615, 0x7611, 0x760a, 0x7614, 0x76b8,
1371 0x7781, 0x777c, 0x7785, 0x7782, 0x776e, 0x7780, 0x776f, 0x777e,
1372 0x7783, 0x78b2, 0x78aa, 0x78b4, 0x78ad, 0x78a8, 0x787e, 0x78ab,
1373 0x789e, 0x78a5, 0x78a0, 0x78ac, 0x78a2, 0x78a4, 0x7998, 0x798a,
1374 0x798b, 0x7996, 0x7995, 0x7994, 0x7993,
1375 /* 0xe3 */
1376 0x7997, 0x7988, 0x7992, 0x7990, 0x7a2b, 0x7a4a, 0x7a30, 0x7a2f,
1377 0x7a28, 0x7a26, 0x7aa8, 0x7aab, 0x7aac, 0x7aee, 0x7b88, 0x7b9c,
1378 0x7b8a, 0x7b91, 0x7b90, 0x7b96, 0x7b8d, 0x7b8c, 0x7b9b, 0x7b8e,
1379 0x7b85, 0x7b98, 0x5284, 0x7b99, 0x7ba4, 0x7b82, 0x7cbb, 0x7cbf,
1380 0x7cbc, 0x7cba, 0x7da7, 0x7db7, 0x7dc2, 0x7da3, 0x7daa, 0x7dc1,
1381 0x7dc0, 0x7dc5, 0x7d9d, 0x7dce, 0x7dc4, 0x7dc6, 0x7dcb, 0x7dcc,
1382 0x7daf, 0x7db9, 0x7d96, 0x7dbc, 0x7d9f, 0x7da6, 0x7dae, 0x7da9,
1383 0x7dal, 0x7dc9, 0x7f73, 0x7fe2, 0x7fe3, 0x7fe5, 0x7fde, 0x8024,
1384 0x805d, 0x805c, 0x8189, 0x8186, 0x8183, 0x8187, 0x818d, 0x818c,
1385 0x818b, 0x8215, 0x8497, 0x84a4, 0x84a1, 0x849f, 0x84ba, 0x84ce,
1386 0x84c2, 0x84ac, 0x84ae, 0x84ab, 0x84b9, 0x84b4, 0x84c1, 0x84cd,
1387 0x84aa, 0x849a, 0x84b1, 0x84d0, 0x849d, 0x84a7, 0x84bb, 0x84a2,
1388 0x8494, 0x84c7, 0x84cc, 0x849b, 0x84a9, 0x84af, 0x84a8, 0x84d6,
1389 0x8498, 0x84b6, 0x84cf, 0x84a0, 0x84d7, 0x84d4, 0x84d2, 0x84db,
1390 0x84b0, 0x8491, 0x8661, 0x8733, 0x8723, 0x8728, 0x876b, 0x8740,
1391 0x872e, 0x871e, 0x8721, 0x8719, 0x871b, 0x8743, 0x872c, 0x8741,
1392 0x873e, 0x8746, 0x8720, 0x8732, 0x872a, 0x872d, 0x873c, 0x8712,
1393 0x873a, 0x8731, 0x8735, 0x8742, 0x8726, 0x8727, 0x8738, 0x8724,
1394 0x871a, 0x8730, 0x8711, 0x88f7, 0x88e7, 0x88f1, 0x88f2, 0x88fa,
1395 0x88fe, 0x88ee, 0x88fc, 0x88f6, 0x88fb,
1396 /* 0xe4 */
1397 0x88f0, 0x88ec, 0x88eb, 0x889d, 0x89a1, 0x899f, 0x899e, 0x89e9,
1398 0x89eb, 0x89e8, 0x8aab, 0x8a99, 0x8a8b, 0x8a92, 0x8a8f, 0x8a96,
1399 0x8c3d, 0x8c68, 0x8c69, 0x8cd5, 0x8ccf, 0x8cd7, 0x8d96, 0x8e09,
1400 0x8e02, 0x8dff, 0x8e0d, 0x8dfd, 0x8e0a, 0x8e03, 0x8e07, 0x8e06,
1401 0x8e05, 0x8dfe, 0x8e00, 0x8e04, 0x8f10, 0x8f11, 0x8f0e, 0x8f0d,
1402 0x9123, 0x911c, 0x9120, 0x9122, 0x911f, 0x911d, 0x911a, 0x9124,
1403 0x9121, 0x911b, 0x917a, 0x9172, 0x9179, 0x9173, 0x92a5, 0x92a4,
1404 0x9276, 0x929b, 0x927a, 0x92a0, 0x9294, 0x92a1, 0x928d, 0x92a6,
1405 0x929a, 0x92ab, 0x9279, 0x9297, 0x927f, 0x92a3, 0x92ee, 0x928e,
1406 0x9282, 0x9295, 0x92a2, 0x927d, 0x9288, 0x92a1, 0x928a, 0x928b,
1407 0x928c, 0x9299, 0x92a7, 0x927e, 0x9287, 0x929d, 0x928d,
1408 0x922d, 0x969e, 0x96a1, 0x96ff, 0x9758, 0x977d, 0x977a, 0x977e,
```



```

1583 0x66d2, 0x6a8d, 0x6a96, 0x6a81, 0x6aa5, 0x6a89, 0x6a9f, 0x6a9b,
1584 0x6aa1, 0x6a9e, 0x6a87, 0x6a93, 0x6a8e,
1585 /* 0xed */
1586 0x6a95, 0x6a83, 0x6aa8, 0x6aa4, 0x6a91, 0x6a7f, 0x6aa6, 0x6a9a,
1587 0x6a85, 0x6a8c, 0x6a92, 0x6b5b, 0x6bad, 0x6c09, 0x6fcc, 0x6fa9,
1588 0x6ff4, 0x6fd4, 0x6fe3, 0x6fdc, 0x6fed, 0x6fe7, 0x6fe6, 0x6fde,
1589 0x6ff2, 0x6fdd, 0x6fe2, 0x6fe8, 0x71e1, 0x71f1, 0x71e8, 0x71f2,
1590 0x71e4, 0x71f0, 0x71e2, 0x7373, 0x736e, 0x736f, 0x7497, 0x74b2,
1591 0x74ab, 0x7490, 0x74aa, 0x74ad, 0x74b1, 0x74a5, 0x74af, 0x7510,
1592 0x7511, 0x7512, 0x750f, 0x7584, 0x7643, 0x7648, 0x7649, 0x7647,
1593 0x76a4, 0x76e9, 0x77b5, 0x77ab, 0x77b2, 0x77b7, 0x77b6, 0x77b4,
1594 0x77b1, 0x77a8, 0x77f0, 0x78f3, 0x78fd, 0x7902, 0x78fb, 0x78fc,
1595 0x78f2, 0x7905, 0x78f9, 0x78fe, 0x7904, 0x79ab, 0x79a8, 0x7a5c,
1596 0x7a5b, 0x7a56, 0x7a58, 0x7a54, 0x7a5a, 0x7abe, 0x7ac0, 0x7ac1,
1597 0x7c05, 0x7c0f, 0x7bf2, 0x7c00, 0x7bff, 0x7bfb, 0x7c0e, 0x7bf4,
1598 0x7c0b, 0x7bf3, 0x7c02, 0x7c09, 0x7c03, 0x7c01, 0x7bf8, 0x7bfd,
1599 0x7c06, 0x7bf0, 0x7bf1, 0x7c10, 0x7c0a, 0x7ce8, 0x7e2d, 0x7e3c,
1600 0x7e42, 0x7e33, 0x9848, 0x7e38, 0x7e2a, 0x7e49, 0x7e40, 0x7e47,
1601 0x7e29, 0x7e4c, 0x7e30, 0x7e3b, 0x7e36, 0x7e44, 0x7e3a, 0x7f45,
1602 0x7f7f, 0x7f7e, 0x7f7d, 0x7ff4, 0x7ff2, 0x802c, 0x81bb, 0x81c4,
1603 0x81cc, 0x81ca, 0x81c5, 0x81c7, 0x81bc, 0x81e9, 0x825b, 0x825a,
1604 0x825c, 0x8583, 0x8580, 0x858f, 0x85a7, 0x8595, 0x85a0, 0x858b,
1605 0x85a3, 0x857b, 0x85a4, 0x859a, 0x859e,
1606 /* 0xee */
1607 0x8577, 0x857c, 0x8589, 0x85a1, 0x857a, 0x8578, 0x8557, 0x858e,
1608 0x8596, 0x8586, 0x858d, 0x8599, 0x859d, 0x8581, 0x85a2, 0x8582,
1609 0x8588, 0x8585, 0x8579, 0x8576, 0x8598, 0x8590, 0x859f, 0x8668,
1610 0x87be, 0x87aa, 0x87ad, 0x87c5, 0x87b0, 0x87ac, 0x87b9, 0x87b5,
1611 0x87bc, 0x87ae, 0x87c9, 0x87c3, 0x87c2, 0x87cc, 0x87b7, 0x87af,
1612 0x87c4, 0x87ca, 0x87b4, 0x87b6, 0x87bf, 0x87b8, 0x87bd, 0x87de,
1613 0x87b2, 0x8935, 0x8933, 0x893c, 0x893e, 0x8941, 0x8952, 0x8937,
1614 0x8942, 0x89ad, 0x89af, 0x89ae, 0x89f2, 0x89f3, 0x8b1e, 0x8b18,
1615 0x8b16, 0x8b11, 0x8b05, 0x8b0b, 0x8b22, 0x8b0f, 0x8b12, 0x8b15,
1616 0x8b07, 0x8b0d, 0x8b08, 0x8b06, 0x8b1c, 0x8b13, 0x8b1a, 0x8c4f,
1617 0x8c70, 0x8c72, 0x8c71, 0x8c6f, 0x8c95, 0x8c94, 0x8cf9, 0x8d6f,
1618 0x8e4e, 0x8e4d, 0x8e53, 0x8e50, 0x8e4c, 0x8e47, 0x8f43, 0x8f40,
1619 0x9085, 0x907e, 0x9138, 0x919a, 0x91a2, 0x919b, 0x9199, 0x919f,
1620 0x91a1, 0x919d, 0x91a0, 0x93a1, 0x9383, 0x93af, 0x9364, 0x9356,
1621 0x9347, 0x937c, 0x9358, 0x935c, 0x9376, 0x9349, 0x9350, 0x9351,
1622 0x9360, 0x936d, 0x938f, 0x934c, 0x936a, 0x9379, 0x9357, 0x9355,
1623 0x9352, 0x934f, 0x9371, 0x9377, 0x937b, 0x9361, 0x935e, 0x9363,
1624 0x9367, 0x9380, 0x934e, 0x9359, 0x95c7, 0x95c0, 0x95c9, 0x95c3,
1625 0x95c5, 0x95b7, 0x96ae, 0x96b0, 0x96ac, 0x9720, 0x971f, 0x9718,
1626 0x971d, 0x9719, 0x979a, 0x97a1, 0x979c,
1627 /* 0xef */
1628 0x979e, 0x979d, 0x97d5, 0x97d4, 0x97f1, 0x9841, 0x9844, 0x984a,
1629 0x9849, 0x9845, 0x9843, 0x9925, 0x992b, 0x992c, 0x992a, 0x9933,
1630 0x9932, 0x992f, 0x992d, 0x9931, 0x9930, 0x9998, 0x99a3, 0x99a1,
1631 0x9a02, 0x99fa, 0x99f4, 0x99f7, 0x99f9, 0x99f8, 0x99f6, 0x99fb,
1632 0x99fd, 0x99fe, 0x99fc, 0x9a03, 0x9abe, 0x9afe, 0x9afd, 0x9b01,
1633 0x9afc, 0x9b48, 0x9b9a, 0x9ba8, 0x9b9e, 0x9b9b, 0x9ba6, 0x9ba1,
1634 0x9ba5, 0x9ba4, 0x9b86, 0x9ba2, 0x9ba0, 0x9baf, 0x9d33, 0x9d41,
1635 0x9d67, 0x9d36, 0x9d2e, 0x9d2f, 0x9d31, 0x9d38, 0x9d30, 0x9d45,
1636 0x9d42, 0x9d43, 0x9d3e, 0x9d37, 0x9d40, 0x9d3d, 0x7ff5, 0x9d2d,
1637 0x9e8a, 0x9e89, 0x9e8d, 0x9eb0, 0x9ec8, 0x9eda, 0x9efb, 0x9eff,
1638 0x9f24, 0x9f23, 0x9f22, 0x9f54, 0x9fa0, 0x5131, 0x512d, 0x512e,
1639 0x5698, 0x569c, 0x5697, 0x5699, 0x569d, 0x5699, 0x5970, 0x5b3c,
1640 0x5c69, 0x5c6a, 0x5dc0, 0x5e6d, 0x5e6e, 0x61d8, 0x61df, 0x61ed,
1641 0x61ee, 0x61f1, 0x61ea, 0x61f0, 0x61eb, 0x61d6, 0x61e9, 0x64ff,
1642 0x6504, 0x64fd, 0x64f8, 0x6501, 0x6503, 0x64fc, 0x6594, 0x65db,
1643 0x66da, 0x66db, 0x66d8, 0x6ac5, 0x6ab9, 0x6abd, 0x6ae1, 0x6ac6,
1644 0x6aba, 0x6ab6, 0x6ab7, 0x6ac7, 0x6ab4, 0x6aad, 0x6b5e, 0x6bc9,
1645 0x6c0b, 0x7007, 0x700c, 0x700d, 0x7001, 0x7005, 0x7014, 0x700e,
1646 0x6fff, 0x7000, 0x6ffb, 0x7026, 0x6ffc, 0x6ff7, 0x700a, 0x7201,
1647 0x71ff, 0x71f9, 0x7203, 0x71fd, 0x7376,
1648 /* 0xf0 */
1649 0x74b8, 0x74c0, 0x74b5, 0x74c1, 0x74be, 0x74b6, 0x74bb, 0x74c2,
1650 0x7514, 0x7513, 0x765c, 0x7664, 0x7659, 0x7650, 0x7653, 0x7657,
1651 0x765a, 0x76a6, 0x76bd, 0x76ec, 0x77c2, 0x77ba, 0x78ff, 0x790c,
1652 0x7913, 0x7914, 0x7909, 0x7910, 0x7912, 0x791f, 0x79ad, 0x79ac,
1653 0x7a5f, 0x7c1c, 0x7c29, 0x7c19, 0x7c20, 0x7c1f, 0x7c2d, 0x7c1d,
1654 0x7c26, 0x7c28, 0x7c22, 0x7c25, 0x7c30, 0x7e5c, 0x7e50, 0x7e56,
1655 0x7e63, 0x7e58, 0x7e62, 0x7e5f, 0x7e51, 0x7e60, 0x7e57, 0x7e53,
1656 0x7fb5, 0x7fb3, 0x7ff7, 0x7ff8, 0x8075, 0x81d1, 0x81d2, 0x81d0,
1657 0x825f, 0x825e, 0x85b4, 0x85c6, 0x85c0, 0x85c3, 0x85c2, 0x85b3,
1658 0x85b5, 0x85bd, 0x85c7, 0x85c4, 0x85bf, 0x85cb, 0x85ce, 0x85c8,
1659 0x85c5, 0x85b1, 0x85b6, 0x85d2, 0x8624, 0x85b8, 0x85b7, 0x85be,
1660 0x8669, 0x87e7, 0x87e6, 0x87e2, 0x87db, 0x87eb, 0x87ea, 0x87e5,
1661 0x87df, 0x87f3, 0x87e4, 0x87d4, 0x87dc, 0x87d3, 0x87ed, 0x87d8,
1662 0x87e3, 0x87a4, 0x87d7, 0x87d9, 0x8801, 0x87f4, 0x87e8, 0x87dd,
1663 0x8953, 0x894b, 0x894f, 0x894c, 0x8946, 0x8950, 0x8951, 0x8949,
1664 0x8b2a, 0x8b27, 0x8b23, 0x8b33, 0x8b30, 0x8b35, 0x8b47, 0x8b2f,
1665 0x8b3c, 0x8b3e, 0x8b31, 0x8b25, 0x8b37, 0x8b26, 0x8b36, 0x8b2e,
1666 0x8b24, 0x8b3b, 0x8b3d, 0x8b3a, 0x8c42, 0x8c75, 0x8c99, 0x8c98,
1667 0x8c97, 0x8cfe, 0x8d04, 0x8d02, 0x8d00, 0x8e5c, 0x8e62, 0x8e60,
1668 0x8e57, 0x8e56, 0x8e5e, 0x8e65, 0x8e67,
1669 /* 0xf1 */

```

```
1670 0x8e5b, 0x8e5a, 0x8e61, 0x8e5d, 0x8e69, 0x8e54, 0x8f46, 0x8f47,
1671 0x8f48, 0x8f4b, 0x9128, 0x913a, 0x913b, 0x913e, 0x91a8, 0x91a5,
1672 0x91a7, 0x91af, 0x91aa, 0x93b5, 0x938c, 0x9392, 0x93b7, 0x939b,
1673 0x939d, 0x9389, 0x93a7, 0x938e, 0x93aa, 0x939e, 0x93a6, 0x9395,
1674 0x9388, 0x9399, 0x939f, 0x938d, 0x93b1, 0x9391, 0x93b2, 0x93a4,
1675 0x93a8, 0x93b4, 0x93a3, 0x93a5, 0x95d2, 0x95d3, 0x95d1, 0x96b3,
1676 0x96d7, 0x96da, 0x5dc2, 0x96df, 0x96d8, 0x96dd, 0x9723, 0x9722,
1677 0x9725, 0x97ac, 0x97ae, 0x97a8, 0x97ab, 0x97a4, 0x97aa, 0x97a2,
1678 0x97a5, 0x97d7, 0x97d9, 0x97d6, 0x97d8, 0x97fa, 0x9850, 0x9851,
1679 0x9852, 0x98b8, 0x9941, 0x993c, 0x993a, 0x9a0f, 0x9a0b, 0x9a09,
1680 0x9a0d, 0x9a04, 0x9a11, 0x9a0a, 0x9a05, 0x9a07, 0x9a06, 0x9ac0,
1681 0x9adc, 0x9b08, 0x9b04, 0x9b05, 0x9b29, 0x9b35, 0x9b4a, 0x9b4c,
1682 0x9b4b, 0x9bc7, 0x9bc6, 0x9bc3, 0x9bbf, 0x9bc1, 0x9bb5, 0x9bb8,
1683 0x9bd3, 0x9bb6, 0x9bc4, 0x9bb9, 0x9bbd, 0x9d5c, 0x9d53, 0x9d4f,
1684 0x9d4a, 0x9d5b, 0x9d4b, 0x9d59, 0x9d56, 0x9d4c, 0x9d57, 0x9d52,
1685 0x9d54, 0x9d5f, 0x9d58, 0x9d5a, 0x9e8e, 0x9e8c, 0x9edf, 0x9f01,
1686 0x9f00, 0x9f16, 0x9f25, 0x9f2b, 0x9f2a, 0x9f29, 0x9f28, 0x9f4c,
1687 0x9f55, 0x5134, 0x5135, 0x5296, 0x52f7, 0x53b4, 0x56ab, 0x56ad,
1688 0x56a6, 0x56a7, 0x56aa, 0x56ac, 0x58da, 0x58dd, 0x58db, 0x5912,
1689 0x5b3d, 0x5b3e, 0x5b3f, 0x5dc3, 0x5e70,
1690 /* 0xf2 */
1691 0x5fbf, 0x61fb, 0x6507, 0x6510, 0x650d, 0x6509, 0x650c, 0x650e,
1692 0x6584, 0x65de, 0x65dd, 0x66de, 0x6ae7, 0x6ae0, 0x6acc, 0x6ad1,
1693 0x6ad9, 0x6acb, 0x6adf, 0x6adc, 0x6ad0, 0x6aeb, 0x6acf, 0x6acd,
1694 0x6ade, 0x6b60, 0x6bb0, 0x6c0c, 0x7019, 0x7027, 0x7020, 0x7016,
1695 0x702b, 0x7021, 0x7022, 0x7023, 0x7029, 0x7017, 0x7024, 0x701c,
1696 0x702a, 0x720c, 0x720a, 0x720a, 0x7207, 0x7202, 0x7205, 0x72a5, 0x72a6,
1697 0x72a4, 0x72a3, 0x72a1, 0x74cb, 0x74c5, 0x74b7, 0x74c3, 0x7516,
1698 0x7660, 0x77c9, 0x77ca, 0x77c4, 0x77f1, 0x791d, 0x791b, 0x7921,
1699 0x791c, 0x7917, 0x791e, 0x791e, 0x79b0, 0x7a67, 0x7a68, 0x7c33, 0x7c3c,
1700 0x7c39, 0x7c2c, 0x7c3b, 0x7cec, 0x7cea, 0x7e76, 0x7e75, 0x7e78,
1701 0x7e70, 0x7e77, 0x7e6f, 0x7e7a, 0x7e72, 0x7e74, 0x7e68, 0x7f4b,
1702 0x7f4a, 0x7f83, 0x7f86, 0x7fb7, 0x7ffd, 0x7ffe, 0x8078, 0x81d7,
1703 0x81d5, 0x8264, 0x8261, 0x8263, 0x85eb, 0x85f1, 0x85ed, 0x85d9,
1704 0x85e1, 0x85e8, 0x85da, 0x85d7, 0x85ec, 0x85f2, 0x85f8, 0x85d8,
1705 0x85df, 0x85e3, 0x85dc, 0x85d1, 0x85f0, 0x85e6, 0x85ef, 0x85de,
1706 0x85e2, 0x8800, 0x87fa, 0x8803, 0x87f6, 0x87f7, 0x8809, 0x880c,
1707 0x880b, 0x8806, 0x87fc, 0x8808, 0x87ff, 0x880a, 0x8802, 0x8962,
1708 0x895a, 0x895b, 0x8957, 0x8957, 0x8961, 0x895c, 0x8958, 0x895d, 0x8959,
1709 0x8988, 0x89b7, 0x89b6, 0x89f6, 0x8b50, 0x8b48, 0x8b4a, 0x8b40,
1710 0x8b53, 0x8b56, 0x8b54, 0x8b4b, 0x8b55,
1711 /* 0xf3 */
1712 0x8b51, 0x8b42, 0x8b52, 0x8b57, 0x8c43, 0x8c77, 0x8c76, 0x8c9a,
1713 0x8d06, 0x8d07, 0x8d09, 0x8dac, 0x8daa, 0x8dad, 0x8dab, 0x8e6d,
1714 0x8e78, 0x8e73, 0x8e6a, 0x8e6f, 0x8e7b, 0x8ec2, 0x8f52, 0x8f51,
1715 0x8f4f, 0x8f50, 0x8f53, 0x8fb4, 0x9140, 0x913f, 0x91b0, 0x91ad,
1716 0x93de, 0x93cf, 0x93cf, 0x93c2, 0x93da, 0x93d0, 0x93f9, 0x93ec,
1717 0x93cc, 0x93d9, 0x93a9, 0x93e6, 0x93ca, 0x93d4, 0x93ee, 0x93e3,
1718 0x93d5, 0x93c4, 0x93ce, 0x93c0, 0x93d2, 0x93e7, 0x957d, 0x95da,
1719 0x95db, 0x96e1, 0x9729, 0x972b, 0x972c, 0x9728, 0x9726, 0x97b3,
1720 0x97b7, 0x97b6, 0x97dd, 0x97de, 0x97df, 0x985c, 0x9859, 0x985d,
1721 0x9857, 0x98bf, 0x98bd, 0x98bb, 0x98be, 0x9948, 0x9947, 0x9943,
1722 0x99a6, 0x99a7, 0x9a1a, 0x9a15, 0x9a25, 0x9a1d, 0x9a24, 0x9a1b,
1723 0x9a22, 0x9a20, 0x9a27, 0x9a23, 0x9a1e, 0x9a1c, 0x9a14, 0x9ac2,
1724 0x9b0b, 0x9b0a, 0x9b0e, 0x9b0c, 0x9b37, 0x9bea, 0x9beb, 0x9be0,
1725 0x9bde, 0x9be4, 0x9be6, 0x9be2, 0x9bf0, 0x9bd4, 0x9bd7, 0x9bec,
1726 0x9bdc, 0x9bd9, 0x9be5, 0x9bd5, 0x9be1, 0x9bda, 0x9d77, 0x9d81,
1727 0x9d8a, 0x9d84, 0x9d88, 0x9d71, 0x9d80, 0x9d78, 0x9d86, 0x9d8b,
1728 0x9d8c, 0x9d7d, 0x9d6b, 0x9d74, 0x9d75, 0x9d70, 0x9d69, 0x9d85,
1729 0x9d73, 0x9d7b, 0x9d82, 0x9d6f, 0x9d79, 0x9d7f, 0x9d87, 0x9d68,
1730 0x9e94, 0x9e91, 0x9ec0, 0x9efc, 0x9f2d, 0x9f40, 0x9f41, 0x9f4d,
1731 0x9f56, 0x9f57, 0x9f58, 0x5337, 0x56b2,
1732 /* 0xf4 */
1733 0x56b5, 0x56b3, 0x58e3, 0x5b45, 0x5dc6, 0x5dc7, 0x5eee, 0x5eef,
1734 0x5fc0, 0x5fc1, 0x61f9, 0x6517, 0x6516, 0x6515, 0x6513, 0x65df,
1735 0x66e8, 0x66e3, 0x66e4, 0x6af3, 0x6af0, 0x6aea, 0x6ae8, 0x6af9,
1736 0x6af1, 0x6ae6, 0x6aef, 0x703c, 0x7035, 0x702f, 0x7037, 0x7034,
1737 0x7031, 0x7042, 0x7038, 0x703f, 0x703a, 0x7039, 0x7040, 0x703b,
1738 0x7033, 0x7041, 0x7213, 0x7214, 0x721a, 0x737d, 0x737c, 0x74ba,
1739 0x76ab, 0x76aa, 0x76be, 0x76ed, 0x77cc, 0x77ce, 0x77cf, 0x77cd,
1740 0x77f2, 0x7925, 0x7923, 0x7927, 0x7927, 0x7924, 0x7929, 0x79b2,
1741 0x7a6e, 0x7a6c, 0x7a6d, 0x7af7, 0x7c49, 0x7c48, 0x7c4a, 0x7c47,
1742 0x7c45, 0x7cee, 0x7e7b, 0x7e7e, 0x7e81, 0x7e80, 0x7fba, 0x7fff,
1743 0x8079, 0x81db, 0x81d9, 0x820b, 0x8268, 0x8269, 0x8622, 0x85ff,
1744 0x8601, 0x85fe, 0x861b, 0x8600, 0x85f6, 0x8604, 0x8609, 0x8605,
1745 0x860c, 0x85fd, 0x8819, 0x8810, 0x8811, 0x8817, 0x8813, 0x8816,
1746 0x8963, 0x8966, 0x89b9, 0x89f7, 0x8b60, 0x8b6a, 0x8b5d, 0x8b68,
1747 0x8b63, 0x8b65, 0x8b67, 0x8b6d, 0x8dae, 0x8e86, 0x8e88, 0x8e84,
1748 0x8f59, 0x8f56, 0x8f57, 0x8f55, 0x8f58, 0x8f5a, 0x908d, 0x9143,
1749 0x9141, 0x91b7, 0x91b5, 0x91b2, 0x91b3, 0x91b0, 0x9113, 0x93fb,
1750 0x9420, 0x940f, 0x9414, 0x93fe, 0x9415, 0x9410, 0x9428, 0x9419,
1751 0x940d, 0x93f5, 0x9400, 0x93f7, 0x9407, 0x940e, 0x9416, 0x9412,
1752 0x93fa, 0x9409, 0x93f8, 0x940a, 0x93ff,
1753 /* 0xf5 */
1754 0x93fc, 0x940c, 0x93f6, 0x9411, 0x9406, 0x95de, 0x95e0, 0x95df,
1755 0x972e, 0x972f, 0x97b9, 0x97bb, 0x97fd, 0x97fe, 0x9860, 0x9862,
1756 0x9863, 0x985f, 0x98c1, 0x98c2, 0x9950, 0x994e, 0x9959, 0x994c,
```



```
1844 0x8ea6, 0x91c3, 0x9474, 0x9478, 0x9476, 0x9475, 0x9a60, 0x9c74,  
1845 0x9c73, 0x9c71, 0x9c75, 0x9e14, 0x9e13, 0x9ef6, 0x9f0a, 0x9fa4,  
1846 0x7068, 0x7065, 0x7cf7, 0x866a, 0x883e, 0x883d, 0x883f, 0x8b9e,  
1847 0x8c9c, 0x8ea9, 0x8ec9, 0x974b, 0x9873, 0x9874, 0x98cc, 0x9961,  
1848 0x99ab, 0x9a64, 0x9a66, 0x9a67, 0x9b24, 0x9e15, 0x9e17, 0x9f48,  
1849 0x6207, 0x6b1e, 0x7227, 0x864c, 0x8ea8, 0x9482, 0x9480, 0x9481,  
1850 0x9a69, 0x9a68, 0x9b2e, 0x9e19, 0x7229, 0x864b, 0x8b9f, 0x9483,  
1851 0x9c79, 0x9eb7, 0x7675, 0x9a6b, 0x9c7a, 0x9e1d, 0x7069, 0x706a,  
1852 0x9ea4, 0x9f7e, 0x9f49, 0x9f98,  
1853 };  
1854  
1855 static int  
1856 big5_mbtowc (conv_t conv, ucs4_t *pwc, const unsigned char *s, int n)  
1857 {  
1858     unsigned char c1 = s[0];  
1859     if ((c1 >= 0xa1 && c1 <= 0xc7) || (c1 >= 0xc9 && c1 <= 0xf9)) {  
1860         if (n >= 2) {  
1861             unsigned char c2 = s[1];  
1862             if ((c2 >= 0x40 && c2 < 0x7f) || (c2 >= 0xa1 && c2 < 0xff)) {  
1863                 unsigned int i = 157 * (c1 - 0xa1) + (c2 - (c2 >= 0xa1 ? 0x62 : 0x40));  
1864                 unsigned short wc = 0xffffd;  
1865                 if (i < 6280) {  
1866                     if (i < 6121)  
1867                         wc = big5_2uni_pageal[i];  
1868                     } else {  
1869                         if (i < 13932)  
1870                             wc = big5_2uni_pagec9[i-6280];  
1871                     }  
1872                     if (wc != 0xffffd) {  
1873                         *pwc = (ucs4_t) wc;  
1874                         return 2;  
1875                     }  
1876                 }  
1877                 return RET_ILSEQ;  
1878             }  
1879             return RET_TOOFEW(0);  
1880         }  
1881         return RET_ILSEQ;  
1882     }  
1883 #endif /* NEED_TOWC */  
1884  
1885 #ifdef NEED_TOMB  
1886 static const unsigned short big5_2charset[13703] = {  
1887 0xa246, 0xa247, 0xa244, 0xa1b1, 0xa258, 0xa1d3, 0xa150, 0xa1d1,  
1888 0xa1d2, 0xa3be, 0xa3bc, 0xa3bd, 0xa3bf, 0xa3bb, 0xa344, 0xa345,  
1889 0xa346, 0xa347, 0xa348, 0xa349, 0xa34a, 0xa34b, 0xa34c, 0xa34d,  
1890 0xa34e, 0xa34f, 0xa350, 0xa351, 0xa352, 0xa353, 0xa354, 0xa355,  
1891 0xa356, 0xa357, 0xa358, 0xa359, 0xa35a, 0xa35b, 0xa35c, 0xa35d,  
1892 0xa35e, 0xa35f, 0xa360, 0xa361, 0xa362, 0xa363, 0xa364, 0xa365,  
1893 0xa366, 0xa367, 0xa368, 0xa369, 0xa36a, 0xa36b, 0xa36c, 0xa36d,  
1894 0xa36e, 0xa36f, 0xa370, 0xa371, 0xa372, 0xa373, 0xc7b3, 0xc7b1,  
1895 0xc7b2, 0xc7b4, 0xc7b5, 0xc7b6, 0xc7b7, 0xc7b8, 0xc7b9, 0xc7ba,  
1896 0xc7bb, 0xc7bc, 0xc7bd, 0xc7be, 0xc7bf, 0xc7c0, 0xc7c1, 0xc7c2,  
1897 0xc7c3, 0xc7c4, 0xc7c5, 0xc7c6, 0xc7c7, 0xc7c8, 0xc7c9, 0xc7ca,  
1898 0xc7cb, 0xc7cc, 0xc7cd, 0xc7cf, 0xc7d0, 0xc7d1, 0xc7d2, 0xc7d3,  
1899 0xc7d4, 0xc7d5, 0xc7d6, 0xc7d7, 0xc7d8, 0xc7d9, 0xc7da, 0xc7db,  
1900 0xc7dc, 0xc7dd, 0xc7de, 0xc7df, 0xc7e0, 0xc7e1, 0xc7e2, 0xc7e3,  
1901 0xc7e4, 0xc7e5, 0xc7e6, 0xc7e7, 0xc7e8, 0xc7e9, 0xa156, 0xa158,  
1902 0xa1a5, 0xa1a6, 0xa1a7, 0xa1a8, 0xa1a9, 0xa14c, 0xa14b, 0xa1ac,  
1903 0xa1ab, 0xa1b0, 0xa1c2, 0xa24a, 0xa1c1, 0xa24b, 0xa2b9, 0xa2ba,  
1904 0xa2bb, 0xa2bc, 0xa2bd, 0xa2be, 0xa2bf, 0xa2c0, 0xa2c1, 0xa2c2,  
1905 0xa1f6, 0xa1f4, 0xa1f7, 0xa1f5, 0xa1f8, 0xa1f9, 0xa1fb, 0xa1fa,  
1906 0xa1d4, 0xa1db, 0xa1e8, 0xa1e7, 0xa1fd, 0xa1fc, 0xa1e4, 0xa1e5,  
1907 0xa1ec, 0xa1ed, 0xa1ef, 0xa1ee, 0xa1e3, 0xa1dc, 0xa1da, 0xa1dd,  
1908 0xa1d8, 0xa1d9, 0xa1e6, 0xa1e9, 0xc7e9, 0xc7ea, 0xc7eb, 0xc7ec,  
1909 0xc7ed, 0xc7ee, 0xc7ef, 0xc7f0, 0xc7f1, 0xc7f2, 0xc7f3, 0xc7f4,  
1910 0xc7f5, 0xc7f6, 0xc7f7, 0xc7f8, 0xc7f9, 0xc7fa, 0xc7fb, 0xc7fc,  
1911 0xa277, 0xa278, 0xa27a, 0xa27b, 0xa27c, 0xa27d, 0xa275, 0xa274,  
1912 0xa273, 0xa272, 0xa271, 0xa2a4, 0xa2a5, 0xa2a7, 0xa2a6, 0xa27e,  
1913 0xa2a1, 0xa2a3, 0xa2a2, 0xa2ac, 0xa2ad, 0xa2ae, 0xa262, 0xa263,  
1914 0xa264, 0xa265, 0xa266, 0xa267, 0xa268, 0xa269, 0xa270, 0xa26f,  
1915 0xa26e, 0xa26d, 0xa26c, 0xa26b, 0xa26a, 0xa276, 0xa279, 0xa1bd,  
1916 0xa1bc, 0xa1b6, 0xa1b5, 0xa1bf, 0xa1be, 0xa1bb, 0xa1ba, 0xa1b3,  
1917 0xa1b7, 0xa1b4, 0xa2a8, 0xa2a9, 0xa2ab, 0xa2aa, 0xa1b9, 0xa1b8,  
1918 0xa1f3, 0xa1f0, 0xa1f2, 0xa1f1, 0xa140, 0xa142, 0xa143, 0xa1b2,  
1919 0xc6a4, 0xa171, 0xa172, 0xa16d, 0xa16e, 0xa175, 0xa176, 0xa179,  
1920 0xa17a, 0xa169, 0xa16a, 0xa245, 0xa165, 0xa166, 0xa1a9, 0xa1aa,  
1921 0xa2c3, 0xa2c4, 0xa2c5, 0xa2c6, 0xa2c7, 0xa2c8, 0xa2c9, 0xa2ca,  
1922 0xa2cb, 0xc6a5, 0xc6a6, 0xc6a7, 0xc6a8, 0xc6a9, 0xc6aa, 0xc6ab,  
1923 0xc6ac, 0xc6ad, 0xc6ae, 0xc6af, 0xc6b0, 0xc6b1, 0xc6b2, 0xc6b3,  
1924 0xc6b4, 0xc6b5, 0xc6b6, 0xc6b7, 0xc6b8, 0xc6b9, 0xc6ba, 0xc6bb,  
1925 0xc6bc, 0xc6bd, 0xc6be, 0xc6bf, 0xc6c0, 0xc6c1, 0xc6c2, 0xc6c3,  
1926 0xc6c4, 0xc6c5, 0xc6c6, 0xc6c7, 0xc6c8, 0xc6c9, 0xc6ca, 0xc6cb,  
1927 0xc6cc, 0xc6cd, 0xc6ce, 0xc6cf, 0xc6d0, 0xc6d1, 0xc6d2, 0xc6d3,  
1928 0xc6d4, 0xc6d5, 0xc6d6, 0xc6d7, 0xc6d8, 0xc6d9, 0xc6da, 0xc6db,  
1929 0xc6dc, 0xc6dd, 0xc6de, 0xc6df, 0xc6e0, 0xc6e1, 0xc6e2, 0xc6e3,  
1930 0xc6e4, 0xc6e5, 0xc6e6, 0xc6e7, 0xc6e8, 0xc6e9, 0xc6ea, 0xc6eb,
```


1931 0xc6ec, 0xc6ed, 0xc6ee, 0xc6ef, 0xc6f0, 0xc6f1, 0xc6f2, 0xc6f3,
1932 0xc6f4, 0xc6f5, 0xc6f6, 0xc6f7, 0xc6a2, 0xc6a3, 0xc6f8, 0xc6f9,
1933 0xc6fa, 0xc6fb, 0xc6fc, 0xc6fd, 0xc6fe, 0xc740, 0xc741, 0xc742,
1934 0xc743, 0xc744, 0xc745, 0xc746, 0xc747, 0xc748, 0xc749, 0xc74a,
1935 0xc74b, 0xc74c, 0xc74d, 0xc74e, 0xc74f, 0xc750, 0xc751, 0xc752,
1936 0xc753, 0xc754, 0xc755, 0xc756, 0xc757, 0xc758, 0xc759, 0xc75a,
1937 0xc75b, 0xc75c, 0xc75d, 0xc75e, 0xc75f, 0xc760, 0xc761, 0xc762,
1938 0xc763, 0xc764, 0xc765, 0xc766, 0xc767, 0xc768, 0xc769, 0xc76a,
1939 0xc76b, 0xc76c, 0xc76d, 0xc76e, 0xc76f, 0xc770, 0xc771, 0xc772,
1940 0xc773, 0xc774, 0xc775, 0xc776, 0xc777, 0xc778, 0xc779, 0xc77a,
1941 0xc77b, 0xc77c, 0xc77d, 0xc77e, 0xc7a1, 0xc7a2, 0xc7a3, 0xc7a4,
1942 0xc7a5, 0xc7a6, 0xc7a7, 0xc7a8, 0xc7a9, 0xc7aa, 0xc7ab, 0xc7ac,
1943 0xc7ad, 0xc7ae, 0xc7af, 0xc7b0, 0xc6a1, 0xa374, 0xa375, 0xa376,
1944 0xa377, 0xa378, 0xa379, 0xa37a, 0xa37b, 0xa37c, 0xa37d, 0xa37e,
1945 0xa3a1, 0xa3a2, 0xa3a3, 0xa3a4, 0xa3a5, 0xa3a6, 0xa3a7, 0xa3a8,
1946 0xa3a9, 0xa3aa, 0xa3ab, 0xa3ac, 0xa3ad, 0xa3ae, 0xa3af, 0xa3b0,
1947 0xa3b1, 0xa3b2, 0xa3b3, 0xa3b4, 0xa3b5, 0xa3b6, 0xa3b7, 0xa3b8,
1948 0xa3b9, 0xa3ba, 0xa1c0, 0xa255, 0xa256, 0xa250, 0xa251, 0xa252,
1949 0xa254, 0xa257, 0xa253, 0xa1eb, 0xa1ea, 0xa24f, 0xa440, 0xa442,
1950 0xa443, 0xc945, 0xa456, 0xa454, 0xa457, 0xa455, 0xc946, 0xa4a3,
1951 0xc94f, 0xc94d, 0xa4a2, 0xa4a1, 0xa542, 0xa541, 0xa540, 0xa543,
1952 0xa4fe, 0xa5e0, 0xa5e1, 0xa8c3, 0xa458, 0xa4a4, 0xc950, 0xa4a5,
1953 0xc963, 0xa6ea, 0xcbb1, 0xa459, 0xa4a6, 0xa544, 0xc964, 0xc940,
1954 0xa444, 0xa45b, 0xc947, 0xa45c, 0xa4a7, 0xa545, 0xa547, 0xa546,
1955 0xa5e2, 0xa5e3, 0xa8c4, 0xadbc, 0xa441, 0xc941, 0xa445, 0xa45e,
1956 0xa45d, 0xa5e4, 0xa8c5, 0xb0ae, 0xd44b, 0xb6c3, 0xdcbb1, 0xdcbb2,
1957 0xa446, 0xa4a9, 0xa8c6, 0xa447, 0xc948, 0xa45f, 0xa4aa, 0xa4ac,
1958 0xc951, 0xa4ad, 0xa4ab, 0xa5e5, 0xa8c7, 0xa8c8, 0xab45, 0xa460,
1959 0xa4ae, 0xa5e6, 0xa5e8, 0xa5e7, 0xa6eb, 0xa8c9, 0xa8ca, 0xab46,
1960 0xab47, 0xadbd, 0xdcbb3, 0xf6d6, 0xa448, 0xa4b0, 0xa4af, 0xc952,
1961 0xa4b1, 0xa4b7, 0xa4b2, 0xa4b3, 0xc954, 0xc953, 0xa4b5, 0xa4b6,
1962 0xa4b4, 0xa54a, 0xa54b, 0xa54c, 0xa54d, 0xa549, 0xa550, 0xc96a,
1963 0xc966, 0xc969, 0xa551, 0xa561, 0xc968, 0xa54e, 0xa54f, 0xa548,
1964 0xc965, 0xc967, 0xa5f5, 0xc9b0, 0xa5f2, 0xa5f6, 0xc9ba, 0xc9ae,
1965 0xa5f3, 0xc9b2, 0xa5f4, 0xa5f7, 0xa5e9, 0xc9b1, 0xa5f8, 0xc9b5,
1966 0xc9b9, 0xc9b6, 0xc9b3, 0xa5ea, 0xa5ec, 0xa5f9, 0xa5ee, 0xc9ab,
1967 0xa5f1, 0xa5ef, 0xa5f0, 0xc9bb, 0xc9b8, 0xc9af, 0xa5ed, 0xc9ac,
1968 0xa5eb, 0xc9b4, 0xc9b7, 0xc9ad, 0xca66, 0xa742, 0xa6f4, 0xca67,
1969 0xa6f1, 0xa744, 0xa6f9, 0xa6f8, 0xca5b, 0xa6fc, 0xa6f7, 0xca60,
1970 0xca68, 0xca64, 0xa6fa, 0xa6fd, 0xa6ee, 0xa747, 0xca5d, 0xcbbd,
1971 0xa6ec, 0xa743, 0xa6ed, 0xa6f5, 0xa6f6, 0xca62, 0xca5e, 0xa6fb,
1972 0xa6f3, 0xca5a, 0xa6ef, 0xca65, 0xa745, 0xa748, 0xa6f2, 0xa740,
1973 0xa746, 0xa6f0, 0xca63, 0xa741, 0xca69, 0xca5c, 0xa6fe, 0xca5f,
1974 0xca61, 0xa8d8, 0xcbbf, 0xcbbc, 0xa8d0, 0xcbbc, 0xa8cb, 0xa8d5,
1975 0xa8ce, 0xcbb9, 0xa8d6, 0xcbb8, 0xcbbc, 0xcbbc, 0xcbc1, 0xa8de,
1976 0xa8d9, 0xcbb3, 0xcbb5, 0xa8db, 0xa8cf, 0xcbb6, 0xcbc2, 0xcbc9,
1977 0xa8d4, 0xcbb4, 0xcbb4, 0xa8d3, 0xcbb7, 0xa8d7, 0xcbbba, 0xa8d2,
1978 0xa8cd, 0xa8dc, 0xcbbc4, 0xa8dd, 0xcbbc8, 0xcbbc6, 0xcbcba, 0xa8da,
1979 0xcbbe, 0xcbb2, 0xcbbc0, 0xa8d1, 0xcbbc5, 0xa8cc, 0xcbc7, 0xab56,
1980 0xab4a, 0xcde0, 0xcde8, 0xab49, 0xab51, 0xab5d, 0xcdee, 0xcdec,
1981 0xcde7, 0xab4b, 0xcded, 0xcde3, 0xab59, 0xab50, 0xab58, 0xcdde,
1982 0xcdea, 0xcde1, 0xab54, 0xcde2, 0xcddd, 0xab5b, 0xab4e, 0xab57,
1983 0xab4d, 0xcddf, 0xcde4, 0xcdeb, 0xab55, 0xab52, 0xcde6, 0xab5a,
1984 0xcde9, 0xcde5, 0xab4f, 0xab5c, 0xab53, 0xab4c, 0xab48, 0xcdef,
1985 0xadd7, 0xadc1, 0xadd1, 0xadd6, 0xd0d0, 0xd0cf, 0xd0d4, 0xd0d5,
1986 0xadc4, 0xadcd, 0xadda, 0xadde, 0xd0c9, 0xadc7, 0xd0ca, 0xaddc,
1987 0xadd3, 0xadbe, 0xadbf, 0xd0dd, 0xb0bf, 0xadcc, 0xadcb, 0xd0cb,
1988 0xadcf, 0xd45b, 0xadc6, 0xd0d6, 0xadd5, 0xadd4, 0xadca, 0xd0ce,
1989 0xd0d7, 0xd0c8, 0xadc9, 0xd0d8, 0xadd2, 0xd0cc, 0xadc0, 0xadc3,
1990 0xadc2, 0xd0d9, 0xadd0, 0xadc5, 0xadd9, 0xaddb, 0xd0d3, 0xadd8,
1991 0xd0db, 0xd0cd, 0xd0dc, 0xd0d1, 0xd0da, 0xd0d2, 0xadc8, 0xd463,
1992 0xd457, 0xb0b3, 0xd45c, 0xd462, 0xb0b2, 0xd455, 0xb0b6, 0xd459,
1993 0xd452, 0xb0b4, 0xd456, 0xb0b9, 0xb0be, 0xd467, 0xd451, 0xb0ba,
1994 0xd466, 0xb0b5, 0xd458, 0xb0b1, 0xd453, 0xd44f, 0xd45d, 0xd450,
1995 0xd44e, 0xd45a, 0xd460, 0xd461, 0xb0b7, 0xd85b, 0xd45e, 0xd44d,
1996 0xd45f, 0xb0c1, 0xd464, 0xb0c0, 0xd44c, 0xd454, 0xd465, 0xb0bc,
1997 0xb0bb, 0xb0b8, 0xb0bd, 0xb0af, 0xb0b0, 0xb3c8, 0xd85e, 0xd857,
1998 0xb3c5, 0xd85f, 0xd855, 0xd858, 0xb3c4, 0xd859, 0xb3c7, 0xd85d,
1999 0xd853, 0xd852, 0xb3c9, 0xb3ca, 0xb3c6, 0xb3cb, 0xd851, 0xd85c,
2000 0xd85a, 0xd854, 0xb3c3, 0xd856, 0xb6ca, 0xb6c4, 0xdcbb7, 0xb6cd,
2001 0xdcbbd, 0xdcce, 0xb6c6, 0xb6c7, 0xdcba, 0xb6c5, 0xdcce3, 0xb6cb,
2002 0xdcce4, 0xdcce, 0xb6cc, 0xdcbb4, 0xb6c9, 0xdcce5, 0xdcce, 0xdcce,
2003 0xdcbb8, 0xb6c8, 0xdcbb6, 0xb6ce, 0xdcbb, 0xdcce2, 0xdcbb9, 0xdcce1,
2004 0xb9b6, 0xb9b3, 0xb9b4, 0xe0f1, 0xb9b2, 0xb9af, 0xe0f2,
2005 0xb9b1, 0xe0f5, 0xe0f7, 0xe0fe, 0xe0fd, 0xe0f8, 0xb9ae, 0xe0f0,
2006 0xb9ac, 0xe0f3, 0xb9b7, 0xe0f6, 0xe0fa, 0xb9b0, 0xb9ad, 0xe0fc,
2007 0xe0fb, 0xb9b5, 0xe0f4, 0xbbf8, 0xe4ec, 0xe4e9, 0xbbf9, 0xbbf7,
2008 0xe4f0, 0xe4ed, 0xe4ee, 0xbbf6, 0xbbf9, 0xe4e7, 0xbbf5, 0xbbfd,
2009 0xe4ea, 0xe4eb, 0xbbf8, 0xbbf9, 0xe4f1, 0xe4ee, 0xe4ef, 0xbeaa,
2010 0xe8f8, 0xbea7, 0xe8f5, 0xbea9, 0xbea8, 0xe8fe, 0xbea8, 0xe8f7,
2011 0xe8f4, 0xc076, 0xecbd, 0xc077, 0xecbb, 0xecbc, 0xecba, 0xecb9,
2012 0xecbe, 0xc075, 0xfbf8, 0xfbf9, 0xe4e8, 0xfbf7, 0xc078, 0xc35f,
2013 0xf1eb, 0xf1ec, 0xc4d7, 0xc4d8, 0xf5c1, 0xf5c0, 0xc56c, 0xc56b,
2014 0xf7d0, 0xa449, 0xa461, 0xa4b9, 0xa4b8, 0xa553, 0xa552, 0xa5fc,
2015 0xa5fb, 0xa5fd, 0xa5fa, 0xa74a, 0xa749, 0xa74b, 0xa8e0, 0xa8df,
2016 0xa8e1, 0xab5e, 0xa259, 0xd0de, 0xa25a, 0xb0c2, 0xa25c, 0xa25b,
2017 0xd860, 0xa25d, 0xb9b8, 0xa25e, 0xa44a, 0xa4ba, 0xa5fe, 0xa8e2,

2018 0xa44b, 0xa4bd, 0xa4bb, 0xa4bc, 0xa640, 0xa74c, 0xa8e4, 0xa8e3,
2019 0xa8e5, 0xaddd, 0xbeac, 0xc94e, 0xa554, 0xa555, 0xa641, 0xca6a,
2020 0xab60, 0xab5f, 0xd0e0, 0xd0df, 0xb0c3, 0xa4be, 0xc955, 0xcbbc,
2021 0xab61, 0xade0, 0xadde, 0xaddf, 0xbead, 0xa556, 0xa642, 0xc9bc,
2022 0xa74d, 0xa74e, 0xca6b, 0xc9ce, 0xa8e6, 0xc9cf, 0xd0e2, 0xd0e3,
2023 0xade3, 0xd0e4, 0xd0e1, 0xade4, 0xade2, 0xade1, 0xd0e5, 0xd468,
2024 0xd861, 0xdcc5, 0xe140, 0xbbf6, 0xbeae, 0xe8f9, 0xa44c, 0xa45a,
2025 0xb0c4, 0xb3cd, 0xb9b9, 0xc942, 0xa4bf, 0xa559, 0xa557, 0xa558,
2026 0xa8e7, 0xa44d, 0xa44e, 0xa462, 0xa4c0, 0xa4c1, 0xa4c2, 0xc9be,
2027 0xa55a, 0xc96b, 0xa646, 0xc9bf, 0xa644, 0xa645, 0xc9bd, 0xa647,
2028 0xa643, 0xca6c, 0xaaec, 0xca6d, 0xca6e, 0xa750, 0xa74f, 0xa753,
2029 0xa751, 0xa752, 0xa8ed, 0xa8ec, 0xc9bd4, 0xc9bd1, 0xc9bd2, 0xc9bd0,
2030 0xa8ee, 0xa8ea, 0xa8e9, 0xa8eb, 0xa8e8, 0xa8ef, 0xab63, 0xcdf0,
2031 0xc9bd3, 0xab68, 0xcdf1, 0xab64, 0xab67, 0xab66, 0xab65, 0xab62,
2032 0xd0e8, 0xade7, 0xd0eb, 0xade5, 0xd0e7, 0xade8, 0xade6, 0xade9,
2033 0xd0e9, 0xd0ea, 0xd0e6, 0xd0ec, 0xb3d1, 0xb0c5, 0xd469, 0xd46b,
2034 0xd46a, 0xd46c, 0xb0c6, 0xb3ce, 0xb3cf, 0xb3d0, 0xb6d0, 0xdcc7,
2035 0xdcc6, 0xdcc8, 0xdcc9, 0xb6d1, 0xb6cf, 0xe141, 0xe142, 0xb9bb,
2036 0xb9ba, 0xe35a, 0xbc40, 0xbc41, 0xbc42, 0xbc44, 0xe4f2, 0xe4f3,
2037 0xbc43, 0xbeaf, 0xb6b0, 0xf1ed, 0xf5c3, 0xf5c2, 0xf7d1, 0xa44f,
2038 0xa55c, 0xa55b, 0xa648, 0xc9c0, 0xa755, 0xa756, 0xa754, 0xa757,
2039 0xca6f, 0xca70, 0xa8f1, 0xc9bd5, 0xa8f0, 0xcdf2, 0xab6c, 0xcdf3,
2040 0xab6b, 0xab69, 0xab6a, 0xd0ed, 0xb0c7, 0xd46e, 0xb0ca, 0xd46d,
2041 0xb1e5, 0xb0c9, 0xb0c8, 0xb3d4, 0xb3d3, 0xb3d2, 0xb6d2, 0xb6d5,
2042 0xb6d6, 0xb6d4, 0xb6d3, 0xe143, 0xe144, 0xe4f5, 0xbc45, 0xe4f4,
2043 0xb6b1, 0xc9bf, 0xc079, 0xf1ee, 0xc455, 0xa463, 0xa4c3, 0xc956,
2044 0xa4c4, 0xa4c5, 0xa55d, 0xa55e, 0xa649, 0xca71, 0xc9bd6, 0xc9bd7,
2045 0xab6d, 0xd0ee, 0xb0cc, 0xb0cb, 0xd863, 0xd862, 0xa450, 0xa4c6,
2046 0xa55f, 0xb0cd, 0xc943, 0xc96c, 0xa560, 0xc9c2, 0xa64b, 0xa64a,
2047 0xc9c1, 0xa758, 0xadea, 0xd46f, 0xb6d7, 0xe145, 0xb9bc, 0xe8fa,
2048 0xf3fd, 0xa4c7, 0xc9bd8, 0xcdf4, 0xb0d0, 0xb0ce, 0xb0cf, 0xa451,
2049 0xa464, 0xa2cd, 0xa4ca, 0xa4c9, 0xa4c8, 0xa563, 0xa562, 0xc96d,
2050 0xc9c3, 0xa8f5, 0xa8f2, 0xa8f4, 0xa8f3, 0xab6e, 0xb3d5, 0xa452,
2051 0xa4cb, 0xa565, 0xa564, 0xca72, 0xa8f6, 0xc957, 0xa567, 0xa566,
2052 0xa64c, 0xa64d, 0xca73, 0xa759, 0xa75a, 0xa8f7, 0xa8f8, 0xa8f9,
2053 0xab6f, 0xcdf5, 0xadeb, 0xc944, 0xa4cc, 0xc9c4, 0xca74, 0xca75,
2054 0xc9bd9, 0xc9bd8, 0xcdf7, 0xcdf6, 0xcdf9, 0xcdf8, 0xab70, 0xd470,
2055 0xaded, 0xd0ef, 0xaded, 0xd864, 0xb3d6, 0xd865, 0xe146, 0xb9bd,
2056 0xb9bc, 0xf1ef, 0xc958, 0xa568, 0xb0d1, 0xa453, 0xa465, 0xa4ce,
2057 0xa4cd, 0xa4cf, 0xa8fb, 0xa8fa, 0xa8fc, 0xab71, 0xadee, 0xe8fb,
2058 0xc24f, 0xa466, 0xa56a, 0xa579, 0xa574, 0xa56f, 0xa56e, 0xa575,
2059 0xa573, 0xa56c, 0xa57a, 0xa57b, 0xa56d, 0xa569, 0xa578, 0xa577, 0xa576,
2060 0xa56b, 0xa572, 0xa571, 0xa57b, 0xa570, 0xa653, 0xa659, 0xa655,
2061 0xa65b, 0xc9c5, 0xa658, 0xa64e, 0xa651, 0xa654, 0xa650, 0xa657,
2062 0xa65a, 0xa64f, 0xa652, 0xa656, 0xa65c, 0xca7e, 0xca7b, 0xa767,
2063 0xca7c, 0xa75b, 0xa75d, 0xa775, 0xa770, 0xcaa5, 0xca7d, 0xa75f,
2064 0xa761, 0xcaa4, 0xa768, 0xca78, 0xa774, 0xa776, 0xa75c, 0xa76d,
2065 0xca76, 0xa773, 0xa764, 0xa764, 0xa76f, 0xca77, 0xa76c, 0xa76a,
2066 0xa76b, 0xa771, 0xcaa1, 0xa75e, 0xa772, 0xcaa3, 0xa766, 0xa763,
2067 0xca7a, 0xa762, 0xcaa6, 0xa765, 0xa769, 0xa760, 0xcaa2, 0xca79,
2068 0xc9beb, 0xc9bea, 0xa94f, 0xc9bed, 0xc9bef, 0xc9be4, 0xc9be7, 0xc9bee,
2069 0xa950, 0xc9be1, 0xc9be5, 0xc9be9, 0xc9ce49, 0xa94b, 0xc9ce4d, 0xa8fd,
2070 0xc9be6, 0xa8fe, 0xa94c, 0xa945, 0xa941, 0xc9be2, 0xa944, 0xa949,
2071 0xa952, 0xc9be3, 0xc9bd8, 0xa943, 0xc9bdd, 0xc9bdf, 0xa946, 0xa948,
2072 0xc9bdb, 0xc9be0, 0xa951, 0xa94d, 0xc9be8, 0xa953, 0xa94a, 0xc9bde,
2073 0xa947, 0xa942, 0xa940, 0xc9bec, 0xa94e, 0xc9ce48, 0xc9dfb, 0xc9ce4b,
2074 0xc9dfd, 0xab78, 0xab78, 0xab78, 0xab74, 0xab7d, 0xab7a, 0xab72,
2075 0xc9dfc, 0xc9ce43, 0xab73, 0xc9ce4f, 0xab75, 0xab79, 0xc9ce45, 0xc9ce42,
2076 0xab77, 0xc9dfa, 0xab76, 0xc9ce4a, 0xab7c, 0xc9ce4c, 0xab7a, 0xab73,
2077 0xab7e, 0xab7b, 0xc9ce40, 0xab71, 0xc9ce47, 0xc9ce47, 0xab7a, 0xab72,
2078 0xab76, 0xab75, 0xc9dfe, 0xc9ce44, 0xc9ce4e, 0xd144, 0xadfb, 0xd0f1,
2079 0xd0f6, 0xadfa, 0xae40, 0xd0f4, 0xadef, 0xadf9, 0xadfe, 0xd0fb,
2080 0xadfa, 0xadfd, 0xd0fe, 0xadf5, 0xd0f5, 0xd142, 0xd143, 0xadf7,
2081 0xd141, 0xadf3, 0xae43, 0xd0f8, 0xadf1, 0xd146, 0xd0f9, 0xd0fd,
2082 0xadf6, 0xae42, 0xd0fa, 0xadfc, 0xd140, 0xd147, 0xd4a1, 0xd145,
2083 0xae44, 0xadf0, 0xd0fc, 0xd0f3, 0xadf8, 0xd0f2, 0xd0f7, 0xd0f0,
2084 0xae41, 0xd477, 0xb0e4, 0xd4a7, 0xb0e2, 0xb0df, 0xd47c, 0xb0db,
2085 0xd4a2, 0xb0e6, 0xd476, 0xd47b, 0xd47a, 0xadf2, 0xb0e1, 0xd4a5,
2086 0xd4a8, 0xd473, 0xb3e8, 0xd4a9, 0xb0e7, 0xb0d9, 0xb0d6, 0xd47e,
2087 0xb0d3, 0xd4a6, 0xb0da, 0xd4aa, 0xd474, 0xd4a4, 0xb0dd, 0xd475,
2088 0xd478, 0xd47d, 0xb0de, 0xb0dc, 0xb0e8, 0xb0e3, 0xb0d7, 0xb1d2,
2089 0xb0d8, 0xd479, 0xb0e5, 0xb0e0, 0xd4a3, 0xb0d5, 0xb0d4, 0xd471,
2090 0xd472, 0xd86a, 0xb3d7, 0xb3da, 0xd875, 0xb3ee, 0xd878, 0xb3d8,
2091 0xd871, 0xb3de, 0xb3e4, 0xb5bd, 0xb3e2, 0xd86e, 0xb3ef, 0xb3db,
2092 0xb3e3, 0xd876, 0xdcd7, 0xd87b, 0xd86f, 0xd866, 0xd873, 0xd86d,
2093 0xb3e1, 0xd879, 0xb3dd, 0xb3f1, 0xb3ea, 0xb3df, 0xb3dc, 0xb3e7,
2094 0xd87a, 0xd86c, 0xd872, 0xd874, 0xd868, 0xd877, 0xb3d9, 0xd867,
2095 0xb3e0, 0xb3f0, 0xb3ec, 0xd869, 0xb3e6, 0xb3ed, 0xb3e9, 0xb3e5,
2096 0xd870, 0xb3eb, 0xdcd5, 0xdcd1, 0xdcd0, 0xdcca, 0xdcd3, 0xb6e5,
2097 0xb6e6, 0xb6e7, 0xdcdc, 0xb6e8, 0xdccf, 0xdcce, 0xdccc, 0xdcde,
2098 0xb6dc, 0xdcd8, 0xdccd, 0xb6df, 0xdcd6, 0xb6da, 0xdcd2, 0xdcd9,
2099 0xdcdb, 0xdcdf, 0xb6e3, 0xdccb, 0xb6dd, 0xdcd0, 0xb6d8, 0xb6e4,
2100 0xdcda, 0xb6e0, 0xb6e1, 0xb6e7, 0xb6db, 0xa25f, 0xb6d9, 0xdcd4,
2101 0xb6e2, 0xdcd1, 0xb9cd, 0xb9c8, 0xe155, 0xe151, 0xe14b, 0xb9c2,
2102 0xb9be, 0xe154, 0xb9bf, 0xe14e, 0xe150, 0xe153, 0xb9c4, 0xb9cb,
2103 0xb9c5, 0xe149, 0xb9c6, 0xb9c7, 0xe14c, 0xb9cc, 0xe14a, 0xe14f,
2104 0xb9c3, 0xe148, 0xb9c9, 0xb9c1, 0xb9c0, 0xe14d, 0xe152, 0xb9ca,

2105 0xe147, 0xbc4d, 0xe547, 0xe544, 0xbc47, 0xbc53, 0xbc54, 0xbc4a,
2106 0xe542, 0xbc4c, 0xe4f9, 0xbc52, 0xe546, 0xbc49, 0xe548, 0xbc48,
2107 0xe543, 0xe545, 0xbc4b, 0xe541, 0xe4fa, 0xe4f7, 0xd86b, 0xe4fd,
2108 0xe4f6, 0xe4fc, 0xe4fb, 0xe4f8, 0xbc4f, 0xbc4e, 0xbc50, 0xe4fe,
2109 0xbbeb2, 0xe540, 0xe945, 0xe8fd, 0xbebe, 0xe942, 0xbbeb6, 0xbbeba,
2110 0xe941, 0xbbeb9, 0xbbeb5, 0xbbeb8, 0xbbeb3, 0xbbebd, 0xe943, 0xe8fe,
2111 0xbbebc, 0xe8fc, 0xbbebb, 0xe944, 0xe940, 0xbc51, 0xbbebf, 0xe946,
2112 0xbbeb7, 0xbbeb4, 0xeccc6, 0xeccc8, 0xc07b, 0xeccc9, 0xeccc7, 0xeccc5,
2113 0xeccc4, 0xc07d, 0xeccc3, 0xc07e, 0xeccc1, 0xeccc2, 0xc07a, 0xc0a1,
2114 0xc07c, 0xeccc0, 0xc250, 0xefbc, 0xefba, 0xefbf, 0xefbd, 0xefbb,
2115 0xefbe, 0xc360, 0xf1f2, 0xf1f3, 0xc456, 0xf1f4, 0xf1f0, 0xf1f5,
2116 0xf1f1, 0xc251, 0xf3fe, 0xf441, 0xc459, 0xf440, 0xc458, 0xc457,
2117 0xc45a, 0xf5c5, 0xf5c6, 0xc4da, 0xc4d9, 0xc4db, 0xf5c4, 0xf6d8,
2118 0xf6d7, 0xc56d, 0xc56e, 0xc56e, 0xf6d9, 0xc5c8, 0xf8a6, 0xc5f1,
2119 0xf8a5, 0xf8ee, 0xc949, 0xc949, 0xa57d, 0xa57c, 0xa65f, 0xa65e, 0xc9c7,
2120 0xa65d, 0xc9c6, 0xa779, 0xcaa9, 0xcaa8, 0xa777, 0xa77a, 0xcaa7,
2121 0xa778, 0xcbf0, 0xcbf1, 0xa954, 0xabaa, 0xd148, 0xd149, 0xae45,
2122 0xae46, 0xd4ac, 0xb0e9, 0xb0eb, 0xd4ab, 0xb0ea, 0xd87c, 0xb3f2,
2123 0xb6e9, 0xb6ea, 0xdce1, 0xb9cf, 0xb9ce, 0xe549, 0xe948, 0xe947,
2124 0xf96b, 0xa467, 0xc959, 0xc96e, 0xc96f, 0xa662, 0xa666, 0xc9c9,
2125 0xa664, 0xa663, 0xc9c8, 0xa665, 0xa661, 0xa660, 0xc9ca, 0xa7a6,
2126 0xa7a3, 0xa77d, 0xcaaa, 0xcaab, 0xa7a1, 0xcaad, 0xa77b, 0xcaae,
2127 0xcaac, 0xa77e, 0xa7a2, 0xa7a5, 0xa7a4, 0xa77c, 0xcaaf, 0xa959,
2128 0xcbfe, 0xa95b, 0xa95a, 0xcc40, 0xa958, 0xa957, 0xcbf5, 0xcbf4,
2129 0xcbf2, 0xcbf7, 0xcbf6, 0xcbf3, 0xcbfc, 0xcbfd, 0xcbfa, 0xcbf8,
2130 0xa956, 0xcbfb, 0xa95c, 0xcc41, 0xcbf9, 0xabab, 0xa955, 0xabac,
2131 0xce54, 0xce5a, 0xab2, 0xce58, 0xce5e, 0xce55, 0xce59, 0xce5b,
2132 0xce5d, 0xce57, 0xce56, 0xce51, 0xce52, 0xabad, 0xabaf, 0xabae,
2133 0xce53, 0xce5c, 0xab1, 0xce50, 0xd153, 0xd152, 0xd157, 0xd14e,
2134 0xd151, 0xd150, 0xd154, 0xd158, 0xae47, 0xae4a, 0xd14f, 0xd155,
2135 0xae49, 0xd14a, 0xab0, 0xd4ba, 0xd156, 0xd14d, 0xae48, 0xd14c,
2136 0xd4b1, 0xb0ec, 0xb0f0, 0xd4c1, 0xd4af, 0xd4bd, 0xb0f1, 0xd4bf,
2137 0xd4c5, 0xd4c9, 0xd4c0, 0xd4b4, 0xd4bc, 0xd4ca, 0xd4c8, 0xd4be,
2138 0xd4b9, 0xd4b2, 0xd8a6, 0xd4b0, 0xb0f5, 0xd4b7, 0xb0f6, 0xb0f2,
2139 0xd4ad, 0xd4c3, 0xd4b5, 0xd4b3, 0xd4c6, 0xb0f3, 0xd4cc, 0xb0ed,
2140 0xb0ef, 0xd4bb, 0xd4b6, 0xae4b, 0xb0ee, 0xd4b8, 0xd4c7, 0xd4cb,
2141 0xd4c2, 0xd4c4, 0xd4ae, 0xd8a1, 0xd8aa, 0xd8a9, 0xb3fa, 0xd8a2,
2142 0xb3fb, 0xb3f9, 0xd8a4, 0xb3f6, 0xd8a8, 0xd8a3, 0xd8a5, 0xd87d,
2143 0xb3f4, 0xd8b2, 0xd8b1, 0xd8ae, 0xb3f3, 0xb3f7, 0xb3f8, 0xd14b,
2144 0xd8ab, 0xb3f5, 0xb0f4, 0xd8ad, 0xd87e, 0xd8b0, 0xd8af, 0xd8b3,
2145 0xdcef, 0xd8ac, 0xd8a7, 0xdce7, 0xb6f4, 0xb6f7, 0xb6f2, 0xdce6,
2146 0xdcea, 0xdce5, 0xb6ec, 0xb6ef, 0xdce2, 0xb6f0, 0xdce9, 0xb6ee,
2147 0xb6ed, 0xdcec, 0xb6ef, 0xdcee, 0xdceb, 0xb6eb, 0xb6f5, 0xdcf0,
2148 0xdce4, 0xdced, 0xdce3, 0xb6f1, 0xb6f3, 0xdce8, 0xdcf1, 0xe15d,
2149 0xb9d0, 0xe163, 0xb9d5, 0xe15f, 0xe166, 0xe157, 0xb9d7, 0xb9d1,
2150 0xe15c, 0xbc55, 0xe15b, 0xe164, 0xb9d2, 0xb9d6, 0xe15a, 0xe160,
2151 0xe165, 0xe156, 0xb9d4, 0xe15e, 0xe162, 0xe168, 0xe158, 0xe161,
2152 0xb9d3, 0xe167, 0xe159, 0xbc59, 0xe54b, 0xbc57, 0xbc56, 0xe54d,
2153 0xe552, 0xe54e, 0xe551, 0xbc5c, 0xbea5, 0xbc5b, 0xe54a, 0xe550,
2154 0xbc5a, 0xe54f, 0xe54c, 0xbc58, 0xe94d, 0xe94f, 0xe94a, 0xbec1,
2155 0xe94c, 0xbec0, 0xe94e, 0xbec3, 0xe950, 0xbec2, 0xe949, 0xe94b,
2156 0xc0a5, 0xeccc, 0xc0a4, 0xeccd, 0xc0a3, 0xeccb, 0xc0a2, 0xecca,
2157 0xc253, 0xc252, 0xf1f6, 0xf1f8, 0xf1f7, 0xc361, 0xc362, 0xc363,
2158 0xf442, 0xc45b, 0xf7d3, 0xf7d2, 0xc5f2, 0xa468, 0xa4d0, 0xa7a7,
2159 0xce5f, 0xb3fc, 0xb3fd, 0xdcf2, 0xb9d8, 0xe169, 0xe553, 0xc95a,
2160 0xcab0, 0xcc42, 0xce60, 0xd159, 0xae4c, 0xf1f9, 0xc4dc, 0xa469,
2161 0xa57e, 0xc970, 0xa667, 0xa668, 0xa95d, 0xb0f7, 0xb9da, 0xb9db,
2162 0xb9d9, 0xa46a, 0xa4d1, 0xa4d3, 0xa4d2, 0xc95b, 0xa4d4, 0xa5a1,
2163 0xc971, 0xa5a2, 0xa669, 0xa66a, 0xc9cb, 0xa7a8, 0xcab1, 0xa961,
2164 0xcc43, 0xa95f, 0xa960, 0xa95e, 0xd15a, 0xabbb, 0xabbb5, 0xabbb7,
2165 0xabbb4, 0xce61, 0xa962, 0xabbb3, 0xae4d, 0xae4e, 0xae4f, 0xd4cd,
2166 0xb3fe, 0xd8b4, 0xb0f8, 0xb6f8, 0xb9dd, 0xb9dc, 0xe16a, 0xb35d,
2167 0xbec4, 0xefc0, 0xf6da, 0xf7d4, 0xa46b, 0xa5a3, 0xa5a4, 0xc9d1,
2168 0xa66c, 0xa66f, 0xc9cf, 0xc9cd, 0xa66e, 0xc9d0, 0xc9d2, 0xc9cc,
2169 0xa671, 0xa670, 0xa66d, 0xa66b, 0xc9ce, 0xa7b3, 0xa7b0, 0xcab6,
2170 0xcab9, 0xcab8, 0xa7aa, 0xa7b2, 0xa7af, 0xcab5, 0xcab3, 0xa7ae,
2171 0xa7a9, 0xa7ac, 0xcab4, 0xcabb, 0xcab7, 0xa7ad, 0xa7b1, 0xa7b4,
2172 0xcab2, 0xcaba, 0xa7ab, 0xa967, 0xa96f, 0xcc4f, 0xcc48, 0xa970,
2173 0xcc53, 0xcc44, 0xcc4b, 0xa966, 0xcc45, 0xa964, 0xcc4c, 0xcc50,
2174 0xa963, 0xcc51, 0xcc4a, 0xcc4d, 0xa972, 0xa969, 0xcc54, 0xcc52,
2175 0xa96e, 0xa96c, 0xcc49, 0xa96b, 0xcc47, 0xcc46, 0xa96a, 0xa968,
2176 0xa971, 0xa96d, 0xa965, 0xcc4e, 0xabbb9, 0xabc0, 0xce6f, 0xabbb8,
2177 0xce67, 0xce63, 0xce73, 0xce62, 0xabbbb, 0xce6c, 0xabbe, 0xabc1,
2178 0xabbc, 0xce70, 0xabbbf, 0xae56, 0xce76, 0xce64, 0xce66, 0xce6d,
2179 0xce71, 0xce75, 0xce72, 0xce6b, 0xce6e, 0xce68, 0xabc3, 0xce6a,
2180 0xce69, 0xce74, 0xabba, 0xce65, 0xabc2, 0xabbbd, 0xae5c, 0xd162,
2181 0xae5b, 0xd160, 0xae50, 0xae55, 0xd15f, 0xd15c, 0xd161, 0xae51,
2182 0xd15b, 0xae54, 0xae52, 0xd163, 0xae53, 0xae57, 0xae58, 0xae5a,
2183 0xae59, 0xd15d, 0xd15e, 0xd164, 0xd4d4, 0xb0f9, 0xd8c2, 0xd4d3,
2184 0xd4e6, 0xb140, 0xd4e4, 0xb0fe, 0xb0fa, 0xd4ed, 0xd4dd, 0xd4e0,
2185 0xb143, 0xd4ea, 0xd4e2, 0xb0fb, 0xb144, 0xd4e7, 0xd4e5, 0xd4d6,
2186 0xd4eb, 0xd4df, 0xd4da, 0xd4d0, 0xd4ec, 0xd4dc, 0xd4cf, 0xb142,
2187 0xd4e1, 0xd4ee, 0xd4de, 0xd4d2, 0xd4d7, 0xd4ce, 0xb141, 0xd4db,
2188 0xd4d8, 0xb0fc, 0xd4d1, 0xd4e9, 0xb0fd, 0xd4d9, 0xd4d5, 0xd4e8,
2189 0xb440, 0xd8bb, 0xd8bb8, 0xd8c9, 0xd8bd, 0xd8ca, 0xb442, 0xd8c6,
2190 0xd8c3, 0xd8c4, 0xd8c7, 0xd8cb, 0xd4e3, 0xd8cd, 0xd447, 0xb443,
2191 0xd8ce, 0xd8b6, 0xd8c0, 0xd8c5, 0xb441, 0xb444, 0xd8cc, 0xd8cf,

2192 0xd8ba, 0xd8b7, 0xd8b9, 0xd8be, 0xd8bc, 0xb445, 0xd8c8, 0xd8bf,
2193 0xd8c1, 0xd8b5, 0xdcfa, 0xdcf8, 0xb742, 0xb740, 0xdd43, 0xdcf9,
2194 0xdd44, 0xdd40, 0xdcf7, 0xdd46, 0xdcf6, 0xdcfd, 0xb6fe, 0xb6fd,
2195 0xb6fc, 0xdcfb, 0xdd41, 0xb6f9, 0xb741, 0xdcf4, 0xdcfe, 0xdcf3,
2196 0xdcfc, 0xb6fa, 0xdd42, 0xdcf5, 0xb6fb, 0xdd45, 0xe16e, 0xb9e2,
2197 0xb9e1, 0xb9e3, 0xe17a, 0xe17a, 0xe170, 0xe176, 0xe16b, 0xe179, 0xe178,
2198 0xe17c, 0xe175, 0xb9de, 0xe174, 0xb9e4, 0xe16d, 0xb9df, 0xe17b,
2199 0xb9e0, 0xe16f, 0xe172, 0xe177, 0xe171, 0xe16c, 0xe173, 0xe555,
2200 0xbc61, 0xe558, 0xe557, 0xe55a, 0xe55c, 0xbc5f, 0xe556, 0xe554,
2201 0xe55d, 0xe55b, 0xe559, 0xe55f, 0xe55e, 0xbc63, 0xbc5e, 0xbc60,
2202 0xbc62, 0xe560, 0xe957, 0xe956, 0xe955, 0xe958, 0xe951, 0xe952,
2203 0xe95a, 0xe953, 0x953, 0x953, 0x953, 0xe95c, 0xe95b, 0xe954, 0xecd1, 0xc0a8,
2204 0xeccf, 0xecd4, 0xecd3, 0xe959, 0xc0a7, 0xecd2, 0xecce, 0xecd6,
2205 0xecd5, 0xc0a6, 0xecd0, 0x953, 0xc254, 0xfefc1, 0xf1fa, 0xf1fb,
2206 0xf1fc, 0xc45c, 0xc45d, 0xf443, 0xf5c8, 0xf5c7, 0xf6db, 0xf6dc,
2207 0xf7d5, 0xf8a7, 0xa46c, 0xa46d, 0xa46e, 0xa4d5, 0xa5a5, 0xc9d3,
2208 0xa672, 0xa673, 0xa7b7, 0xa7b8, 0xa7b6, 0xa7b5, 0xa973, 0xcc55,
2209 0xa975, 0xa974, 0xa974, 0xcc56, 0xabc4, 0xae5d, 0xd165, 0xd4f0, 0xb145,
2210 0xb447, 0xd4ef, 0xb446, 0xb9e5, 0xe17d, 0x953, 0xc0a9, 0xecd7,
2211 0xc45e, 0xc570, 0xc972, 0xa5a6, 0xc973, 0xa676, 0xa674, 0xa675,
2212 0xe677, 0xa7ba, 0xa7ba, 0xcabc, 0xa7bb, 0xcabd, 0xcc57, 0xcc58,
2213 0xa976, 0xa978, 0xa97a, 0xa977, 0xa97b, 0xa979, 0xabc8, 0xabc5,
2214 0xabc7, 0xabc9, 0xabc6, 0xd166, 0xce77, 0xd168, 0xd167, 0xae63,
2215 0xae5f, 0xae60, 0xae62, 0xae64, 0xae61, 0xae66, 0xae65, 0xb14a,
2216 0xd4f2, 0xd4f1, 0xb149, 0xb148, 0xb147, 0xb14b, 0xb146, 0xd8d5,
2217 0xd8d2, 0xb449, 0xd8d1, 0xd8d6, 0xb44b, 0xd8d4, 0xb448, 0xb44a,
2218 0xd8d3, 0xdd48, 0xdd49, 0xdd4a, 0xb9e6, 0xb9ee, 0xe17e, 0xb9e8,
2219 0xb9ec, 0xe1a1, 0xb9ed, 0xb9e9, 0xb9ea, 0xb9e7, 0xb9eb, 0xbc66,
2220 0xd8d0, 0xbc67, 0xbc65, 0xbc64, 0xe95d, 0x953, 0xecd8, 0xecd9,
2221 0xc364, 0xc45c, 0xa46f, 0xa678, 0xabca, 0xd169, 0xae67, 0xb14e,
2222 0xb14d, 0xb14c, 0xb44c, 0xb44d, 0xd8d7, 0xb9ef, 0x953, 0xa470,
2223 0xc95c, 0xa4d6, 0xc974, 0xc9d4, 0xa679, 0xa97c, 0xdd4b, 0xa471,
2224 0xa4d7, 0xc9d5, 0xcabe, 0xcabf, 0xa7bc, 0xd8d8, 0xb44e, 0xdd4c,
2225 0xc0aa, 0xa472, 0xa4a8, 0xa4d8, 0xc975, 0xa5a7, 0xa7c0, 0xa7bf,
2226 0xa7bd, 0xa7be, 0xcc59, 0xa97e, 0xa9a1, 0xcc5a, 0xa97d, 0xabce,
2227 0xce78, 0xabcd, 0xabcb, 0xabcc, 0xae6a, 0xae68, 0xd16b, 0xae69,
2228 0xd16a, 0xae5e, 0xd4f3, 0xb150, 0xb151, 0xb14f, 0xb9f0, 0xe1a2,
2229 0xbc68, 0xbc69, 0xe561, 0xc0ab, 0xfefc2, 0xfefc3, 0xc4dd, 0xf8a8,
2230 0xc94b, 0xa4d9, 0xa473, 0xc977, 0xc976, 0xa67a, 0xc9d7, 0xc9d8,
2231 0xc9d6, 0xc9d9, 0xcac7, 0xcac2, 0xcac4, 0xcac6, 0xcac3, 0xa7c4,
2232 0xcac0, 0xcac1, 0xa7c1, 0xa7c2, 0xcac5, 0xcac8, 0xa7c3, 0xcac9,
2233 0xcc68, 0xcc62, 0xcc5d, 0xa9a3, 0xcc65, 0xcc63, 0xcc5c, 0xcc69,
2234 0xcc6c, 0xcc67, 0xcc60, 0xa9a5, 0xcc66, 0xa9a6, 0xcc61, 0xcc64,
2235 0xcc5b, 0xcc5f, 0xcc6b, 0xa9a7, 0xa9a8, 0xcc5e, 0xcc6a, 0xa9a2,
2236 0xa9a4, 0xceab, 0xceaa, 0xcea3, 0xcea5, 0xce7d, 0xce7b,
2237 0xceac, 0xcea9, 0xce79, 0xabd0, 0xcea7, 0xcea8, 0xcea6, 0xce7c,
2238 0xce7a, 0xabcf, 0xcea2, 0xce7e, 0xcea1, 0xcead, 0xae6f, 0xae6e,
2239 0xd16c, 0xae6b, 0xd16e, 0xae70, 0xd16f, 0xae73, 0xae71, 0xd170,
2240 0xceae, 0xd172, 0xae6d, 0xae6c, 0xd16d, 0xd171, 0xae72, 0xb153,
2241 0xb152, 0xd4f5, 0xd4f9, 0xd4fb, 0xb154, 0xd4fe, 0xb158, 0xd541,
2242 0xb15a, 0xb156, 0xb15e, 0xb15b, 0xd4f7, 0xb155, 0xd4f6, 0xd4f4,
2243 0xd543, 0xd4f8, 0xb157, 0xd542, 0xb15c, 0xd4fd, 0xd4fc, 0xb15d,
2244 0xd4fa, 0xb159, 0xd544, 0xd540, 0xd8e7, 0xd8ee, 0xd8e3, 0xb451,
2245 0xd8df, 0xd8ef, 0xd8d9, 0xd8ec, 0xd8ea, 0xd8e4, 0xd8ed, 0xd8e6,
2246 0xd8de, 0xd8f0, 0xd8dc, 0xd8e9, 0xd8da, 0xd8f1, 0xb452, 0xd8eb,
2247 0xdd4f, 0xd8dd, 0xb44f, 0xd8e1, 0xb450, 0xd8e0, 0xd8e5, 0xd8e2,
2248 0xd8e8, 0xdd53, 0xdd56, 0xdd4e, 0xdd50, 0xdd55, 0xdd54, 0xb743,
2249 0xd8db, 0xdd52, 0xb744, 0xdd4d, 0xdd51, 0xe1a9, 0xe1b0, 0xe1a7,
2250 0xe1ae, 0xe1a5, 0xe1ad, 0xe1b1, 0xe1a4, 0xe1a8, 0xe1a3, 0xb9f1,
2251 0xe1a6, 0xb9f2, 0xe1ac, 0xe1ab, 0xe1aa, 0xe1af, 0xe565, 0xe567,
2252 0xbc6b, 0xe568, 0xe563, 0xe562, 0xe56c, 0xe56a, 0xbc6a, 0xe56d,
2253 0xe564, 0xe569, 0xe56b, 0xe566, 0xe961, 0xe966, 0xe960, 0xe965,
2254 0xe95e, 0xe968, 0xe964, 0xe969, 0xe963, 0xe95f, 0xe967, 0xe96a,
2255 0xe962, 0xecd4, 0xc0af, 0xc0ad, 0xc0ac, 0xc0ae, 0xfefc4, 0xf172,
2256 0xf1fd, 0xf444, 0xf445, 0xc460, 0xf5c9, 0xc4de, 0xf5ca, 0xf6de,
2257 0xc572, 0xc571, 0xf6dd, 0xc5c9, 0xf7d6, 0xa474, 0xa67b, 0xc9da,
2258 0xcaca, 0xa8b5, 0xb15f, 0xa475, 0xa5aa, 0xa5a9, 0xa5a8, 0xa7c5,
2259 0xae74, 0xdd57, 0xa476, 0xa477, 0xa478, 0xa4da, 0xabd1, 0xceaf,
2260 0xb453, 0xa479, 0xc95d, 0xa5ab, 0xa5ac, 0xc978, 0xa67c, 0xcacb,
2261 0xa7c6, 0xcacc, 0xa9ae, 0xcc6e, 0xa9ac, 0xa9ab, 0xcc6d, 0xa9a9,
2262 0xcc6f, 0xa9aa, 0xa9ad, 0xabd2, 0xabd4, 0x953, 0x953, 0x953,
2263 0x953, 0x953, 0x953, 0x953, 0x953, 0x953, 0x953, 0x953,
2264 0xd546, 0xb161, 0xb163, 0xb160, 0xb455, 0xd545, 0xb456, 0xd8f3,
2265 0xb457, 0xd8f2, 0xb454, 0xdd5a, 0xdd5c, 0xb745, 0xdd5b, 0xdd59,
2266 0xdd58, 0xe1b4, 0xb9f5, 0xb9f6, 0xb9f7, 0xe1b2, 0xe1b3, 0xb9f3,
2267 0xe571, 0xe56f, 0xbc6d, 0xe570, 0xbc6e, 0xbc6c, 0xb9f4, 0xe96d,
2268 0xe96b, 0xe96c, 0xe96e, 0xe56e, 0xecd4, 0xc0b0, 0xecd4, 0xfefc5, 0xfefc6,
2269 0xe96e, 0xf1fe, 0xa47a, 0xa5ad, 0xa67e, 0xc9db, 0xa67d, 0xa9af,
2270 0xb746, 0xa4db, 0xa5ae, 0xabd5, 0xb458, 0xc979, 0xc97a, 0xc9dc,
2271 0xa7c8, 0xcad0, 0xcace, 0xa7c9, 0xcacd, 0xcacf, 0xcad1, 0xa7c7,
2272 0xa9b3, 0xa9b4, 0xa9b1, 0xa9b0, 0x953, 0xa9b2, 0xabd6, 0x953,
2273 0x953, 0x953, 0x953, 0x953, 0x953, 0x953, 0x953, 0x953,
2274 0xd178, 0xae78, 0xd176, 0x953, 0xd547, 0xd54a, 0xd54b, 0xd548,
2275 0xb167, 0xb166, 0xb164, 0xb165, 0xd549, 0xb168, 0xb45a, 0xb45b,
2276 0xb45c, 0xdd5d, 0xdd5f, 0xdd61, 0xb748, 0xb747, 0xb459, 0xdd60,
2277 0xdd5e, 0xe1b8, 0xe1b6, 0xe1bc, 0xb9f8, 0xe1bd, 0xe1ba, 0xb9f9,
2278 0xe1b7, 0xe1b5, 0xe1bb, 0xbc70, 0xe573, 0xe1b9, 0xbc72, 0xe574,

2279 0xbc71, 0xbc74, 0xe575, 0xbc6f, 0xbc73, 0xe973, 0xe971, 0xe970,
2280 0xe972, 0xe96f, 0xc366, 0xf446, 0xf447, 0xf5cb, 0xf6df, 0xc655,
2281 0xa9b5, 0xa7ca, 0xabd8, 0xa47b, 0xa4dc, 0xa5af, 0xc9dd, 0xa7cb,
2282 0xcad2, 0xcebb, 0xabd9, 0xb9fa, 0xa47c, 0xa6a1, 0xb749, 0xa47d,
2283 0xa4dd, 0xa4de, 0xa5b1, 0xa5b0, 0xc9de, 0xa6a2, 0xcad3, 0xa7cc,
2284 0xcc71, 0xcc72, 0xcc73, 0xa9b6, 0xa9b7, 0xcc70, 0xa9b8, 0xabda,
2285 0xcebc, 0xd17a, 0xae7a, 0xd179, 0xb169, 0xd54c, 0xb16a, 0xd54d,
2286 0xb45d, 0xdd62, 0xe1bf, 0xe1be, 0xb9fb, 0xbc75, 0xe576, 0xbeca,
2287 0xe974, 0xc0b1, 0xc573, 0xf7d8, 0xcc74, 0xcabd, 0xb16b, 0xd8f4,
2288 0xb74a, 0xc255, 0xa7ce, 0xa7cd, 0xabdb, 0xd17b, 0xb16d, 0xb343,
2289 0xb16e, 0xb16c, 0xb45e, 0xe1c0, 0xb9fc, 0xbc76, 0xc94c, 0xc9df,
2290 0xcad5, 0xa7cf, 0xcad4, 0xa7d0, 0xa9bc, 0xcc77, 0xcc76, 0xa9bb,
2291 0xa9b9, 0xa9ba, 0xcc75, 0xabdd, 0xcebe, 0xabe0, 0xabdc, 0xabe2,
2292 0xabde, 0xabdf, 0xabe1, 0xae7d, 0xae7c, 0xae7b, 0xd54f, 0xb16f,
2293 0xb172, 0xb170, 0xd54e, 0xb175, 0xb171, 0xd550, 0xb174, 0xb173,
2294 0xd8f6, 0xd8f5, 0xb461, 0xb45f, 0xb460, 0xd8f7, 0xb74b, 0xdd64,
2295 0xb74c, 0xdd63, 0xe577, 0xbc78, 0xe1c1, 0xbc77, 0xb9fd, 0xecd,
2296 0xe975, 0xc0b2, 0xc573, 0xf240, 0xf448, 0xf449, 0xa4df, 0xa5b2,
2297 0xc97b, 0xa7d2, 0xa7d4, 0xc9e2, 0xcad8, 0xcad7, 0xcad6, 0xc9e1,
2298 0xc9e0, 0xa6a4, 0xa7d3, 0xa7d1, 0xa6a3, 0xa9bd, 0xcc78, 0xa9be,
2299 0xcadd, 0xcadf, 0xcade, 0xcc79, 0xcada, 0xa7d8, 0xa7d6, 0xcad9,
2300 0xcadb, 0xcae1, 0xa7d5, 0xcadc, 0xcae5, 0xa9c0, 0xcae2, 0xa7d7,
2301 0xcae0, 0xcae3, 0xa9bf, 0xa9c1, 0xcae4, 0xcacf, 0xcaca, 0xcc7e,
2302 0xc9ae, 0xc9aa, 0xabe7, 0xa9c2, 0xc9aa, 0xc9ad, 0xabe3, 0xc9ac,
2303 0xa9c3, 0xa9c8, 0xa9c6, 0xc9a3, 0xcc7c, 0xc9a5, 0xa9cd, 0xc9cb,
2304 0xabe4, 0xc9a6, 0xabe5, 0xa9c9, 0xc9a8, 0xc9cd, 0xabe6, 0xcc7b,
2305 0xa9ca, 0xabe8, 0xa9cb, 0xa9c7, 0xa9cc, 0xc9a4, 0xcc7a, 0xc9cb,
2306 0xa9c4, 0xcc7d, 0xc9a4, 0xc9a1, 0xa9c5, 0xc9ef, 0xc9c0, 0xc9ca,
2307 0xd1a1, 0xc9cb, 0xabe6, 0xc9ce, 0xc9c4, 0xabe7, 0xc9c6, 0xc9c7,
2308 0xc9c9, 0xabe9, 0xaea3, 0xc9c5, 0xc9c1, 0xaea4, 0xc9cf, 0xae7e,
2309 0xd17d, 0xc9c8, 0xd17c, 0xc9c3, 0xc9cc, 0xabe8, 0xaea1, 0xabf2,
2310 0xaea2, 0xc9cd, 0xd17e, 0xabe9, 0xaea6, 0xabf1, 0xabf0, 0xabe7,
2311 0xaea5, 0xc9d1, 0xaea7, 0xabea, 0xc9c2, 0xb176, 0xd1a4, 0xd1a6,
2312 0xd1a8, 0xaea8, 0xaeae, 0xd553, 0xd1ac, 0xd1a3, 0xb178, 0xd551,
2313 0xaead, 0xaeab, 0xd1ae, 0xd552, 0xd1a5, 0xaeac, 0xd1a9, 0xaeaf,
2314 0xd1ab, 0xaeaa, 0xd1aa, 0xd1ad, 0xd1a7, 0xaea9, 0xb179, 0xd1a2,
2315 0xb177, 0xb17a, 0xd555, 0xd55e, 0xb464, 0xb17c, 0xb1a3, 0xb465,
2316 0xd560, 0xb1aa, 0xd8f9, 0xd556, 0xb1a2, 0xb1a5, 0xb17e, 0xd554,
2317 0xd562, 0xd565, 0xd949, 0xd563, 0xd8fd, 0xb1a1, 0xb1a8, 0xb1ac,
2318 0xd55d, 0xd8f8, 0xd561, 0xb17b, 0xd8fa, 0xd564, 0xd8fc, 0xd559,
2319 0xb462, 0xd557, 0xd558, 0xb1a7, 0xb1a6, 0xd55b, 0xb1ab, 0xd55f,
2320 0xb1a4, 0xd55c, 0xb1a9, 0xb466, 0xb463, 0xd8fb, 0xd55a, 0xb17d,
2321 0xb46b, 0xb46f, 0xd940, 0xb751, 0xb46d, 0xd944, 0xb471, 0xdd65,
2322 0xd946, 0xb753, 0xb469, 0xb46c, 0xd947, 0xd948, 0xd94e, 0xb473,
2323 0xb754, 0xd94a, 0xd94f, 0xd943, 0xb75e, 0xb755, 0xb472, 0xd941,
2324 0xd950, 0xb75d, 0xb470, 0xb74e, 0xd94d, 0xb474, 0xd945, 0xd8fe,
2325 0xb46a, 0xd942, 0xd94b, 0xb74d, 0xb752, 0xb467, 0xd94c, 0xb750,
2326 0xb468, 0xb75c, 0xe1c3, 0xdd70, 0xdd68, 0xe1c2, 0xdd6c, 0xdd6e,
2327 0xdd6b, 0xb75b, 0xdd6a, 0xb75f, 0xe1d2, 0xb75a, 0xba40, 0xdd71,
2328 0xe1c4, 0xb758, 0xdd69, 0xdd6d, 0xb9fe, 0xb74f, 0xdd66, 0xdd67,
2329 0xba41, 0xb757, 0xb759, 0xb756, 0xdd6f, 0xe1c8, 0xe1c9, 0xe1ce,
2330 0xbc7d, 0xe1d5, 0xba47, 0xba46, 0xe1d0, 0xbc7c, 0xe1c5, 0xba45,
2331 0xe1d4, 0xba43, 0xba44, 0xe1d1, 0xe5aa, 0xbc7a, 0xb46e, 0xe1d3,
2332 0xbca3, 0xe1cb, 0xbc7b, 0xbca2, 0xe1c6, 0xe1ca, 0xe1c7, 0xe1cd,
2333 0xba48, 0xbc79, 0xba42, 0xe57a, 0xe1cf, 0xbca1, 0xbca4, 0xe1cc,
2334 0xbc7e, 0xe579, 0xe57e, 0xbee, 0xe578, 0xe9a3, 0xe5a9, 0xbca8,
2335 0xbca6, 0xbee, 0xe5a6, 0xe5a2, 0xbcac, 0xe978, 0xbcaa, 0xe5a1,
2336 0xe976, 0xe5a5, 0xe5a8, 0xe57d, 0xbcab, 0xbca5, 0xe977, 0xbecd,
2337 0xe5a7, 0xbca7, 0xbca9, 0xe5a4, 0xbcad, 0xe5a3, 0xe57c, 0xe57b,
2338 0xbecb, 0xe5ab, 0xe97a, 0xe97e, 0xbed0, 0xe9a2, 0xe97e, 0xe9e1,
2339 0xbed1, 0xe9a1, 0xe97c, 0xc0b4, 0xecd, 0xe979, 0xe97b, 0xc0b5,
2340 0xbed3, 0xc0b3, 0xbed2, 0xc0b7, 0xe97d, 0xbecf, 0xfcf, 0xfcf7,
2341 0xe9e7, 0xfcf8, 0xe9e3, 0xc256, 0xe9e5, 0xe9e4, 0xc0b6, 0xe9e2,
2342 0xe9e6, 0xfcf0, 0xfcf, 0xfcf, 0xfcf9, 0xfcf, 0xfcf, 0xfcfb,
2343 0xc367, 0xc36a, 0xc369, 0xc368, 0xc461, 0xf44a, 0xc462, 0xf241,
2344 0xc4df, 0xf5cc, 0xc4e0, 0xc574, 0xc5ca, 0xf7d9, 0xf7da, 0xf7db,
2345 0xf9ba, 0xa4e0, 0xc97c, 0xa5b3, 0xa6a6, 0xa6af, 0xa6a5, 0xa6a8,
2346 0xa7da, 0xa7d9, 0xc9cb1, 0xa9cf, 0xa9ce, 0xd1af, 0xb1ad, 0xb1ae,
2347 0xb475, 0xdd72, 0xb760, 0xb761, 0xdd74, 0xdd76, 0xdd75, 0xe1d7,
2348 0xe1d6, 0xba49, 0xe1d8, 0xe5ac, 0xbcae, 0xbed4, 0xc0b8, 0xc257,
2349 0xc0b9, 0xa4e1, 0xc9ae6, 0xc9cb2, 0xa9d1, 0xa9d0, 0xa9d2, 0xabf3,
2350 0xc9d2, 0xc9d3, 0xd1b0, 0xaeb0, 0xb1af, 0xb476, 0xd951, 0xa4e2,
2351 0xa47e, 0xa4e3, 0xc97d, 0xa5b7, 0xa5b6, 0xa5b4, 0xa5b5, 0xa6ab,
2352 0xc9e9, 0xc9eb, 0xa6aa, 0xc9e3, 0xc9e4, 0xc9e6, 0xc9e6, 0xc9e8,
2353 0xa6a9, 0xc9e5, 0xc9ec, 0xc9e7, 0xa7e1, 0xa7ea, 0xa7e8, 0xc9f0,
2354 0xcaed, 0xc9f5, 0xa7e6, 0xc9f6, 0xa7df, 0xc9f3, 0xa7e5, 0xc9ef,
2355 0xc9ee, 0xa7e3, 0xc9f4, 0xa7e4, 0xa9d3, 0xa7de, 0xc9f1, 0xc9e9,
2356 0xa7db, 0xa7ee, 0xc9ec, 0xc9f2, 0xa7e0, 0xa7e2, 0xc9e8, 0xc9e9,
2357 0xc9ea, 0xa7ed, 0xa7e7, 0xa7ec, 0xc9eb, 0xa7eb, 0xa7dd, 0xa7dc,
2358 0xa7e9, 0xa9e1, 0xc9cb, 0xc9cb7, 0xa9dc, 0xa9ef, 0xc9cb3, 0xc9cb,
2359 0xc9cb, 0xc9cbf, 0xa9ea, 0xc9cb, 0xc9cb4, 0xa9e8, 0xc9cb8, 0xc9cc0,
2360 0xa9d9, 0xc9cb, 0xa9e3, 0xa9e2, 0xc9cb6, 0xa9d7, 0xa9d8, 0xa9d6,
2361 0xa9ee, 0xa9e6, 0xa9e0, 0xa9d4, 0xc9cb9, 0xa9df, 0xa9d5, 0xa9e7,
2362 0xa9f0, 0xc9d4, 0xa9e4, 0xc9cb5, 0xa9da, 0xa9dd, 0xa9de, 0xa9ec,
2363 0xa9ed, 0xa9eb, 0xa9e5, 0xa9e9, 0xa9db, 0xabf4, 0xc9da, 0xc941,
2364 0xabf8, 0xabfa, 0xc940, 0xc9e6, 0xabfd, 0xd1b1, 0xaeb1, 0xc943,
2365 0xc9d7, 0xc9df, 0xabfe, 0xc9de, 0xc9db, 0xc9e3, 0xc9e5, 0xabf7,

2366 0xabfb, 0xac42, 0xaeb3, 0xcee0, 0xabf9, 0xac45, 0xcded9, 0xabfc,
2367 0xaeb2, 0xabf6, 0xcded6, 0xcdedd, 0xcded5, 0xcded8, 0xcdedc, 0xd1b2,
2368 0xac44, 0xcee1, 0xcee2, 0xcee4, 0xabf5, 0xaecl, 0xd1be, 0xaebf,
2369 0xaecl, 0xd1b4, 0xd1c4, 0xaeb6, 0xd566, 0xd1c6, 0xd1c0, 0xd1b7,
2370 0xd1c9, 0xd1ba, 0xaebc, 0xd57d, 0xd1bd, 0xaebe, 0xaeb5, 0xd1cb,
2371 0xd1bf, 0xaeb8, 0xd1b8, 0xd1b5, 0xd1b6, 0xaeb9, 0xd1c5, 0xd1cc,
2372 0xaebb, 0xd1bc, 0xd1bb, 0xaecl, 0xaecl, 0xaeb4, 0xaeba, 0xaebd,
2373 0xd1c8, 0xd1c2, 0xaeb7, 0xd1b3, 0xd1ca, 0xd1c1, 0xd1c3, 0xd1c7,
2374 0xd567, 0xb1b7, 0xb1cb, 0xb1ca, 0xb1bf, 0xd579, 0xd575, 0xd572,
2375 0xd5a6, 0xb1ba, 0xb1b2, 0xd577, 0xb4a8, 0xb1b6, 0xd5a1, 0xb1cc,
2376 0xb1c9, 0xd57b, 0xd56a, 0xb1c8, 0xd5a3, 0xd569, 0xb1bd, 0xb1c1,
2377 0xd5a2, 0xd573, 0xb1c2, 0xb1bc, 0xd568, 0xb478, 0xd5a5, 0xd571,
2378 0xb1c7, 0xd574, 0xd5a4, 0xb1c6, 0xd952, 0xb1b3, 0xd56f, 0xb1b8,
2379 0xb1c3, 0xb1be, 0xd578, 0xd56e, 0xd56c, 0xd57e, 0xb1b0, 0xb1c4,
2380 0xb1b4, 0xb477, 0xd57c, 0xd57c, 0xb1b5, 0xb1b1, 0xb1c0, 0xb1bb, 0xb1b9,
2381 0xd570, 0xb1c5, 0xd56d, 0xd57a, 0xd576, 0xd954, 0xd953, 0xd56b,
2382 0xd964, 0xb47a, 0xd96a, 0xd959, 0xd967, 0xdd77, 0xb47d, 0xd96b,
2383 0xd96e, 0xb47c, 0xd95c, 0xd96d, 0xd96c, 0xb47e, 0xd955, 0xb479,
2384 0xb4a3, 0xb4a1, 0xd969, 0xd95f, 0xb4a5, 0xd970, 0xd968, 0xd971,
2385 0xb4ad, 0xb4ab, 0xd966, 0xd965, 0xd963, 0xd95d, 0xb4a4, 0xb4a2,
2386 0xd1b9, 0xd956, 0xddb7, 0xd957, 0xb47b, 0xb4aa, 0xdd79, 0xb4a6,
2387 0xb4a7, 0xd958, 0xd96f, 0xdd78, 0xd960, 0xd95b, 0xb4a9, 0xd961,
2388 0xd95e, 0xb4ae, 0xb770, 0xdd7c, 0xddb1, 0xddb6, 0xddaa, 0xb76c,
2389 0xddbb, 0xb769, 0xdd7a, 0xdd7b, 0xb762, 0xb76b, 0xdda4, 0xb76e,
2390 0xb76f, 0xdda5, 0xddb2, 0xddb8, 0xb76a, 0xb764, 0xdda3, 0xdd7d,
2391 0xddba, 0xdda8, 0xdda9, 0xdd7e, 0xddb4, 0xddab, 0xddb5, 0xddad,
2392 0xb765, 0xe1d9, 0xb768, 0xb766, 0xddb9, 0xddb0, 0xddac, 0xdda1,
2393 0xba53, 0xddaf, 0xb76d, 0xdda7, 0xdda6, 0xb767, 0xb763, 0xe1ee,
2394 0xddb3, 0xddae, 0xdda2, 0xe1e9, 0xe1da, 0xe1e5, 0xe1ec, 0xba51,
2395 0xb4ac, 0xe1ea, 0xba4c, 0xba4b, 0xe1f1, 0xe1db, 0xe1e8, 0xe1dc,
2396 0xe1e7, 0xba4f, 0xe1eb, 0xd962, 0xe1f2, 0xe1e3, 0xba52, 0xe5ba,
2397 0xbcaf, 0xe1f0, 0xe1ef, 0xba54, 0xe5ad, 0xbcb0, 0xe5ae, 0xe1df,
2398 0xe1e0, 0xe1dd, 0xe1e2, 0xe1de, 0xe1f3, 0xba4e, 0xbcb1, 0xba50,
2399 0xba55, 0xe1e1, 0xe1ed, 0xe1e6, 0xe5b1, 0xba4a, 0xbcb4, 0xe9aa,
2400 0xe5b6, 0xe5b5, 0xe5b7, 0xe5b4, 0xbcb5, 0xbcbb, 0xbcb8, 0xbcb9,
2401 0xe5af, 0xe5b2, 0xe5bc, 0xbcc1, 0xbcbf, 0xe5b3, 0xd95a, 0xbcb2,
2402 0xe5b9, 0xe5b0, 0xbcc2, 0xe5b8, 0xba4d, 0xbcb7, 0xe1e4, 0xbcbca,
2403 0xbcbce, 0xbccc, 0xbcbda, 0xbcbcb, 0xbcb6, 0xe5bb, 0xbcb3, 0xbccc,
2404 0xbcd8, 0xbcd9, 0xe9a9, 0xbcd7, 0xbcd6, 0xbcd6, 0xbcd6, 0xe9ab,
2405 0xbcd8, 0xbcd5, 0xbcd6, 0xe9a8, 0xc0bb, 0xbcd7, 0xbcd6, 0xc0ba,
2406 0xe9a7, 0xe9a6, 0xbcd6, 0xbcd6, 0xe9a5, 0xe9a4, 0xc0bc, 0xe9ae,
2407 0xbcd6, 0xe9ac, 0xc0bd, 0xc0c2, 0xe9ca, 0xe9cc, 0xc0bf, 0xe9cd,
2408 0xe9ce, 0xe9cb, 0xc0c0, 0xc0c3, 0xe9ce, 0xc0be, 0xc0c1, 0xc259,
2409 0xe9ad, 0xc258, 0xc25e, 0xefd4, 0xc25c, 0xc25d, 0xefd7, 0xefd3,
2410 0xc25a, 0xefd1, 0xc36b, 0xefd5, 0xefd6, 0xefd2, 0xc25b, 0xf242,
2411 0xf245, 0xf246, 0xf244, 0xf247, 0xc36c, 0xf243, 0xf44e, 0xc464,
2412 0xf44d, 0xf44c, 0xf44b, 0xc463, 0xc465, 0xf5cd, 0xc4e2, 0xc4e1,
2413 0xf6e1, 0xf6e0, 0xf6e3, 0xc5cb, 0xc575, 0xf7dd, 0xf6e2, 0xf7dc,
2414 0xc5cd, 0xc5cc, 0xc5f3, 0xf8a9, 0xf8ef, 0xa4e4, 0xd972, 0xe9af,
2415 0xa6ac, 0xcaf7, 0xa7f1, 0xa7ef, 0xa7f0, 0xc0c1, 0xa9f1, 0xac46,
2416 0xcee7, 0xcee8, 0xac47, 0xd1ce, 0xaecl, 0xaecl, 0xd1cd, 0xb1d3,
2417 0xb1cf, 0xd5a7, 0xb1d6, 0xb1d5, 0xb1ce, 0xb1d1, 0xb1d4, 0xb1d0,
2418 0xd976, 0xb1cd, 0xb4af, 0xb4b1, 0xb4b2, 0xd975, 0xd978, 0xb4b0,
2419 0xd973, 0xd977, 0xd974, 0xb771, 0xddbc, 0xba56, 0xe1f4, 0xbcd3,
2420 0xbcc4, 0xe5bd, 0xbcc5, 0xbcc6, 0xe5bf, 0xe5be, 0xe5c0, 0xe9b1,
2421 0xe9b0, 0xecef, 0xecee, 0xc0c4, 0xc0c5, 0xf248, 0xa4e5, 0xd979,
2422 0xb4b4, 0xb4b3, 0xddbd, 0xefd8, 0xc4e3, 0xf7de, 0xa4e6, 0xaecl,
2423 0xb1d8, 0xb1d7, 0xd97a, 0xd97b, 0xb772, 0xe1f5, 0xba57, 0xe9b2,
2424 0xa4e7, 0xa5b8, 0xa9f2, 0xc0c2, 0xc0c2, 0xc0c2, 0xc0c2, 0xb1d9, 0xd97c,
2425 0xb4b5, 0xb773, 0xe5c1, 0xe5c2, 0xecf0, 0xc25f, 0xf8f0, 0xa4e8,
2426 0xc0c3, 0xa9f3, 0xac49, 0xceea, 0xaecl, 0xd1d2, 0xd1d0, 0xd1d1,
2427 0xaecl, 0xd1cf, 0xb1db, 0xb1dc, 0xd5a8, 0xb1dd, 0xb1da, 0xd97d,
2428 0xd97e, 0xddbe, 0xba59, 0xbcd6, 0xecf1, 0xefd9, 0xf24a, 0xf249,
2429 0xf44f, 0xc95e, 0xac4a, 0xa4e9, 0xa5b9, 0xa6ae, 0xa6ad, 0xa6af,
2430 0xa6b0, 0xc95e, 0xc95e, 0xcaf8, 0xa7f2, 0xcafb, 0xcafa, 0xcaf9,
2431 0xcafc, 0xa9f4, 0xc0c3, 0xc0c5, 0xc0c6, 0xa9fb, 0xa9f9, 0xc0ca,
2432 0xc0c6, 0xc0cd, 0xa9f8, 0xaa40, 0xc0c8, 0xc0c4, 0xa9fe, 0xc0cb,
2433 0xa9f7, 0xc0cc, 0xa9fa, 0xa9fc, 0xc0cd, 0xc0cf, 0xc0c7, 0xa9f6,
2434 0xa9f5, 0xa9fd, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf,
2435 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf,
2436 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf,
2437 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf,
2438 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf,
2439 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf,
2440 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf,
2441 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf,
2442 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf,
2443 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf,
2444 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf,
2445 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf,
2446 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf,
2447 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf,
2448 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf,
2449 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf,
2450 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf,
2451 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf,
2452 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf, 0xc0cf,

2453 0xa6b3, 0xccd2, 0xaa42, 0xaa41, 0xcef9, 0xcefa, 0xd1d7, 0xd1d8,
2454 0xaed2, 0xaed3, 0xaed4, 0xd5af, 0xb1e6, 0xb4c2, 0xb4c1, 0xddc8,
2455 0xdf7a, 0xe1fb, 0xe9bd, 0xc261, 0xc467, 0xa4ec, 0xa5bc, 0xa5bd,
2456 0xa5bb, 0xa5be, 0xa5ba, 0xa6b6, 0xc9f6, 0xa6b5, 0xa6b7, 0xc9f1,
2457 0xc9f0, 0xc9f3, 0xc9f2, 0xc9f5, 0xa6b4, 0xc9ef, 0xc9f4, 0xcafd,
2458 0xa7fd, 0xcafe, 0xc43, 0xa7fc, 0xcb47, 0xcb42, 0xcb45, 0xa7f5,
2459 0xa7f6, 0xa7f7, 0xa7f8, 0xa840, 0xcb41, 0xa7fa, 0xa841, 0xcb40,
2460 0xcb46, 0xa7f9, 0xcb44, 0xa7fb, 0xa7f4, 0xa7fe, 0xaa57, 0xccd4,
2461 0xaa43, 0xaa44, 0xaa4e, 0xaa46, 0xaa58, 0xaa48, 0xccdc, 0xaa53,
2462 0xccd7, 0xaa49, 0xcce6, 0xcce7, 0xccdf, 0ccd8, 0xaa56, 0xcce4,
2463 0xaa51, 0xaa4f, 0xcce5, 0xcce3, 0ccdb, 0ccd3, 0ccda, 0xaa4a,
2464 0xaa50, 0xaa44, 0xcce, 0ccdd, 0ccd5, 0xaa52, 0xcce1, 0ccd6,
2465 0xaa55, 0xcce8, 0xaa45, 0xaa4c, 0ccd9, 0xcce2, 0xaa54, 0xaa47,
2466 0xaa4b, 0xcce0, 0xcf5b, 0xac5c, 0xac69, 0xcf56, 0xcf4c, 0xac62,
2467 0xcf4a, 0xac5b, 0xcf45, 0xac65, 0xcf52, 0xcfe, 0xcf41, 0xcf44,
2468 0xcfeb, 0xcf51, 0xcf61, 0xac60, 0xcf46, 0xcf58, 0xcfed, 0xcf5f,
2469 0xcf60, 0xcf63, 0xcf5a, 0xcf4b, 0xcf53, 0xac66, 0xac59, 0xac61,
2470 0xac6d, 0xac56, 0xac58, 0xcf43, 0xac6a, 0xac63, 0xcf5d, 0xcf40,
2471 0xac6c, 0xac67, 0xcf49, 0xac6b, 0xcf50, 0xcf48, 0xac64, 0xcf5c,
2472 0xcf54, 0xac5e, 0xcf62, 0xcf47, 0xac5a, 0xcf59, 0xcf4f, 0xac5f,
2473 0xcf55, 0xac57, 0xcfc, 0xac68, 0xae3, 0xac5d, 0xcf4e, 0xcf4d,
2474 0xcf42, 0xcf5e, 0xcf57, 0xac55, 0xd1ec, 0xaea, 0xd1ed, 0xd1e1,
2475 0xaedf, 0xaeb, 0xd1da, 0xd1e3, 0xd1e, 0xd1d9, 0xd1f4, 0xaed5,
2476 0xd1f3, 0xd1ee, 0xd1ef, 0xaedd, 0xae8, 0xd1e5, 0xd1e6, 0xd1f0,
2477 0xd1e7, 0xd1e2, 0xd1dc, 0xd1dd, 0xd1ea, 0xd1e4, 0xaed6, 0xaeda,
2478 0xd1f2, 0xd1de, 0xae6, 0xae2, 0xae5, 0xaeec, 0xaedb, 0xae7,
2479 0xd1e9, 0xae9, 0xaed8, 0xaed7, 0xd1db, 0xd1df, 0xae0, 0xd1f1,
2480 0xd1e8, 0xd1e0, 0xae4, 0xae1, 0xaed9, 0xaedc, 0xd5c4, 0xd5b4,
2481 0xd5b5, 0xd5b9, 0xd5c8, 0xd5c5, 0xd5be, 0xd5bd, 0xb1ed, 0xd5c1,
2482 0xd5d0, 0xd5b0, 0xd5d1, 0xd5c3, 0xd5d5, 0xd5c9, 0xb1ec, 0xd5c7,
2483 0xb1e7, 0xb1fc, 0xb1f2, 0xb1f6, 0xb1f5, 0xd5b1, 0xd5ce, 0xd5d4,
2484 0xd5cc, 0xd5d3, 0xd5c0, 0xd5b2, 0xd5d2, 0xd5c2, 0xb1ea, 0xb1f7,
2485 0xd5cb, 0xb1f0, 0xd5ca, 0xd5b3, 0xb1f8, 0xb1fa, 0xd5cd, 0xb1fb,
2486 0xb1e9, 0xd5ba, 0xd5cf, 0xb1ef, 0xb1f9, 0xd5bc, 0xd5c6, 0xd5b7,
2487 0xd5bb, 0xb1f4, 0xd5b6, 0xb1e8, 0xb1f1, 0xb1ee, 0xd5bf, 0xaede,
2488 0xd9c0, 0xb1eb, 0xb1f3, 0xd9c3, 0xd9d9, 0xd9ce, 0xb4d6, 0xb4d1,
2489 0xd9bd, 0xb4d2, 0xd9cd, 0xd9c6, 0xd9d3, 0xb4ce, 0xd9ab, 0xd9d5,
2490 0xb4c4, 0xd9b3, 0xb4c7, 0xb4c6, 0xb4d7, 0xd9ad, 0xd9cf, 0xd9d0,
2491 0xb4c9, 0xb4c5, 0xd9bb, 0xb4d0, 0xd9b6, 0xd9d1, 0xb4cc, 0xd9c9,
2492 0xd9d6, 0xd9b0, 0xd9b5, 0xd9af, 0xb4cb, 0xd9c2, 0xddde, 0xd9b1,
2493 0xb4cf, 0xd9ba, 0xd9d2, 0xb4ca, 0xd9b7, 0xd9b4, 0xd9c5, 0xb4cd,
2494 0xb4c3, 0xb4d9, 0xd9c8, 0xd9c7, 0xd9ac, 0xb4c8, 0xd9d4, 0xd9bc,
2495 0xd9be, 0xd9cb, 0xd9ca, 0xd9aa, 0xb4d3, 0xb4d5, 0xd9b2, 0xd9b9,
2496 0xd9c1, 0xb4d4, 0xd9b8, 0xd9c4, 0xd9d7, 0xd9cc, 0xd9d8, 0xd9ae,
2497 0xddf2, 0xb7a6, 0xddf0, 0xdddb, 0xddde, 0xddd9, 0xddec, 0xddcb,
2498 0xddd2, 0xddea, 0xddf4, 0xdddc, 0xddcf, 0xddde2, 0xddde7, 0xddd3,
2499 0xddde4, 0xddd0, 0xddd7, 0xddd8, 0xb7a8, 0xddde, 0xddde9, 0xddcc,
2500 0xddde, 0xdddf, 0xddf1, 0xb7ac, 0xb7a4, 0xd5b8, 0xddd4, 0xddde6,
2501 0xddd5, 0xb7a1, 0xb7b1, 0xddde, 0xb7af, 0xb7ab, 0xddca, 0xb7a3,
2502 0xddcd, 0xb7b0, 0xdddd, 0xddc9, 0xb7a9, 0xddde1, 0xddd1, 0xb7aa,
2503 0xddda, 0xb77e, 0xb4d8, 0xddde3, 0xd9bf, 0xddce, 0xddde8, 0xb7a5,
2504 0xddde5, 0xb7a2, 0xdddf, 0xb7ad, 0xddd6, 0xddf3, 0xb7a7, 0xddce6,
2505 0xb7ae, 0xe24a, 0xe248, 0xe25e, 0xe246, 0xe258, 0xb77d, 0xba5f,
2506 0xe242, 0xe25d, 0xe247, 0xe255, 0xba64, 0xba5d, 0xe25b, 0xe240,
2507 0xe25a, 0xba6f, 0xe251, 0xe261, 0xba6d, 0xe249, 0xba5e, 0xe24b,
2508 0xe259, 0xba67, 0xe244, 0xba6b, 0xba61, 0xe24d, 0xe243, 0xe1fc,
2509 0xe257, 0xba68, 0xe260, 0xe1fd, 0xba65, 0xe253, 0xba66, 0xe245,
2510 0xe250, 0xe24c, 0xe24e, 0xba60, 0xe25f, 0xba6e, 0xe24f, 0xe262,
2511 0xe1fe, 0xe254, 0xba63, 0xba6c, 0xba6a, 0xe241, 0xe256, 0xba69,
2512 0xba62, 0xe252, 0xe25c, 0xe5d5, 0xe5d1, 0xe5cd, 0xe5e1, 0xe5de,
2513 0xbccd, 0xe5e5, 0xe5d4, 0xbcd8, 0xe5db, 0xe5d0, 0xe5da, 0xbcd5,
2514 0xe5ee, 0xe5eb, 0xe5dd, 0xe5ce, 0xe5e2, 0xe5e4, 0xbcd1, 0xe5d8,
2515 0xe5d3, 0xe5ca, 0xbcce, 0xbcd6, 0xe5e7, 0xbcd7, 0xe5cb, 0xe5ed,
2516 0xe5e0, 0xe5e6, 0xbcd4, 0xe5e3, 0xe5ea, 0xbcd9, 0xbcd3, 0xe5dc,
2517 0xe5cf, 0xe5ef, 0xe5cc, 0xe5e8, 0xbcd0, 0xe5d6, 0xe5d7, 0xbccf,
2518 0xbccc, 0xe5d2, 0xbcd2, 0xbccb, 0xe5e9, 0xe5ec, 0xe5d9, 0xe9ca,
2519 0xe9c2, 0xe9be, 0xbef6, 0xbeeb, 0xbef0, 0xbeec, 0xe9cc, 0xe9d7,
2520 0xbee, 0xe9c4, 0xe9cd, 0xe5df, 0xe9ce, 0xbef1, 0xe9dd, 0xbef5,
2521 0xbef8, 0xe9c0, 0xbef4, 0xe9db, 0xe9dc, 0xe9d2, 0xe9d1, 0xe9c9,
2522 0xe9d3, 0xe9da, 0xe9d9, 0xbeef, 0xbeed, 0xe9cb, 0xe9c8, 0xe9c5,
2523 0xe9d8, 0xbef7, 0xe9d6, 0xbef3, 0xbef2, 0xe9d0, 0xe9b9, 0xe9c1,
2524 0xe9c3, 0xe9d5, 0xe9cf, 0beeee, 0xe9c6, 0xe9d4, 0xe9c7, 0xc0cf,
2525 0xed45, 0xc0c8, 0xecf5, 0xed41, 0xc0ca, 0xed4a, 0xecfc, 0xecf7,
2526 0xed49, 0xecf3, 0xecfe, 0xc0d1, 0xed44, 0xed48, 0xecfd, 0xc0c9,
2527 0xed40, 0xecf4, 0xc0d0, 0xed47, 0xecf9, 0xc0cc, 0xecfb, 0xecf8,
2528 0xc0d2, 0xecfa, 0xc0cb, 0xc0ce, 0xed43, 0xecf6, 0xed46, 0xed42,
2529 0xc263, 0xefe7, 0xc268, 0xc269, 0xc262, 0xefe6, 0xefe3, 0xefe4,
2530 0xc266, 0xefe, 0xefe2, 0xc265, 0xefdf, 0xc267, 0xc264, 0xefdd,
2531 0xefe1, 0xefe5, 0xf251, 0xf24e, 0xf257, 0xf256, 0xf254, 0xf24f,
2532 0xc372, 0xf250, 0xc371, 0xc0cd, 0xf253, 0xc370, 0xf258, 0xf252,
2533 0xf24d, 0xefe0, 0xc36f, 0xf24c, 0xf456, 0xf455, 0xf255, 0xc468,
2534 0xf459, 0xf45a, 0xf454, 0xf458, 0xf453, 0xf5d1, 0xf457, 0xc4e7,
2535 0xc4e5, 0xf5cf, 0xf5d2, 0xf5ce, 0xf5d0, 0xc4e6, 0xf6e5, 0xf6e6,
2536 0xc576, 0xf6e4, 0xf7e2, 0xc5cf, 0xf7e0, 0xf7e1, 0xf8ac, 0xc656,
2537 0xf8f3, 0xf8f1, 0xf8f2, 0xf8f4, 0xf9bb, 0xa4ed, 0xa6b8, 0xaa59,
2538 0xcce9, 0xcf64, 0xd1f5, 0xd1f7, 0xd1f6, 0xd1f8, 0xb1fd, 0xd5d7,
2539 0xd1f9, 0xd5d6, 0xd5d8, 0xd5d9, 0xd9da, 0xb4db, 0xd9db, 0xd9dd,

2540 0xb4dc, 0xb4da, 0xd9dc, 0xddfa, 0xddf8, 0xddf7, 0xddf6, 0xddf5,
2541 0xb7b2, 0xddf9, 0xba70, 0xe263, 0xe265, 0xba71, 0xe264, 0xbcbd,
2542 0xbccda, 0xe5f0, 0xe9df, 0xe9de, 0xe9e0, 0xbef9, 0xed4b, 0xc0d3,
2543 0xefef8, 0xc26a, 0xf259, 0xc577, 0xa4ee, 0xa5bf, 0xa6b9, 0xa842,
2544 0xaa5a, 0xaa5b, 0xac6e, 0xd1fa, 0xb7b3, 0xe6d1, 0xbefa, 0xc26b,
2545 0xa4ef, 0xa6ba, 0xcceb, 0xaa5c, 0xccea, 0xcf65, 0xac6f, 0xcf66,
2546 0xac70, 0xd1fc, 0xaeee, 0xaeed, 0xd5de, 0xd5dc, 0xd5dd, 0xd5db,
2547 0xd5da, 0xd9de, 0xd9e1, 0xb4de, 0xd9df, 0xb4dd, 0xd9e0, 0xddfb,
2548 0xe266, 0xe267, 0xe268, 0xe5f3, 0xe5f2, 0xbcdc, 0xe5f1, 0xe5f4,
2549 0xe9e1, 0xe9e2, 0xe9e3, 0xed4c, 0xc0d4, 0xc26c, 0xf25a, 0xc4e8,
2550 0xc95f, 0xac71, 0xcf67, 0xaeeef, 0xb1fe, 0xb4df, 0xd9e2, 0xb7b5,
2551 0xb7b4, 0xe269, 0xe26a, 0xbccd, 0xbcdc, 0xe9e5, 0xe9e4, 0xefef9,
2552 0xf7e3, 0xa4f0, 0xc960, 0xa5c0, 0xa843, 0xcb48, 0xac72, 0xb7b6,
2553 0xa4f1, 0xcf68, 0xac73, 0xcf69, 0xc0d5, 0xa4f2, 0xccec, 0xcf6a,
2554 0xd242, 0xd241, 0xd1fe, 0xd1fd, 0xd243, 0xd240, 0xb240, 0xb241,
2555 0xb4e0, 0xd9e3, 0xd9e4, 0xd9e5, 0xde41, 0xde42, 0xde40, 0xddfd,
2556 0xddfe, 0xb7b7, 0xe26b, 0xe5f7, 0xe5f6, 0xe5f5, 0xe5f8, 0xe9e7,
2557 0xe9e6, 0xbefb, 0xe9e8, 0xc0d6, 0xed4d, 0xfea, 0xf25b, 0xf6e7,
2558 0xa4f3, 0xa5c2, 0xa5c1, 0xaa5d, 0xc961, 0xc97e, 0xa6bb, 0xc9f7,
2559 0xcb49, 0xcb4a, 0xaa5e, 0xccec, 0xac74, 0xcf6b, 0xcf6c, 0xaef0,
2560 0xaef4, 0xd244, 0xaef3, 0xaef1, 0xaef2, 0xd5df, 0xb242, 0xb4e3,
2561 0xb4e1, 0xb4e2, 0xd9e6, 0xba72, 0xa4f4, 0xc9a1, 0xa5c3, 0xc9a4,
2562 0xa5c6, 0xc9a3, 0xa5c5, 0xa5c4, 0xa844, 0xc9a2, 0xc9f8, 0xc9fc,
2563 0xc9fe, 0xca40, 0xa6c5, 0xa6c6, 0xc9fb, 0xa6c1, 0xc9f9, 0xc9fd,
2564 0xa6c2, 0xa6bd, 0xa6be, 0xa6c4, 0xc9fa, 0xa6bc, 0xa845, 0xa6bf,
2565 0xa6c0, 0xa6c3, 0xcb5b, 0xcb59, 0xcb4c, 0xa851, 0xcb53, 0xa84c,
2566 0xcb4d, 0xcb55, 0xcb52, 0xa84f, 0xcb51, 0xa856, 0xcb5a, 0xa858,
2567 0xa85a, 0xcb4b, 0xa84d, 0xcb5c, 0xa854, 0xa857, 0xcd45, 0xa847,
2568 0xa85e, 0xa855, 0xcb4e, 0xa84a, 0xa859, 0xcb56, 0xa848, 0xa849,
2569 0xcd43, 0xcb4f, 0xa850, 0xa85b, 0xcb5d, 0xcb50, 0xa84e, 0xa853,
2570 0xccee, 0xa85c, 0xcb57, 0xa852, 0xa85d, 0xa846, 0xcb54, 0xa84b,
2571 0xcb58, 0xcd44, 0xaa6a, 0xaa7a, 0xccf5, 0xaa71, 0xcd4b, 0xaa62,
2572 0xaa65, 0xcd42, 0xccf3, 0xccf7, 0xaa6d, 0xaa6f, 0ccfa, 0xaa76,
2573 0xaa68, 0xaa66, 0xaa67, 0xaa75, 0xcd47, 0xaa70, 0ccf9, 0ccfb,
2574 0xaa6e, 0xaa73, 0xccf6, 0xcd4a, 0xac75, 0xaa79, 0xaa63, 0xcd49,
2575 0xcd4d, 0xccf8, 0xcd4f, 0xcd40, 0xaa6c, 0xccf4, 0xaa6b, 0xaa7d,
2576 0xaa72, 0xccf2, 0xcf75, 0xaa78, 0xaa7c, 0xcd41, 0xcd46, 0xaa7e,
2577 0xaa77, 0xaa69, 0xaa5f, 0xaa64, 0xccf6, 0xaa60, 0xcd4e, 0xccf0,
2578 0xccef, 0xccfd, 0xccf1, 0xaa7b, 0xaef5, 0xaa74, 0ccfe, 0xaa61,
2579 0xaca6, 0xcd4c, 0xcf7c, 0xcfa1, 0xcfa4, 0xcf77, 0xcfa7, 0xcfaa,
2580 0xcfac, 0xcf74, 0xac76, 0xac7b, 0xd249, 0xacad, 0xcfa5, 0xcfad,
2581 0xcf7b, 0xcf73, 0xd264, 0xac7e, 0xcfa2, 0xcf78, 0xcf7a, 0xaca5,
2582 0xcf7d, 0xac7d, 0xcf70, 0xcfa8, 0xcfab, 0xac7a, 0xaca8, 0xcf6d,
2583 0xacaa, 0xac78, 0xaca, 0xcfa9, 0xcf6f, 0xacab, 0xd25e, 0xcd48,
2584 0xac7c, 0xac77, 0xcf76, 0xcf6e, 0xacac, 0xaca4, 0xcfa3, 0xaca9,
2585 0xaca7, 0xcf79, 0xaca1, 0xcf71, 0xaca2, 0xaca3, 0xcf72, 0xcfa6,
2586 0xac79, 0xcf7e, 0xd24c, 0xaefd, 0xaf43, 0xd255, 0xd25b, 0xd257,
2587 0xd24a, 0xd24d, 0xd246, 0xd247, 0xaf4a, 0xaefa, 0xd256, 0xd25f,
2588 0xaf45, 0xaef6, 0xaf40, 0xd24e, 0xaf42, 0xd24f, 0xd259, 0xaf44,
2589 0xd268, 0xd248, 0xaefc, 0xaefb, 0xaf48, 0xd245, 0xd266, 0xd25a,
2590 0xd267, 0xd261, 0xd253, 0xd262, 0xd25c, 0xd265, 0xd263, 0xaf49,
2591 0xd254, 0xaef9, 0xaef8, 0xaf41, 0xaf47, 0xd260, 0xaf46, 0xd251,
2592 0xb243, 0xd269, 0xd250, 0xd24b, 0xaefe, 0xaf4b, 0xaef7, 0xd258,
2593 0xd25d, 0xb265, 0xd5e1, 0xd5e5, 0xb252, 0xb250, 0xb247, 0xd5e3,
2594 0xd5e2, 0xb25b, 0xd5e8, 0xb255, 0xd5fa, 0xd647, 0xb244, 0xd5f7,
2595 0xd5f0, 0xb267, 0xd5e0, 0xd5fc, 0xb264, 0xb258, 0xb263, 0xb24e,
2596 0xd5ec, 0xd5fe, 0xd5f6, 0xb24f, 0xb249, 0xd645, 0xd5fd, 0xd640,
2597 0xb251, 0xb259, 0xd642, 0xd5ea, 0xd5fb, 0xd5ef, 0xd644, 0xb25e,
2598 0xb246, 0xb25c, 0xd5f4, 0xd5f2, 0xd5f3, 0xb253, 0xd5ee, 0xd5ed,
2599 0xb248, 0xd5e7, 0xd646, 0xb24a, 0xd5f1, 0xb268, 0xb262, 0xd5e6,
2600 0xb25f, 0xb25d, 0xb266, 0xd5f8, 0xb261, 0xd252, 0xd5f9, 0xb260,
2601 0xd641, 0xb245, 0xd5f5, 0xb257, 0xd5e9, 0xb256, 0xb254, 0xb24c,
2602 0xb24b, 0xd9e7, 0xd643, 0xd5eb, 0xd9fc, 0xb24d, 0xb541, 0xb25a,
2603 0xb4ee, 0xd9f6, 0xb4fc, 0xd9ea, 0xb4eb, 0xb4e7, 0xda49, 0xb4ed,
2604 0xb4f1, 0xb4ec, 0xb4f5, 0xda4d, 0xda4a, 0xd9f1, 0xb4fa, 0xb4f4,
2605 0xd9fd, 0xb4e4, 0xda4a, 0xda43, 0xb4e8, 0xd9f7, 0xb4f7, 0xda55,
2606 0xda56, 0xb4e5, 0xda48, 0xb4f9, 0xd9fb, 0xd9ed, 0xd9ee, 0xb4fd,
2607 0xd9f2, 0xd9f9, 0xd9f3, 0xb4fb, 0xb544, 0xd9ef, 0xd9e8, 0xd9e9,
2608 0xd9eb, 0xb4ea, 0xd9f8, 0xb4f8, 0xb542, 0xd9fa, 0xda53, 0xda4b,
2609 0xb4e6, 0xda51, 0xb4f2, 0xb4f0, 0xda57, 0xb4ef, 0xda41, 0xd9f4,
2610 0xd9fe, 0xb547, 0xda45, 0xda42, 0xd9f0, 0xb543, 0xda4f, 0xda4c,
2611 0xda54, 0xb4e9, 0xda40, 0xb546, 0xda47, 0xb4f3, 0xb4f6, 0xda46,
2612 0xb545, 0xd9f5, 0xd5e4, 0xda50, 0xda4e, 0xda52, 0xd9ec, 0xb540,
2613 0xde61, 0xde60, 0xde46, 0xb7bd, 0xde5f, 0xde49, 0xde4a, 0xb7c7,
2614 0xde68, 0xb7c2, 0xde5e, 0xde43, 0xb7c8, 0xb7be, 0xde52, 0xde48,
2615 0xde4b, 0xde63, 0xb7b8, 0xde6a, 0xde62, 0xb7c1, 0xde57, 0xb7cc,
2616 0xb7cb, 0xb7c5, 0xde69, 0xb7b9, 0xde55, 0xde4c, 0xde59, 0xde65,
2617 0xb7cd, 0xb7bb, 0xde54, 0xde4d, 0xb7c4, 0xb7c3, 0xde50, 0xde5a,
2618 0xde64, 0xde47, 0xde51, 0xb7bc, 0xde5b, 0xb7c9, 0xb7c0, 0xde4e,
2619 0xb7bf, 0xde45, 0xde53, 0xde67, 0xb4fe, 0xbab0, 0xde56, 0xe26c,
2620 0xde58, 0xde66, 0xb7c6, 0xde4f, 0xb7ba, 0xb7ca, 0xbcf0, 0xde44,
2621 0xde5d, 0xde5c, 0xe2aa, 0xbaad, 0xe27d, 0xe2a4, 0xbaa2, 0xe26e,
2622 0xbaaf, 0xba77, 0xe26d, 0xe2b0, 0xbab1, 0xe271, 0xe2a3, 0xe273,
2623 0xe2b3, 0xe2af, 0xba75, 0xbaa1, 0xe653, 0xbaae, 0xba7d, 0xe26f,
2624 0xe2ae, 0xbaa3, 0xe2ab, 0xe2b8, 0xe275, 0xe27e, 0xe2b6, 0xe2ac,
2625 0xba7c, 0xe27c, 0xba76, 0xba74, 0xbaa8, 0xe27a, 0xe277, 0xe278,
2626 0xe2b2, 0xe2b7, 0xe2b5, 0xba7a, 0xe2b9, 0xba7e, 0xbaa7, 0xe270,

2627 0xe5fa, 0xe279, 0xba78, 0xbaac, 0xbaa9, 0xba7b, 0xe2a5, 0xe274,
2628 0xbaaa, 0xe2a7, 0xbaa4, 0xbaa6, 0xba73, 0xe2a9, 0xe2a1, 0xe272,
2629 0xbaa5, 0xe2b1, 0xe2b4, 0xe27b, 0xe2a8, 0xba79, 0xbcdf, 0xe2a6,
2630 0xe5f9, 0xe2ad, 0xe276, 0xe644, 0xe64e, 0xbce2, 0xe64d, 0xe659,
2631 0xbce4, 0xe64b, 0xe64f, 0xbcef, 0xe646, 0xbce7, 0xe652, 0xe9f0,
2632 0xbcf3, 0xbcf2, 0xbcf4, 0xe654, 0xe643, 0xe65e, 0xbced, 0xbce3, 0xe657,
2633 0xe65b, 0xe660, 0xe655, 0xe649, 0xbce6, 0xbce9, 0xbcf1, 0xbcec,
2634 0xe64c, 0xe2a2, 0xe648, 0xe65f, 0xbce8, 0xbceb, 0xe661, 0xbce0,
2635 0xe656, 0xe5fb, 0xe65c, 0xc0df, 0xe64a, 0xbce1, 0xe645, 0xbce5,
2636 0xe5fc, 0xbaab, 0xe641, 0xe65a, 0xe642, 0xe640, 0xbcea, 0xe658,
2637 0xe5fe, 0xe651, 0xe650, 0xe65d, 0xe647, 0xbcee, 0xe9f3, 0xbf49,
2638 0xbefe, 0xea40, 0xe9eb, 0xbf41, 0xe9f7, 0xbf48, 0xbf43, 0xe9f5,
2639 0xed4f, 0xe9fb, 0xea42, 0xe9fa, 0xe9e9, 0xe9f8, 0xea44, 0xea46,
2640 0xbefd, 0xea45, 0xbf44, 0xbf4a, 0xbf47, 0xe9fe, 0xbf46, 0xe9f9,
2641 0xe9ed, 0xe9f2, 0xe9fd, 0xbf45, 0xbf42, 0xbefc, 0xbf40, 0xe9f1,
2642 0xe5fd, 0xe9ec, 0xe9ef, 0xea41, 0xe9f4, 0xe9ea, 0xed4e, 0xea43,
2643 0xe9ee, 0xe9fc, 0xed51, 0xc0e3, 0xc0d7, 0xc0db, 0xed53, 0xed59,
2644 0xed57, 0xc0d9, 0xc0da, 0xc0e1, 0xed5a, 0xed52, 0xc0dc, 0xed56,
2645 0xed55, 0xed5b, 0xc0e2, 0xc0dd, 0xc0e0, 0xed54, 0xc0e4, 0xc0de,
2646 0xc0e5, 0xc0d8, 0xed58, 0xed50, 0xeff7, 0xc271, 0xeff4, 0xeff6,
2647 0xc26f, 0xeff2, 0xeff3, 0xfee, 0xe9f6, 0xefe, 0xc270, 0xefe,
2648 0xc26d, 0xeff8, 0xc26e, 0xfec, 0xfed, 0xeff1, 0xc273, 0xc272,
2649 0xef0, 0xc378, 0xf25f, 0xf265, 0xc379, 0xf25c, 0xc376, 0xc373,
2650 0xf267, 0xc377, 0xc374, 0xf25e, 0xf261, 0xf262, 0xf263, 0xf266,
2651 0xeff5, 0xf25d, 0xc375, 0xf264, 0xf268, 0xf260, 0xf45d, 0xc46a,
2652 0xf460, 0xc46b, 0xf468, 0xf45f, 0xf45c, 0xf45e, 0xf462, 0xf465,
2653 0xf464, 0xf467, 0xf45b, 0xc469, 0xf463, 0xf466, 0xf469, 0xf461,
2654 0xf5d3, 0xf5d4, 0xf5d8, 0xf5d9, 0xf5d6, 0xf5d7, 0xf5d5, 0xc4e9,
2655 0xc578, 0xf6eb, 0xf6e8, 0xf6e9, 0xf6ea, 0xc579, 0xf7e5, 0xf7e4,
2656 0xf8af, 0xc5f4, 0xf8ad, 0xf8ba, 0xf8b0, 0xf8ae, 0xf8f5, 0xc657, 0xc665,
2657 0xf9a3, 0xf96c, 0xf9a2, 0xf9d0, 0xf9d1, 0xa4f5, 0xa6c7, 0xca41,
2658 0xcb5e, 0xa85f, 0xa862, 0xcb5f, 0xa860, 0xa861, 0xcd58, 0xcd5a,
2659 0xcd55, 0xcd52, 0xcd54, 0xaa4, 0xaa2, 0xcd56, 0xaa3, 0xcd53,
2660 0xcd50, 0xaa1, 0xcd57, 0xcd51, 0xaa5, 0xcd59, 0xcfaf, 0xcfb3,
2661 0xacb7, 0xcfb6, 0xaca, 0xacb2, 0xacb4, 0xacb6, 0xacb3, 0xcfb2,
2662 0xcfb1, 0xacb1, 0xcfb4, 0xcfb5, 0xcfae, 0xacb5, 0xacb0, 0xcfb0,
2663 0xd277, 0xd278, 0xd279, 0xaf50, 0xaf4c, 0xd26e, 0xd276, 0xd27b,
2664 0xaf51, 0xd26c, 0xd272, 0xd26b, 0xd275, 0xd271, 0xaf4d, 0xaf4f,
2665 0xd27a, 0xd26a, 0xd26d, 0xd273, 0xd274, 0xd27c, 0xd270, 0xaf4e,
2666 0xb26d, 0xd64e, 0xd650, 0xd64c, 0xd658, 0xd64a, 0xd657, 0xb269,
2667 0xd648, 0xda5b, 0xd652, 0xb26c, 0xd653, 0xd656, 0xd65a, 0xd64f,
2668 0xd654, 0xb26a, 0xb26b, 0xd659, 0xd64d, 0xd649, 0xd65b, 0xd651,
2669 0xd655, 0xd64b, 0xb548, 0xb549, 0xda65, 0xb54f, 0xda59, 0xda62,
2670 0xda58, 0xb54c, 0xda60, 0xda5e, 0xda5f, 0xb54a, 0xda63, 0xda5c,
2671 0xda5a, 0xb54b, 0xda5d, 0xda61, 0xb54d, 0xda64, 0xde70, 0xde77,
2672 0xde79, 0xdea1, 0xb7da, 0xde6b, 0xb7d2, 0xde7a, 0xb7d7, 0xdea2,
2673 0xb7ce, 0xde7d, 0xde6d, 0xde7e, 0xde6c, 0xb7dc, 0xde78, 0xb7cf,
2674 0xdea3, 0xb7d4, 0xde71, 0xb7d9, 0xde7c, 0xde6f, 0xde76, 0xde72,
2675 0xde6e, 0xb7d1, 0xb7d8, 0xb7d6, 0xb7d3, 0xb7db, 0xb7d0, 0xde75,
2676 0xb7d5, 0xb54e, 0xde7b, 0xde73, 0xde74, 0xe2c1, 0xbab4, 0xe2bd,
2677 0xe2c3, 0xe2bf, 0xbab6, 0xe2be, 0xe2c2, 0xe2ba, 0xe2bc, 0xbab5,
2678 0xe2c0, 0xe2bb, 0xbab7, 0xbab2, 0xe2c4, 0xbab3, 0xe667, 0xe664,
2679 0xe670, 0xe66a, 0xe66c, 0xbcf4, 0xe666, 0xe66e, 0xe66d, 0xe66b,
2680 0xe671, 0xbcf7, 0xe668, 0xe66f, 0xbcf5, 0xe663, 0xe665, 0xbcf6,
2681 0xe662, 0xe672, 0xe669, 0xea4a, 0xbf51, 0xea55, 0xea53, 0xbf4b,
2682 0xea49, 0xea4c, 0xea4d, 0xea48, 0xbf55, 0xbf56, 0xea47, 0xea56,
2683 0xea51, 0xbf4f, 0xbf4c, 0xea50, 0xea4e, 0xbf52, 0xea52, 0xbf4d,
2684 0xbf4e, 0xea4f, 0xbf50, 0xea4b, 0xea54, 0xbf53, 0xea57, 0xea58,
2685 0xbf54, 0xc0e7, 0xc0ee, 0xed5c, 0xed62, 0xed60, 0xc0ea, 0xc0e9,
2686 0xc0e6, 0xed5e, 0xc0ec, 0xc0eb, 0xc0e8, 0xed61, 0xed5d, 0xed5f,
2687 0xc0ed, 0xc277, 0xeffb, 0xc274, 0xc275, 0xeffd, 0xc276, 0xeffa,
2688 0xeff9, 0xf26c, 0xeffc, 0xf26d, 0xc37a, 0xf26b, 0xf26a, 0xf269,
2689 0xc37b, 0xc46c, 0xf46a, 0xf46b, 0xf5dc, 0xf5db, 0xc4ea, 0xf5da,
2690 0xf6ec, 0xf6ed, 0xf7e6, 0xf8b1, 0xf8f6, 0xf9bc, 0xc679, 0xf9c6,
2691 0xa4f6, 0xaa6, 0xaa7, 0xacb8, 0xc0ef, 0xa4f7, 0xaa8, 0xaf52,
2692 0xb7dd, 0xa4f8, 0xb26e, 0xbab8, 0xc962, 0xcfb7, 0xd27d, 0xe2c5,
2693 0xc0f0, 0xa4f9, 0xaa9, 0xcfb8, 0xcfb9, 0xda66, 0xb550, 0xdea4,
2694 0xb7de, 0xe2c6, 0xbcf8, 0xc37c, 0xa4fa, 0xda67, 0xa4fb, 0xa6c9,
2695 0xca42, 0xa6c8, 0xa865, 0xa864, 0xa863, 0xcb60, 0xaaaa, 0xaaab,
2696 0xcd5b, 0xcfbba, 0xcfbdb, 0xacba, 0xcfbbb, 0xacb9, 0xcfbcb, 0xacbb,
2697 0xd2a2, 0xd2a1, 0xd27e, 0xaf53, 0xd65d, 0xd65e, 0xb26f, 0xd65c,
2698 0xd65f, 0xb552, 0xb270, 0xb551, 0xda6b, 0xda6a, 0xda68, 0xda69,
2699 0xda6c, 0xdea6, 0xdea5, 0xdea9, 0xdea8, 0xdea7, 0xbab9, 0xe2c9,
2700 0xe2c8, 0xbaba, 0xe2c7, 0xe673, 0xe674, 0xbcf9, 0xea59, 0xea5a,
2701 0xf272, 0xc37d, 0xf271, 0xf270, 0xf26e, 0xf26f, 0xc4eb, 0xf46c,
2702 0xf6ee, 0xf8f7, 0xa4fc, 0xc9a5, 0xa5c7, 0xc9a6, 0xca43, 0xca44,
2703 0xcb66, 0xcb62, 0xcb61, 0xaaac, 0xcb65, 0xa867, 0xcb63, 0xa866,
2704 0xcb67, 0xcb64, 0xcd5f, 0xcfbe, 0xcd5d, 0xcd64, 0xaaad, 0xaab0,
2705 0xcd65, 0xcd61, 0xcd62, 0xcd5c, 0xaaaf, 0xcd5e, 0xaaae, 0xcd63,
2706 0xcd60, 0xcfc2, 0xcacbd, 0xcacbe, 0xcfc5, 0xcfbf, 0xcfc4, 0xcfc0,
2707 0xcacbc, 0xcfc3, 0xcfc1, 0xd2a8, 0xd2a5, 0xd2a7, 0xaf58, 0xaf57,
2708 0xaf55, 0xd2a4, 0xd2a9, 0xaf54, 0xaf56, 0xd2a6, 0xd667, 0xd2a3,
2709 0xd2aa, 0xd662, 0xd666, 0xd665, 0xda6e, 0xda79, 0xd668, 0xd663,
2710 0xda6d, 0xb274, 0xb273, 0xd661, 0xd664, 0xb275, 0xb272, 0xb271,
2711 0xd660, 0xd669, 0xda70, 0xda77, 0xb554, 0xda76, 0xda73, 0xb556,
2712 0xda75, 0xda6f, 0xda71, 0xda74, 0xda72, 0xb555, 0xda78, 0xb553,
2713 0xb7df, 0xdead, 0xdead, 0xdead, 0xb7e2, 0xb7e1, 0xdead, 0xdead,

2714 0xe2ca, 0xbabb, 0xb7e0, 0xdeb0, 0xdeaf, 0xe2cd, 0xe2cb, 0xbcfaf,
2715 0xbabc, 0xe2cc, 0xe676, 0xbcbf, 0xe675, 0xe67e, 0xe67d, 0xe67b,
2716 0xe67a, 0xe677, 0xe678, 0xe679, 0xe67c, 0xe6a1, 0xea5f, 0xea5c,
2717 0xea5d, 0xbf57, 0xea5b, 0xea61, 0xea60, 0xea5e, 0xed64, 0xed65,
2718 0xc0f1, 0xc0f2, 0xed63, 0xc279, 0xe6fe, 0xc278, 0xc37e, 0xc3a1,
2719 0xc46d, 0xf46e, 0xf46d, 0xf5dd, 0xf6ef, 0xc57a, 0xf7e8, 0xf7e7,
2720 0xf7e9, 0xa5c8, 0xcfc6, 0xaf59, 0xb276, 0xd66a, 0xa5c9, 0xc9a7,
2721 0xa4fd, 0xca45, 0xcb6c, 0xcb6a, 0xcb6b, 0xcb68, 0xa868, 0xcb69,
2722 0xcd6d, 0xaab3, 0xcd6b, 0xcd67, 0xcd6a, 0xcd66, 0xaab5, 0xcd69,
2723 0xaab2, 0xaab1, 0xaab4, 0xcd6c, 0xcd68, 0xacc2, 0xacc5, 0xcfce,
2724 0xcfd, 0xcfcc, 0xcacf, 0xcfd5, 0xcfcf, 0xacc1, 0xd2af, 0xcfd2,
2725 0xcfd0, 0xacc4, 0xcfc8, 0xcfd3, 0xcfcf, 0xcfd4, 0xcfd1, 0xcfc9,
2726 0xacc0, 0xcfd6, 0xcfc7, 0xacc3, 0xd2b4, 0xd2ab, 0xd2b6, 0xd2ae,
2727 0xd2b9, 0xd2ba, 0xd2ac, 0xd2b8, 0xd2b5, 0xd2b3, 0xd2b7, 0xaf5f,
2728 0xaf5d, 0xd2b1, 0xd2ad, 0xd2b0, 0xd2bb, 0xd2b2, 0xaf5e, 0xcfcf,
2729 0xaf5a, 0xaf5c, 0xd678, 0xd66d, 0xd66b, 0xd66c, 0xd673, 0xd674,
2730 0xd670, 0xb27b, 0xd675, 0xd672, 0xd66f, 0xb279, 0xd66e, 0xb277,
2731 0xb27a, 0xd671, 0xd679, 0xaf5b, 0xb278, 0xd677, 0xd676, 0xb27c,
2732 0xda7e, 0xdaa1, 0xb560, 0xdaa7, 0xdaa9, 0xdaa2, 0xb55a, 0xdaa6,
2733 0xdaa5, 0xb55b, 0xb561, 0xb562, 0xdaa8, 0xb558, 0xda7d, 0xda7b,
2734 0xdaa3, 0xda7a, 0xb55f, 0xda7c, 0xdaa4, 0xdaa0, 0xb559, 0xb55e,
2735 0xb55c, 0xb55d, 0xb557, 0xb7e9, 0xdeb7, 0xb7e8, 0xdeb, 0xdeb1,
2736 0xdeb, 0xdeb2, 0xdeb3, 0xdeb, 0xdeb, 0xdeb, 0xdeb8, 0xdeb9, 0xdeb5,
2737 0xdeb4, 0xdeb, 0xb7e5, 0xdeb6, 0xb7e4, 0xb7e4, 0xb7e, 0xb7ec,
2738 0xb7e7, 0xb7e6, 0xe2ce, 0xbabe, 0xbabd, 0xe2d3, 0xbcf, 0xbabf,
2739 0xbac1, 0xe2d4, 0xb7e3, 0xbac0, 0xe2d0, 0xe2d2, 0xe2cf, 0xe2d1,
2740 0xe6ab, 0xe6aa, 0xe6a7, 0xbd40, 0xea62, 0xbd41, 0xea6a, 0xbcf,
2741 0xe6a8, 0xe6a5, 0xe6a2, 0xe6a9, 0xe6a3, 0xe6a4, 0xbcf, 0xed69,
2742 0xea66, 0xea65, 0xea67, 0xed66, 0xbf5a, 0xea63, 0xbf58, 0xbf5c,
2743 0xbf5b, 0xea64, 0xea68, 0xbf59, 0xed6d, 0xc0f5, 0xc27a, 0xc0f6,
2744 0xc0f3, 0xed6a, 0xed68, 0xed6b, 0xed6e, 0xc0f4, 0xed6c, 0xed67,
2745 0xf042, 0xf045, 0xf275, 0xf040, 0xf46f, 0xf046, 0xc3a2, 0xf044,
2746 0xc27b, 0xf041, 0xf043, 0xf043, 0xf047, 0xf276, 0xf274, 0xc3a3, 0xf273,
2747 0xc46e, 0xc4ed, 0xf6f1, 0xc4ec, 0xf6f3, 0xf6f0, 0xf6f2, 0xc5d0,
2748 0xf8b2, 0xa5ca, 0xcd6e, 0xd2bc, 0xd2bd, 0xb27d, 0xdeb, 0xbf5d,
2749 0xc3a4, 0xc57b, 0xf8b3, 0xa5cb, 0xcd6f, 0xa260, 0xcfd7, 0xcfd8,
2750 0xd2be, 0xd2bf, 0xb27e, 0xb2a1, 0xdaab, 0xdec2, 0xdec1, 0xdec0,
2751 0xe2d5, 0xe2d6, 0xe2d7, 0xbac2, 0xe6ad, 0xe6ac, 0xea69, 0xbf5e,
2752 0xbf5f, 0xed72, 0xed6f, 0xed70, 0xed71, 0xf049, 0xf048, 0xc27c,
2753 0xf277, 0xf5de, 0xa5cc, 0xacc6, 0xb2a2, 0xdec3, 0xa5cd, 0xd2c0,
2754 0xb2a3, 0xb563, 0xb564, 0xa5ce, 0xa5cf, 0xca46, 0xa86a, 0xa869,
2755 0xacc7, 0xcfd9, 0xdaac, 0xa5d0, 0xa5d1, 0xa5d2, 0xa5d3, 0xa86b,
2756 0xa86c, 0xcb6e, 0xcb6d, 0xaab6, 0xcd72, 0xcd70, 0xcd71, 0xcfd, 0xcfd,
2757 0xcfdb, 0xaccb, 0xacc9, 0xacc, 0xacc, 0xaf60, 0xaf64, 0xaf63,
2758 0xd2c1, 0xaf62, 0xaf61, 0xd2c2, 0xb2a6, 0xd67b, 0xd67a, 0xb2a4,
2759 0xb2a5, 0xb566, 0xb565, 0xdaae, 0xdaad, 0xb2a7, 0xb7ed, 0xdec5,
2760 0xb7ee, 0xdec4, 0xe2d8, 0xe6ae, 0xbd42, 0xea6a, 0xed73, 0xc3a6,
2761 0xc3a5, 0xc57c, 0xa5d4, 0xcd73, 0xb2a8, 0xe2d9, 0xbac3, 0xcb6f,
2762 0xcb70, 0xcd74, 0xaab8, 0xaab9, 0xaab7, 0xaccf, 0xacc0, 0xaccd,
2763 0xacce, 0xcfd, 0xcfd, 0xacc, 0xd2c3, 0xaf68, 0xaf69, 0xb2ab,
2764 0xd2c9, 0xaf6e, 0xaf6c, 0xd2ca, 0xd2c5, 0xaf6b, 0xaf6a, 0xaf65,
2765 0xd2c8, 0xd2c7, 0xd2c4, 0xaf6d, 0xd2c6, 0xaf66, 0xaf67, 0xb2ac,
2766 0xd6a1, 0xd6a2, 0xb2ad, 0xd67c, 0xd67e, 0xd6a4, 0xd6a3, 0xd67d,
2767 0xb2a9, 0xb2aa, 0xdab6, 0xb56b, 0xb56a, 0xdab0, 0xb568, 0xdab3,
2768 0xb56c, 0xdab4, 0xb56d, 0xdab1, 0xb567, 0xb569, 0xdab5, 0xdab2,
2769 0xdaaf, 0xded2, 0xdec7, 0xb7f0, 0xb7f3, 0xb7f2, 0xb7f7, 0xb7f6,
2770 0xded3, 0xded1, 0xdec, 0xdec, 0xdec, 0xb7f4, 0xded0, 0xdec,
2771 0xded4, 0xdec, 0xb7f5, 0xb7ef, 0xb7f1, 0xdec9, 0xe2db, 0xbac7,
2772 0xe2df, 0xbac6, 0xe2dc, 0xbac5, 0xdec8, 0xdecf, 0xe2de, 0xbac8,
2773 0xe2e0, 0xe2dd, 0xe2da, 0xe6b1, 0xe6b5, 0xe6b7, 0xe6b3, 0xe6b2,
2774 0xe6b0, 0xbd45, 0xbd43, 0xbd48, 0xbd49, 0xe6b4, 0xbd46, 0xe6af,
2775 0xbd47, 0xbac4, 0xe6b6, 0xbd44, 0xea6c, 0xea6b, 0xea73, 0xea6d,
2776 0xea72, 0xea6f, 0xbf60, 0xea71, 0xbf61, 0xbf62, 0xea70, 0xea6e,
2777 0xc0f8, 0xed74, 0xc0f7, 0xed77, 0xed75, 0xed76, 0xc0f9, 0xf04d,
2778 0xc2a1, 0xf04e, 0xc27d, 0xf04f, 0xc27e, 0xf04c, 0xf050, 0xf04a,
2779 0xc3a7, 0xf278, 0xc3a8, 0xc46f, 0xf04b, 0xc470, 0xc4ee, 0xf5df,
2780 0xc57e, 0xf6f4, 0xc57d, 0xf7ea, 0xc5f5, 0xc5f6, 0xf9cc, 0xacd1,
2781 0xcfd, 0xb56e, 0xb56f, 0xa5d5, 0xa6ca, 0xca47, 0xcb71, 0xa86d,
2782 0xaaba, 0xacd2, 0xacd3, 0xacd4, 0xd6a6, 0xd2cb, 0xaf6f, 0xb2ae,
2783 0xd6a5, 0xdab8, 0xb571, 0xdab7, 0xb570, 0xded5, 0xbd4a, 0xe6bb,
2784 0xe6b8, 0xe6b9, 0xe6ba, 0xed78, 0xf051, 0xf471, 0xf470, 0xf6f5,
2785 0xa5d6, 0xcd75, 0xaf70, 0xb572, 0xded6, 0xe2e1, 0xbd4b, 0xea74,
2786 0xf052, 0xf472, 0xa5d7, 0xaabb, 0xacd7, 0xcfd, 0xacd8, 0xacd6,
2787 0xacd5, 0xd2cc, 0xaf71, 0xaf72, 0xaf73, 0xb2b0, 0xd6a7, 0xb2af,
2788 0xdab9, 0xb2b1, 0xb573, 0xded7, 0xb7f8, 0xb7f9, 0xbac9, 0xbaca,
2789 0xbd4c, 0xbf64, 0xea75, 0xbf63, 0xed79, 0xc0fa, 0xf053, 0xf473,
2790 0xa5d8, 0xa86e, 0xcd78, 0xcd77, 0xaabc, 0xcd76, 0xaabd, 0xcd79,
2791 0xcfe5, 0xacdb, 0xacda, 0xcfe7, 0xcfe6, 0xacdf, 0xacde, 0xacd9,
2792 0xcfe1, 0xcfe2, 0xcfe3, 0xace0, 0xcfe0, 0xacdc, 0xcfe4, 0xacdd,
2793 0xd2cf, 0xd2d3, 0xd2d1, 0xd2d0, 0xd2d4, 0xd2d5, 0xd2d6, 0xd2ce,
2794 0xd2cd, 0xaf75, 0xaf76, 0xd2d7, 0xd2d2, 0xd6b0, 0xd2d8, 0xaf77,
2795 0xaf74, 0xd6aa, 0xd6a9, 0xd6ab, 0xd6ac, 0xd6ae, 0xd6ad, 0xd6b2,
2796 0xb2b5, 0xb2b2, 0xb2b6, 0xd6a8, 0xb2b7, 0xd6b1, 0xb2b4, 0xd6af,
2797 0xb2b3, 0xdabc, 0xdabe, 0xdaba, 0xdabb, 0xdabf, 0xdac1, 0xdac2,
2798 0xdabd, 0xdac0, 0xb574, 0xdedb, 0xdec, 0xded8, 0xdedc, 0xdec1,
2799 0xdedd, 0xb7fa, 0xb843, 0xb7fd, 0xded9, 0xdeda, 0xbace, 0xb846,
2800 0xb7fe, 0xb844, 0xb7fc, 0xdedf, 0xb845, 0xdede, 0xb841, 0xb7fb,

2801 0xb842, 0xdee2, 0xe2e6, 0xe2e8, 0xb840, 0xe2e3, 0xbacc, 0xe2e9,
2802 0xbacd, 0xe2e7, 0xe2e2, 0xe2e5, 0xe2ea, 0xbacb, 0xe2e4, 0xbd4e,
2803 0xe6bf, 0xe6be, 0xbd51, 0xbd4f, 0xe6bc, 0xbd4d, 0xe6bd, 0xbd50,
2804 0xea7d, 0xeaa1, 0xea7e, 0xea76, 0xea7a, 0xea79, 0xea77, 0xbf66,
2805 0xbf67, 0xbf65, 0xea78, 0xea7b, 0xea7c, 0xbf68, 0xc140, 0xeda3,
2806 0xc0fc, 0xed7b, 0xc0fe, 0xc141, 0xc0fd, 0xeda2, 0xed7c, 0xc0fb,
2807 0xeda1, 0xed7a, 0xed7e, 0xed7d, 0xf055, 0xc2a4, 0xc2a5, 0xc2a2,
2808 0xc2a3, 0xf054, 0xf27b, 0xc3a9, 0xf279, 0xf27a, 0xf474, 0xf477,
2809 0xf475, 0xf476, 0xf5e0, 0xc4ef, 0xf7eb, 0xf8b4, 0xc5f7, 0xf8f8,
2810 0xf8f9, 0xc666, 0xa5d9, 0xace1, 0xdac3, 0xdee3, 0xa5da, 0xa86f,
2811 0xaabe, 0xcfe8, 0xcfe9, 0xaf78, 0xdac4, 0xb575, 0xb847, 0xc142,
2812 0xeda4, 0xf27c, 0xf478, 0xa5db, 0xcda1, 0xcd7a, 0xcd7c, 0xcd7e,
2813 0xcd7d, 0xcd7b, 0xaabf, 0xace2, 0xcff2, 0xcfed, 0xcfea, 0xcff1,
2814 0xace4, 0xace5, 0xcff0, 0xcfef, 0xcfee, 0xcfeb, 0xcfec, 0xcff3,
2815 0xace3, 0xaf7c, 0xaf44, 0xaf43, 0xd2e1, 0xd2db, 0xd2d9, 0xafaf,
2816 0xd6b9, 0xaf7a, 0xd2de, 0xd2e2, 0xd2e4, 0xd2e0, 0xd2da, 0xafaf,
2817 0xd2df, 0xd2dd, 0xaf79, 0xd2e5, 0xafaf, 0xd2e3, 0xaf7d, 0xd2dc,
2818 0xaf7e, 0xaf7b, 0xb2b9, 0xd6ba, 0xd6b3, 0xd6b5, 0xd6b7, 0xd6b8,
2819 0xd6b6, 0xb2ba, 0xd6bb, 0xd6b4, 0xdac8, 0xb576, 0xdad0, 0xdac5,
2820 0xdad1, 0xdac6, 0xdac7, 0xdacf, 0xdace, 0xdacb, 0xb2b8, 0xb577,
2821 0xdac9, 0xdacc, 0xb578, 0xdacd, 0xdaca, 0xdee, 0xdf2, 0xb84e,
2822 0xe2f0, 0xb851, 0xdef0, 0xdead, 0xdee8, 0xdea, 0xdeb, 0xee4,
2823 0xb84d, 0xb84c, 0xb848, 0xee7, 0xb84f, 0xb850, 0xee6, 0xee9,
2824 0xdef1, 0xb84a, 0xb84b, 0xdef, 0xee5, 0xe2f2, 0xbad0, 0xe2f4,
2825 0xdec, 0xe2f6, 0xbad4, 0xe2f7, 0xe2f3, 0xbad1, 0xe2ef, 0xbad3,
2826 0xe2ec, 0xe2f1, 0xe2f5, 0xe2ee, 0xb849, 0xe2eb, 0xbad2, 0xe2ed,
2827 0xbd54, 0xe6c1, 0xbd58, 0xbd56, 0xbacf, 0xe6c8, 0xe6c9, 0xbd53,
2828 0xe6c7, 0xe6ca, 0xbd55, 0xbd52, 0xe6c3, 0xe6c0, 0xe6c5, 0xe6c2,
2829 0xbd59, 0xe6c4, 0xe6c6, 0xbd57, 0xbf6a, 0xea8, 0xea2, 0xea6,
2830 0xeaac, 0xeaad, 0xea9, 0xea7, 0xea4, 0xbf6c, 0xbf69,
2831 0xea3, 0xea5, 0xbf6b, 0xeaab, 0xc146, 0xeda, 0xeda5, 0xc145,
2832 0xc143, 0xedac, 0xc144, 0xeda8, 0xeda9, 0xeda6, 0xedad, 0xf056,
2833 0xc147, 0xeda7, 0xeda, 0xedab, 0xf05a, 0xf057, 0xc2a6, 0xf05b,
2834 0xf05d, 0xf05c, 0xf058, 0xf059, 0xf2a3, 0xc3aa, 0xf27e, 0xf2a2,
2835 0xf27d, 0xf2a4, 0xf2a1, 0xf47a, 0xf47d, 0xf479, 0xc471, 0xf47b,
2836 0xf47c, 0xf47e, 0xc472, 0xc473, 0xc474, 0xf5e1, 0xf5e3, 0xf5e2,
2837 0xf6f6, 0xf8b5, 0xf8fa, 0xa5dc, 0xcb72, 0xaac0, 0xcda3, 0xaac1,
2838 0xaac2, 0xcda2, 0xcff8, 0xcff7, 0xace6, 0xace9, 0xace8, 0xace7,
2839 0xcff4, 0xcff6, 0xcff5, 0xd2e8, 0xafaf, 0xd2ec, 0xd2eb, 0xd2ea,
2840 0xd2e6, 0xafaf, 0xafaf, 0xafad, 0xafae, 0xd2e7, 0xd2e9, 0xafaf,
2841 0xafaf, 0xafaf, 0xafaf, 0xd6c2, 0xd6c0, 0xd6bc, 0xb2bb, 0xd6bd,
2842 0xb2bc, 0xd6be, 0xd6bf, 0xd6c1, 0xb2bd, 0xdad5, 0xdad4, 0xdad3,
2843 0xdad2, 0xdef6, 0xb852, 0xdef3, 0xdef5, 0xb853, 0xb854, 0xdef4,
2844 0xe341, 0xe2f9, 0xe2fa, 0xbad7, 0xbad5, 0xbad6, 0xe343, 0xe342,
2845 0xe2fe, 0xe2fd, 0xe2fc, 0xe2fb, 0xe340, 0xe2f8, 0xe6cb, 0xe6d0,
2846 0xe6ce, 0xe6cd, 0xe6cc, 0xe6cf, 0xeaee, 0xbf6d, 0xc148, 0xedb0,
2847 0xc149, 0xedaf, 0xf05f, 0xf05e, 0xc2a7, 0xf2a5, 0xc3ab, 0xf4a1,
2848 0xc5a1, 0xf6f7, 0xf8b7, 0xf8b6, 0xc9a8, 0xace, 0xaceb, 0xd6c3,
2849 0xb856, 0xa5dd, 0xa872, 0xa871, 0xa870, 0xcda4, 0xaac4, 0xaac3,
2850 0xacee, 0xcffa, 0xcffd, 0xcffb, 0xacec, 0xaced, 0xcff9, 0xcffc,
2851 0xafb5, 0xd2f3, 0xd2f5, 0xd2f4, 0xafb2, 0xd2ef, 0xafb0, 0xafaf,
2852 0xafb3, 0xafb1, 0xafb4, 0xd2f2, 0xd2ed, 0xd2ee, 0xd2f1, 0xd2f0,
2853 0xd6c6, 0xd6c7, 0xd6c5, 0xd6c4, 0xb2be, 0xb57d, 0xdad6, 0xdad8,
2854 0xdada, 0xb57c, 0xb57a, 0xdad7, 0xb57b, 0xdad9, 0xb579, 0xdf41,
2855 0xdef7, 0xdefa, 0xdefe, 0xb85a, 0xdefc, 0xdefb, 0xdef8, 0xdef9,
2856 0xb858, 0xdf40, 0xb857, 0xb85c, 0xb85b, 0xb859, 0xdefd, 0xe349,
2857 0xe348, 0xe344, 0xbad8, 0xe347, 0xe346, 0xbad9, 0xb5e, 0xe6d2,
2858 0xbd5f, 0xbd5b, 0xbd5d, 0xbd5a, 0xbd5c, 0xeaaf, 0xbf70, 0xeab1,
2859 0xeab0, 0xe345, 0xbf72, 0xbf71, 0xbf6e, 0xbf6f, 0xedb5, 0xedb3,
2860 0xc14a, 0xedb4, 0xedb6, 0xedb2, 0xedb1, 0xf060, 0xc2aa, 0xc2a8,
2861 0xc2a9, 0xf2a6, 0xf2a7, 0xc3ad, 0xc3ac, 0xf4a3, 0xf4a4, 0xf4a2,
2862 0xf6f8, 0xf6f9, 0xa5de, 0xca48, 0xa873, 0xcda5, 0xaac6, 0xaac5,
2863 0xcda6, 0xd040, 0xacef, 0xcffe, 0xacf0, 0xafb6, 0xd2f8, 0xd2f6,
2864 0xd2fc, 0xafb7, 0xd2f7, 0xd2fb, 0xd2f9, 0xd2fa, 0xd6c8, 0xd6ca,
2865 0xb2bf, 0xd6c9, 0xb2c0, 0xb5a1, 0xb5a2, 0xb57e, 0xdadb, 0xdf44,
2866 0xb85d, 0xb85e, 0xdf43, 0xdf42, 0xe34a, 0xbadb, 0xbada, 0xe34b,
2867 0xe34c, 0xbd61, 0xbd60, 0xeab5, 0xe6d3, 0xe6d5, 0xe6d4, 0xeab4,
2868 0xeab2, 0xeab6, 0xeab3, 0xbf73, 0xedb7, 0xc14b, 0xedb8, 0xedb9,
2869 0xc2ab, 0xc2ac, 0xc475, 0xc5d1, 0xa5df, 0xd041, 0xd2fd, 0xafb8,
2870 0xb3ba, 0xb3b9, 0xb5a4, 0xdadd, 0xb5a3, 0xdadac, 0xdf45, 0xbadc,
2871 0xe34d, 0xbadd, 0xc476, 0xf4a5, 0xa6cb, 0xaac7, 0xcda7, 0xacf2,
2872 0xacf1, 0xd042, 0xd043, 0xd340, 0xd342, 0xafb9, 0xd344, 0xd347,
2873 0xd345, 0xd346, 0xd343, 0xd2fe, 0xafba, 0xd348, 0xd341, 0xd6d3,
2874 0xb2c6, 0xd6dc, 0xb2c3, 0xd6d5, 0xb2c7, 0xb2c1, 0xd6d0, 0xd6dd,
2875 0xd6d1, 0xd6ce, 0xb2c5, 0xb2c2, 0xd6d4, 0xd6d7, 0xb2c4, 0xd6d8,
2876 0xb2c8, 0xd6d9, 0xd6dc, 0xd6de, 0xd6da, 0xd6db, 0xd6cd, 0xd6cb,
2877 0xd6db, 0xdadf, 0xdae4, 0xdae0, 0xdae6, 0xb5a7, 0xd6cc, 0xdae1,
2878 0xb5a5, 0xdade, 0xb5ac, 0xdae2, 0xb5ab, 0xdae3, 0xb5ad, 0xb5a8,
2879 0xb5ae, 0xb5a9, 0xb5aa, 0xb5a6, 0xdae5, 0xb861, 0xdf50, 0xdf53,
2880 0xdf47, 0xdf4c, 0xdf46, 0xb863, 0xdf4a, 0xdf48, 0xb862, 0xdf4f,
2881 0xdf4e, 0xdf4b, 0xdf4d, 0xdf49, 0xbae1, 0xdf52, 0xb85f, 0xdf51,
2882 0xe35d, 0xbae8, 0xe358, 0xbae7, 0xe34e, 0xe350, 0xbae0, 0xe355,
2883 0xe354, 0xe357, 0xbae5, 0xe352, 0xe351, 0xbae4, 0xbadf, 0xe353,
2884 0xbae2, 0xe359, 0xe35b, 0xe356, 0xe34f, 0xbae3, 0xbd69, 0xbade,
2885 0xe35c, 0xe6d9, 0xbd62, 0xe6db, 0xbd63, 0xbd65, 0xe6de, 0xe6de,
2886 0xbae6, 0xe6dc, 0xe6d8, 0xb860, 0xbd68, 0xbd64, 0xbd66, 0xbd67,
2887 0xbf76, 0xe6dd, 0xe6d7, 0xbd6a, 0xe6da, 0xeac0, 0xeabb, 0xeac5,

2888 0xbf74, 0xeabd, 0xbf78, 0xeac3, 0xeaba, 0xeab7, 0xeac6, 0xc151,
2889 0xbf79, 0xeac2, 0xeab8, 0xbf77, 0xeabc, 0xbf7b, 0xeab9, 0xeabe,
2890 0xbf7a, 0xeac1, 0xeac4, 0xedcb, 0xedcc, 0xedbc, 0xedc3, 0xedc1,
2891 0xc14f, 0xedc8, 0xeabf, 0xedbf, 0xedc9, 0xc14e, 0xedbe, 0xedbd,
2892 0xedc7, 0xedc4, 0xedc6, 0xedba, 0xedca, 0xc14c, 0xedc5, 0xedce,
2893 0xedc2, 0xc150, 0xc14d, 0xedc0, 0xedbb, 0xedcd, 0xbf75, 0xf063,
2894 0xf061, 0xf067, 0xc2b0, 0xf065, 0xf064, 0xc2b2, 0xf06a, 0xc2b1,
2895 0xf06b, 0xf068, 0xc2ae, 0xf069, 0xf062, 0xc2af, 0xc2ad, 0xf2ab,
2896 0xf066, 0xf06c, 0xf2a8, 0xc3b2, 0xc3b0, 0xf2aa, 0xf2ac, 0xf2a9,
2897 0xc3b1, 0xc3ae, 0xc3af, 0xc3b3, 0xc478, 0xf4aa, 0xf4a9, 0xf4a7,
2898 0xf4a6, 0xf4a8, 0xc477, 0xc479, 0xc4f0, 0xf5e5, 0xf5e4, 0xf6fa,
2899 0xf6fc, 0xf6fe, 0xf6fd, 0xf6fb, 0xc5a3, 0xc5a2, 0xc5d3, 0xc5d2,
2900 0xc5d4, 0xf7ed, 0xf7ec, 0xf8fb, 0xf8b8, 0xf8fc, 0xc658, 0xc659,
2901 0xf96d, 0xc67e, 0xa6cc, 0xcda8, 0xd045, 0xd046, 0xd044, 0xacf3,
2902 0xd047, 0xd048, 0xd049, 0xd049, 0xd349, 0xd34f, 0xd34d, 0xafbb, 0xd34b,
2903 0xd34c, 0xd34e, 0xd34a, 0xb2c9, 0xd6de, 0xb2cb, 0xd6e0, 0xb2ca,
2904 0xd6df, 0xdae8, 0xb5af, 0xdaea, 0xdae7, 0xd6e1, 0xb5b0, 0xdae9,
2905 0xdf56, 0xb864, 0xdf54, 0xb865, 0xdf55, 0xb866, 0xbae9, 0xe361,
2906 0xe35e, 0xe360, 0xbaea, 0xbaeb, 0xe35f, 0xe6df, 0xe6e0, 0xbd6b,
2907 0xe6e2, 0xe6e1, 0xa261, 0xeaca, 0xeacb, 0xeac7, 0xeac8, 0xbf7c,
2908 0xbf7d, 0xeac9, 0xc157, 0xc153, 0xc158, 0xc154, 0xc156, 0xc152,
2909 0xc155, 0xc2b3, 0xedcf, 0xf2ae, 0xf2ad, 0xf4ab, 0xc47a, 0xc47b,
2910 0xf741, 0xf5e6, 0xf740, 0xf8fd, 0xf9a4, 0xa6cd, 0xa74, 0xcda9,
2911 0xaac8, 0xacf6, 0xacf6, 0xd04c, 0xacf4, 0xd04a, 0xacf9, 0xacf5, 0xacfa,
2912 0xacf8, 0xd04b, 0xacf7, 0xafbf, 0xafbe, 0xd35a, 0xafc7, 0xd353,
2913 0xd359, 0xafc3, 0xd352, 0xd358, 0xd356, 0xafc2, 0xafc4, 0xd355,
2914 0xafbd, 0xd354, 0xafc8, 0xafc5, 0xafc9, 0xafc6, 0xd351, 0xd350,
2915 0xd357, 0xafc0, 0xafbc, 0xafc1, 0xd6f0, 0xd6e9, 0xb5b5, 0xd6e8,
2916 0xb2cf, 0xb2d6, 0xb2d3, 0xb2d9, 0xb2d8, 0xb2d4, 0xd6e2, 0xd6e5,
2917 0xd6e4, 0xb2d0, 0xd6e6, 0xd6ef, 0xb2d1, 0xd6e3, 0xd6ec, 0xd6ed,
2918 0xb2d2, 0xd6ea, 0xb2d7, 0xb2cd, 0xb2d5, 0xd6e7, 0xb2cc, 0xd6eb,
2919 0xd6ee, 0xdafb, 0xdaf2, 0xb5b2, 0xdaf9, 0xdaf6, 0xdaee, 0xdaf7,
2920 0xb5b4, 0xdaef, 0xdaeb, 0xb86c, 0xdaf4, 0xb5b1, 0xdafa, 0xb5b8,
2921 0xb5ba, 0xdaed, 0xb5b9, 0xdaf0, 0xb5b3, 0xdaf8, 0xdaf1, 0xdaf5,
2922 0xdaf3, 0xb5b6, 0xdaec, 0xb5bb, 0xb2ce, 0xb5b7, 0xb5bc, 0xb868,
2923 0xdf5d, 0xdf5f, 0xdf61, 0xdf65, 0xdf5b, 0xdf59, 0xb86a, 0xdf60,
2924 0xdf64, 0xdf5c, 0xdf58, 0xdf57, 0xdf62, 0xdf5a, 0xdf5e, 0xb86b,
2925 0xb869, 0xdf66, 0xb867, 0xdf63, 0xe372, 0xbaee, 0xe36a, 0xbd78,
2926 0xe374, 0xbaf1, 0xe378, 0xbaf7, 0xe365, 0xe375, 0xe362, 0xe377,
2927 0xe366, 0xbafe, 0xbafb, 0xe376, 0xe370, 0xbaed, 0xbaf5, 0xbaf4,
2928 0xbaf3, 0xbaf9, 0xe363, 0xbafa, 0xe371, 0xbaf6, 0xbaec, 0xe373,
2929 0xbaef, 0xbaf0, 0xbaf8, 0xe368, 0xe367, 0xe364, 0xe36c, 0xe369,
2930 0xe36d, 0xbafd, 0xe379, 0xbaf2, 0xe36e, 0xe36f, 0xe36b, 0xbafc,
2931 0xe6e7, 0xbd70, 0xbd79, 0xbd75, 0xe6e4, 0xbd72, 0xbd76, 0xe6f0,
2932 0xbd6c, 0xe6e8, 0xbd74, 0xe6eb, 0xe6e6, 0xbd73, 0xbd77, 0xe6e5,
2933 0xbd71, 0xe6ef, 0xbd6e, 0xe6ee, 0xe6ed, 0xbd7a, 0xe572, 0xbd6d,
2934 0xe6ec, 0xe6e3, 0xbd7b, 0xe6ea, 0xbd6f, 0xe6e9, 0xbfa2, 0xbfa7,
2935 0xbf7e, 0xead8, 0xeacf, 0xeadb, 0xead3, 0xead9, 0xbfa8, 0xbfa1,
2936 0xeacc, 0xead2, 0xeadc, 0xead5, 0xeada, 0xeace, 0xead6, 0xbfa3,
2937 0xead4, 0xbfa6, 0xbfa5, 0xead0, 0xead1, 0xeacd, 0xead7, 0xbfa4,
2938 0xeade, 0xeadd, 0xeadda, 0xedd6, 0xc15f, 0xedd0, 0xc159, 0xc169,
2939 0xeddc, 0xc161, 0xc15d, 0xedd3, 0xc164, 0xc167, 0xedde, 0xc15c,
2940 0xedd5, 0xc165, 0xede0, 0xeddd, 0xedd1, 0xc160, 0xc15a, 0xc168,
2941 0xedd8, 0xc163, 0xedd2, 0xc15e, 0xeddf, 0xc162, 0xc15b, 0xedd9,
2942 0xc166, 0xedd7, 0xeddb, 0xf06e, 0xf074, 0xc2b9, 0xf077, 0xc2b4,
2943 0xc2b5, 0xf06f, 0xf076, 0xf071, 0xc2ba, 0xc2b7, 0xf06d, 0xc2b6,
2944 0xf073, 0xf075, 0xc2b8, 0xf072, 0xf070, 0xf2b8, 0xc3b7, 0xc3b8,
2945 0xc3b4, 0xc3b5, 0xf2b4, 0xf2b2, 0xf2b6, 0xc3ba, 0xf2b7, 0xf2b0,
2946 0xf2af, 0xf2b3, 0xf2b1, 0xc3b6, 0xf2b5, 0xf4ac, 0xc47e, 0xc47d,
2947 0xf4ad, 0xf4af, 0xf4ae, 0xc4a1, 0xf5eb, 0xf5e8, 0xf5e9, 0xf5e7,
2948 0xf5ea, 0xc4f2, 0xf5ec, 0xc4f1, 0xf742, 0xc5d5, 0xc5d7, 0xf7ee,
2949 0xc5d6, 0xf8b9, 0xf940, 0xf942, 0xf8fe, 0xf941, 0xc66c, 0xa6ce,
2950 0xacfb, 0xd26f, 0xafca, 0xb2da, 0xdafc, 0xdafd, 0xeadf, 0xc16a,
2951 0xede1, 0xc2bb, 0xf2ba, 0xf2b9, 0xc4a2, 0xf5ed, 0xf743, 0xc5f8,
2952 0xca49, 0xaac9, 0xa875, 0xd04d, 0xd360, 0xd35b, 0xd35f, 0xd35d,
2953 0xafcb, 0xd35e, 0xd35c, 0xd6f1, 0xdafe, 0xdb40, 0xdf69, 0xdf6a,
2954 0xb86e, 0xb86f, 0xdf68, 0xdf6b, 0xdf67, 0xb86d, 0xbb40, 0xb870,
2955 0xe37a, 0xbd7c, 0xe6f1, 0xbd7d, 0xbfa9, 0xeae2, 0xeae0, 0xeae1,
2956 0xede4, 0xede3, 0xede2, 0xf2bb, 0xc3b9, 0xf2bc, 0xf744, 0xc5f9,
2957 0xf8ba, 0xa6cf, 0xaacb, 0xaaca, 0xd04f, 0xacfc, 0xd04e, 0xd362,
2958 0xafcc, 0xd6f2, 0xd361, 0xb2dc, 0xd6f5, 0xd6f3, 0xd6f4, 0xb2db,
2959 0xdb42, 0xdb43, 0xdb41, 0xb873, 0xdf6d, 0xdf6c, 0xdf6e, 0xb872,
2960 0xb871, 0xe6f2, 0xe6f4, 0xbd7e, 0xe6f3, 0xeae3, 0xbfaa, 0xf079,
2961 0xf078, 0xc3bb, 0xf2bd, 0xc3bc, 0xc3bc, 0xf4b0, 0xf5ee, 0xc4f3,
2962 0xa6d0, 0xd050, 0xacfd, 0xd365, 0xafce, 0xd364, 0xd363, 0xafcd,
2963 0xd6fb, 0xd6fd, 0xd6fe, 0xd6f7, 0xb2dd, 0xd6f8, 0xb2de, 0xd6fc,
2964 0xd6f9, 0xd6fa, 0xb2df, 0xb5be, 0xb5bf, 0xdb44, 0xdf6f, 0xdf70,
2965 0xe37e, 0xbb43, 0xbb41, 0xbb42, 0xe37b, 0xe37c, 0xe37d, 0xe6f9,
2966 0xe6fa, 0xbda1, 0xe6f7, 0xe6f6, 0xe6f8, 0xe6f5, 0xbfad, 0xeae4,
2967 0xbfab, 0xbfac, 0xede6, 0xc16b, 0xede5, 0xfaf8, 0xf07a, 0xf07b,
2968 0xc2bc, 0xc2bd, 0xc16c, 0xf2be, 0xf2bf, 0xf4b1, 0xc4a3, 0xa6d1,
2969 0xa6d2, 0xacfe, 0xaacc, 0xafcf, 0xd051, 0xb5c0, 0xa6d3, 0xad41,
2970 0xd052, 0xd053, 0xad40, 0xad42, 0xa6d4, 0xd054, 0xafd1, 0xd366,
2971 0xafd3, 0xafd0, 0xafd2, 0xd741, 0xb2e0, 0xd740, 0xd6fe, 0xdf71,
2972 0xe3a1, 0xbda2, 0xbfae, 0xeae6, 0xeae5, 0xede7, 0xf5ef, 0xa6d5,
2973 0xcb73, 0xcdaa, 0xad43, 0xd055, 0xd368, 0xafd4, 0xd367, 0xafd5,
2974 0xd743, 0xb2e2, 0xd742, 0xd744, 0xb2e1, 0xdb46, 0xdb47, 0xdb45,

2975 0xb5c1, 0xb874, 0xb875, 0xbb45, 0xe3a3, 0xe3a2, 0xbb44, 0xe6fb,
2976 0xe6fc, 0xee7, 0xc170, 0xc16f, 0xc16d, 0xc16e, 0xc171, 0xf07c,
2977 0xc2bf, 0xc2be, 0xf2c0, 0xf4b2, 0xc5a5, 0xc5a4, 0xa6d6, 0xd1fb,
2978 0xb877, 0xb5c2, 0xb876, 0xbb46, 0xa6d7, 0xc9a9, 0xa6d8, 0xa6d9,
2979 0xcdab, 0xcb76, 0xcb77, 0xa877, 0xcb74, 0xa876, 0xa879, 0xcb75,
2980 0xa87b, 0xa87a, 0xcb78, 0xa878, 0xaad1, 0xaacf, 0xcdad, 0xaace,
2981 0xaad3, 0xaad5, 0xaad2, 0xcdb0, 0xcdac, 0xaad6, 0xaad0, 0xa87c,
2982 0xaad4, 0xcdaf, 0xcdae, 0xaacd, 0xd05b, 0xad47, 0xad48, 0xd05d,
2983 0xd057, 0xd05a, 0xd063, 0xd061, 0xad49, 0xd067, 0xad4c, 0xd064,
2984 0xd05c, 0xd059, 0xdb49, 0xd062, 0xad44, 0xd065, 0xd056, 0xd05f,
2985 0xad46, 0xad4b, 0xd060, 0xad4f, 0xad4d, 0xd058, 0xad4a, 0xd05e,
2986 0xad4e, 0xad45, 0xd066, 0xafda, 0xafef, 0xafd8, 0xafd6, 0xd36a,
2987 0xafde, 0xafdb, 0xd36c, 0xafdd, 0xd36b, 0xd369, 0xd36e, 0xafef,
2988 0xafef, 0xdb48, 0xd36f, 0xd36d, 0xafd7, 0xafd9, 0xafdc, 0xafdf,
2989 0xafef, 0xd74e, 0xb2e4, 0xd745, 0xd747, 0xd748, 0xd750, 0xd74c,
2990 0xd74a, 0xd74d, 0xd751, 0xb2e5, 0xb2e9, 0xd746, 0xd74f, 0xb2e7,
2991 0xb2e6, 0xd74b, 0xd749, 0xb2e3, 0xb2e8, 0xb5c8, 0xdb51, 0xdb4f,
2992 0xb5ca, 0xdb4a, 0xdfa1, 0xb5c9, 0xdb4e, 0xdb4b, 0xb5c5, 0xb5cb,
2993 0xdb50, 0xb5c7, 0xdb4d, 0xbb47, 0xb5c6, 0xdb4c, 0xb5cc, 0xb5c4,
2994 0xb5c3, 0xdf77, 0xdf75, 0xdf7b, 0xdf73, 0xdfa2, 0xdf78, 0xdf72,
2995 0xb87b, 0xb8a3, 0xdf7d, 0xdf76, 0xb87e, 0xb87c, 0xdf7e, 0xb879,
2996 0xb878, 0xdf79, 0xb87d, 0xb5cd, 0xdf7c, 0xdf74, 0xb87a, 0xb8a1,
2997 0xb8a2, 0xbb4c, 0xbb48, 0xbb4d, 0xe3a6, 0xe3a5, 0xe3a7, 0xbb4a,
2998 0xe3a4, 0xbb4b, 0xe3aa, 0xe3a9, 0xe3a8, 0xbb49, 0xe741, 0xe744,
2999 0xbda8, 0xe743, 0xbda7, 0xbda3, 0xbda4, 0xbda5, 0xe740, 0xe6fe,
3000 0xbda6, 0xe742, 0xe6fd, 0xeae9, 0xeaf3, 0xbfb1, 0xbfb0, 0xeaed,
3001 0xeaeef, 0xeaea, 0xeaeef, 0xeae8, 0xeaf1, 0xbfaf, 0xeaf0, 0xeaec,
3002 0xeaf2, 0xeaeef, 0xc174, 0xede8, 0xede, 0xc178, 0xc17a, 0xc177,
3003 0xc176, 0xc175, 0xc173, 0xede9, 0xede, 0xc172, 0xede, 0xc179,
3004 0xede, 0xede, 0xc2c0, 0xc2c1, 0xf0a1, 0xf07d, 0xf07e, 0xf2c2,
3005 0xf2c1, 0xc3be, 0xf4b4, 0xc4a4, 0xf4b3, 0xf5f0, 0xf745, 0xc5a6,
3006 0xf943, 0xf944, 0xc5d8, 0xa6da, 0xaad7, 0xdb52, 0xbb4e, 0xc17b,
3007 0xedef, 0xa6db, 0xafef, 0xafef, 0xafef, 0xafef, 0xa6dc, 0xad50,
3008 0xdb54, 0xdb55, 0xdb56, 0xbb4f, 0xbfb2, 0xa6dd, 0xaad8, 0xd068,
3009 0xafef, 0xd370, 0xb2ea, 0xdb57, 0xb8a4, 0xbb50, 0xbfb3, 0xc17c,
3010 0xc2c2, 0xf4b5, 0xa6de, 0xaad9, 0xafef, 0xd752, 0xb5ce, 0xbb51,
3011 0xe3ab, 0xe745, 0xa6df, 0xb5cf, 0xdfa3, 0xbb52, 0xa6e0, 0xcdb1,
3012 0xd069, 0xad51, 0xd372, 0xafef, 0xafef, 0xafef, 0xafef, 0xd371,
3013 0xd757, 0xd754, 0xd756, 0xb2eb, 0xb2ed, 0xb2ec, 0xd753, 0xb2ee,
3014 0xd755, 0xdb58, 0xdb59, 0xdb5a, 0xdfa6, 0xdfa7, 0xdfa5, 0xdfa8,
3015 0xb8a5, 0xdfa4, 0xbb53, 0xe74a, 0xe746, 0xe749, 0xe74b, 0xe748,
3016 0xe747, 0xeaf5, 0xeaf6, 0xeaf7, 0xbfb4, 0xbfb5, 0xdf1, 0xdf0,
3017 0xdf2, 0xf0a3, 0xf0a2, 0xf2c4, 0xf2c5, 0xf2c3, 0xc4a5, 0xf4b6,
3018 0xf4b7, 0xf746, 0xf7ef, 0xf8bb, 0xa6e1, 0xa87d, 0xc17d, 0xa6e2,
3019 0xd758, 0xdb5b, 0xc641, 0xca4a, 0xca4b, 0xca4d, 0xa6e3, 0xca4e,
3020 0xca4c, 0xcba2, 0xcba3, 0xcb7b, 0xcba1, 0xa8a1, 0xa8a2, 0xcb7c,
3021 0xcb7a, 0xcb79, 0xcb7d, 0xa87e, 0xcb7e, 0xd06a, 0xcdb6, 0xaadc,
3022 0xcdb5, 0xcdb7, 0xaadb, 0xcdbc, 0xaadf, 0xcdb2, 0xcdb0, 0xcdb6,
3023 0xaae6, 0xcdb3, 0xaae3, 0xcdb9, 0xcdbf, 0xcdb1, 0xcdb4, 0xaae2,
3024 0xaadd, 0xcdba, 0xaae4, 0xaae7, 0xaae1, 0xaada, 0xcdb, 0xcdb8,
3025 0xcdb5, 0xaae9, 0xaae5, 0xaae0, 0xcdbd, 0xafef, 0xcdbb, 0xaade,
3026 0xaae8, 0xcdb3, 0xcdb2, 0xcdb4, 0xad62, 0xad5c, 0xad64, 0xad61,
3027 0xd071, 0xd074, 0xad5d, 0xd06b, 0xad56, 0xad60, 0xad63, 0xad65,
3028 0xd0a2, 0xd077, 0xad55, 0xd0a1, 0xad59, 0xad57, 0xad52, 0xd06f,
3029 0xd07e, 0xd073, 0xd076, 0xd0a5, 0xad66, 0xd07d, 0xad5e, 0xd078,
3030 0xd0a4, 0xd075, 0xd079, 0xd07c, 0xd06d, 0xd0a3, 0xd07b, 0xd06c,
3031 0xd070, 0xad5f, 0xad5a, 0xad53, 0xad58, 0xad54, 0xad67, 0xd06e,
3032 0xd3a5, 0xad5b, 0xd07a, 0xce41, 0xd3a8, 0xaffa, 0xd376, 0xd3a3,
3033 0xd37d, 0xd3b2, 0xd3aa, 0xd37e, 0xd3a9, 0xd378, 0xd37c, 0xd3b5,
3034 0xaffd, 0xd3ad, 0xd3a4, 0xafed, 0xd3b3, 0xd374, 0xd3ac, 0xafff,
3035 0xafff, 0xd373, 0xafff, 0xafff, 0xafff, 0xd3ab, 0xafff, 0xafff,
3036 0xd072, 0xdb5c, 0xd3a6, 0xd37a, 0xafff, 0xd37b, 0xd3a1, 0xafff,
3037 0xd375, 0xd3af, 0xd3ae, 0xd3b6, 0xafff, 0xafff, 0xd3b4, 0xd3b0,
3038 0xd3a7, 0xd3a2, 0xafff, 0xafff, 0xd377, 0xafef, 0xd3b1, 0xafef,
3039 0xd379, 0xd75e, 0xd760, 0xd765, 0xd779, 0xb2fc, 0xb2f2, 0xd75d,
3040 0xb2fd, 0xb2fe, 0xd768, 0xd76f, 0xd775, 0xd762, 0xd769, 0xb340,
3041 0xd777, 0xd772, 0xb2fa, 0xb2f8, 0xd76e, 0xd76a, 0xd75c, 0xb2ef,
3042 0xd761, 0xd759, 0xb2f7, 0xb2f9, 0xd766, 0xd763, 0xb2f4, 0xd773,
3043 0xb2f1, 0xd764, 0xd77a, 0xd76c, 0xd76b, 0xb2f0, 0xb2fb, 0xb2f3,
3044 0xd75a, 0xd75f, 0xd770, 0xd776, 0xb341, 0xd75b, 0xd767, 0xd76d,
3045 0xb2f6, 0xd778, 0xd771, 0xd774, 0xb2f5, 0xdb6c, 0xdb60, 0xb5d7,
3046 0xdb7d, 0xdba7, 0xdbaa, 0xb5d5, 0xdb68, 0xdba3, 0xdb69, 0xdb77,
3047 0xb5e2, 0xdb73, 0xb5df, 0xdb74, 0xdb5d, 0xdba4, 0xb5e8, 0xdba1,
3048 0xdb75, 0xdbac, 0xdb70, 0xdfc8, 0xdbaf, 0xb5e6, 0xdb6e, 0xdb7a,
3049 0xb5e9, 0xb5d4, 0xdb72, 0xdbad, 0xdb6b, 0xdb64, 0xdb6f, 0xdb63,
3050 0xdb61, 0xb5d0, 0xdba5, 0xdb6a, 0xdba6, 0xdba9, 0xb5d8, 0xb5dd,
3051 0xb5d9, 0xb5e1, 0xdb7e, 0xb5da, 0xdb76, 0xdb66, 0xb5d2, 0xdb5e,
3052 0xdba2, 0xdbab, 0xdb65, 0xb5e0, 0xdbb0, 0xdb71, 0xdb6d, 0xb5d1,
3053 0xb5e5, 0xdb7c, 0xb5e7, 0xdb78, 0xb5dc, 0xb5de, 0xb5d3, 0xb5d3,
3054 0xb5e4, 0xdb79, 0xdb67, 0xdb7b, 0xdb62, 0xdba6, 0xdbae, 0xdb5f,
3055 0xdfc7, 0xdfdd, 0xb855, 0xdfcc, 0xdfca, 0xdfb5, 0xb8a9, 0xdfc5,
3056 0xdfd9, 0xdfc1, 0xb8b1, 0xdfd8, 0xdfbf, 0xb5e3, 0xdfcf, 0xdfc0,
3057 0xdfd6, 0xb8b0, 0xb8a8, 0xdfaa, 0xdfb2, 0xdfcb, 0xdfc3, 0xdfdc,
3058 0xdfc6, 0xb8b6, 0xdfd7, 0xb8ad, 0xdfc9, 0xdfd1, 0xdfb6, 0xdfd0,
3059 0xdfef, 0xdfb1, 0xdfd2, 0xdfdf, 0xdfab, 0xb5db, 0xdfb9, 0xdfb8,
3060 0xb8af, 0xdfbc, 0xdfbe, 0xdfcd, 0xdfde, 0xb8b2, 0xb8b3, 0xdfb0,
3061 0xb8ab, 0xdfb4, 0xdfda, 0xb8b4, 0xb8ac, 0xb8ae, 0xb8b5, 0xdfef,

3062 0xdfd3, 0xdfce, 0xdfbb, 0xdfba, 0xb8aa, 0xdfac, 0xb8a7, 0xdfc4,
3063 0xdfad, 0xdfc2, 0xdfb7, 0xdfdb, 0xb8a6, 0xdfb3, 0xdfaf, 0xdfd5,
3064 0xdfae, 0xbb60, 0xe3d3, 0xe3c2, 0xe3ac, 0xe3ca, 0xbb58, 0xe3bb,
3065 0xe3c5, 0xbb5b, 0xe3be, 0xbb59, 0xe3af, 0xe3cd, 0xe3ae, 0xe3c1,
3066 0xe3ad, 0xe3bf, 0xe3c8, 0xe3c6, 0xe3ba, 0xe3b5, 0xe3b3, 0xe3b4,
3067 0xe3c7, 0xe3d2, 0xe3bc, 0xbb5a, 0xe3b7, 0xe3cb, 0xbb5d, 0xe3b6,
3068 0xe3b0, 0xe3c0, 0xbb61, 0xbb55, 0xbb5e, 0xe3b8, 0xe3b2, 0xbb57,
3069 0xdfd4, 0xbb56, 0xe3c3, 0xbb54, 0xbb63, 0xbb5c, 0xe3c4, 0xe3b9,
3070 0xe3b1, 0xe3cc, 0xe3bd, 0xbb62, 0xe3d0, 0xbb5f, 0xe3cf, 0xe3c9,
3071 0xe3ce, 0xe3d1, 0xe773, 0xe774, 0xe767, 0xe766, 0xe762, 0xbdb4,
3072 0xbdac, 0xe776, 0xe775, 0xdfa9, 0xe75f, 0xe763, 0xe75d, 0xe770,
3073 0xe761, 0xe777, 0xe75a, 0xe758, 0xe764, 0xe76e, 0xe769, 0xbdb6,
3074 0xe74f, 0xe76d, 0xbdb7, 0xdfbd, 0xe75b, 0xe752, 0xe755, 0xe77b,
3075 0xe75c, 0xe753, 0xe751, 0xe74e, 0xbdb0, 0xe765, 0xbdaf, 0xbdb3,
3076 0xe760, 0xe768, 0xbda9, 0xe778, 0xe77c, 0xbdad, 0xe757, 0xe76b,
3077 0xe76f, 0xe754, 0xe779, 0xbdb2, 0xbdb1, 0xe74c, 0xbdb5, 0xe772,
3078 0xe756, 0xe76a, 0xe750, 0xe75e, 0xe759, 0xbdad, 0xbdae, 0xe76c,
3079 0xe77d, 0xe77a, 0xe771, 0xe74d, 0xbdaa, 0xeb49, 0xeb40, 0xeb43,
3080 0xbfbf, 0xeb45, 0xeaf9, 0xeb41, 0xeb47, 0xbfb8, 0xbfbf, 0xbfb6,
3081 0xeafb, 0xeb4c, 0xeb46, 0xeafc, 0xeb55, 0xeb4f, 0xeaf8, 0xee46,
3082 0xeafe, 0xbfb7, 0xeb4a, 0xeb54, 0xbfbf, 0xeb51, 0xeafd, 0xeb44,
3083 0xeb48, 0xeb42, 0xeb56, 0xeb53, 0xeb50, 0xbfb9, 0xbfbf, 0xbfb6,
3084 0xeafa, 0xeb57, 0xbfbf, 0xeb4d, 0xeb4b, 0xeb4e, 0xee53, 0xee40,
3085 0xee45, 0xee52, 0xee44, 0xedfb, 0xee41, 0xc1a2, 0xedf4, 0xee4d,
3086 0xee4f, 0xedf3, 0xc1a1, 0xee51, 0xee49, 0xc1a8, 0xee50, 0xee42,
3087 0xc1aa, 0xedf9, 0xeb52, 0xee4a, 0xee47, 0xedf5, 0xee55, 0xc1a4,
3088 0xc1a5, 0xedf7, 0xee48, 0xee54, 0xee4b, 0xedfd, 0xc1a7, 0xc1a3,
3089 0xee4c, 0xedfe, 0xee56, 0xedf8, 0xee43, 0xee4e, 0xedfa, 0xedfc,
3090 0xc2cb, 0xedf6, 0xc1a9, 0xc2c4, 0xc17e, 0xc1a6, 0xc2c8, 0xf0b3,
3091 0xf0a9, 0xf0a4, 0xf0aa, 0xf0b4, 0xf0b8, 0xf0b7, 0xc2ca, 0xc2c9,
3092 0xf0ab, 0xf0b9, 0xf0ae, 0xf0a6, 0xf0a8, 0xf0a7, 0xf0ad, 0xf0b2,
3093 0xf0a5, 0xf0ac, 0xf0b1, 0xc2c7, 0xf0af, 0xc2c5, 0xf0b0, 0xc2c3,
3094 0xc2c6, 0xf2d5, 0xf0b5, 0xc3c2, 0xf2cd, 0xf2d1, 0xf2c9, 0xf2cc,
3095 0xf2d4, 0xc3c0, 0xf2d9, 0xf2d2, 0xf2ca, 0xf2da, 0xf2d3, 0xc3c3,
3096 0xc3c4, 0xf2d7, 0xf2cb, 0xc3bf, 0xc3c1, 0xf2c6, 0xf2ce, 0xf2c8,
3097 0xf2d8, 0xf2d6, 0xf2c7, 0xf2cf, 0xf4be, 0xc3c5, 0xf2d0, 0xc4a7,
3098 0xc4a9, 0xc4a6, 0xf4c3, 0xf4bb, 0xf4b9, 0xf4bd, 0xf4ba, 0xf4bf,
3099 0xf4c1, 0xc4aa, 0xc4ac, 0xf4c0, 0xc4ad, 0xc4ab, 0xf4c2, 0xc4a8,
3100 0xc4f4, 0xf5f1, 0xf5f7, 0xc4f6, 0xf4bc, 0xf5f6, 0xf5fd, 0xf5f4,
3101 0xf5fb, 0xf5fa, 0xf4b8, 0xf5f5, 0xf0b6, 0xf5fe, 0xf5f3, 0xf5f8,
3102 0xf5fc, 0xf5f2, 0xf74a, 0xc4f5, 0xf5f9, 0xf7f4, 0xf74b, 0xf749,
3103 0xf747, 0xf748, 0xf74c, 0xc5d9, 0xf7f2, 0xf7f0, 0xf7f5, 0xf7f3,
3104 0xf7f6, 0xc5da, 0xf7f1, 0xf7bc, 0xf945, 0xf946, 0xf947, 0xf9c7,
3105 0xf9bd, 0xca4f, 0xaaea, 0xad68, 0xd3b8, 0xd3b7, 0xb040, 0xb342,
3106 0xd77c, 0xd77b, 0xb5ea, 0xb8b8, 0xb8b7, 0xb8b9, 0xb3d4, 0xe77e,
3107 0xeb58, 0xeb5a, 0xeb59, 0xc1ab, 0xee57, 0xf0ba, 0xf9a5, 0xa6e4,
3108 0xcdc9, 0xcdca, 0xcdc8, 0xcdc7, 0xaaeb, 0xd0a9, 0xd0a7, 0xd0a6,
3109 0xad69, 0xad6b, 0xad6a, 0xd0a8, 0xd3c4, 0xd3c1, 0xd3bf, 0xb041,
3110 0xd3c2, 0xb046, 0xd3bc, 0xd3cb, 0xd3cd, 0xd3bd, 0xb043, 0xd3ce,
3111 0xd3c9, 0xd3bb, 0xd3c0, 0xd3ca, 0xd3c6, 0xd3c3, 0xb048, 0xd3cc,
3112 0xd3be, 0xd3c7, 0xd3b9, 0xb047, 0xb044, 0xd3c5, 0xd3c8, 0xd3ba,
3113 0xb045, 0xb042, 0xb34c, 0xd7a5, 0xb34b, 0xd7a8, 0xd7ab, 0xb348,
3114 0xb346, 0xd77e, 0xd7a9, 0xd7a7, 0xd7a4, 0xd7ac, 0xd7ad, 0xd7af,
3115 0xd7b0, 0xd77d, 0xb345, 0xd7a2, 0xd7a1, 0xd7ae, 0xb347, 0xd7a3,
3116 0xb349, 0xb344, 0xd7a6, 0xb34d, 0xb34a, 0xd7aa, 0xb5f1, 0xdbbf,
3117 0xdbb4, 0xb5ee, 0xdfe7, 0xdbbd, 0xdbb1, 0xb5ec, 0xdbb6, 0xb5ef,
3118 0xdbbba, 0xdbb8, 0xb5f2, 0xb5eb, 0xdbb2, 0xdbb5, 0xb5f0, 0xdbb3,
3119 0xdbbe, 0xdbbc, 0xdbb7, 0xdbb9, 0xdbbb, 0xb5ed, 0xdfe8, 0xdfee,
3120 0xdfe4, 0xdfea, 0xb8ba, 0xdfe6, 0xb8c0, 0xb8bf, 0xb8be, 0xdfed,
3121 0xb8c1, 0xb8c2, 0xdfe3, 0xdf0, 0xb8c3, 0xb8bd, 0xb8bc, 0xdfec,
3122 0xb8c4, 0xdfe2, 0xdfe5, 0xdfef, 0xdfef, 0xe3f4, 0xe3e9, 0xb8bb,
3123 0xbb6a, 0xe3dd, 0xe3f2, 0xe3de, 0xbb65, 0xe3db, 0xe3e4, 0xe3dc,
3124 0xbb67, 0xe3d6, 0xe3f1, 0xbb68, 0xe3ee, 0xe3ef, 0xe3d7, 0xbb6d,
3125 0xe3e6, 0xe3e0, 0xe3e7, 0xe3da, 0xe3f3, 0xe3eb, 0xe3e5, 0xe3d5,
3126 0xbb69, 0xe3ec, 0xbb6c, 0xe3f0, 0xe3ea, 0xbb66, 0xe3e8, 0xe3e2,
3127 0xbb64, 0xe3d9, 0xe3e1, 0xe3ed, 0xe3df, 0xe3e3, 0xbdc1, 0xdfe9,
3128 0xe7b2, 0xe7bb, 0xe7b1, 0xe7ad, 0xe7aa, 0xbdc2, 0xe7a8, 0xbb6b,
3129 0xe7a1, 0xbdc0, 0xe7a7, 0xbdbf, 0xe7ac, 0xe7a9, 0xe7b9, 0xe7b4,
3130 0xe7ae, 0xe7b3, 0xbdbb, 0xe7ab, 0xe7be, 0xe7a2, 0xe7a3, 0xe7ba,
3131 0xbdbc, 0xe7bf, 0xbdbe, 0xe7c0, 0xe7b0, 0xe3d8, 0xe7b6, 0xe7af,
3132 0xe7b8, 0xe7b5, 0xe7a6, 0xbdb9, 0xe7bd, 0xbdba, 0xe7a4, 0xbdbd,
3133 0xeb64, 0xe7b7, 0xe7bc, 0xeb61, 0xbdb8, 0xbfc0, 0xeb6b, 0xeb67,
3134 0xeb65, 0xeb60, 0xeb6f, 0xbfc4, 0xeb5c, 0xeb68, 0xeb69, 0xeb5f,
3135 0xeb5e, 0xeb6c, 0xeb62, 0xeb5d, 0xeb53, 0xeb6e, 0xeb5b, 0xeb6d,
3136 0xeb6a, 0xbfc2, 0xbfc1, 0xbfc3, 0xeb66, 0xf0cb, 0xee59, 0xc1b1,
3137 0xee5d, 0xee5a, 0xee61, 0xee67, 0xee5e, 0xee70, 0xc1ae, 0xee6a,
3138 0xee5f, 0xee6b, 0xee66, 0xee6d, 0xee5e, 0xc1b3, 0xc1b2, 0xee60,
3139 0xee6e, 0xee58, 0xee6c, 0xc1ac, 0xee64, 0xee63, 0xee68, 0xee5b,
3140 0xc1b0, 0xc1b4, 0xee62, 0xee69, 0xc1b5, 0xee65, 0xc1ad, 0xc1af,
3141 0xf0c7, 0xf0c5, 0xf0cc, 0xf0c9, 0xf0cd, 0xf0be, 0xf0c6, 0xf0d1,
3142 0xee6f, 0xf0c2, 0xc2cf, 0xe7a5, 0xf0bd, 0xf0ca, 0xf0c4, 0xf0c1,
3143 0xf0bc, 0xf0bb, 0xf0d0, 0xf0c0, 0xf0bf, 0xc2cd, 0xf0c8, 0xc2cc,
3144 0xc2ce, 0xf0c3, 0xf0cf, 0xf0cf, 0xf2de, 0xf2df, 0xc3c9, 0xf2dc, 0xc3c6,
3145 0xf2e4, 0xc3ca, 0xf2e6, 0xf2db, 0xf0ce, 0xf2e8, 0xf2dd, 0xc3c7,
3146 0xf2e3, 0xf2e5, 0xf2e0, 0xf2e7, 0xf2e2, 0xf2e1, 0xc3c8, 0xf4c5,
3147 0xf4c6, 0xf4c8, 0xf4ae, 0xf4af, 0xf4c9, 0xf4c7, 0xf4c4, 0xf642,
3148 0xf645, 0xf641, 0xc4fa, 0xf643, 0xc4f9, 0xc4f8, 0xc4f7, 0xf644,

3149 0xf751, 0xf74f, 0xf74e, 0xf640, 0xf750, 0xf646, 0xf74d, 0xf7f9,
3150 0xf7d7, 0xf7f7, 0xc5db, 0xf7f8, 0xf7fa, 0xf8bf, 0xc5fa, 0xf8be,
3151 0xf8bd, 0xc5fb, 0xc65a, 0xf96e, 0xf9a7, 0xf9a6, 0xf9a8, 0xa6e5,
3152 0xd0aa, 0xd3cf, 0xd3d0, 0xdbc0, 0xf647, 0xf8c0, 0xa6e6, 0xad6c,
3153 0xd0ab, 0xd7b1, 0xb34e, 0xdbc2, 0xdbc1, 0xb5f3, 0xb8c5, 0xe7c1,
3154 0xbdc3, 0xbdc4, 0xbfc5, 0xc5fc, 0xa6e7, 0xd0ac, 0xaaed, 0xd0ae,
3155 0xd0ad, 0xad6d, 0xd3d1, 0xd3d8, 0xb049, 0xd3d6, 0xd3d4, 0xd3db,
3156 0xd3d2, 0xd3d3, 0xb04a, 0xb04e, 0xd3dc, 0xb04d, 0xd3da, 0xd3d7,
3157 0xd3d5, 0xb04b, 0xb04c, 0xd3d9, 0xb350, 0xd7b2, 0xb355, 0xd7c2,
3158 0xb354, 0xd7c4, 0xd7b8, 0xb352, 0xd7c3, 0xd7b3, 0xb353, 0xd7bf,
3159 0xd7bb, 0xd7bd, 0xd7b7, 0xd7be, 0xb34f, 0xd7ba, 0xd7b9, 0xd7b5,
3160 0xd7c0, 0xd7bc, 0xd7b4, 0xd7b6, 0xb351, 0xd7c1, 0xb5f6, 0xdbcd,
3161 0xdbc9, 0xdbcb, 0xdbc6, 0xdbc5, 0xdbc3, 0xdbc4, 0xdbcc, 0xdbc8,
3162 0xdbc7, 0xb5f4, 0xb5f5, 0xdbcf, 0xb8cd, 0xdf2, 0xdf8, 0xdf3,
3163 0xdf4, 0xdf9, 0xb8cf, 0xb8c7, 0xb8ce, 0xdf1, 0xdbc4, 0xb8ca,
3164 0xb8c8, 0xdf7, 0xdf6, 0xb8c9, 0xb8cb, 0xdf5, 0xb8c6, 0xb8cc,
3165 0xe3f6, 0xbb74, 0xe442, 0xe441, 0xe3fb, 0xbb76, 0xe440, 0xe3f7,
3166 0xe3f8, 0xbb6e, 0xbb70, 0xe3fd, 0xe3f5, 0xbb72, 0xbb71, 0xe3f9,
3167 0xe3fe, 0xe3fc, 0xbb73, 0xe3fa, 0xdbcce, 0xbb6f, 0xe7c2, 0xe7c9,
3168 0xbdc6, 0xe7cd, 0xbdc4, 0xe7c5, 0xe7c3, 0xe7cc, 0xbdc5, 0xe7cb,
3169 0xbdc7, 0xbdc8, 0xe7c4, 0xbdc9, 0xe7ca, 0xe7c6, 0xe7c7, 0xe7c8,
3170 0xbb75, 0xeb70, 0xeb7c, 0xbfca, 0xeb77, 0xeb79, 0xbfc8, 0xeb71,
3171 0xeb75, 0xeb78, 0xbfc6, 0xbfc9, 0xeb7b, 0xeb73, 0xeb74, 0xeb7a,
3172 0xeb72, 0xeb76, 0xbfc7, 0xee72, 0xee71, 0xc1b7, 0xee77, 0xc1b9,
3173 0xc1b6, 0xee73, 0xc1ba, 0xee74, 0xee75, 0xee78, 0xc1b8, 0xf0d6,
3174 0xf0d9, 0xf0d3, 0xf0d5, 0xf0d4, 0xf0d7, 0xf0d8, 0xee76, 0xf0d2,
3175 0xc3cd, 0xc3ce, 0xc3cb, 0xf2ed, 0xf2e9, 0xf4ca, 0xc4b0, 0xf4cb,
3176 0xf649, 0xc4fb, 0xf64b, 0xc4fc, 0xf648, 0xf64a, 0xc5a8, 0xf752,
3178 0xc5a7, 0xf7fd, 0xf7fc, 0xf7fb, 0xf948, 0xf949, 0xf94b, 0xf94a,
3179 0xca50, 0xa6e8, 0xad6e, 0xd7c5, 0xb5f7, 0xdf2, 0xc2d0, 0xf2f2,
3180 0xa8a3, 0xb357, 0xb356, 0xdbd0, 0xb5f8, 0xdbd2, 0xdbd1, 0xdf2,
3181 0xb8d0, 0xe443, 0xe446, 0xe445, 0xe444, 0xe7ce, 0xe7d0, 0xe7cf,
3182 0xbfcc, 0xbfc3, 0xc1bb, 0xee79, 0xee7b, 0xee7a, 0xc2d1, 0xf2f4,
3183 0xf2f3, 0xf4cc, 0xc4b1, 0xc4fd, 0xf754, 0xf753, 0xc65b, 0xa8a4,
3184 0xd0af, 0xad6f, 0xd7c8, 0xd7c6, 0xd7c7, 0xdbd4, 0xdbd5, 0xe043,
3185 0xdbd3, 0xdf2, 0xe041, 0xe040, 0xe042, 0xb8d1, 0xdf2, 0xdf2,
3186 0xe044, 0xe449, 0xe447, 0xe448, 0xe7d3, 0xe7d1, 0xe7d2, 0xeb7d,
3187 0xee7c, 0xee7d, 0xc2d2, 0xf2f5, 0xf4cd, 0xc4b2, 0xf64c, 0xf755,
3188 0xc5a9, 0xf7fe, 0xf94c, 0xa8a5, 0xad71, 0xad72, 0xd0b0, 0xd0b1,
3189 0xad70, 0xb054, 0xb052, 0xb051, 0xb058, 0xb050, 0xb059, 0xd3dd,
3190 0xb056, 0xb053, 0xb057, 0xb055, 0xb04f, 0xb35f, 0xb359, 0xd7cc,
3191 0xb35e, 0xb360, 0xb35a, 0xb35b, 0xd7ca, 0xb358, 0xd7cb, 0xb35d,
3192 0xd7c9, 0xb35c, 0xb644, 0xb646, 0xdbd8, 0xb645, 0xb5f9, 0xb5fd,
3193 0xb8e4, 0xe049, 0xdbda, 0xb5fe, 0xdbdd, 0xdbde, 0xb643, 0xdbbe,
3194 0xdb2, 0xdb3, 0xdbd7, 0xdbd6, 0xdb2, 0xb642, 0xdb2, 0xdbdf,
3195 0xb640, 0xb5fb, 0xb647, 0xdbdb, 0xdbdc, 0xdbd9, 0xb641, 0xb5fc,
3196 0xb5fa, 0xe048, 0xb8df, 0xb8da, 0xb8d5, 0xb8e5, 0xb8d6, 0xb8d2,
3197 0xb8e1, 0xb8de, 0xb8e0, 0xb8d7, 0xb8dc, 0xb8d3, 0xb8d4, 0xe050,
3198 0xe04d, 0xe045, 0xe04a, 0xb8e2, 0xe051, 0xb8e3, 0xb8d9, 0xe047,
3199 0xe04f, 0xe04b, 0xe04e, 0xe04c, 0xb8dd, 0xe046, 0xb8d8, 0xe04c,
3200 0xbb78, 0xbb7b, 0xe44e, 0xbba5, 0xe44d, 0xbb7d, 0xbdcf, 0xe44f,
3201 0xbba4, 0xe44b, 0xbba6, 0xbb79, 0xb8db, 0xbb7c, 0xbb7a, 0xbb7e,
3202 0xbba2, 0xbb77, 0xbba7, 0xbba3, 0xbba1, 0xe44a, 0xbdd6, 0xbdd2,
3203 0xbdd9, 0xe7d6, 0xbdda, 0xe7e2, 0xe7db, 0xbdc3, 0xe7e3, 0xe7dd,
3204 0xbdd5, 0xe7de, 0xbdd4, 0xe7e1, 0xbdcce, 0xe7df, 0xe7d5, 0xbdc3,
3205 0xebaa, 0xbdd3, 0xbdd0, 0xbdd8, 0xe7d4, 0xe7d8, 0xbdcc, 0xe7d7,
3206 0xe7d9, 0xe7da, 0xbdd7, 0xe7dc, 0xe7e0, 0xe7e4, 0xbddb, 0xbfd2,
3207 0xeba5, 0xebab, 0xeba8, 0xeb7e, 0xebac, 0xeba1, 0xeba7, 0xbfd3,
3208 0xbfd3, 0xebad, 0xbfcf, 0xbfd9, 0xbfd4, 0xebaf, 0xeba9, 0xbfd0,
3209 0xeba2, 0xbfd4, 0xeba3, 0xeba4, 0xbfdb, 0xbfd8, 0xbdd1, 0xbfce,
3210 0xebb0, 0xbfd5, 0xebae, 0xbfd1, 0xbfd6, 0xbfd7, 0xc1c3,
3211 0xeea4, 0xeead, 0xeeaa, 0xc1c0, 0xeea5, 0xeeab, 0xc1bc,
3212 0xeea7, 0xc1c4, 0xeea3, 0xeea8, 0xeeaf, 0xeba6, 0xeea9, 0xeea2,
3213 0xc1bd, 0xeea1, 0xc1be, 0xeeb0, 0xc1bf, 0xeeae, 0xc1c2, 0xee7e,
3214 0xc1c1, 0xeea6, 0xf0dc, 0xf0ea, 0xf0e5, 0xf0e7, 0xf0db, 0xc2d3,
3215 0xf0da, 0xc2d6, 0xc2d5, 0xf0e9, 0xf0e1, 0xf0de, 0xf0e4, 0xf0dd,
3216 0xf0df, 0xf0e8, 0xf0e6, 0xc2d4, 0xf0ed, 0xf0eb, 0xf0e2, 0xf0ec,
3217 0xf0e3, 0xf2f9, 0xc3cf, 0xf341, 0xf64f, 0xc3d6, 0xf0e0, 0xf2f7,
3218 0xc3d2, 0xf2f8, 0xf2fd, 0xc3d4, 0xc3d5, 0xf2f6, 0xf340, 0xf342,
3219 0xf2fa, 0xf2fe, 0xf2fb, 0xf343, 0xc3d1, 0xc3d7, 0xc3d3,
3220 0xc3d0, 0xf4d0, 0xc4b7, 0xf4ce, 0xf4d2, 0xf4d3, 0xc4b5, 0xf4d4,
3221 0xf4d1, 0xf4cf, 0xc4b8, 0xc4b4, 0xf4d5, 0xc4b6, 0xc4b3, 0xc4fe,
3222 0xc540, 0xf64e, 0xf64d, 0xf650, 0xf651, 0xc541, 0xf756, 0xf75b,
3223 0xc5aa, 0xf758, 0xf757, 0xf75a, 0xf759, 0xf843, 0xc5dc, 0xf842,
3224 0xf840, 0xf841, 0xc5fe, 0xc5fd, 0xf8c1, 0xf8c2, 0xc640, 0xf94d,
3225 0xf94e, 0xc667, 0xc66d, 0xf9a9, 0xf9c8, 0xa8a6, 0xd7cd, 0xd7ce,
3226 0xe052, 0xe450, 0xe7e5, 0xc1c6, 0xc1c5, 0xf0ee, 0xf344, 0xf844,
3227 0xa8a7, 0xd3de, 0xb05a, 0xb361, 0xe054, 0xe053, 0xbddc, 0xe7e6,
3228 0xbddd, 0xeeb1, 0xc2d7, 0xc676, 0xa8a8, 0xc3cb, 0xd3df, 0xb362,
3229 0xd7cf, 0xd7d0, 0xdb2, 0xb648, 0xb8e6, 0xe056, 0xe055, 0xe057,
3230 0xe451, 0xe452, 0xbba8, 0xbfd, 0xbdd, 0xbfd, 0xeeb5, 0xeeb2,
3231 0xeeb4, 0xeeb3, 0xc1c7, 0xf0ef, 0xf346, 0xf345, 0xcba4, 0xb05c,
3232 0xb05b, 0xd3e0, 0xd7d1, 0xdb2, 0xdb2, 0xb649, 0xe059, 0xe05a,
3233 0xe058, 0xb8e8, 0xb8e7, 0xbbaa, 0xbba9, 0xe7e7, 0xebb3, 0xebb1,
3234 0xebb2, 0xbfd, 0xeeb7, 0xeeb6, 0xf0f2, 0xf0f1, 0xf0f0, 0xf347,
3235 0xf9aa, 0xa8a9, 0xad73, 0xad74, 0xb05d, 0xb05e, 0xd3e2, 0xd3e1,

3236 0xd7d2, 0xb368, 0xb366, 0xb363, 0xb367, 0xb365, 0xb364, 0xb64a,
3237 0xdbea, 0xb8ed, 0xb64c, 0xb651, 0xdbec, 0xb653, 0xb652, 0xb655,
3238 0xbbeb, 0xb8eb, 0xb64f, 0xb64b, 0xb64d, 0xb64e, 0xb654, 0xb650,
3239 0xb64e, 0xb8ef, 0xb8ee, 0xb8ec, 0xb8f0, 0xb8ea, 0xb8eb, 0xb8e9,
3240 0xe05b, 0xe454, 0xbbac, 0xbbad, 0xbbab, 0xe453, 0xe455, 0xe7ea,
3241 0xe7ec, 0xbde7, 0xe7ed, 0xbde0, 0xe7e9, 0xbddf, 0xbde9, 0xbde5,
3242 0xbde6, 0xbde2, 0xe7e8, 0xbde1, 0xe7ee, 0xe7eb, 0xbde8, 0xbde3,
3243 0xbde4, 0xebb5, 0xebb7, 0xebb6, 0xebb8, 0xbfe0, 0xebb4, 0xc1cb,
3244 0xeeb8, 0xc1c8, 0xc1cc, 0xc1ca, 0xc1c9, 0xf0f3, 0xf0f6, 0xf0f5,
3245 0xf0f4, 0xc2d8, 0xf348, 0xf349, 0xc3d8, 0xf34a, 0xc3d9, 0xc4ba,
3246 0xc4b9, 0xf652, 0xc542, 0xf653, 0xf75c, 0xc5ab, 0xc5ac, 0xf845,
3247 0xc642, 0xa8aa, 0xb36a, 0xb369, 0xe05c, 0xe05d, 0xbbae, 0xebb9,
3248 0xbdea, 0xebba, 0xeeb9, 0xa8ab, 0xd0b2, 0xad76, 0xad75, 0xd3e3,
3249 0xb05f, 0xd3e4, 0xd7d5, 0xd7d4, 0xd7d3, 0xdbee, 0xb658, 0xdbed,
3250 0xb657, 0xdbef, 0xb656, 0xe05f, 0xe062, 0xe060, 0xe061, 0xe065,
3251 0xe05e, 0xe066, 0xe063, 0xe064, 0xbbb0, 0xe456, 0xbbaaf, 0xe7f2,
3252 0xe7f0, 0xbdeb, 0xe7ef, 0xe7f1, 0xbdec, 0xebbb, 0xebbc, 0xc1cd,
3253 0xf34c, 0xf34e, 0xf34b, 0xf34d, 0xf4d6, 0xf654, 0xf96f, 0xa8ac,
3254 0xad77, 0xd3e5, 0xd3e7, 0xd3e6, 0xd7d8, 0xb36c, 0xd7d6, 0xb36b,
3255 0xd7d9, 0xd7da, 0xd7d7, 0xdbfb, 0xb660, 0xdbf3, 0xdbf9, 0xb65b,
3256 0xb65e, 0xdbf2, 0xb659, 0xdbf6, 0xe06c, 0xb65d, 0xdbf1, 0xdbf7,
3257 0xdbf4, 0xdbfa, 0xdbf0, 0xdbf8, 0xb65c, 0xb65f, 0xdbf5, 0xb65a,
3258 0xb8f2, 0xe068, 0xb8f1, 0xe06f, 0xe06e, 0xb8f8, 0xb8f9, 0xe070,
3259 0xb8f3, 0xe06d, 0xb8f7, 0xe072, 0xe069, 0xe06b, 0xb8f4, 0xe067,
3260 0xe06a, 0xe071, 0xb8f5, 0xe073, 0xb8f6, 0xbbb1, 0xe45b, 0xe461,
3261 0xe459, 0xe462, 0xe458, 0xe45d, 0xe463, 0xe460, 0xe45f, 0xe45e,
3262 0xe457, 0xe45c, 0xe45a, 0xbdf1, 0xbdee, 0xe7fb, 0xe841, 0xe843,
3263 0xe840, 0xe7f8, 0xe7fa, 0xe845, 0xe842, 0xe7fc, 0xe846, 0xe7f9,
3264 0xe844, 0xbdef, 0xbdf5, 0xbdf3, 0xe7f3, 0xbdf4, 0xbdf0, 0xe7f4,
3265 0xe7f6, 0xe7f5, 0xe7fd, 0xe7fe, 0xbdf2, 0xbded, 0xe7f7, 0xebc6,
3266 0xbfe2, 0xebbd, 0xbfe3, 0xbfe6, 0xebc2, 0xebbf, 0xbfe5, 0xebc3,
3267 0xebc4, 0xebbe, 0xebc7, 0xebc0, 0xebc5, 0xbfe4, 0xbfe1, 0xebc1,
3268 0xeebf, 0xc1d0, 0xc1ce, 0xc1cf, 0xeebe, 0xeebb, 0xeeba,
3269 0xeebd, 0xeebc, 0xf145, 0xc2de, 0xf0fb, 0xf0fa, 0xc2d9, 0xf141,
3270 0xf140, 0xf0f7, 0xf143, 0xf0fc, 0xc2dd, 0xf0f9, 0xf142, 0xf0f8,
3271 0xc2da, 0xc2dc, 0xf0fd, 0xc2db, 0xf0fe, 0xf144, 0xf352, 0xc3de,
3272 0xf34f, 0xf353, 0xc3db, 0xf351, 0xc3e0, 0xc3dd, 0xf350, 0xc3df,
3273 0xf354, 0xc3da, 0xc4bc, 0xc4be, 0xf4d9, 0xc4bd, 0xf4d7, 0xc3dc,
3274 0xf4d8, 0xc4bb, 0xc543, 0xc545, 0xf656, 0xc544, 0xf655, 0xf761,
3275 0xc5ad, 0xf760, 0xc5ae, 0xf75e, 0xf75d, 0xf762, 0xf763, 0xf846,
3276 0xf75f, 0xf8c6, 0xf8c3, 0xf8c4, 0xf8c5, 0xc65c, 0xf951, 0xf950,
3277 0xf94f, 0xf970, 0xf9be, 0xf9ab, 0xc66e, 0xa8ad, 0xb060, 0xb8fa,
3278 0xbdf6, 0xebc8, 0xc2df, 0xf355, 0xf9ac, 0xa8ae, 0xaaee, 0xad79,
3279 0xad78, 0xb063, 0xd3e8, 0xb061, 0xd3e9, 0xb062, 0xd7df, 0xd7db,
3280 0xb36d, 0xd7de, 0xd7dd, 0xd7dc, 0xb36e, 0xd7e0, 0xd7e1, 0xdc43,
3281 0xdc41, 0xdc45, 0xdc46, 0xdc4c, 0xdc48, 0xdc4a, 0xdc42, 0xdbfc,
3282 0xdc49, 0xdc4b, 0xdc44, 0xdc47, 0xdbfd, 0xb662, 0xdc40, 0xdbfe,
3283 0xb661, 0xb663, 0xb8fd, 0xe075, 0xe077, 0xe076, 0xe07b, 0xb8fb,
3284 0xe078, 0xe074, 0xe079, 0xe07a, 0xb8fc, 0xb8fe, 0xe07c, 0xe467,
3285 0xe466, 0xe464, 0xe465, 0xbbb3, 0xbbb5, 0xbbb2, 0xbbb4, 0xe84d,
3286 0xe84e, 0xe849, 0xe84a, 0xbdf8, 0xbdfd, 0xbdf7, 0xbdfc, 0xbdf9,
3287 0xe84b, 0xe84c, 0xe848, 0xbe40, 0xbdfb, 0xbdfa, 0xbdfc, 0xe847,
3288 0xebca, 0xbfe8, 0xebcc, 0xbfea, 0xebcf, 0xebcb, 0xebc9, 0xebce,
3289 0xbfe9, 0xebcd, 0xbfe7, 0xc1d3, 0xc1d6, 0xeecl, 0xc1d4, 0xeecl,
3290 0xc1d2, 0xc1d5, 0xf146, 0xf147, 0xf148, 0xc2e0, 0xf149, 0xc2e1,
3291 0xc3e2, 0xf358, 0xf359, 0xf357, 0xf356, 0xf35a, 0xc3e1, 0xf4dd,
3292 0xf4db, 0xf4dc, 0xf4de, 0xf4df, 0xf4da, 0xf4df, 0xf658, 0xf659, 0xf657,
3293 0xc546, 0xf764, 0xc5af, 0xf765, 0xf848, 0xf847, 0xa8af, 0xb664,
3294 0xb940, 0xbbb6, 0xbfec, 0xbfeb, 0xc3e3, 0xc47c, 0xc547, 0xa8b0,
3295 0xb064, 0xb941, 0xf35b, 0xcba6, 0xa8b1, 0xa8b4, 0xa8b3, 0xa8b2,
3296 0xcba5, 0xcddc, 0xcddc, 0xaaef, 0xaaef, 0xcddc, 0xcddc, 0xaaef,
3297 0xcdd1, 0xcdd0, 0xcdd2, 0xd0b6, 0xd0b4, 0xad7c, 0xd0b3, 0xada3,
3298 0xad7e, 0xad7b, 0xada4, 0xad7d, 0xada2, 0xada1, 0xd0b5, 0xad7a,
3299 0xb06a, 0xd3eb, 0xd3f1, 0xb067, 0xb06e, 0xb069, 0xd3ee, 0xd3f0,
3300 0xb06c, 0xd3ea, 0xd3ed, 0xb068, 0xb065, 0xd3ec, 0xb06b, 0xd3ef,
3301 0xb06d, 0xb066, 0xd7e3, 0xd7e6, 0xb370, 0xb37a, 0xb376, 0xd7e4,
3302 0xb37e, 0xb377, 0xb37c, 0xb372, 0xb36f, 0xb371, 0xb37d, 0xd7e5,
3303 0xb375, 0xb378, 0xb374, 0xb379, 0xd7e7, 0xb37b, 0xb373, 0xd7e2,
3304 0xdc4d, 0xb665, 0xdc4f, 0xb667, 0xb669, 0xdc4e, 0xb666, 0xb66a,
3305 0xb668, 0xb947, 0xe0a3, 0xb94f, 0xe07e, 0xb950, 0xb945, 0xe0a1,
3306 0xb94a, 0xe0a2, 0xb943, 0xb942, 0xb94d, 0xb94c, 0xb94b, 0xb949,
3307 0xb94e, 0xe07d, 0xb944, 0xb946, 0xb948, 0xb948, 0xb948, 0xb949,
3308 0xb949, 0xb949, 0xb949, 0xb949, 0xb949, 0xb949, 0xb949, 0xb949,
3309 0xb949, 0xb949, 0xb949, 0xb949, 0xb949, 0xb949, 0xb949, 0xb949,
3310 0xb949, 0xb949, 0xb949, 0xb949, 0xb949, 0xb949, 0xb949, 0xb949,
3311 0xb949, 0xb949, 0xb949, 0xb949, 0xb949, 0xb949, 0xb949, 0xb949,
3312 0xc1d9, 0xeecl, 0xebd3, 0xc2e2, 0xc2e4, 0xc3e4, 0xc3e5, 0xf4e0,
3313 0xc5de, 0xc5dd, 0xa8b6, 0xca55, 0xb06f, 0xca52, 0xca53, 0xca51,
3314 0xca54, 0xcbaa, 0xcba7, 0xcba5, 0xcba8, 0xa8b7, 0xa8ba, 0xcba9,
3315 0xa8b9, 0xcbaa, 0xa8b8, 0xcdd5, 0xcdd7, 0xaaaf, 0xcdd3, 0xcdd6,
3316 0xcdd4, 0xaaaf, 0xaaaf, 0xaaaf, 0xd0b8, 0xd0bc, 0xd0b9, 0xada7,
3317 0xada8, 0xd0bb, 0xd0bd, 0xd0bf, 0xada5, 0xd0be, 0xada6, 0xd7ee,
3318 0xd0ba, 0xd3f2, 0xd3fb, 0xd3f9, 0xd3fd, 0xd3fe, 0xd3fa, 0xd3fc,
3319 0xb071, 0xd3f7, 0xd3f3, 0xb070, 0xb072, 0xd3f6, 0xd3fd, 0xd3f8,
3320 0xb3a1, 0xd7f1, 0xd7e9, 0xd7ef, 0xd7f0, 0xb3a2, 0xd7e8, 0xd7ea,
3321 0xd0b7, 0xd7ec, 0xd7ed, 0xd7eb, 0xb66c, 0xdc56, 0xebd4, 0xdc57,
3322 0xdc54, 0xb3a3, 0xb66e, 0xdc53, 0xdc59, 0xdc58, 0xb66b, 0xdc5c,

3323 0xdc52, 0xdc5b, 0xdc50, 0xdc5a, 0xdc55, 0xb66d, 0xe0aa, 0xe0a5,
3324 0xe0ab, 0xe0a6, 0xe0a4, 0xe0a7, 0xb951, 0xe0a9, 0xe0a8, 0xb952,
3325 0xbbc1, 0xbbc0, 0xbbc0, 0xe46e, 0xe471, 0xe469, 0xe46d, 0xbbc2, 0xe46c,
3326 0xe46a, 0xe470, 0xe46b, 0xe468, 0xe46f, 0xe859, 0xbe48, 0xf14a,
3327 0xe856, 0xe857, 0xe855, 0xdc51, 0xbe47, 0xe85a, 0xe854, 0xbe46,
3328 0xbe49, 0xe858, 0xe85d, 0xbff3, 0xebd5, 0xbff3, 0xebd6, 0xebd7, 0xeec4, 0xc1dd,
3329 0xf14b, 0xf14c, 0xf14d, 0xf35d, 0xf35c, 0xf4e2, 0xf4e1, 0xf65b,
3330 0xf65c, 0xf65a, 0xf766, 0xc5b0, 0xa8bb, 0xadaa, 0xada9, 0xb075,
3331 0xb074, 0xd440, 0xd441, 0xd3fe, 0xb073, 0xd7f5, 0xd7f6, 0xd7f2,
3332 0xb3a4, 0xd7f3, 0xd7f4, 0xdc5f, 0xdc61, 0xdc5d, 0xdc60, 0xb66f,
3333 0xdc5e, 0xb670, 0xdd73, 0xb955, 0xb954, 0xb953, 0xe0ac, 0xe0ad,
3334 0xe473, 0xe475, 0xbbc6, 0xbbc3, 0xbbc5, 0xbbc4, 0xe474, 0xe472,
3335 0xe861, 0xe85e, 0xe85f, 0xbe4d, 0xe860, 0xe85b, 0xe85c, 0xbe4a,
3336 0xbe4b, 0xe85d, 0xbe4c, 0xebdb, 0xebdc, 0xebd9, 0xebda, 0xbff4,
3337 0xebd8, 0xeec8, 0xeec5, 0xeec7, 0xc1e0, 0xeecb, 0xc1df, 0xeec9,
3338 0xeec6, 0xeeca, 0xeec6, 0xc1de, 0xf14f, 0xf150, 0xf14e, 0xf152,
3339 0xc2e5, 0xc2e6, 0xf35f, 0xc3e7, 0xf151, 0xf35e, 0xc3e6, 0xf4e5,
3340 0xf4e6, 0xc4bf, 0xc4bf, 0xf4e4, 0xf4e3, 0xf65d, 0xc548, 0xf849, 0xf8c8,
3341 0xf8c7, 0xc643, 0xc65d, 0xf8c9, 0xf971, 0xc66f, 0xa8bc, 0xaaf6,
3342 0xb956, 0xc4c0, 0xa8bd, 0xadab, 0xb3a5, 0xb671, 0xc2e7, 0xaaf7,
3343 0xd0c1, 0xd0c0, 0xd0c0, 0xd442, 0xb078, 0xb076, 0xb07a, 0xd444, 0xb079,
3344 0xb077, 0xd443, 0xb3a8, 0xd7fc, 0xb3a7, 0xb3a9, 0xd842, 0xb3ab,
3345 0xd7fe, 0xd840, 0xd7f7, 0xb3aa, 0xd843, 0xd7f9, 0xd7fa, 0xd7f8,
3346 0xb3a6, 0xd841, 0xd7fb, 0xd7fd, 0xdc6d, 0xdc6c, 0xdc6a, 0xdc62,
3347 0xdc71, 0xdc65, 0xdc6f, 0xdc76, 0xdc6e, 0xb679, 0xb675, 0xdc63,
3348 0xdc69, 0xb677, 0xdc68, 0xb678, 0xb67a, 0xdc6b, 0xb672, 0xb673,
3349 0xdc77, 0xdc75, 0xdc74, 0xdc66, 0xdc72, 0xb676, 0xb674, 0xdc73,
3350 0xdc64, 0xdc67, 0xdc70, 0xe4ba, 0xe0b7, 0xe0b0, 0xe0c3, 0xe0cc,
3351 0xe0b3, 0xb961, 0xe0c0, 0xb957, 0xb959, 0xb965, 0xe0b1, 0xb95a,
3352 0xb95c, 0xb966, 0xb95b, 0xb964, 0xe0b9, 0xe0ae, 0xb962, 0xe0b8,
3353 0xb95e, 0xe0ca, 0xb963, 0xe0c8, 0xe0bc, 0xe0c6, 0xb960, 0xe0af,
3354 0xe0c9, 0xe0c4, 0xe0cb, 0xb958, 0xb967, 0xb95d, 0xe0b5, 0xe0bd,
3355 0xe0c1, 0xe0c5, 0xe0c2, 0xb95f, 0xe0b4, 0xe0b2, 0xe0be, 0xe0bb, 0xe0ba,
3356 0xe0bf, 0xe0c2, 0xe0c7, 0xe478, 0xbbc7, 0xe4a4, 0xe47a, 0xbbcc,
3357 0xbbd0, 0xe4ad, 0xe4b5, 0xe4a6, 0xbbc8, 0xe4aa, 0xe0b6, 0xbbc9,
3358 0xe4b1, 0xe4b6, 0xe4ae, 0xe4b0, 0xe4b9, 0xe4b2, 0xe47e, 0xe4a9,
3359 0xbbd1, 0xbbcd, 0xe47c, 0xe4ab, 0xbbcb, 0xe4a5, 0xbbca, 0xe4b3,
3360 0xe4a2, 0xe479, 0xbbce, 0xe4b8, 0xe47b, 0xe4af, 0xe4ac, 0xe4a7,
3361 0xe477, 0xe476, 0xe4a1, 0xe4b4, 0xbbcf, 0xe4b7, 0xe47d, 0xe4a3,
3362 0xbe52, 0xbe5a, 0xbe55, 0xe8a4, 0xe8a1, 0xe867, 0xbe50, 0xbe4f,
3363 0xbe56, 0xe865, 0xbe54, 0xe871, 0xe863, 0xe864, 0xbe4e, 0xe8a3,
3364 0xbe58, 0xe874, 0xe879, 0xe873, 0xebee, 0xe86f, 0xe877, 0xe875,
3365 0xe868, 0xe862, 0xe87d, 0xbe57, 0xe87e, 0xe878, 0xe86d, 0xe86b,
3366 0xe866, 0xe86e, 0xe87b, 0xe86a, 0xe87a, 0xe8a2, 0xbe53, 0xe876,
3367 0xe87c, 0xe872, 0xe86c, 0xbe51, 0xe4a8, 0xe870, 0xbe59, 0xe869,
3368 0xebf4, 0xbff7, 0xebf3, 0xebf0, 0xec44, 0xbffb, 0xec41, 0xebf8,
3369 0xec43, 0xebe9, 0xebf6, 0xbffd, 0xebe1, 0xebdf, 0xec42, 0xec40,
3370 0xebfe, 0xebed, 0xebec, 0xebe2, 0xc040, 0xebe8, 0xebf2, 0xebfd,
3371 0xc043, 0xec45, 0xc1e8, 0xc045, 0xbffe, 0xebe6, 0xebef, 0xebde,
3372 0xebe0, 0xbff5, 0xc042, 0xbffa, 0xebe7, 0xebf7, 0xebf1, 0xc041,
3373 0xebdd, 0xc1e3, 0xebf9, 0xebfc, 0xebfb, 0xebeb, 0xc044, 0xbff9,
3374 0xbff8, 0xebf5, 0xebfb, 0xbff6, 0xebe4, 0xebfa, 0xebe5, 0xebea,
3375 0xeed2, 0xeed7, 0xc1e5, 0xc1e7, 0xeedd, 0xc1e1, 0xeec6, 0xeec3,
3376 0xeed8, 0xeed9, 0xeec2, 0xc1ee, 0xeec1, 0xeed1, 0xeec0, 0xeed4,
3377 0xeec3, 0xc1ed, 0xc1eb, 0xeed5, 0xeec8, 0xeeda, 0xeec7, 0xeec9,
3378 0xeed0, 0xc1e6, 0xeec4, 0xeed6, 0xc1ea, 0xeedb, 0xc1ec, 0xeec4,
3379 0xc1e4, 0xeed6, 0xeec5, 0xeedf, 0xebe3, 0xeec6, 0xeed3, 0xc1e9,
3380 0xeecb, 0xc1e2, 0xeec6, 0xf160, 0xf159, 0xc2e9, 0xf154, 0xf163,
3381 0xf15b, 0xeedc, 0xf165, 0xf155, 0xc2e8, 0xf15f, 0xc2ea, 0xc2f2,
3382 0xc2f0, 0xf161, 0xc2f1, 0xf157, 0xf158, 0xf15d, 0xf162, 0xeecd,
3383 0xc2eb, 0xf16a, 0xf167, 0xf16b, 0xf15e, 0xf15a, 0xf168, 0xf36a,
3384 0xf15c, 0xc2ee, 0xc2ed, 0xeecf, 0xc2ef, 0xf164, 0xf166, 0xc2ec,
3385 0xf169, 0xf153, 0xf156, 0xf373, 0xf363, 0xc3eb, 0xf371, 0xf361,
3386 0xc3ec, 0xf36c, 0xf368, 0xc3f1, 0xf372, 0xf362, 0xf365, 0xc3e9,
3387 0xf374, 0xf36d, 0xf370, 0xc3ef, 0xc3f4, 0xc3f2, 0xf369, 0xf364,
3388 0xc3ed, 0xc3ee, 0xf360, 0xc3ea, 0xc3e8, 0xc3f0, 0xf36f, 0xc3f3,
3389 0xf36b, 0xf375, 0xc3f5, 0xf367, 0xf36e, 0xf4f3, 0xf542, 0xf4f5,
3390 0xf4fc, 0xf366, 0xf4fa, 0xf4e9, 0xf540, 0xc4c3, 0xf4ed, 0xf4fe,
3391 0xf4f4, 0xc4c2, 0xf544, 0xf4f6, 0xf4fb, 0xf4fd, 0xf4e7, 0xf541,
3392 0xf4f2, 0xf4f7, 0xf4eb, 0xf4ef, 0xf543, 0xf4f9, 0xf4e8, 0xf4ec,
3393 0xf4ee, 0xf4f8, 0xc4c1, 0xf4f1, 0xf4ea, 0xf4f0, 0xf661, 0xf666,
3394 0xc54f, 0xf668, 0xc549, 0xf664, 0xf66a, 0xc54e, 0xc54a, 0xc54b,
3395 0xf660, 0xf667, 0xc54d, 0xf665, 0xc54c, 0xf65f, 0xf663, 0xf662,
3396 0xf65e, 0xf669, 0xc5b1, 0xf76d, 0xf770, 0xf76c, 0xf76e, 0xf76f,
3397 0xf769, 0xf76a, 0xf767, 0xf76b, 0xf768, 0xc5b2, 0xc5b3, 0xf84b,
3398 0xf84d, 0xf84c, 0xf84e, 0xc5e0, 0xf84a, 0xc5df, 0xc5e1, 0xf8bc,
3399 0xf8cc, 0xc644, 0xf8ca, 0xf953, 0xf952, 0xf954, 0xc65f, 0xf955,
3400 0xc65e, 0xf956, 0xf972, 0xf975, 0xf974, 0xc668, 0xf973, 0xc672,
3401 0xc670, 0xc671, 0xc677, 0xf9c0, 0xf9c1, 0xf9bf, 0xf9c9, 0xaaf8,
3402 0xd844, 0xc78, 0xe8a5, 0xf376, 0xaaf9, 0xadac, 0xb07b, 0xd845,
3403 0xd846, 0xb3ac, 0xb67d, 0xdc7a, 0xdc79, 0xb6a3, 0xb67c, 0xdc7b,
3404 0xb67e, 0xb6a2, 0xb6a1, 0xb67b, 0xb968, 0xe0d0, 0xe0ce, 0xe0cf,
3405 0xe0cd, 0xbbd2, 0xbbd5, 0xbbd7, 0xbbd6, 0xbbd3, 0xbbd4, 0xe8a7,
3406 0xe8a6, 0xbe5b, 0xe8a8, 0xe8a9, 0xbe5c, 0xec4d, 0xec4b, 0xeef3,
3407 0xec49, 0xec4a, 0xc046, 0xec46, 0xec4e, 0xec48, 0xec4c, 0xeef0,
3408 0xeef1, 0xeef2, 0xc1f3, 0xeef5, 0xc1f2, 0xeef0, 0xc1ef, 0xc1f0,
3409 0xc1f1, 0xec47, 0xc2f5, 0xf16e, 0xf16c, 0xf16d, 0xc2f3, 0xc2f6,

3410 0xc2f4, 0xf377, 0xf378, 0xc3f6, 0xf545, 0xf547, 0xf546, 0xc4c4,
3411 0xc550, 0xf66d, 0xf66c, 0xf66b, 0xaaafa, 0xc9aa, 0xca58, 0xa6e9,
3412 0xca56, 0xca59, 0xca57, 0xcbae, 0xa8c1, 0xa8c2, 0xcbb0, 0xa8bf,
3413 0xcbaaf, 0xcbad, 0xa8c0, 0xa8be, 0xcdd8, 0xcddb, 0xaaaf, 0xcdda,
3414 0xcdd9, 0xaaaf, 0xaaaf, 0xab40, 0xcddc, 0xaaaf, 0xd0c6, 0xadae,
3415 0xadaf, 0xadab0, 0xd0c7, 0xd0c3, 0xadad, 0xd0c4, 0xd0c5, 0xd0c2,
3416 0xb0a4, 0xb0a1, 0xd445, 0xb0a2, 0xb0a5, 0xd446, 0xb07e, 0xb07c,
3417 0xb07d, 0xb0a3, 0xb3ad, 0xd849, 0xb3b5, 0xd848, 0xd84b, 0xb3b1,
3418 0xd84a, 0xb6ab, 0xb3af, 0xb3b2, 0xb3ae, 0xb3b3, 0xb3b4, 0xb3b0,
3419 0xd847, 0xb6a7, 0xdc7d, 0xdc3, 0xdc2, 0xb6ac, 0xb6a8, 0xb6a9,
3420 0xdc7c, 0xdc7e, 0xdc1, 0xb6a4, 0xb6a6, 0xb6aa, 0xb6a5, 0xe0d3,
3421 0xe0d1, 0xe0d2, 0xb96a, 0xb96b, 0xe0d4, 0xb969, 0xbbd8, 0xbdda,
3422 0xbbd9, 0xe4bb, 0xe4bc, 0xe8ab, 0xe8aa, 0xc047, 0xc048, 0xec4f,
3423 0xc049, 0xeef6, 0xeef4, 0xeef5, 0xc1f4, 0xf1f6, 0xc3f7, 0xc1f5,
3424 0xab41, 0xb0a6, 0xd447, 0xd84c, 0xb3b6, 0xb6ad, 0xdca4, 0xdca6,
3425 0xb6af, 0xb6ae, 0xb6b0, 0xb6b1, 0xdca5, 0xb96e, 0xb96f, 0xb96d,
3426 0xbdb, 0xb96c, 0xe0d5, 0xbdb, 0xe8ac, 0xec50, 0xc04a, 0xc1f6,
3427 0xf170, 0xf174, 0xc2f9, 0xf171, 0xc2fa, 0xc2f8, 0xf175, 0xc2fb,
3428 0xf173, 0xf379, 0xc2f7, 0xc3f8, 0xf8cd, 0xab42, 0xb3b8, 0xb3b7,
3429 0xb6b2, 0xdca8, 0xca7, 0xb6b3, 0xe0d9, 0xb973, 0xb970, 0xe0d8,
3430 0xb972, 0xe0d6, 0xb971, 0xe0d7, 0xe4bd, 0xbdb, 0xe8af, 0xbe5d,
3431 0xe8ad, 0xbe5e, 0xbe5f, 0xe8ae, 0xbe60, 0xec51, 0xc04e, 0xc04b,
3432 0xc050, 0xec53, 0xc04c, 0xec52, 0xc04f, 0xc04d, 0xeef9, 0xeefb,
3433 0xc1f7, 0xeefa, 0xc1f8, 0xeef8, 0xeef7, 0xf177, 0xf176, 0xc2fc,
3434 0xf178, 0xf37e, 0xc3fa, 0xf37d, 0xf37a, 0xc3f9, 0xf37b, 0xf37c,
3435 0xf548, 0xf549, 0xc4c5, 0xc553, 0xf66e, 0xc551, 0xc552, 0xf66f,
3436 0xc5b4, 0xc5b5, 0xf771, 0xc645, 0xf8cf, 0xc647, 0xf8ce, 0xf8d0,
3437 0xc646, 0xf957, 0xf9ad, 0xab43, 0xb974, 0xe4be, 0xe8b0, 0xc051,
3438 0xc052, 0xab44, 0xbe61, 0xc3fb, 0xabd1, 0xc053, 0xc5e2, 0xabd2,
3439 0xd84d, 0xdca9, 0xdcab, 0xdcaa, 0xe0dd, 0xe0da, 0xb975, 0xb976,
3440 0xe0db, 0xe0dc, 0xe4c0, 0xe4c5, 0xbdb, 0xe4bf, 0xe4c1, 0xe4c8,
3441 0xe4c3, 0xe4c7, 0xe4c4, 0xe4c2, 0xe4c6, 0xbdb, 0xe8b3, 0xe8b1,
3442 0xbe63, 0xbe62, 0xe8b2, 0xbe64, 0xec56, 0xec55, 0xc054, 0xec54,
3443 0xeefc, 0xeefe, 0xef41, 0xef40, 0xc1f9, 0xeefd, 0xf1a1, 0xc2fd,
3444 0xf17d, 0xf1a2, 0xc2fe, 0xf17b, 0xf17e, 0xf17c, 0xf179, 0xc340,
3445 0xf17a, 0xf3a1, 0xf3a3, 0xf3a2, 0xf54a, 0xf54b, 0xf670, 0xc5b7,
3446 0xc5b6, 0xf84f, 0xf850, 0xc648, 0xf8d1, 0xc669, 0xabd3, 0xb6b4,
3447 0xe4ca, 0xe4c9, 0xe8b5, 0xe8b4, 0xc1fa, 0xef43, 0xef42, 0xf1a5,
3448 0xf1a3, 0xf1a6, 0xf1a4, 0xc3fc, 0xf3a4, 0xf3a5, 0xf3a6, 0xf671,
3449 0xf772, 0xf8d2, 0xabd4, 0xec57, 0xef44, 0xabd5, 0xbbe0, 0xec58,
3450 0xc341, 0xf1a7, 0xc3fd, 0xf54c, 0xf54d, 0xc554, 0xf851, 0xabd6,
3451 0xb3bb, 0xb3bc, 0xd84e, 0xb6b5, 0xb6b6, 0xdcac, 0xb6b7, 0xb97a,
3452 0xb97c, 0xe0df, 0xe0e0, 0xe0de, 0xb977, 0xb978, 0xb97b, 0xb979,
3453 0xe4cb, 0xbbe1, 0xbbe2, 0xe8bc, 0xbe67, 0xe8b7, 0xe8b6, 0xe8bb,
3454 0xbe65, 0xc05b, 0xe8b8, 0xe8bd, 0xe8ba, 0xe8b9, 0xbe66, 0xc059,
3455 0xec5a, 0xc055, 0xec5b, 0xec59, 0xc058, 0xc056, 0xc05a, 0xc057,
3456 0xef45, 0xef4a, 0xef46, 0xef49, 0xc1fb, 0xedd4, 0xef48, 0xef47,
3457 0xc344, 0xc342, 0xc345, 0xc343, 0xf1a8, 0xf1a9, 0xf1aa, 0xc346,
3458 0xf3aa, 0xc440, 0xf3a8, 0xc441, 0xf3a7, 0xf3a9, 0xc3fe, 0xf551,
3459 0xf54e, 0xf54f, 0xf550, 0xf672, 0xc556, 0xc555, 0xf774, 0xf773,
3460 0xc5b8, 0xc5e3, 0xc649, 0xc660, 0xf958, 0xf959, 0xf9af, 0xabd7,
3461 0xdcad, 0xe0e1, 0xe4cc, 0xe4cd, 0xbbe3, 0xbbe4, 0xe8be, 0xbe68,
3462 0xc1fc, 0xf1ab, 0xc347, 0xf3ad, 0xc442, 0xf3ac, 0xf3ae, 0xf3ab,
3463 0xf675, 0xf552, 0xf553, 0xc4c6, 0xf674, 0xf673, 0xf775, 0xf9b0,
3464 0xabd8, 0xabd9, 0xb0a7, 0xd448, 0xd84f, 0xb6b8, 0xb6bb, 0xb6b9,
3465 0xdcae, 0xb6bd, 0xb6ba, 0xb6bc, 0xb97e, 0xe0e2, 0xe0e3, 0xe8c0,
3466 0xb97d, 0xb9a1, 0xb9a2, 0xe4cf, 0xe4ce, 0xbbe5, 0xbbe6, 0xe4d0,
3467 0xe8bf, 0xbbe8, 0xbe69, 0xbbe7, 0xc05c, 0xe8c1, 0xbe6b, 0xbe6a,
3468 0xe8c2, 0xe8c5, 0xe8c3, 0xe8c4, 0xbe6c, 0xc061, 0xc05f, 0xc05e,
3469 0xec5d, 0xc060, 0xec5c, 0xf4b, 0xec5e, 0xc05d, 0xec5f, 0xf4e,
3470 0xf4c, 0xf4d, 0xf52, 0xc34b, 0xf51, 0xf54, 0xf53, 0xf50,
3471 0xf4f, 0xc1fd, 0xf1ae, 0xf1ad, 0xc34a, 0xc348, 0xc349, 0xf1ac,
3472 0xf3b1, 0xc443, 0xf3b0, 0xf3af, 0xc444, 0xf558, 0xf557, 0xf555,
3473 0xf554, 0xc4c8, 0xc4c7, 0xf559, 0xf776, 0xc5b9, 0xf677, 0xc557,
3474 0xf676, 0xf556, 0xf777, 0xc5e4, 0xc661, 0xf959, 0xf9b1, 0xabda,
3475 0xd850, 0xf555, 0xabdb, 0xe4d2, 0xe4d1, 0xec60, 0xf57, 0xf56,
3476 0xc34c, 0xf3b2, 0xf3b3, 0xc4c9, 0xf9b2, 0xb0a8, 0xb6bf, 0xb6be,
3477 0xe0e4, 0xe0e6, 0xb9a4, 0xe0e5, 0xb9a3, 0xb9a5, 0xe0e7, 0xe4d4,
3478 0xe4d6, 0xe4d5, 0xe4d8, 0xbbe9, 0xe4d7, 0xe4d3, 0xe4d9, 0xe8cc,
3479 0xe8cf, 0xe8d1, 0xe8c7, 0xe8cb, 0xe8c8, 0xbe6e, 0xbe71, 0xbe73,
3480 0xe8c9, 0xe8ca, 0xbe72, 0xe8cd, 0xe8d0, 0xe8ce, 0xbe74, 0xbe70,
3481 0xe8c6, 0xbe6d, 0xbe6f, 0xc063, 0xec66, 0xec64, 0xec63, 0xec69,
3482 0xec68, 0xec67, 0xec62, 0xc062, 0xec61, 0xec65, 0xc064, 0xf5a,
3483 0xf5e, 0xf5b, 0xf5d, 0xf5c, 0xf59, 0xf5f, 0xf62, 0xf60,
3484 0xf61, 0xc240, 0xc1fe, 0xf58, 0xf63, 0xf1b3, 0xf1b6, 0xf1b8,
3485 0xf1b7, 0xf1b1, 0xf1b5, 0xf1b0, 0xf1b2, 0xc34d, 0xf1af, 0xf1b4,
3486 0xf3c0, 0xf3b5, 0xc445, 0xc446, 0xf3b4, 0xf3b9, 0xf3bf, 0xf3b7,
3487 0xf3be, 0xf3bb, 0xf3ba, 0xf3bd, 0xf3b8, 0xf3b6, 0xf3bc, 0xf560,
3488 0xf55e, 0xc4ca, 0xf55d, 0xf563, 0xf561, 0xc4cb, 0xf55c, 0xf55a,
3489 0xf55b, 0xc4cd, 0xf55f, 0xc4cc, 0xf562, 0xf678, 0xf67e, 0xf679,
3490 0xc55b, 0xf6a1, 0xc55a, 0xf67d, 0xf67c, 0xc559, 0xf67b, 0xc558,
3491 0xf67a, 0xf77d, 0xf7a1, 0xf77e, 0xf77b, 0xc55b, 0xf778, 0xf77c,
3492 0xf7a3, 0xf7a2, 0xf779, 0xf77a, 0xc5ba, 0xf852, 0xc5e7, 0xf853,
3493 0xc5e5, 0xc5e6, 0xf8d3, 0xc64a, 0xf976, 0xc66a, 0xf9b3, 0xc66b,
3494 0xf9b4, 0xf9b5, 0xf9c3, 0xf9c2, 0xc67a, 0xf9cd, 0xb0a9, 0xe0e9,
3495 0xe0e8, 0xbbea, 0xbbeb, 0xe4da, 0xe8d2, 0xec6c, 0xbe75, 0xc065,
3496 0xec6a, 0xec6d, 0xc066, 0xf64, 0xec6b, 0xf1b9, 0xc34e, 0xf3c1,

3497 0xf566, 0xf564, 0xf565, 0xf6a2, 0xc55c, 0xf7a4, 0xc5ea, 0xc5bc,
3498 0xc5e8, 0xc5e9, 0xf8d4, 0xc662, 0xb0aa, 0xf1ba, 0xd449, 0xb9a6,
3499 0xe4db, 0xbbec, 0xe4dc, 0xe8d4, 0xe8d3, 0xc068, 0xbe76, 0xbe77,
3500 0xe8d7, 0xe8d6, 0xe8d5, 0xec6e, 0xec71, 0xec70, 0xec6f, 0xc067,
3501 0xef68, 0xef66, 0xef65, 0xef67, 0xc34f, 0xf1bc, 0xf1bd, 0xc350,
3502 0xf1bb, 0xf3c3, 0xf3c2, 0xf3c5, 0xc447, 0xf3c4, 0xf567, 0xf569,
3503 0xf568, 0xf6a3, 0xf6a6, 0xf6a4, 0xf6a5, 0xf7a5, 0xc5bd, 0xf854,
3504 0xf855, 0xf856, 0xc64b, 0xc663, 0xf9b6, 0xb0ab, 0xbe78, 0xc069,
3505 0xf1be, 0xf7a6, 0xf9c4, 0xd44a, 0xc67b, 0xb0ac, 0xec72, 0xf1bf,
3506 0xf3c6, 0xf6a7, 0xf7a7, 0xb0ad, 0xe4dd, 0xe4de, 0xbbed, 0xbbee,
3507 0xe8d9, 0xbe7a, 0xbe79, 0xe8d8, 0xef69, 0xf1c0, 0xf1c2, 0xf1c1,
3508 0xc353, 0xc352, 0xc351, 0xc55e, 0xf6a8, 0xc55d, 0xf7a9, 0xf7a8,
3509 0xc64c, 0xf8d5, 0xb3bd, 0xe0ea, 0xe4e1, 0xe4df, 0xe4e0, 0xe8e2,
3510 0xe8dd, 0xe8da, 0xe8e1, 0xe8e3, 0xbe7c, 0xe8e0, 0xe8dc, 0xe8db,
3511 0xe8df, 0xe8de, 0xbe7b, 0xec7d, 0xec78, 0xec76, 0xecal, 0xec77,
3512 0xec73, 0xec79, 0xec74, 0xef72, 0xec75, 0xeca2, 0xec7c, 0xc06a,
3513 0xec7b, 0xec7a, 0xec7e, 0xef6a, 0xef6d, 0xef6c, 0xef74, 0xef6f,
3514 0xef73, 0xef71, 0xef70, 0xef6e, 0xef6b, 0xc243, 0xc242, 0xc244,
3515 0xc241, 0xef75, 0xf1c8, 0xf1cb, 0xf1c9, 0xf1cd, 0xf1ce, 0xf1c6,
3516 0xc358, 0xf1c7, 0xf1c5, 0xf1cc, 0xf1c4, 0xf1c3, 0xc357, 0xc355,
3517 0xc354, 0xf1ca, 0xf1cf, 0xf3cf, 0xf3d5, 0xc44a, 0xf3d0, 0xf3d3, 0xf3d7,
3518 0xc44b, 0xf3d2, 0xf3ca, 0xf3c9, 0xf3d6, 0xf3cd, 0xf3cb, 0xf3d4,
3519 0xf3cc, 0xc449, 0xc448, 0xf3c7, 0xf3c8, 0xf3d1, 0xf3ce, 0xf56c,
3520 0xf56f, 0xc356, 0xf56d, 0xf573, 0xf571, 0xf56b, 0xf576, 0xf56a,
3521 0xc4cf, 0xf572, 0xf56e, 0xc4ce, 0xf575, 0xf574, 0xf6ab, 0xf6aa,
3522 0xf6b1, 0xf6ad, 0xf6b0, 0xc560, 0xf6ae, 0xf6af, 0xf6a9, 0xf6ac,
3523 0xc55f, 0xc5bf, 0xf7b4, 0xf7af, 0xf7b3, 0xf7b6, 0xf7b2, 0xf7ae,
3524 0xc5c1, 0xf7b1, 0xf7b5, 0xc5c0, 0xf7ac, 0xf570, 0xf7b0, 0xf7ad,
3525 0xf7aa, 0xf7ab, 0xc5be, 0xf85a, 0xf85c, 0xf85f, 0xf85b, 0xf860,
3526 0xf859, 0xf857, 0xc5eb, 0xf85d, 0xc5ed, 0xc5ec, 0xf858, 0xf85e,
3527 0xf8da, 0xc64d, 0xf8db, 0xf8d9, 0xf8d6, 0xf8d8, 0xf8d7, 0xf95a,
3528 0xf95c, 0xf95b, 0xf979, 0xf978, 0xf977, 0xf97a, 0xc673, 0xc674,
3529 0xf9ca, 0xf9ce, 0xb3be, 0xdcaf, 0xe0ed, 0xb9a7, 0xe0eb, 0xe0ec,
3530 0xe4e2, 0xe4e3, 0xbbf1, 0xbbef, 0xe4e4, 0xbbf0, 0xe8e8, 0xe8eb,
3531 0xe8e5, 0xe8ec, 0xe8e4, 0xe8e6, 0xe8e7, 0xe8ea, 0xbeal, 0xe8ef,
3532 0xe8ee, 0xbe7d, 0xe8e9, 0xe8ed, 0xbe7e, 0xecac, 0xc06f, 0xeca7,
3533 0xc06b, 0xeca4, 0xeca, 0xecad, 0xc070, 0xeca9, 0xeca6, 0xeca, 0xeca5,
3534 0xecab, 0xc06c, 0xeca3, 0xc06d, 0xc06e, 0xeca8, 0xefa9,
3535 0xef7a, 0xef7b, 0xef7e, 0xef7c, 0xef76, 0xef79, 0xefa5, 0xef7d,
3536 0xc245, 0xefa7, 0xefa4, 0xc246, 0xefa6, 0xef77, 0xefa2, 0xefa3,
3537 0xefa1, 0xf1d2, 0xf1d4, 0xf1d7, 0xf1d1, 0xc359, 0xf1d9, 0xf1d0,
3538 0xf1da, 0xf1d6, 0xf1d8, 0xf1dc, 0xf1d5, 0xf1dd, 0xf1d3, 0xf1c, 0xf1db,
3539 0xc35a, 0xf1db, 0xc35b, 0xc44d, 0xef78, 0xf3f1, 0xf3e8, 0xc44f,
3540 0xf3e4, 0xc450, 0xf3ed, 0xf3e7, 0xf3dd, 0xc44e, 0xf3ea, 0xf3e5,
3541 0xf3e6, 0xf3d8, 0xf3df, 0xf3ee, 0xf3eb, 0xf3e3, 0xf3ef, 0xf3de,
3542 0xf3d9, 0xf3ec, 0xf3db, 0xf3e9, 0xf3e0, 0xf3f0, 0xf3dc, 0xc44c,
3543 0xf3da, 0xf3e1, 0xf3e2, 0xf37d, 0xf37b, 0xf37a, 0xf37c, 0xf37e,
3544 0xf37c, 0xf378, 0xf37d, 0xf37e, 0xf37a, 0xf37b, 0xf37c, 0xf37e,
3545 0xf37a, 0xf376, 0xf377, 0xf378, 0xf379, 0xf37a, 0xf37b, 0xf37c,
3546 0xf37d, 0xf37e, 0xf37f, 0xf380, 0xf381, 0xf382, 0xf383, 0xf384,
3547 0xf385, 0xf386, 0xf387, 0xf388, 0xf389, 0xf38a, 0xf38b, 0xf38c,
3548 0xf38d, 0xf38e, 0xf38f, 0xf390, 0xf391, 0xf392, 0xf393, 0xf394,
3549 0xf395, 0xf396, 0xf397, 0xf398, 0xf399, 0xf39a, 0xf39b, 0xf39c,
3550 0xf39d, 0xf39e, 0xf39f, 0xf3a0, 0xf3a1, 0xf3a2, 0xf3a3, 0xf3a4,
3551 0xf3a5, 0xf3a6, 0xf3a7, 0xf3a8, 0xf3a9, 0xf3aa, 0xf3ab, 0xf3ac,
3552 0xf3ad, 0xf3ae, 0xf3af, 0xf3b0, 0xf3b1, 0xf3b2, 0xf3b3, 0xf3b4,
3553 0xf3b5, 0xf3b6, 0xf3b7, 0xf3b8, 0xf3b9, 0xf3ba, 0xf3bb, 0xf3bc,
3554 0xf3bd, 0xf3be, 0xf3bf, 0xf3c0, 0xf3c1, 0xf3c2, 0xf3c3, 0xf3c4,
3555 0xf3c5, 0xf3c6, 0xf3c7, 0xf3c8, 0xf3c9, 0xf3ca, 0xf3cb, 0xf3cc,
3556 0xf3cd, 0xf3ce, 0xf3cf, 0xf3d0, 0xf3d1, 0xf3d2, 0xf3d3, 0xf3d4,
3557 0xf3d5, 0xf3d6, 0xf3d7, 0xf3d8, 0xf3d9, 0xf3da, 0xf3db, 0xf3dc,
3558 0xf3dd, 0xf3de, 0xf3df, 0xf3e0, 0xf3e1, 0xf3e2, 0xf3e3, 0xf3e4,
3559 0xf3e5, 0xf3e6, 0xf3e7, 0xf3e8, 0xf3e9, 0xf3ea, 0xf3eb, 0xf3ec,
3560 0xf3ed, 0xf3ee, 0xf3ef, 0xf3f0, 0xf3f1, 0xf3f2, 0xf3f3, 0xf3f4,
3561 0xf3f5, 0xf3f6, 0xf3f7, 0xf3f8, 0xf3f9, 0xf3fa, 0xf3fb, 0xf3fc,
3562 0xf3fd, 0xf3fe, 0xf3ff, 0xf400, 0xf401, 0xf402, 0xf403, 0xf404,
3563 0xf405, 0xf406, 0xf407, 0xf408, 0xf409, 0xf40a, 0xf40b, 0xf40c,
3564 0xf40d, 0xf40e, 0xf40f, 0xf410, 0xf411, 0xf412, 0xf413, 0xf414,
3565 0xf415, 0xf416, 0xf417, 0xf418, 0xf419, 0xf41a, 0xf41b, 0xf41c,
3566 0xf41d, 0xf41e, 0xf41f, 0xf420, 0xf421, 0xf422, 0xf423, 0xf424,
3567 0xf425, 0xf426, 0xf427, 0xf428, 0xf429, 0xf42a, 0xf42b, 0xf42c,
3568 0xf42d, 0xf42e, 0xf42f, 0xf430, 0xf431, 0xf432, 0xf433, 0xf434,
3569 0xf435, 0xf436, 0xf437, 0xf438, 0xf439, 0xf43a, 0xf43b, 0xf43c,
3570 0xf43d, 0xf43e, 0xf43f, 0xf440, 0xf441, 0xf442, 0xf443, 0xf444,
3571 0xf445, 0xf446, 0xf447, 0xf448, 0xf449, 0xf44a, 0xf44b, 0xf44c,
3572 0xf44d, 0xf44e, 0xf44f, 0xf450, 0xf451, 0xf452, 0xf453, 0xf454,
3573 0xf455, 0xf456, 0xf457, 0xf458, 0xf459, 0xf45a, 0xf45b, 0xf45c,
3574 0xf45d, 0xf45e, 0xf45f, 0xf460, 0xf461, 0xf462, 0xf463, 0xf464,
3575 0xf465, 0xf466, 0xf467, 0xf468, 0xf469, 0xf46a, 0xf46b, 0xf46c,
3576 0xf46d, 0xf46e, 0xf46f, 0xf470, 0xf471, 0xf472, 0xf473, 0xf474,
3577 0xf475, 0xf476, 0xf477, 0xf478, 0xf479, 0xf47a, 0xf47b, 0xf47c,
3578 0xf47d, 0xf47e, 0xf47f, 0xf480, 0xf481, 0xf482, 0xf483, 0xf484,
3579 0xf485, 0xf486, 0xf487, 0xf488, 0xf489, 0xf48a, 0xf48b, 0xf48c,
3580 0xf48d, 0xf48e, 0xf48f, 0xf490, 0xf491, 0xf492, 0xf493, 0xf494,
3581 0xf495, 0xf496, 0xf497, 0xf498, 0xf499, 0xf49a, 0xf49b, 0xf49c,
3582 0xf49d, 0xf49e, 0xf49f, 0xf4a0, 0xf4a1, 0xf4a2, 0xf4a3, 0xf4a4,
3583 0xf4a5, 0xf4a6, 0xf4a7, 0xf4a8, 0xf4a9, 0xf4aa, 0xf4ab, 0xf4ac,

```

3584 0xa16c, 0xa16f, 0xa170, 0xa173, 0xa174, 0xa177, 0xa178, 0xa17b,
3585 0xa17c, 0xa1c6, 0xa1c7, 0xa1ca, 0xa1cb, 0xa1c8, 0xa1c9, 0xa15c,
3586 0xa14d, 0xa14f, 0xa151, 0xa152, 0xa153, 0xa154, 0xa17d, 0xa17e,
3587 0xa1a1, 0xa1a2, 0xa1a3, 0xa1a4, 0xa1cc, 0xa1cd, 0xa1ce, 0xa1de,
3588 0xa1df, 0xa1e0, 0xa1e1, 0xa1e2, 0xa24c, 0xa24d, 0xa24e, 0xa149,
3589 0xa1ad, 0xa243, 0xa248, 0xa1ae, 0xa15d, 0xa15e, 0xa1af, 0xa1cf,
3590 0xa141, 0xa1d0, 0xa144, 0xa241, 0xa2af, 0xa2b0, 0xa2b1, 0xa2b2,
3591 0xa2b3, 0xa2b4, 0xa2b5, 0xa2b6, 0xa2b7, 0xa2b8, 0xa147, 0xa146,
3592 0xa1d5, 0xa1d7, 0xa1d6, 0xa148, 0xa249, 0xa2cf, 0xa2d0, 0xa2d1,
3593 0xa2d2, 0xa2d3, 0xa2d4, 0xa2d5, 0xa2d6, 0xa2d7, 0xa2d8, 0xa2d9,
3594 0xa2da, 0xa2db, 0xa2dc, 0xa2dd, 0xa2de, 0xa2df, 0xa2e0, 0xa2e1,
3595 0xa2e2, 0xa2e3, 0xa2e4, 0xa2e5, 0xa2e6, 0xa2e7, 0xa2e8, 0xa242,
3596 0xa1c4, 0xa2e9, 0xa2ea, 0xa2eb, 0xa2ec, 0xa2ed, 0xa2ee, 0xa2ef,
3597 0xa2f0, 0xa2f1, 0xa2f2, 0xa2f3, 0xa2f4, 0xa2f5, 0xa2f6, 0xa2f7,
3598 0xa2f8, 0xa2f9, 0xa2fa, 0xa2fb, 0xa2fc, 0xa2fd, 0xa2fe, 0xa340,
3599 0xa341, 0xa342, 0xa343, 0xa161, 0xa155, 0xa162, 0xa14e,
3600 };
3601
3602 static const Summary16 big5_uni2indx_page00[16] = {
3603 /* 0x0000 */
3604 { 0, 0x0000 }, { 0, 0x0000 }, { 0, 0x0000 }, { 0, 0x0000 },
3605 { 0, 0x0000 }, { 0, 0x0000 }, { 0, 0x0000 }, { 0, 0x0000 },
3606 { 0, 0x0000 }, { 0, 0x0000 }, { 0, 0x00ac }, { 4, 0x0083 },
3607 { 7, 0x0000 }, { 7, 0x0080 }, { 8, 0x0000 }, { 8, 0x0080 },
3608 };
3609 static const Summary16 big5_uni2indx_page02[38] = {
3610 /* 0x0200 */
3611 { 9, 0x0000 }, { 9, 0x0000 }, { 9, 0x0000 }, { 9, 0x0000 },
3612 { 9, 0x0000 }, { 9, 0x0000 }, { 9, 0x0000 }, { 9, 0x0000 },
3613 { 9, 0x0000 }, { 9, 0x0000 }, { 9, 0x0000 }, { 9, 0x0000 },
3614 { 9, 0x0e80 }, { 13, 0x0200 }, { 14, 0x0000 }, { 14, 0x0000 },
3615 /* 0x0300 */
3616 { 14, 0x0000 }, { 14, 0x0000 }, { 14, 0x0000 }, { 14, 0x0000 },
3617 { 14, 0x0000 }, { 14, 0x0000 }, { 14, 0x0000 }, { 14, 0x0000 },
3618 { 14, 0x0000 }, { 14, 0xffff }, { 29, 0x03fb }, { 38, 0xffff },
3619 { 53, 0x03fb }, { 62, 0x0000 }, { 62, 0x0000 }, { 62, 0x0000 },
3620 /* 0x0400 */
3621 { 62, 0x0002 }, { 63, 0x1ff0 }, { 72, 0xffff8 }, { 85, 0xffff },
3622 { 101, 0xffff }, { 117, 0x0002 },
3623 };
3624 static const Summary16 big5_uni2indx_page20[44] = {
3625 /* 0x2000 */
3626 { 118, 0x0000 }, { 118, 0x3318 }, { 124, 0x0064 }, { 127, 0x4824 },
3627 { 131, 0x0000 }, { 131, 0x0000 }, { 131, 0x0000 }, { 131, 0x0000 },
3628 { 131, 0x0000 }, { 131, 0x0000 }, { 131, 0x0000 }, { 131, 0x0000 },
3629 { 131, 0x0000 }, { 131, 0x0000 }, { 131, 0x0000 }, { 131, 0x0000 },
3630 /* 0x2100 */
3631 { 131, 0x0228 }, { 134, 0x0000 }, { 134, 0x0000 }, { 134, 0x0000 },
3632 { 134, 0x0000 }, { 134, 0x0000 }, { 134, 0x03ff }, { 144, 0x0000 },
3633 { 144, 0x0000 }, { 144, 0x03cf }, { 152, 0x0000 }, { 152, 0x0000 },
3634 { 152, 0x0000 }, { 152, 0x0000 }, { 152, 0x0000 }, { 152, 0x0000 },
3635 /* 0x2200 */
3636 { 152, 0x0000 }, { 152, 0xc400 }, { 155, 0x4e29 }, { 162, 0x1030 },
3637 { 165, 0x0000 }, { 165, 0x0004 }, { 166, 0x00c3 }, { 170, 0x0000 },
3638 { 170, 0x0000 }, { 170, 0x0000 }, { 170, 0x0020 }, { 171, 0x8000 },
3639 };
3640 static const Summary16 big5_uni2indx_page24[37] = {
3641 /* 0x2400 */
3642 { 172, 0x0000 }, { 172, 0x0000 }, { 172, 0x0000 }, { 172, 0x0000 },
3643 { 172, 0x0000 }, { 172, 0x0000 }, { 172, 0x03ff }, { 182, 0x3ff0 },
3644 { 192, 0x0000 }, { 192, 0x0000 }, { 192, 0x0000 }, { 192, 0x0000 },
3645 { 192, 0x0000 }, { 192, 0x0000 }, { 192, 0x0000 }, { 192, 0x0000 },
3646 /* 0x2500 */
3647 { 192, 0x1005 }, { 195, 0x1111 }, { 199, 0x1010 }, { 201, 0x1010 },
3648 { 203, 0x0000 }, { 203, 0x4001 }, { 205, 0xe402 }, { 210, 0x000f },
3649 { 214, 0xffff }, { 229, 0x0030 }, { 231, 0x0003 }, { 233, 0x300c },
3650 { 237, 0xc8c0 }, { 242, 0x0000 }, { 242, 0x003c }, { 246, 0x0000 },
3651 /* 0x2600 */
3652 { 246, 0x0260 }, { 249, 0x0000 }, { 249, 0x0000 }, { 249, 0x0000 },
3653 { 249, 0x0007 },
3654 };
3655 static const Summary16 big5_uni2indx_page30[62] = {
3656 /* 0x3000 */
3657 { 252, 0xff2f }, { 265, 0x6037 }, { 272, 0x03fe }, { 281, 0x0000 },
3658 { 281, 0xffff }, { 296, 0xffff }, { 312, 0xffff }, { 328, 0xffff },
3659 { 344, 0xffff }, { 360, 0x600f }, { 366, 0xffff }, { 381, 0xffff },
3660 { 397, 0xffff }, { 413, 0xffff }, { 429, 0xffff }, { 445, 0x407f },
3661 /* 0x3100 */
3662 { 453, 0xffe0 }, { 464, 0xffff }, { 480, 0x03ff }, { 490, 0x0000 },
3663 { 490, 0x0000 }, { 490, 0x0000 }, { 490, 0x0000 }, { 490, 0x0000 },
3664 { 490, 0x0000 }, { 490, 0x0000 }, { 490, 0x0000 }, { 490, 0x0000 },
3665 { 490, 0x0000 }, { 490, 0x0000 }, { 490, 0x0000 }, { 490, 0x0000 },
3666 /* 0x3200 */
3667 { 490, 0x0000 }, { 490, 0x0000 }, { 490, 0x0000 }, { 490, 0x0000 },
3668 { 490, 0x0000 }, { 490, 0x0000 }, { 490, 0x0000 }, { 490, 0x0000 },
3669 { 490, 0x0000 }, { 490, 0x0000 }, { 490, 0x0008 }, { 491, 0x0000 },
3670 { 491, 0x0000 }, { 491, 0x0000 }, { 491, 0x0000 }, { 491, 0x0000 },

```

```
3671 /* 0x3300 */
3672 { 491, 0x0000 }, { 491, 0x0000 }, { 491, 0x0000 }, { 491, 0x0000 },
3673 { 491, 0x0000 }, { 491, 0x0000 }, { 491, 0x0000 }, { 491, 0x0000 },
3674 { 491, 0xc000 }, { 493, 0x7000 }, { 496, 0x0002 }, { 497, 0x0000 },
3675 { 497, 0x4010 }, { 499, 0x0026 },
3676 };
3677 static const Summary16 big5_uni2indx_page4e[1307] = {
3678 /* 0x4e00 */
3679 { 502, 0xff8b }, { 514, 0xc373 }, { 523, 0x6840 }, { 527, 0x1b0f },
3680 { 535, 0xe9ac }, { 544, 0xf34c }, { 553, 0x0200 }, { 554, 0xc008 },
3681 { 557, 0x795c }, { 566, 0xca3e }, { 575, 0x7976 }, { 585, 0x0648 },
3682 { 589, 0x2fdf }, { 601, 0xf7f0 }, { 612, 0x033a }, { 618, 0xa8ff },
3683 /* 0x4f00 */
3684 { 629, 0xef37 }, { 641, 0x233f }, { 650, 0xb004 }, { 654, 0xfd59 },
3685 { 665, 0xf3ca }, { 675, 0xffff }, { 691, 0xde9f }, { 703, 0xffff9 },
3686 { 717, 0xabff }, { 730, 0x7df7 }, { 743, 0xc000 }, { 745, 0x8eec },
3687 { 754, 0xeebf }, { 767, 0xffdb }, { 781, 0xd003 }, { 786, 0x45fa },
3688 /* 0x5000 */
3689 { 795, 0xfae1 }, { 805, 0xdffe }, { 819, 0xbfef }, { 833, 0x10ab },
3690 { 839, 0xffeb }, { 853, 0xfcaa }, { 863, 0xef3f }, { 876, 0x24fd },
3691 { 885, 0x78ad }, { 894, 0xf7f6 }, { 906, 0xf00c }, { 912, 0xedff },
3692 { 926, 0xcff6 }, { 938, 0x2cfa }, { 947, 0xf7f9 }, { 960, 0xeb6b },
3693 /* 0x5100 */
3694 { 971, 0x1ffd }, { 983, 0x95bf }, { 994, 0x6677 }, { 1004, 0xbfbf },
3695 { 1018, 0x3bfb }, { 1030, 0xfeb4 }, { 1041, 0x7bae }, { 1052, 0x11e2 },
3696 { 1058, 0xa681 }, { 1064, 0x41be }, { 1072, 0x1435 }, { 1078, 0x72c3 },
3697 { 1086, 0x7d70 }, { 1095, 0x7191 }, { 1102, 0x0003 }, { 1104, 0x276b },
3698 /* 0x5200 */
3699 { 1113, 0x57cb }, { 1123, 0x70cf }, { 1132, 0x4732 }, { 1139, 0x0def },
3700 { 1149, 0x7eda }, { 1160, 0xfc74 }, { 1170, 0xfe06 }, { 1179, 0xbd84 },
3701 { 1189, 0x3f9f }, { 1201, 0x8bca }, { 1209, 0x7e49 }, { 1218, 0x5800 },
3702 { 1221, 0x228f }, { 1228, 0xebec }, { 1239, 0x8a5c }, { 1246, 0xddbb },
3703 /* 0x5300 */
3704 { 1258, 0xef60 }, { 1267, 0xb6e7 }, { 1278, 0xa40f }, { 1285, 0xf293 },
3705 { 1294, 0x37bb }, { 1305, 0x549e }, { 1313, 0xd04b }, { 1320, 0x9baf },
3706 { 1331, 0xc414 }, { 1336, 0xf7d4 }, { 1347, 0x30b0 }, { 1352, 0xa14 },
3707 { 1356, 0x2f08 }, { 1362, 0x88d0 }, { 1367, 0xff7e }, { 1381, 0x192f },
3708 /* 0x5400 */
3709 { 1389, 0xffda }, { 1402, 0xfb07 }, { 1412, 0x7ff1 }, { 1424, 0x7beb },
3710 { 1436, 0xc5ef }, { 1447, 0x0010 }, { 1448, 0x99ff }, { 1460, 0xfdf },
3711 { 1475, 0x79d7 }, { 1486, 0x0567 }, { 1493, 0xffe7 }, { 1507, 0xfdc },
3712 { 1519, 0xc3ff }, { 1531, 0x4040 }, { 1533, 0x6ff7 }, { 1546, 0xbd8e },
3713 /* 0x5500 */
3714 { 1556, 0xdffa }, { 1569, 0x0497 }, { 1575, 0xf4c0 }, { 1582, 0x5bff },
3715 { 1595, 0xed7b }, { 1607, 0xd0e7 }, { 1616, 0x047e }, { 1623, 0xf8e0 },
3716 { 1631, 0xff9f }, { 1645, 0xb73e }, { 1656, 0x7dfe }, { 1669, 0x882e },
3717 { 1675, 0xffffd }, { 1690, 0xbe7f }, { 1703, 0x83fe }, { 1713, 0xf6c4 },
3718 /* 0x5600 */
3719 { 1722, 0xf357 }, { 1733, 0xb8fd }, { 1744, 0xd680 }, { 1750, 0xef7d },
3720 { 1763, 0x5767 }, { 1773, 0x4788 }, { 1779, 0xff7d }, { 1793, 0xc3df },
3721 { 1804, 0xf0ff }, { 1816, 0x37a9 }, { 1825, 0x7de0 }, { 1834, 0x70fc },
3722 { 1843, 0x3f6f }, { 1855, 0xec9a }, { 1864, 0x4cb3 }, { 1872, 0x8681 },
3723 /* 0x5700 */
3724 { 1877, 0x3f9e }, { 1888, 0xdd5c }, { 1898, 0xf70d }, { 1908, 0x4819 },
3725 { 1913, 0xfea3 }, { 1924, 0x0007 }, { 1927, 0xaf56 }, { 1937, 0x38ff },
3726 { 1948, 0x980d }, { 1954, 0xefb8 }, { 1965, 0x403d }, { 1971, 0xb760 },
3727 { 1979, 0xd8ce }, { 1988, 0x9035 }, { 1994, 0x72bf }, { 2005, 0x3fff },
3728 /* 0x5800 */
3729 { 2019, 0x7ff7 }, { 2033, 0x7a11 }, { 2040, 0xf7bb }, { 2053, 0xabff },
3730 { 2066, 0xff00 }, { 2074, 0x6f8e }, { 2086, 0xa93c }, { 2094, 0xfe72 },
3731 { 2105, 0xcfef }, { 2118, 0xf11b }, { 2127, 0xdb6b }, { 2138, 0xf40a },
3732 { 2145, 0xc3e6 }, { 2154, 0xef7e }, { 2167, 0x9b9c }, { 2176, 0xf610 },
3733 /* 0x5900 */
3734 { 2183, 0xf048 }, { 2189, 0x16f4 }, { 2197, 0xfeb5 }, { 2209, 0x5182 },
3735 { 2214, 0xc7b1 }, { 2223, 0x15bb }, { 2232, 0x6e87 }, { 2241, 0xfbd },
3736 { 2255, 0xe43f }, { 2265, 0x63cd }, { 2274, 0xc1ff }, { 2285, 0x7e7e },
3737 { 2297, 0xfdeb }, { 2310, 0x7d5f }, { 2322, 0x777b }, { 2334, 0xfefe },
3738 /* 0x5a00 */
3739 { 2347, 0x960b }, { 2354, 0xdbea }, { 2365, 0x6229 }, { 2371, 0x53e8 },
3740 { 2379, 0x37df }, { 2391, 0xfdef }, { 2405, 0x36f5 }, { 2415, 0xbd81 },
3741 { 2423, 0xdc18 }, { 2430, 0xfcdb }, { 2442, 0xd2e4 }, { 2450, 0xffff },
3742 { 2466, 0x3fd7 }, { 2478, 0xffe0 }, { 2489, 0x7f6f }, { 2502, 0xabf8 },
3743 /* 0x5b00 */
3744 { 2512, 0x9bae }, { 2522, 0x6ed9 }, { 2532, 0xf5fb }, { 2545, 0xf115 },
3745 { 2553, 0x79a9 }, { 2562, 0xbdfb }, { 2575, 0x5a3c }, { 2583, 0xadaf },
3746 { 2594, 0xdbba }, { 2605, 0x1fac }, { 2614, 0x71fc }, { 2624, 0x8379 },
3747 { 2632, 0x7cf7 }, { 2644, 0xc35f }, { 2654, 0xdfff }, { 2669, 0x0567 },
3748 /* 0x5c00 */
3749 { 2676, 0xff9a }, { 2688, 0x8467 }, { 2695, 0x1534 }, { 2701, 0xdf8b },
3750 { 2712, 0xf9f3 }, { 2724, 0x3373 }, { 2733, 0xf7bd }, { 2746, 0x5e1a },
3751 { 2754, 0xbf40 }, { 2762, 0xa03f }, { 2770, 0xffff }, { 2786, 0x01eb },
3752 { 2793, 0xdfc0 }, { 2802, 0xcfd }, { 2814, 0x7500 }, { 2819, 0xabd3 },
3753 /* 0x5d00 */
3754 { 2829, 0xf8c3 }, { 2838, 0xeed6 }, { 2849, 0x43fd }, { 2859, 0xb7ff },
3755 { 2873, 0x5eaf }, { 2884, 0x4227 }, { 2890, 0x9bac }, { 2899, 0xf686 },
3756 { 2908, 0x27d7 }, { 2918, 0xf6bc }, { 2929, 0xf787 }, { 2940, 0x35b7 },
3757 { 2950, 0xaacd }, { 2959, 0xe176 }, { 2968, 0x49e7 }, { 2977, 0xe29f },
```

```

3758 /* 0x5e00 */
3759 { 2987, 0x545c }, { 2994, 0xaff2 }, { 3005, 0x2b3f }, { 3015, 0x61d8 },
3760 { 3022, 0xfc3b }, { 3033, 0xbbbb }, { 3043, 0xffcf }, { 3057, 0x7b7d },
3761 { 3069, 0xbf95 }, { 3080, 0x1ce0 }, { 3086, 0x7dfd }, { 3099, 0x43ff },
3762 { 3110, 0x5fff6 }, { 3122, 0xffffe }, { 3137, 0xd3ef }, { 3149, 0xc4ce },
3763 /* 0x5f00 */
3764 { 3157, 0x8db6 }, { 3166, 0xadbc }, { 3176, 0x63dc }, { 3185, 0x11eb },
3765 { 3193, 0xdf59 }, { 3204, 0x23d0 }, { 3210, 0xbeb4 }, { 3220, 0xf3db },
3766 { 3232, 0x1fe7 }, { 3243, 0xdbc7 }, { 3254, 0xff63 }, { 3266, 0xfae4 },
3767 { 3276, 0xb22b }, { 3284, 0x63f7 }, { 3295, 0xed3b }, { 3306, 0xadba },
3768 /* 0x6000 */
3769 { 3316, 0xfe01 }, { 3324, 0x7eff }, { 3338, 0xffff7 }, { 3353, 0x02bc },
3770 { 3359, 0x32ff }, { 3370, 0xef39 }, { 3381, 0xffffc }, { 3395, 0x8005 },
3771 { 3398, 0x77fb }, { 3411, 0xbcf5 }, { 3422, 0x010d }, { 3426, 0xffff7 },
3772 { 3441, 0xffffb }, { 3456, 0xbf3a }, { 3467, 0x0057 }, { 3472, 0xdfff },
3773 /* 0x6100 */
3774 { 3487, 0xef7b }, { 3500, 0xbd7d }, { 3512, 0xdb88 }, { 3520, 0xc8d4 },
3775 { 3527, 0xffff3 }, { 3541, 0xed7c }, { 3552, 0x5dee }, { 3563, 0x56ff },
3776 { 3575, 0x7e0d }, { 3584, 0xac5f }, { 3594, 0xff96 }, { 3606, 0xd57f },
3777 { 3618, 0x3fee }, { 3630, 0xc140 }, { 3634, 0x6ff9 }, { 3646, 0xffe7 },
3778 /* 0x6200 */
3779 { 3660, 0x779b }, { 3671, 0x8e77 }, { 3681, 0x6ebf }, { 3693, 0xe45d },
3780 { 3702, 0x6fcf }, { 3714, 0x5f1f }, { 3725, 0xe07f }, { 3735, 0xfedf },
3781 { 3749, 0xd7db }, { 3761, 0x01fe }, { 3769, 0xff00 }, { 3777, 0xfb7b },
3782 { 3790, 0xffd4 }, { 3802, 0x1fdf }, { 3814, 0xf800 }, { 3819, 0xffff },
3783 /* 0x6300 */
3784 { 3835, 0xfb8f }, { 3847, 0x007b }, { 3853, 0xbf00 }, { 3860, 0x7f5c },
3785 { 3871, 0xfffff }, { 3887, 0x07f3 }, { 3896, 0xeba0 }, { 3904, 0x3de7 },
3786 { 3915, 0xf7bf }, { 3929, 0xfbd7 }, { 3942, 0xfbf }, { 3957, 0x6003 },
3787 { 3961, 0xffffd }, { 3976, 0xbfed }, { 3989, 0xfbb }, { 4002, 0x027f },
3788 /* 0x6400 */
3789 { 4010, 0xfe40 }, { 4018, 0xddfd }, { 4031, 0xfdf }, { 4046, 0xe2f9 },
3790 { 4056, 0x680b }, { 4062, 0xfb1f }, { 4074, 0xfbe3 }, { 4086, 0xaaffd },
3791 { 4099, 0x9fa4 }, { 4108, 0xf7ed }, { 4121, 0x7a7d }, { 4132, 0xf80f },
3792 { 4141, 0xeebe }, { 4153, 0x0fd5 }, { 4162, 0xbb5d }, { 4173, 0xfd9f },
3793 /* 0x6500 */
3794 { 4186, 0xf2db }, { 4197, 0x3bf9 }, { 4208, 0xfe7f }, { 4222, 0xebcc },
3795 { 4232, 0x876a }, { 4240, 0x73fa }, { 4251, 0x95fc }, { 4261, 0x9ffc },
3796 { 4273, 0x109f }, { 4280, 0xfaf7 }, { 4293, 0xddb7 }, { 4305, 0xbbcd },
3797 { 4316, 0xf87e }, { 4327, 0xeccd }, { 4337, 0xf366 }, { 4347, 0x3c3f },
3798 /* 0x6600 */
3799 { 4357, 0xffffd }, { 4372, 0xb03f }, { 4381, 0xe9f7 }, { 4393, 0x067e },
3800 { 4401, 0x96ae }, { 4410, 0xfe06 }, { 4419, 0xd576 }, { 4429, 0x5fd7 },
3801 { 4441, 0x3fd1 }, { 4451, 0xa3f3 }, { 4461, 0xc07 }, { 4470, 0x6fb7 },
3802 { 4482, 0x9fd1 }, { 4492, 0x7f44 }, { 4501, 0x7b59 }, { 4511, 0xd3dd },
3803 /* 0x6700 */
3804 { 4522, 0xaf3b }, { 4533, 0xa9bd }, { 4543, 0x7dcf }, { 4555, 0xff3a },
3805 { 4567, 0xfbe0 }, { 4577, 0xf6eb }, { 4589, 0xb401 }, { 4594, 0xffff },
3806 { 4610, 0x7afa }, { 4621, 0xb7bf }, { 4634, 0xc000 }, { 4636, 0x0ffd },
3807 { 4647, 0xff7f }, { 4662, 0xff1f }, { 4675, 0xfefc }, { 4688, 0x95ff },
3808 /* 0x6800 */
3809 { 4700, 0x0000 }, { 4700, 0xb5dc }, { 4710, 0xef63 }, { 4721, 0x3f3e },
3810 { 4732, 0xfb7f }, { 4746, 0x001b }, { 4750, 0xe800 }, { 4754, 0xfbf6 },
3811 { 4767, 0x9eef }, { 4779, 0xb8df }, { 4790, 0xff9f }, { 4804, 0x003f },
3812 { 4810, 0x7bd0 }, { 4819, 0xf5ff }, { 4833, 0xfdb }, { 4846, 0x3fff },
3813 /* 0x6900 */
3814 { 4860, 0xfdf0 }, { 4871, 0x00bf }, { 4878, 0x8420 }, { 4881, 0xbbbd },
3815 { 4893, 0xdf37 }, { 4905, 0xffde }, { 4919, 0xff6d }, { 4932, 0x0ff3 },
3816 { 4942, 0x604c }, { 4947, 0x5efb }, { 4959, 0xffffb }, { 4974, 0xfafb },
3817 { 4987, 0xfe5e }, { 4999, 0x0219 }, { 5003, 0x79f4 }, { 5013, 0xf9de },
3818 /* 0x6a00 */
3819 { 5025, 0xa7f7 }, { 5037, 0xebfa }, { 5049, 0x01eb }, { 5056, 0xff34 },
3820 { 5067, 0xebd3 }, { 5078, 0xef73 }, { 5090, 0xafd7 }, { 5102, 0xc040 },
3821 { 5105, 0x72bb }, { 5115, 0xdcff }, { 5128, 0xf17f }, { 5140, 0x2fd8 },
3822 { 5149, 0xb8ec }, { 5158, 0xfe0b }, { 5168, 0xdda3 }, { 5178, 0x1f0b },
3823 /* 0x6b00 */
3824 { 5186, 0x8f1d }, { 5195, 0x47cf }, { 5205, 0xb12b }, { 5213, 0xffde },
3825 { 5227, 0x7fee }, { 5240, 0xda73 }, { 5250, 0x24ff }, { 5260, 0xc4 },
3826 { 5268, 0xf75d }, { 5280, 0xcbf2 }, { 5290, 0xecfd }, { 5302, 0xb4ed },
3827 { 5312, 0xbf9f }, { 5325, 0x4ddd }, { 5335, 0x99dd }, { 5345, 0xfb8d },
3828 /* 0x6c00 */
3829 { 5356, 0xbb7f }, { 5369, 0xaf7b }, { 5381, 0xddfb }, { 5394, 0xc959 },
3830 { 5402, 0xfc4f }, { 5413, 0xfab5 }, { 5424, 0xafe3 }, { 5435, 0x6d5f },
3831 { 5446, 0xfffff }, { 5462, 0x3f7d }, { 5474, 0x7800 }, { 5478, 0xfdb },
3832 { 5492, 0xb6ff }, { 5505, 0x7eff }, { 5519, 0xfba }, { 5532, 0x022f },
3833 /* 0x6d00 */
3834 { 5538, 0xff9b }, { 5551, 0xfec7 }, { 5563, 0xffa5 }, { 5575, 0xffff },
3835 { 5591, 0x0007 }, { 5594, 0xc700 }, { 5599, 0xf7ff }, { 5614, 0xffff1 },
3836 { 5627, 0x7ffd }, { 5641, 0x01bf }, { 5649, 0xdc00 }, { 5654, 0xfdbc },
3837 { 5666, 0xbf5 }, { 5679, 0xfffff }, { 5695, 0xff7f }, { 5710, 0x3eff },
3838 /* 0x6e00 */
3839 { 5723, 0x0029 }, { 5726, 0xbe00 }, { 5732, 0xf9ff }, { 5746, 0xff7f },
3840 { 5761, 0x6efb }, { 5773, 0xfd7e }, { 5786, 0xcbf }, { 5799, 0x039e },
3841 { 5806, 0xe300 }, { 5811, 0xfbdd }, { 5824, 0xccff }, { 5836, 0xf6df },
3842 { 5849, 0xfffff }, { 5865, 0x117f }, { 5874, 0xf800 }, { 5879, 0xfbf6 },
3843 /* 0x6f00 */
3844 { 5892, 0xe7ef }, { 5905, 0xd73c }, { 5915, 0xfeef }, { 5929, 0xdfef },

```

```
3845 { 5943, 0xc00b }, { 5948, 0xedbf }, { 5961, 0xfedf }, { 5975, 0xfdcd },
3846 { 5987, 0x7bf5 }, { 5999, 0x40fd }, { 6007, 0xffff }, { 6023, 0xb75f },
3847 { 6035, 0xffdf }, { 6050, 0xf930 }, { 6058, 0xfbdf }, { 6072, 0xdc97 },
3848 /* 0x7000 */
3849 { 6082, 0xfef3 }, { 6095, 0xbff2 }, { 6107, 0x8fdf }, { 6119, 0xdfbf },
3850 { 6133, 0x177f }, { 6144, 0xede6 }, { 6155, 0x0f7f }, { 6166, 0x3553 },
3851 { 6174, 0x447c }, { 6181, 0x877e }, { 6191, 0xfa12 }, { 6199, 0x45bb },
3852 { 6208, 0xede0 }, { 6217, 0x779e }, { 6228, 0x8017 }, { 6233, 0xbfd9 },
3853 /* 0x7100 */
3854 { 6245, 0x7e55 }, { 6255, 0xde89 }, { 6264, 0xc16f }, { 6273, 0x0447 },
3855 { 6278, 0x7ade }, { 6289, 0xf75d }, { 6301, 0x57ff }, { 6314, 0x2905 },
3856 { 6319, 0x86f7 }, { 6329, 0xfe95 }, { 6340, 0x97b3 }, { 6350, 0xf32f },
3857 { 6361, 0xcfff }, { 6375, 0x9f75 }, { 6386, 0x71f7 }, { 6397, 0xfb17 },
3858 /* 0x7200 */
3859 { 6408, 0x34ee }, { 6417, 0xee19 }, { 6426, 0x37cc }, { 6435, 0xef61 },
3860 { 6445, 0x9fd6 }, { 6456, 0xef4c }, { 6466, 0xd68f }, { 6476, 0xfbdd },
3861 { 6489, 0x7b73 }, { 6500, 0x6def }, { 6512, 0xd7fe }, { 6525, 0xa431 },
3862 { 6531, 0xe7f7 }, { 6543, 0x97d7 }, { 6554, 0x0f5b }, { 6563, 0xfdf8 },
3863 /* 0x7300 */
3864 { 6575, 0x9d83 }, { 6583, 0x7bce }, { 6594, 0x22ec }, { 6601, 0xdcff },
3865 { 6614, 0x763d }, { 6624, 0xef87 }, { 6635, 0xdfe7 }, { 6648, 0xfded },
3866 { 6661, 0x4fff }, { 6674, 0xa0fc }, { 6682, 0x3b77 }, { 6693, 0xdbfc },
3867 { 6705, 0x3ded }, { 6716, 0x7fdc }, { 6728, 0x6fa9 }, { 6738, 0xf570 },
3868 /* 0x7400 */
3869 { 6747, 0x3ffb }, { 6760, 0x2c40 }, { 6764, 0xff7f }, { 6779, 0x847f },
3870 { 6788, 0xec57 }, { 6798, 0xdeb7 }, { 6810, 0xe69c }, { 6819, 0xf22f },
3871 { 6829, 0x0feb }, { 6839, 0xd5b5 }, { 6849, 0xafeb }, { 6861, 0xede7 },
3872 { 6873, 0x8c2f }, { 6881, 0xffff }, { 6893, 0x537f }, { 6904, 0xe8f0 },
3873 /* 0x7500 */
3874 { 6912, 0xb99d }, { 6922, 0xb5ff }, { 6935, 0xff66 }, { 6947, 0xe78f },
3875 { 6958, 0xd981 }, { 6965, 0xbe10 }, { 6972, 0x9c7c }, { 6981, 0xe3c1 },
3876 { 6989, 0x9cd1 }, { 6997, 0x2733 }, { 7005, 0x0cbc }, { 7012, 0xff6d },
3877 { 7025, 0xfcb7 }, { 7037, 0xefb7 }, { 7050, 0xadaf }, { 7059, 0xffff },
3878 /* 0x7600 */
3879 { 7075, 0xbf0b }, { 7085, 0xfe7b }, { 7098, 0xa3ff }, { 7110, 0x353f },
3880 { 7120, 0x13cc }, { 7127, 0x97cd }, { 7137, 0x7637 }, { 7147, 0xfb27 },
3881 { 7158, 0xcfd6 }, { 7169, 0x7e6c }, { 7179, 0xec50 }, { 7186, 0xed31 },
3882 { 7195, 0x677c }, { 7205, 0xfc1c }, { 7214, 0xf6fa }, { 7226, 0x5fbf },
3883 /* 0x7700 */
3884 { 7239, 0x0fba }, { 7248, 0xae2f }, { 7258, 0xa3ad }, { 7267, 0x7ffe },
3885 { 7281, 0xfc0f }, { 7291, 0xde74 }, { 7301, 0xffef }, { 7316, 0xf200 },
3886 { 7321, 0xfbbf }, { 7335, 0xfea2 }, { 7345, 0x3daf }, { 7356, 0xbccf },
3887 { 7369, 0xf694 }, { 7378, 0x5fb9 }, { 7389, 0xf3ad }, { 7400, 0x3f8f },
3888 /* 0x7800 */
3889 { 7411, 0xf26c }, { 7420, 0xa01f }, { 7427, 0xffef }, { 7442, 0x01bf },
3890 { 7450, 0x7728 }, { 7458, 0x7005 }, { 7463, 0xff35 }, { 7475, 0xda03 },
3891 { 7482, 0xd2f9 }, { 7492, 0xc7fa }, { 7503, 0x3fbf }, { 7516, 0x5c1d },
3892 { 7524, 0xff3a }, { 7536, 0xec33 }, { 7545, 0xb7af }, { 7557, 0xfe9c },
3893 /* 0x7900 */
3894 { 7568, 0x5236 }, { 7575, 0x7a9f }, { 7586, 0xbffa }, { 7599, 0xe722 },
3895 { 7607, 0x9ff7 }, { 7620, 0xfcff }, { 7634, 0x2fbb }, { 7645, 0xb61d },
3896 { 7654, 0xed06 }, { 7662, 0x1dfd }, { 7673, 0x7dd7 }, { 7685, 0xefdf },
3897 { 7699, 0xeb23 }, { 7708, 0xf166 }, { 7717, 0x7ed9 }, { 7728, 0x0dc0 },
3898 /* 0x7a00 */
3899 { 7733, 0x3d3d }, { 7743, 0xdfbf }, { 7757, 0xc945 }, { 7764, 0xba83 },
3900 { 7772, 0x7dd1 }, { 7782, 0x9dd0 }, { 7790, 0x7b87 }, { 7800, 0xcf73 },
3901 { 7811, 0x9ff3 }, { 7823, 0xc3f5 }, { 7833, 0xdf0d }, { 7843, 0xc5fe },
3902 { 7854, 0x0cb3 }, { 7861, 0x8302 }, { 7865, 0xe879 }, { 7874, 0xaec0 },
3903 /* 0x7b00 */
3904 { 7881, 0xc773 }, { 7891, 0x6f0f }, { 7901, 0xfd7d }, { 7914, 0x093f },
3905 { 7922, 0xffff }, { 7935, 0x0157 }, { 7941, 0x62fb }, { 7951, 0x01ff },
3906 { 7960, 0xfdb4 }, { 7971, 0x3bf3 }, { 7982, 0xb013 }, { 7988, 0x43b2 },
3907 { 7995, 0x5ed3 }, { 8005, 0xff30 }, { 8015, 0x0fff }, { 8027, 0xeb9f },
3908 /* 0x7c00 */
3909 { 8039, 0xfeef }, { 8053, 0xf203 }, { 8060, 0x3fef }, { 8073, 0xfb89 },
3910 { 8083, 0x37a9 }, { 8092, 0x9e99 }, { 8101, 0xdef9 }, { 8113, 0xa72c },
3911 { 8121, 0x3733 }, { 8130, 0xc1f6 }, { 8139, 0x812e }, { 8145, 0xfe3e },
3912 { 8157, 0x5d20 }, { 8163, 0xf2f7 }, { 8175, 0xd585 }, { 8183, 0x69d7 },
3913 /* 0x7d00 */
3914 { 8193, 0xffff }, { 8209, 0xffff }, { 8225, 0xdb07 }, { 8234, 0xff6f },
3915 { 8248, 0xc4ff }, { 8259, 0xd97f }, { 8271, 0xfecf }, { 8283, 0xbe0f },
3916 { 8293, 0xf17b }, { 8304, 0xf05e }, { 8313, 0xf6cf }, { 8325, 0xffb7 },
3917 { 8339, 0x5ef7 }, { 8351, 0xef84 }, { 8360, 0xd7cb }, { 8371, 0x0edf },
3918 /* 0x7e00 */
3919 { 8381, 0xff08 }, { 8390, 0xfcff }, { 8404, 0xee3f }, { 8416, 0xffff },
3920 { 8432, 0x13ff }, { 8443, 0xd7ff }, { 8457, 0xaf0f }, { 8467, 0x7ffd },
3921 { 8481, 0xbdc7 }, { 8492, 0x1ffa }, { 8503, 0x0000 }, { 8503, 0x0000 },
3922 { 8503, 0x0000 }, { 8503, 0x0000 }, { 8503, 0x0000 }, { 8503, 0x0000 },
3923 /* 0x7f00 */
3924 { 8503, 0x0000 }, { 8503, 0x0000 }, { 8503, 0x0000 }, { 8503, 0xe740 },
3925 { 8510, 0xbd38 }, { 8519, 0xf933 }, { 8529, 0x7feb }, { 8542, 0xfeed },
3926 { 8555, 0x7fe8 }, { 8566, 0x7c76 }, { 8576, 0xb3f7 }, { 8588, 0xffef },
3927 { 8603, 0xfeaf }, { 8616, 0xd8b7 }, { 8626, 0xff6f }, { 8640, 0xfbbf },
3928 /* 0x8000 */
3929 { 8654, 0xf8fb }, { 8666, 0xdbf7 }, { 8679, 0x1752 }, { 8686, 0xe2f9 },
3930 { 8696, 0x85c8 }, { 8702, 0x7547 }, { 8711, 0x9090 }, { 8715, 0xe3ef },
3931 { 8727, 0x9ef4 }, { 8737, 0x3f6d }, { 8748, 0xee2e }, { 8758, 0x0536 },
```

```

3932 { 8764, 0xf7bc }, { 8776, 0x7ff3 }, { 8789, 0xa07b }, { 8797, 0x7f3f },
3933 /* 0x8100 */
3934 { 8810, 0x0567 }, { 8817, 0xeb60 }, { 8825, 0xbabe }, { 8836, 0x6601 },
3935 { 8841, 0xfcd8 }, { 8851, 0x583f }, { 8860, 0xcaf7 }, { 8871, 0x87df },
3936 { 8882, 0xbfcd }, { 8894, 0xffa0 }, { 8904, 0x5bcd }, { 8914, 0xfebf },
3937 { 8928, 0xb6fd }, { 8940, 0xefa7 }, { 8952, 0x77ef }, { 8965, 0xdf9c },
3938 /* 0x8200 */
3939 { 8976, 0x3fb7 }, { 8988, 0xf877 }, { 8999, 0x9d27 }, { 9008, 0xb7fc },
3940 { 9020, 0xcab5 }, { 9029, 0xdfef }, { 9043, 0xfb5a }, { 9054, 0xflb6 },
3941 { 9064, 0xec39 }, { 9073, 0xef1f }, { 9085, 0xfbbf }, { 9099, 0x7ffb },
3942 { 9113, 0x000d }, { 9116, 0xdafe }, { 9128, 0xbdfb }, { 9141, 0x4e7f },
3943 /* 0x8300 */
3944 { 9152, 0x33ff }, { 9164, 0x5ac0 }, { 9170, 0xbff5 }, { 9183, 0x9ffe },
3945 { 9196, 0xffbf }, { 9211, 0x005f }, { 9217, 0x0000 }, { 9217, 0xdfd8 },
3946 { 9229, 0xffca }, { 9241, 0x6ffd }, { 9254, 0xcffd }, { 9267, 0xa001 },
3947 { 9270, 0xdfff }, { 9285, 0xfb2f }, { 9297, 0xdfbf }, { 9311, 0xff7f },
3948 /* 0x8400 */
3949 { 9326, 0xfeda }, { 9338, 0x080f }, { 9343, 0xba08 }, { 9349, 0xbfff },
3950 { 9364, 0x7afd }, { 9376, 0xead7 }, { 9388, 0xfbeb }, { 9401, 0x67f9 },
3951 { 9412, 0xe044 }, { 9417, 0xff93 }, { 9429, 0xdf97 }, { 9441, 0x9f57 },
3952 { 9452, 0xfef7 }, { 9466, 0x08df }, { 9474, 0xdf80 }, { 9482, 0xfedf },
3953 /* 0x8500 */
3954 { 9496, 0xffc5 }, { 9508, 0xf7fe }, { 9522, 0xfffb }, { 9537, 0x6803 },
3955 { 9542, 0x67fb }, { 9554, 0x6bfa }, { 9565, 0x7fff }, { 9580, 0x5fe2 },
3956 { 9590, 0xffff }, { 9606, 0xff73 }, { 9619, 0x87df }, { 9630, 0xe7fb },
3957 { 9643, 0xebfd }, { 9656, 0xf7a7 }, { 9668, 0xbf7e }, { 9681, 0xefc7 },
3958 /* 0x8600 */
3959 { 9693, 0x1ef3 }, { 9703, 0xdf82 }, { 9712, 0x76ff }, { 9725, 0xdf7e },
3960 { 9738, 0x79c9 }, { 9747, 0xda7d }, { 9758, 0xefbe }, { 9771, 0x1e9b },
3961 { 9780, 0x7ce0 }, { 9788, 0x77fb }, { 9801, 0x87be }, { 9811, 0xffff },
3962 { 9826, 0xlbff }, { 9838, 0xffdb }, { 9852, 0x3f5c }, { 9862, 0x4fe0 },
3963 /* 0x8700 */
3964 { 9870, 0x7fff }, { 9885, 0x5f0e }, { 9894, 0x77ff }, { 9908, 0xddbf },
3965 { 9921, 0xf04f }, { 9930, 0xffff }, { 9946, 0xffff }, { 9962, 0x0ff8 },
3966 { 9971, 0xa3be }, { 9981, 0xfddf }, { 9995, 0xfclc }, { 10004, 0xffffd },
3967 { 10019, 0x1f7d }, { 10030, 0xfb9e }, { 10042, 0xbddf }, { 10056, 0xdcdc },
3968 /* 0x8800 */
3969 { 10067, 0x3f6f }, { 10079, 0xbafb }, { 10091, 0xdf7f }, { 10105, 0xfbef },
3970 { 10119, 0x7dlb }, { 10129, 0x2eec }, { 10138, 0xaf8e }, { 10148, 0xf2f7 },
3971 { 10160, 0x7b0f }, { 10170, 0xcfee }, { 10182, 0x1d96 }, { 10190, 0x77c6 },
3972 { 10200, 0x7e07 }, { 10209, 0xffff5 }, { 10223, 0xd982 }, { 10230, 0x7fdf },
3973 /* 0x8900 */
3974 { 10244, 0x5ee6 }, { 10254, 0xc7ff }, { 10267, 0xfeee }, { 10280, 0x79ef },
3975 { 10292, 0x9a56 }, { 10300, 0xffcf }, { 10314, 0xfe5f }, { 10327, 0xde5e },
3976 { 10338, 0x896e }, { 10346, 0xf9e8 }, { 10356, 0xf45e }, { 10366, 0xe6c4 },
3977 { 10374, 0x0001 }, { 10375, 0xbe7c }, { 10386, 0x3b7f }, { 10398, 0xdddf },
3978 /* 0x8a00 */
3979 { 10411, 0xd59d }, { 10421, 0xe9ef }, { 10433, 0x34ac }, { 10440, 0xde53 },
3980 { 10450, 0xf573 }, { 10461, 0x4bf7 }, { 10472, 0x7b4f }, { 10483, 0x9eff },
3981 { 10496, 0xb8fe }, { 10507, 0x476e }, { 10516, 0x0dfb }, { 10526, 0xff45 },
3982 { 10537, 0xabfd }, { 10549, 0xfbfef }, { 10563, 0xe9d7 }, { 10574, 0xddff },
3983 /* 0x8b00 */
3984 { 10588, 0xedf7 }, { 10601, 0x7fff }, { 10616, 0xddfd }, { 10629, 0x7eeb },
3985 { 10641, 0xcfe7 }, { 10653, 0xb7ff }, { 10667, 0xbde9 }, { 10678, 0xef91 },
3986 { 10688, 0x5d75 }, { 10698, 0xd77c }, { 10709, 0x0000 }, { 10709, 0x0000 },
3987 { 10709, 0x0000 }, { 10709, 0x0000 }, { 10709, 0x0000 }, { 10709, 0x0000 },
3988 /* 0x8c00 */
3989 { 10709, 0x0000 }, { 10709, 0x0000 }, { 10709, 0x0000 }, { 10709, 0xfa80 },
3990 { 10716, 0xffee }, { 10730, 0xb4f1 }, { 10739, 0xbf76 }, { 10751, 0x2fef },
3991 { 10763, 0xb677 }, { 10774, 0x77bf }, { 10787, 0x9fbf }, { 10800, 0xffffd },
3992 { 10815, 0x95bf }, { 10826, 0xf6ae }, { 10837, 0x75ff }, { 10850, 0x7f3b },
3993 /* 0x8d00 */
3994 { 10862, 0xa7f5 }, { 10873, 0x0af9 }, { 10881, 0x0000 }, { 10881, 0x0000 },
3995 { 10881, 0x0000 }, { 10881, 0x0000 }, { 10881, 0xfbd0 }, { 10891, 0x2bdd },
3996 { 10901, 0xf633 }, { 10911, 0x9a7f }, { 10922, 0xfdad }, { 10934, 0xd6fc },
3997 { 10945, 0xf9e6 }, { 10956, 0xbfeb }, { 10969, 0xdfdf }, { 10983, 0xf41f },
3998 /* 0x8e00 */
3999 { 10993, 0xa6fd }, { 11004, 0xffff }, { 11020, 0x4aff }, { 11031, 0xf37b },
4000 { 11043, 0x7fb7 }, { 11056, 0xfef9 }, { 11069, 0xb6ff }, { 11082, 0x1d5c },
4001 { 11090, 0x7ff6 }, { 11103, 0xe5ff }, { 11116, 0x1f7b }, { 11127, 0x2404 },
4002 { 11130, 0xbe05 }, { 11138, 0xf99e }, { 11149, 0xdb3e }, { 11160, 0xdff2 },
4003 /* 0x8f00 */
4004 { 11172, 0x6fef }, { 11185, 0xfdff }, { 11200, 0xd679 }, { 11210, 0xcbfc },
4005 { 11221, 0xebfd }, { 11234, 0xffff }, { 11249, 0x001f }, { 11254, 0x0000 },
4006 { 11254, 0x0000 }, { 11254, 0x9800 }, { 11257, 0xe148 }, { 11263, 0x8017 },
4007 { 11268, 0x6a74 }, { 11276, 0x00fe }, { 11283, 0x6d7f }, { 11295, 0xfdf1 },
4008 /* 0x9000 */
4009 { 11307, 0xb87f }, { 11318, 0xfef3 }, { 11331, 0xe01f }, { 11339, 0xf176 },
4010 { 11349, 0xee96 }, { 11359, 0x7b3f }, { 11371, 0xeb8d }, { 11381, 0xffffd },
4011 { 11396, 0xadff }, { 11409, 0xcbb3 }, { 11419, 0x84ef }, { 11428, 0xe17f },
4012 { 11439, 0x4daa }, { 11447, 0xbff0 }, { 11458, 0xbf3f }, { 11471, 0xfe3f },
4013 /* 0x9100 */
4014 { 11484, 0xebff }, { 11498, 0xffd7 }, { 11512, 0xffdf }, { 11527, 0xcf7f },
4015 { 11540, 0xffffb }, { 11555, 0x85ed }, { 11564, 0xd73f }, { 11576, 0x07bc },
4016 { 11584, 0xaef }, { 11597, 0xfe0f }, { 11608, 0xfdaf }, { 11621, 0x76bf },
4017 { 11633, 0xfaef }, { 11646, 0x37bb }, { 11657, 0x7fdc }, { 11669, 0xa3ba },
4018 /* 0x9200 */

```



```

4019 { 11678, 0xb6ff }, { 11691, 0x56f7 }, { 11702, 0x60f8 }, { 11709, 0xe7df },
4020 { 11722, 0xff61 }, { 11733, 0x4cdf }, { 11743, 0xb0fb }, { 11753, 0xff45 },
4021 { 11764, 0x7ded }, { 11776, 0x3ffa }, { 11788, 0x1fff }, { 11801, 0x18fc },
4022 { 11809, 0xffff }, { 11825, 0xe3af }, { 11836, 0xc7d3 }, { 11846, 0xdf83 },
4023 /* 0x9300 */
4024 { 11856, 0xfb57 }, { 11868, 0xef7d }, { 11881, 0xffff }, { 11896, 0x1378 },
4025 { 11903, 0xfec0 }, { 11912, 0x5ff7 }, { 11925, 0x34bb }, { 11934, 0x5ee3 },
4026 { 11944, 0xf70d }, { 11954, 0xef66 }, { 11967, 0xd7fe }, { 11980, 0x00bf },
4027 { 11987, 0xf59d }, { 11998, 0xf7f7 }, { 12012, 0x51de }, { 12021, 0xffe0 },
4028 /* 0x9400 */
4029 { 12032, 0xfec9 }, { 12043, 0x037f }, { 12052, 0x5f01 }, { 12059, 0xbfef },
4030 { 12073, 0x9ff1 }, { 12084, 0x60a7 }, { 12091, 0xef1d }, { 12102, 0xf1ff },
4031 { 12115, 0x000f }, { 12119, 0x0000 }, { 12119, 0x0000 }, { 12119, 0x0000 },
4032 { 12119, 0x0000 }, { 12119, 0x0000 }, { 12119, 0x0000 }, { 12119, 0x0000 },
4033 /* 0x9500 */
4034 { 12119, 0x0000 }, { 12119, 0x0000 }, { 12119, 0x0000 }, { 12119, 0x0000 },
4035 { 12119, 0x0000 }, { 12119, 0x0000 }, { 12119, 0x0000 }, { 12119, 0x3c80 },
4036 { 12124, 0xfb4d }, { 12135, 0xd91f }, { 12145, 0x7b3a }, { 12155, 0xffe3 },
4037 { 12167, 0x3fe9 }, { 12178, 0xdc7f }, { 12190, 0x003f }, { 12196, 0x0000 },
4038 /* 0x9600 */
4039 { 12196, 0x0000 }, { 12196, 0x5000 }, { 12198, 0xf51f }, { 12209, 0xbe07 },
4040 { 12218, 0xfc1d }, { 12228, 0xf91b }, { 12238, 0xbc1e }, { 12247, 0x71ff },
4041 { 12259, 0x6ff9 }, { 12271, 0x5bbe }, { 12282, 0x5796 }, { 12291, 0x9b1b },
4042 { 12300, 0x7fff }, { 12315, 0xffff }, { 12329, 0x872e }, { 12337, 0xaf7e },
4043 /* 0x9700 */
4044 { 12349, 0xebf5 }, { 12361, 0xf34f }, { 12372, 0xdffd }, { 12386, 0xe725 },
4045 { 12395, 0x0bdc }, { 12403, 0x5d44 }, { 12410, 0x5747 }, { 12419, 0xfddd },
4046 { 12432, 0xed3f }, { 12444, 0x7790 }, { 12452, 0x7d7f }, { 12465, 0x8ac8 },
4047 { 12471, 0xfafa }, { 12483, 0xf3f9 }, { 12495, 0x202a }, { 12499, 0xef4b },
4048 /* 0x9800 */
4049 { 12510, 0xf5ff }, { 12524, 0x79cf }, { 12535, 0xabd3 }, { 12545, 0x0ba5 },
4050 { 12552, 0xf77a }, { 12564, 0xfb8f }, { 12576, 0x8ebd }, { 12586, 0x001f },
4051 { 12591, 0x0000 }, { 12591, 0x0000 }, { 12591, 0xf300 }, { 12597, 0xfd4e },
4052 { 12608, 0x1a57 }, { 12616, 0x8800 }, { 12618, 0xaeac }, { 12627, 0x7654 },
4053 /* 0x9900 */
4054 { 12635, 0x17ad }, { 12644, 0xcdff }, { 12657, 0xffb2 }, { 12669, 0xf42f },
4055 { 12679, 0x5baa }, { 12688, 0xdbff }, { 12702, 0x0002 }, { 12703, 0x0000 },
4056 { 12703, 0x0000 }, { 12703, 0x73c0 }, { 12710, 0xf9ea }, { 12721, 0x2e3f },
4057 { 12731, 0xfa8e }, { 12741, 0xbbff }, { 12755, 0x76bc }, { 12765, 0xffd3 },
4058 /* 0x9a00 */
4059 { 12778, 0xeefe }, { 12791, 0x7e72 }, { 12801, 0x7ebd }, { 12813, 0xe7f7 },
4060 { 12826, 0xf77f }, { 12840, 0xcefd }, { 12852, 0x0ff5 }, { 12862, 0x0000 },
4061 { 12862, 0x0000 }, { 12862, 0x0000 }, { 12862, 0xa900 }, { 12866, 0xdb9b },
4062 { 12877, 0xa4c7 }, { 12885, 0x917f }, { 12895, 0xf8ca }, { 12904, 0x7ece },
4063 /* 0x9b00 */
4064 { 12915, 0x7d7a }, { 12926, 0xc7e7 }, { 12937, 0xcbbd }, { 12948, 0xdcae },
4065 { 12958, 0xfd7e }, { 12971, 0x8f76 }, { 12981, 0x91d3 }, { 12989, 0x7cf3 },
4066 { 13000, 0x01e5 }, { 13006, 0x4c2f }, { 13014, 0xed77 }, { 13026, 0xa360 },
4067 { 13032, 0x07db }, { 13041, 0x5ef8 }, { 13051, 0x1df7 }, { 13062, 0x2181 },
4068 /* 0x9c00 */
4069 { 13066, 0x6be0 }, { 13074, 0x309c }, { 13080, 0x3b3a }, { 13089, 0xfade },
4070 { 13101, 0x7f53 }, { 13112, 0xc3f5 }, { 13122, 0x61cd }, { 13130, 0x07ba },
4071 { 13138, 0x0000 }, { 13138, 0x0000 }, { 13138, 0x0000 }, { 13138, 0x0000 },
4072 { 13138, 0x0000 }, { 13138, 0x0000 }, { 13138, 0x26e0 }, { 13144, 0xbefe },
4073 /* 0x9d00 */
4074 { 13157, 0x03f9 }, { 13165, 0xebb5 }, { 13176, 0xe36d }, { 13186, 0xe9cb },
4075 { 13196, 0x9c2f }, { 13205, 0xbfde }, { 13218, 0x9f83 }, { 13227, 0xabbf },
4076 { 13239, 0x1fff }, { 13251, 0xffd5 }, { 13264, 0xb7df }, { 13277, 0xdffe },
4077 { 13291, 0xfdae }, { 13303, 0xffef }, { 13318, 0xfb7e }, { 13331, 0xfffd },
4078 /* 0x9e00 */
4079 { 13345, 0xaaff }, { 13357, 0x6ebf }, { 13369, 0x0000 }, { 13369, 0x0000 },
4080 { 13369, 0x0000 }, { 13369, 0x0000 }, { 13369, 0x0000 }, { 13369, 0xb620 },
4081 { 13375, 0x7fcd }, { 13387, 0xbe9e }, { 13398, 0x62b3 }, { 13406, 0x58f1 },
4082 { 13414, 0xf10d }, { 13422, 0xfd7b }, { 13435, 0xe9f1 }, { 13445, 0xbefd },
4083 /* 0x9f00 */
4084 { 13458, 0xc6c3 }, { 13466, 0x5f6d }, { 13477, 0xff3d }, { 13490, 0x69ff },
4085 { 13502, 0xffcf }, { 13516, 0xfb44 }, { 13528, 0xdcfb }, { 13540, 0x4ff7 },
4086 { 13552, 0x2000 }, { 13553, 0x1137 }, { 13560, 0x0015 },
4087 };
4088 static const Summary16 big5_uni2indx_pagefa[1] = {
4089 /* 0xfa00 */
4090 { 13563, 0x3000 },
4091 };
4092 static const Summary16 big5_uni2indx_pagefe[23] = {
4093 /* 0xfe00 */
4094 { 13565, 0x0000 }, { 13565, 0x0000 }, { 13565, 0x0000 }, { 13565, 0xffff },
4095 { 13580, 0xfe1f }, { 13592, 0xfef5 }, { 13605, 0x0e7f }, { 13615, 0x0000 },
4096 { 13615, 0x0000 }, { 13615, 0x0000 }, { 13615, 0x0000 }, { 13615, 0x0000 },
4097 { 13615, 0x0000 }, { 13615, 0x0000 }, { 13615, 0x0000 }, { 13615, 0x0000 },
4098 /* 0xff00 */
4099 { 13615, 0xff7a }, { 13628, 0xffff }, { 13644, 0xffff }, { 13660, 0x97ff },
4100 { 13673, 0xfffe }, { 13688, 0x3fff }, { 13702, 0x0010 },
4101 };
4102
4103 static int
4104 big5_wctomb (conv_t conv, unsigned char *r, ucs4_t wc, int n)
4105 {

```

```

4106 if (n >= 2) {
4107     const Summary16 *summary = NULL;
4108     if (wc < 0x0100)
4109         summary = &big5_uni2indx_page00[(wc>>4)];
4110     else if (wc >= 0x0200 && wc < 0x0460)
4111         summary = &big5_uni2indx_page02[(wc>>4)-0x020];
4112     else if (wc >= 0x2000 && wc < 0x22c0)
4113         summary = &big5_uni2indx_page20[(wc>>4)-0x200];
4114     else if (wc >= 0x2400 && wc < 0x2650)
4115         summary = &big5_uni2indx_page24[(wc>>4)-0x240];
4116     else if (wc >= 0x3000 && wc < 0x33e0)
4117         summary = &big5_uni2indx_page30[(wc>>4)-0x300];
4118     else if (wc >= 0x4e00 && wc < 0x9fb0)
4119         summary = &big5_uni2indx_page4e[(wc>>4)-0x4e0];
4120     else if (wc >= 0xfa00 && wc < 0xfa10)
4121         summary = &big5_uni2indx_pagefa[(wc>>4)-0xfa0];
4122     else if (wc >= 0xfe00 && wc < 0xff70)
4123         summary = &big5_uni2indx_pagefe[(wc>>4)-0xfe0];
4124     if (summary) {
4125         unsigned short used = summary->used;
4126         unsigned int i = wc & 0x0f;
4127         if (used & ((unsigned short) 1 << i)) {
4128             unsigned short c;
4129             /* Keep in 'used' only the bits 0..i-1. */
4130             used &= ((unsigned short) 1 << i) - 1;
4131             /* Add 'summary->indx' and the number of bits set in 'used'. */
4132             used = (used & 0x5555) + ((used & 0xaaaa) >> 1);
4133             used = (used & 0x3333) + ((used & 0xcccc) >> 2);
4134             used = (used & 0x0f0f) + ((used & 0xf0f0) >> 4);
4135             used = (used & 0x00ff) + (used >> 8);
4136             c = big5_2charset[summary->indx + used];
4137             r[0] = (c >> 8); r[1] = (c & 0xff);
4138             return 2;
4139         }
4140     }
4141     return RET_ILSEQ;
4142 }
4143 return RET_TOOSMALL;
4144 }
4145 #endif /* NEED_TOMB */

```

32.202 big5_emacs.h

```

1 /* $XFree86: xc/lib/X11/lcUniConv/big5_emacs.h,v 1.1 2000/11/28 18:50:06 dawes Exp $ */
2
3 /*
4 * BIG5-0 and BIG5-1
5 */
6
7 /*
8 BIG5 with its 13494 characters doesn't fit in a single 94x94 or 96x96
9 block. Therefore Emacs/Mule developers, in a typically Japanese way of
10 thinking, have developed an alternative encoding of BIG5 in two 94x94
11 planes, very similar to the SHIFT_JIS encoding for JISX0208.
12
13 Conversion between BIG5 codes (s1,s2) and BIG5-0 codes (c1,c2):
14 Example. (s1,s2) = 0xA140, (c1,c2) = 0x2121.
15 0xA1 <= s1 <= 0xC7, 0x40 <= s2 <= 0x7E || 0xA1 <= s2 <= 0xFE,
16 0x21 <= c1 <= 0x62, 0x21 <= c2 <= 0x7E.
17 Invariant:
18 157*(s1-0xA1) + (s2 < 0x80 ? s2-0x40 : s2-0x62)
19 = 94*(c1-0x21)+(c2-0x21)
20 Conversion (s1,s2) -> (c1,c2):
21 t := 157*(s1-0xA1) + (s2 < 0x80 ? s2-0x40 : s2-0x62)
22 c1 := (t div 94) + 0x21
23 c2 := (t mod 94) + 0x21
24 Conversion (c1,c2) -> (s1,s2):
25 t := 94*(c1-0x21)+(c2-0x21)
26 t2 := t mod 157
27 s1 := (t div 157) + 0xA1
28 s2 := (t2 < 0x3F ? t2+0x40 : t2+0x62)
29
30 Conversion between BIG5 codes (s1,s2) and BIG5-1 codes (c1,c2):
31 Example. (s1,s2) = 0xC940, (c1,c2) = 0x2121.
32 0xC9 <= s1 <= 0xF9, 0x40 <= s2 <= 0x7E || 0xA1 <= s2 <= 0xFE,
33 0x21 <= c1 <= 0x72, 0x21 <= c2 <= 0x7E.
34 Invariant:
35 157*(s1-0xC9) + (s2 < 0x80 ? s2-0x40 : s2-0x62)
36 = 94*(c1-0x21)+(c2-0x21)
37 Conversion (s1,s2) -> (c1,c2):
38 t := 157*(s1-0xC9) + (s2 < 0x80 ? s2-0x40 : s2-0x62)
39 c1 := (t div 94) + 0x21
40 c2 := (t mod 94) + 0x21
41 Conversion (c1,c2) -> (s1,s2):
42 t := 94*(c1-0x21)+(c2-0x21)

```

```

43 t2 := t mod 157
44 s1 := (t div 157) + 0xC9
45 s2 := (t2 < 0x3F ? t2+0x40 : t2+0x62)
46 */
47
48 static int
49 big5_0_mbtowc (conv_t conv, ucs4_t *pwc, const unsigned char *s, int n)
50 {
51     unsigned char c1 = s[0];
52     if (c1 >= 0x21 && c1 <= 0x62) {
53         if (n >= 2) {
54             unsigned char c2 = s[1];
55             if (c2 >= 0x21 && c2 <= 0x7e) {
56                 unsigned int i = 94 * (c1 - 0x21) + (c2 - 0x21);
57                 if (0) {
58                     /* Unoptimized. */
59                     unsigned char buf[2];
60                     buf[0] = (i / 157) + 0xa1;
61                     i = i % 157;
62                     buf[1] = i + (i < 0x3f ? 0x40 : 0x62);
63                     return big5_mbtowc(conv,pwc,buf,2);
64                 } else {
65                     /* Inline the implementation of big5_mbtowc. */
66                     if (i < 6121) {
67                         unsigned short wc = big5_2uni_pagea1[i];
68                         if (wc != 0xffffd) {
69                             *pwc = (ucs4_t) wc;
70                             return 2;
71                         }
72                     }
73                 }
74             }
75             return RET_ILSEQ;
76         }
77         return RET_TOOFEW(0);
78     }
79     return RET_ILSEQ;
80 }
81
82 static int
83 big5_1_mbtowc (conv_t conv, ucs4_t *pwc, const unsigned char *s, int n)
84 {
85     unsigned char c1 = s[0];
86     if (c1 >= 0x21 && c1 <= 0x72) {
87         if (n >= 2) {
88             unsigned char c2 = s[1];
89             if (c2 >= 0x21 && c2 <= 0x7e) {
90                 unsigned int i = 94 * (c1 - 0x21) + (c2 - 0x21);
91                 if (0) {
92                     /* Unoptimized. */
93                     unsigned char buf[2];
94                     buf[0] = (i / 157) + 0xc9;
95                     i = i % 157;
96                     buf[1] = i + (i < 0x3f ? 0x40 : 0x62);
97                     return big5_mbtowc(conv,pwc,buf,2);
98                 } else {
99                     /* Inline the implementation of big5_mbtowc. */
100                    if (i < 7652) {
101                        unsigned short wc = big5_2uni_pagec9[i];
102                        if (wc != 0xffffd) {
103                            *pwc = (ucs4_t) wc;
104                            return 2;
105                        }
106                    }
107                }
108            }
109            return RET_ILSEQ;
110        }
111        return RET_TOOFEW(0);
112    }
113    return RET_ILSEQ;
114 }
115
116 static int
117 big5_0_wctomb (conv_t conv, unsigned char *r, ucs4_t wc, int n)
118 {
119     if (n >= 2) {
120         unsigned char buf[2];
121         int ret = big5_wctomb(conv,buf,wc,2);
122         if (ret != RET_ILSEQ) {
123             unsigned char s1, s2;
124             if (ret != 2) abort();
125             s1 = buf[0];
126             s2 = buf[1];
127             if (!(s1 >= 0xa1)) abort();
128             if (!(s2 >= 0x40 && s2 <= 0x7e) || (s2 >= 0xa1 && s2 <= 0xfe)) abort();
129             if (s1 < 0xc9) {

```

```

130     unsigned int t = 157 * (s1 - 0xa1) + s2 - (s2 < 0x80 ? 0x40 : 0x62);
131     r[0] = (t / 94) + 0x21;
132     r[1] = (t % 94) + 0x21;
133     return 2;
134 }
135 }
136 return RET_ILSEQ;
137 }
138 return RET_TOOSMALL;
139 }
140
141 static int
142 big5_l_wctomb (conv_t conv, unsigned char *r, ucs4_t wc, int n)
143 {
144     if (n >= 2) {
145         unsigned char buf[2];
146         int ret = big5_wctomb(conv, buf, wc, 2);
147         if (ret != RET_ILSEQ) {
148             unsigned char s1, s2;
149             if (ret != 2) abort();
150             s1 = buf[0];
151             s2 = buf[1];
152             if (!(s1 <= 0xf9)) abort();
153             if (!(s2 >= 0x40 && s2 <= 0x7e) || (s2 >= 0xa1 && s2 <= 0xfe)) abort();
154             if (s1 >= 0xc9) {
155                 unsigned int t = 157 * (s1 - 0xc9) + s2 - (s2 < 0x80 ? 0x40 : 0x62);
156                 r[0] = (t / 94) + 0x21;
157                 r[1] = (t % 94) + 0x21;
158                 return 2;
159             }
160         }
161         return RET_ILSEQ;
162     }
163     return RET_TOOSMALL;
164 }

```

32.203 cp1133.h

```

1 /* $XFree86: xc/lib/X11/lcUniConv/cp1133.h,v 1.3 2000/11/29 17:40:28 dawes Exp $ */
2
3 /*
4 * IBM-CP1133
5 */
6
7 static const unsigned short cp1133_2uni_1[64] = {
8     /* 0xa0 */
9     0x00a0, 0x0e81, 0x0e82, 0x0e84, 0x0e87, 0x0e88, 0x0eaa, 0x0e8a,
10    0x0e8d, 0x0e94, 0x0e95, 0x0e96, 0x0e97, 0x0e99, 0x0e9a, 0x0e9b,
11    /* 0xb0 */
12    0x0e9c, 0x0e9d, 0x0e9e, 0x0e9f, 0x0ea1, 0x0ea2, 0x0ea3, 0x0ea5,
13    0x0ea7, 0x0eab, 0x0ead, 0x0eae, 0xfffd, 0xfffd, 0xfffd, 0xeaf,
14    /* 0xc0 */
15    0x0eb0, 0x0eb2, 0x0eb3, 0x0eb4, 0x0eb5, 0x0eb6, 0x0eb7, 0x0eb8,
16    0x0eb9, 0x0ebc, 0x0eb1, 0xebb, 0x0ebd, 0xfffd, 0xfffd, 0xfffd,
17    /* 0xd0 */
18    0x0ec0, 0x0ec1, 0x0ec2, 0x0ec3, 0x0ec4, 0x0ec8, 0x0ec9, 0x0eca,
19    0x0ecb, 0x0ecc, 0x0ecd, 0x0ec6, 0xfffd, 0x0edc, 0x0edd, 0x20ad,
20 };
21 static const unsigned short cp1133_2uni_2[16] = {
22     /* 0xf0 */
23     0x0ed0, 0x0ed1, 0x0ed2, 0x0ed3, 0x0ed4, 0x0ed5, 0x0ed6, 0x0ed7,
24     0x0ed8, 0x0ed9, 0xfffd, 0xfffd, 0x00a2, 0x00ac, 0x00a6, 0xfffd,
25 };
26
27 static int
28 cp1133_mbtowc (conv_t conv, ucs4_t *pwc, const unsigned char *s, int n)
29 {
30     unsigned char c = *s;
31     if (c < 0xa0) {
32         *pwc = (ucs4_t) c;
33         return 1;
34     }
35     else if (c < 0xe0) {
36         unsigned short wc = cp1133_2uni_1[c-0xa0];
37         if (wc != 0xfffd) {
38             *pwc = (ucs4_t) wc;
39             return 1;
40         }
41     }
42     else if (c < 0xf0) {
43     }
44     else {
45         unsigned short wc = cp1133_2uni_2[c-0xf0];
46         if (wc != 0xfffd) {
47             *pwc = (ucs4_t) wc;

```

```

48     return 1;
49     }
50 }
51 return RET_ILSEQ;
52 }
53
54 static const unsigned char cp1133_page00[16] = {
55     0xa0, 0x00, 0xfc, 0x00, 0x00, 0x00, 0xfe, 0x00, /* 0xa0-0xa7 */
56     0x00, 0x00, 0x00, 0x00, 0xfd, 0x00, 0x00, 0x00, /* 0xa8-0xaf */
57 };
58 static const unsigned char cp1133_page0e[96] = {
59     0x00, 0xa1, 0xa2, 0x00, 0xa3, 0x00, 0x00, 0xa4, /* 0x80-0x87 */
60     0xa5, 0x00, 0xa7, 0x00, 0x00, 0xa8, 0x00, 0x00, /* 0x88-0x8f */
61     0x00, 0x00, 0x00, 0x00, 0xa9, 0xaa, 0xab, 0xac, /* 0x90-0x97 */
62     0x00, 0xad, 0xae, 0xaf, 0xb0, 0xb1, 0xb2, 0xb3, /* 0x98-0x9f */
63     0x00, 0xb4, 0xb5, 0xb6, 0x00, 0xb7, 0x00, 0xb8, /* 0xa0-0xa7 */
64     0x00, 0x00, 0xa6, 0xb9, 0x00, 0xba, 0xbb, 0xbf, /* 0xa8-0xaf */
65     0xc0, 0xca, 0xc1, 0xc2, 0xc3, 0xc4, 0xc5, 0xc6, /* 0xb0-0xb7 */
66     0xc7, 0xc8, 0x00, 0xcb, 0xc9, 0xcc, 0x00, 0x00, /* 0xb8-0xbf */
67     0xd0, 0xd1, 0xd2, 0xd3, 0xd4, 0x00, 0xdb, 0x00, /* 0xc0-0xc7 */
68     0xd5, 0xd6, 0xd7, 0xd8, 0xd9, 0xda, 0x00, 0x00, /* 0xc8-0xcf */
69     0xf0, 0xf1, 0xf2, 0xf3, 0xf4, 0xf5, 0xf6, 0xf7, /* 0xd0-0xd7 */
70     0xf8, 0xf9, 0x00, 0x00, 0xdd, 0xde, 0x00, 0x00, /* 0xd8-0xdf */
71 };
72
73 static int
74 cp1133_wctomb (conv_t conv, unsigned char *r, ucs4_t wc, int n)
75 {
76     unsigned char c = 0;
77     if (wc < 0x00a0) {
78         *r = wc;
79         return 1;
80     }
81     else if (wc >= 0x00a0 && wc < 0x00b0)
82         c = cp1133_page00[wc-0x00a0];
83     else if (wc >= 0x0e80 && wc < 0x0ee0)
84         c = cp1133_page0e[wc-0x0e80];
85     else if (wc == 0x20ad)
86         c = 0xdf;
87     if (c != 0) {
88         *r = c;
89         return 1;
90     }
91     return RET_ILSEQ;
92 }

```

32.204 cp1251.h

```

1 /* $XFree86: xc/lib/X11/lcUniConv/cp1251.h,v 1.1 2000/12/04 18:49:32 dawes Exp $ */
2
3 /*
4  * CP1251
5  */
6 #ifdef NEED_TOWC
7
8 static const unsigned short cp1251_2uni[128] = {
9     /* 0x80 */
10    0x0402, 0x0403, 0x201a, 0x0453, 0x201e, 0x2026, 0x2020, 0x2021,
11    0x20ac, 0x2030, 0x0409, 0x2039, 0x040a, 0x040c, 0x040b, 0x040f,
12    /* 0x90 */
13    0x0452, 0x2018, 0x2019, 0x201c, 0x201d, 0x2022, 0x2013, 0x2014,
14    0xffff, 0x2122, 0x0459, 0x203a, 0x045a, 0x045c, 0x045b, 0x045f,
15    /* 0xa0 */
16    0x00a0, 0x040e, 0x045e, 0x0408, 0x00a4, 0x0490, 0x00a6, 0x00a7,
17    0x0401, 0x00a9, 0x0404, 0x00ab, 0x00ac, 0x00ad, 0x00ae, 0x0407,
18    /* 0xb0 */
19    0x00b0, 0x00b1, 0x0406, 0x0456, 0x0491, 0x00b5, 0x00b6, 0x00b7,
20    0x0451, 0x2116, 0x0454, 0x00bb, 0x0458, 0x0405, 0x0455, 0x0457,
21    /* 0xc0 */
22    0x0410, 0x0411, 0x0412, 0x0413, 0x0414, 0x0415, 0x0416, 0x0417,
23    0x0418, 0x0419, 0x041a, 0x041b, 0x041c, 0x041d, 0x041e, 0x041f,
24    /* 0xd0 */
25    0x0420, 0x0421, 0x0422, 0x0423, 0x0424, 0x0425, 0x0426, 0x0427,
26    0x0428, 0x0429, 0x042a, 0x042b, 0x042c, 0x042d, 0x042e, 0x042f,
27    /* 0xe0 */
28    0x0430, 0x0431, 0x0432, 0x0433, 0x0434, 0x0435, 0x0436, 0x0437,
29    0x0438, 0x0439, 0x043a, 0x043b, 0x043c, 0x043d, 0x043e, 0x043f,
30    /* 0xf0 */
31    0x0440, 0x0441, 0x0442, 0x0443, 0x0444, 0x0445, 0x0446, 0x0447,
32    0x0448, 0x0449, 0x044a, 0x044b, 0x044c, 0x044d, 0x044e, 0x044f,
33 };
34
35 static int
36 cp1251_mbtowc (conv_t conv, ucs4_t *pwc, const unsigned char *s, int n)
37 {

```

```

38 unsigned char c = *s;
39 if (c < 0x80) {
40     *pwc = (ucs4_t) c;
41     return 1;
42 }
43 else {
44     unsigned short wc = cp1251_2uni[c-0x80];
45     if (wc != 0xffff) {
46         *pwc = (ucs4_t) wc;
47         return 1;
48     }
49 }
50 return RET_ILSEQ;
51 }
52 #endif /* NEED_TOWC */
53
54 #ifdef NEED_TOMB
55 static const unsigned char cp1251_page00[32] = {
56     0xa0, 0x00, 0x00, 0x00, 0xa4, 0x00, 0xa6, 0xa7, /* 0xa0-0xa7 */
57     0x00, 0xa9, 0x00, 0xab, 0xac, 0xad, 0xae, 0x00, /* 0xa8-0xaf */
58     0xb0, 0xb1, 0x00, 0x00, 0x00, 0x00, 0xb5, 0xb6, 0xb7, /* 0xb0-0xb7 */
59     0x00, 0x00, 0x00, 0xbb, 0x00, 0x00, 0x00, 0x00, /* 0xb8-0xbf */
60 };
61 static const unsigned char cp1251_page04[152] = {
62     0x00, 0xa8, 0x80, 0x81, 0xaa, 0xbd, 0xb2, 0xaf, /* 0x00-0x07 */
63     0xa3, 0x8a, 0x8c, 0x8e, 0x8d, 0x00, 0xa1, 0x8f, /* 0x08-0x0f */
64     0xc0, 0xc1, 0xc2, 0xc3, 0xc4, 0xc5, 0xc6, 0xc7, /* 0x10-0x17 */
65     0xc8, 0xc9, 0xca, 0xcb, 0xcc, 0xcd, 0xce, 0xcf, /* 0x18-0x1f */
66     0xd0, 0xd1, 0xd2, 0xd3, 0xd4, 0xd5, 0xd6, 0xd7, /* 0x20-0x27 */
67     0xd8, 0xd9, 0xda, 0xdb, 0xdc, 0xdd, 0xde, 0xdf, /* 0x28-0x2f */
68     0xe0, 0xe1, 0xe2, 0xe3, 0xe4, 0xe5, 0xe6, 0xe7, /* 0x30-0x37 */
69     0xe8, 0xe9, 0xea, 0xeb, 0xec, 0xed, 0xee, 0xef, /* 0x38-0x3f */
70     0xf0, 0xf1, 0xf2, 0xf3, 0xf4, 0xf5, 0xf6, 0xf7, /* 0x40-0x47 */
71     0xf8, 0xf9, 0xfa, 0xfb, 0xfc, 0xfd, 0xfe, 0xff, /* 0x48-0x4f */
72     0x00, 0xb8, 0x90, 0x83, 0xba, 0xbe, 0xb3, 0xbf, /* 0x50-0x57 */
73     0xbc, 0x9a, 0x9c, 0x9e, 0x9d, 0x00, 0xa2, 0x9f, /* 0x58-0x5f */
74     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x60-0x67 */
75     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x68-0x6f */
76     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x70-0x77 */
77     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x78-0x7f */
78     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x80-0x87 */
79     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x88-0x8f */
80     0xa5, 0xb4, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x90-0x97 */
81 };
82 static const unsigned char cp1251_page20[48] = {
83     0x00, 0x00, 0x00, 0x96, 0x97, 0x00, 0x00, 0x00, /* 0x10-0x17 */
84     0x91, 0x92, 0x82, 0x00, 0x93, 0x94, 0x84, 0x00, /* 0x18-0x1f */
85     0x86, 0x87, 0x95, 0x00, 0x00, 0x00, 0x85, 0x00, /* 0x20-0x27 */
86     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x28-0x2f */
87     0x89, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x30-0x37 */
88     0x00, 0x8b, 0x9b, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x38-0x3f */
89 };
90
91 static int
92 cp1251_wctomb (conv_t conv, unsigned char *r, ucs4_t wc, int n)
93 {
94     unsigned char c = 0;
95     if (wc < 0x0080) {
96         *r = wc;
97         return 1;
98     }
99     else if (wc >= 0x00a0 && wc < 0x00c0)
100         c = cp1251_page00[wc-0x00a0];
101     else if (wc >= 0x0400 && wc < 0x0498)
102         c = cp1251_page04[wc-0x0400];
103     else if (wc >= 0x2010 && wc < 0x2040)
104         c = cp1251_page20[wc-0x2010];
105     else if (wc == 0x20ac)
106         c = 0x88;
107     else if (wc == 0x2116)
108         c = 0xb9;
109     else if (wc == 0x2122)
110         c = 0x99;
111     if (c != 0) {
112         *r = c;
113         return 1;
114     }
115     return RET_ILSEQ;
116 }
117 #endif /* NEED_TOMB */

```

32.205 cp1255.h

```

1 /* $XFree86: xc/lib/X11/lcUniConv/cp1255.h,v 1.1 2000/12/04 18:49:33 dawes Exp $ */
2

```

```

3  /*
4  * CP1255
5  */
6
7  static const unsigned short cp1255_2uni[128] = {
8      /* 0x80 */
9      0x20ac, 0xffff, 0x201a, 0x0192, 0x201e, 0x2026, 0x2020, 0x2021,
10     0x02c6, 0x2030, 0xffff, 0x2039, 0xffff, 0xffff, 0xffff, 0xffff,
11     /* 0x90 */
12     0xffff, 0x2018, 0x2019, 0x201c, 0x201d, 0x2022, 0x2013, 0x2014,
13     0x02dc, 0x2122, 0xffff, 0x203a, 0xffff, 0xffff, 0xffff, 0xffff,
14     /* 0xa0 */
15     0x00a0, 0x00a1, 0x00a2, 0x00a3, 0x20aa, 0x00a5, 0x00a6, 0x00a7,
16     0x00a8, 0x00a9, 0x00d7, 0x00ab, 0x00ac, 0x00ad, 0x00ae, 0x00af,
17     /* 0xb0 */
18     0x00b0, 0x00b1, 0x00b2, 0x00b3, 0x00b4, 0x00b5, 0x00b6, 0x00b7,
19     0x00b8, 0x00b9, 0x00f7, 0x00bb, 0x00bc, 0x00bd, 0x00be, 0x00bf,
20     /* 0xc0 */
21     0x05b0, 0x05b1, 0x05b2, 0x05b3, 0x05b4, 0x05b5, 0x05b6, 0x05b7,
22     0x05b8, 0x05b9, 0xffff, 0x05bb, 0x05bc, 0x05bd, 0x05be, 0x05bf,
23     /* 0xd0 */
24     0x05c0, 0x05c1, 0x05c2, 0x05c3, 0x05f0, 0x05f1, 0x05f2, 0x05f3,
25     0x05f4, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
26     /* 0xe0 */
27     0x05d0, 0x05d1, 0x05d2, 0x05d3, 0x05d4, 0x05d5, 0x05d6, 0x05d7,
28     0x05d8, 0x05d9, 0x05da, 0x05db, 0x05dc, 0x05dd, 0x05de, 0x05df,
29     /* 0xf0 */
30     0x05e0, 0x05e1, 0x05e2, 0x05e3, 0x05e4, 0x05e5, 0x05e6, 0x05e7,
31     0x05e8, 0x05e9, 0x05ea, 0xffff, 0xffff, 0x200e, 0x200f, 0xffff,
32 };
33
34 static int
35 cp1255_mbtowc (conv_t conv, ucs4_t *pwc, const unsigned char *s, int n)
36 {
37     unsigned char c = *s;
38     if (c < 0x80) {
39         *pwc = (ucs4_t) c;
40         return 1;
41     }
42     else {
43         unsigned short wc = cp1255_2uni[c-0x80];
44         if (wc != 0xffff) {
45             *pwc = (ucs4_t) wc;
46             return 1;
47         }
48     }
49     return RET_ILSEQ;
50 }
51
52 static const unsigned char cp1255_page00[88] = {
53     0xa0, 0xa1, 0xa2, 0xa3, 0x00, 0xa5, 0xa6, 0xa7, /* 0xa0-0xa7 */
54     0xa8, 0xa9, 0x00, 0xab, 0xac, 0xad, 0xae, 0xaf, /* 0xa8-0xaf */
55     0xb0, 0xb1, 0xb2, 0xb3, 0xb4, 0xb5, 0xb6, 0xb7, /* 0xb0-0xb7 */
56     0xb8, 0xb9, 0x00, 0xbb, 0xbc, 0xbd, 0xbe, 0xbf, /* 0xb8-0xbf */
57     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xc0-0xc7 */
58     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xc8-0xcf */
59     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xaa, /* 0xd0-0xd7 */
60     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xd8-0xdf */
61     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xe0-0xe7 */
62     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xe8-0xef */
63     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xba, /* 0xf0-0xf7 */
64 };
65 static const unsigned char cp1255_page02[32] = {
66     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x88, 0x00, /* 0xc0-0xc7 */
67     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xc8-0xcf */
68     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xd0-0xd7 */
69     0x00, 0x00, 0x00, 0x00, 0x98, 0x00, 0x00, 0x00, /* 0xd8-0xdf */
70 };
71 static const unsigned char cp1255_page05[72] = {
72     0xc0, 0xc1, 0xc2, 0xc3, 0xc4, 0xc5, 0xc6, 0xc7, /* 0xb0-0xb7 */
73     0xc8, 0xc9, 0x00, 0xcb, 0xcc, 0xcd, 0xce, 0xcf, /* 0xb8-0xbf */
74     0xd0, 0xd1, 0xd2, 0xd3, 0x00, 0x00, 0x00, 0x00, /* 0xc0-0xc7 */
75     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xc8-0xcf */
76     0xe0, 0xe1, 0xe2, 0xe3, 0xe4, 0xe5, 0xe6, 0xe7, /* 0xd0-0xd7 */
77     0xe8, 0xe9, 0xea, 0xeb, 0xec, 0xed, 0xee, 0xef, /* 0xd8-0xdf */
78     0xf0, 0xf1, 0xf2, 0xf3, 0xf4, 0xf5, 0xf6, 0xf7, /* 0xe0-0xe7 */
79     0xf8, 0xf9, 0xfa, 0xfb, 0x00, 0x00, 0x00, 0x00, /* 0xe8-0xef */
80     0xd4, 0xd5, 0xd6, 0xd7, 0xd8, 0x00, 0x00, 0x00, /* 0xf0-0xf7 */
81 };
82 static const unsigned char cp1255_page20[56] = {
83     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xfd, 0xfe, /* 0x08-0x0f */
84     0x00, 0x00, 0x00, 0x96, 0x97, 0x00, 0x00, 0x00, /* 0x10-0x17 */
85     0x91, 0x92, 0x82, 0x00, 0x93, 0x94, 0x84, 0x00, /* 0x18-0x1f */
86     0x86, 0x87, 0x95, 0x00, 0x00, 0x00, 0x85, 0x00, /* 0x20-0x27 */
87     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x28-0x2f */
88     0x89, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x30-0x37 */
89     0x00, 0x8b, 0x9b, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x38-0x3f */

```

```

90 };
91
92 static int
93 cp1255_wctomb (conv_t conv, unsigned char *r, ucs4_t wc, int n)
94 {
95     unsigned char c = 0;
96     if (wc < 0x0080) {
97         *r = wc;
98         return 1;
99     }
100    else if (wc >= 0x00a0 && wc < 0x00f8)
101        c = cp1255_page00[wc-0x00a0];
102    else if (wc == 0x0192)
103        c = 0x83;
104    else if (wc >= 0x02c0 && wc < 0x02e0)
105        c = cp1255_page02[wc-0x02c0];
106    else if (wc >= 0x05b0 && wc < 0x05f8)
107        c = cp1255_page05[wc-0x05b0];
108    else if (wc >= 0x2008 && wc < 0x2040)
109        c = cp1255_page20[wc-0x2008];
110    else if (wc == 0x20aa)
111        c = 0xa4;
112    else if (wc == 0x20ac)
113        c = 0x80;
114    else if (wc == 0x2122)
115        c = 0x99;
116    if (c != 0) {
117        *r = c;
118        return 1;
119    }
120    return RET_ILSEQ;
121 }

```

32.206 cp1256.h

```

1 /* $XFree86: xc/lib/X11/lcUniConv/cp1256.h,v 1.1 2000/12/04 18:49:34 dawes Exp $ */
2
3 /*
4  * CP1256
5  */
6
7 static const unsigned short cp1256_2uni[128] = {
8     /* 0x80 */
9     0x20ac, 0x067e, 0x201a, 0x0192, 0x201e, 0x2026, 0x2020, 0x2021,
10    0x02c6, 0x2030, 0x0679, 0x2039, 0x0152, 0x0686, 0x0698, 0x0688,
11    /* 0x90 */
12    0x06af, 0x2018, 0x2019, 0x201c, 0x201d, 0x2022, 0x2013, 0x2014,
13    0x06a9, 0x2122, 0x0691, 0x203a, 0x0153, 0x200c, 0x200d, 0x06ba,
14    /* 0xa0 */
15    0x00a0, 0x060c, 0x00a2, 0x00a3, 0x00a4, 0x00a5, 0x00a6, 0x00a7,
16    0x00a8, 0x00a9, 0x06be, 0x00ab, 0x00ac, 0x00ad, 0x00ae, 0x00af,
17    /* 0xb0 */
18    0x00b0, 0x00b1, 0x00b2, 0x00b3, 0x00b4, 0x00b5, 0x00b6, 0x00b7,
19    0x00b8, 0x00b9, 0x061b, 0x00bb, 0x00bc, 0x00bd, 0x00be, 0x061f,
20    /* 0xc0 */
21    0x06c1, 0x0621, 0x0622, 0x0623, 0x0624, 0x0625, 0x0626, 0x0627,
22    0x0628, 0x0629, 0x062a, 0x062b, 0x062c, 0x062d, 0x062e, 0x062f,
23    /* 0xd0 */
24    0x0630, 0x0631, 0x0632, 0x0633, 0x0634, 0x0635, 0x0636, 0x00d7,
25    0x0637, 0x0638, 0x0639, 0x063a, 0x0640, 0x0641, 0x0642, 0x0643,
26    /* 0xe0 */
27    0x00e0, 0x0644, 0x00e2, 0x0645, 0x0646, 0x0647, 0x0648, 0x00e7,
28    0x00e8, 0x00e9, 0x00ea, 0x00eb, 0x0649, 0x064a, 0x00ee, 0x00ef,
29    /* 0xf0 */
30    0x064b, 0x064c, 0x064d, 0x064e, 0x00f4, 0x064f, 0x0650, 0x00f7,
31    0x0651, 0x00f9, 0x0652, 0x00fb, 0x00fc, 0x200e, 0x200f, 0x06d2,
32 };
33
34 static int
35 cp1256_mbtowc (conv_t conv, ucs4_t *pwc, const unsigned char *s, int n)
36 {
37     unsigned char c = *s;
38     if (c < 0x80)
39         *pwc = (ucs4_t) c;
40     else
41         *pwc = (ucs4_t) cp1256_2uni[c-0x80];
42     return 1;
43 }
44
45 static const unsigned char cp1256_page00[96] = {
46     0xa0, 0x00, 0xa2, 0xa3, 0xa4, 0xa5, 0xa6, 0xa7, /* 0xa0-0xa7 */
47     0xa8, 0xa9, 0x00, 0xab, 0xac, 0xad, 0xae, 0xaf, /* 0xa8-0xaf */
48     0xb0, 0xb1, 0xb2, 0xb3, 0xb4, 0xb5, 0xb6, 0xb7, /* 0xb0-0xb7 */
49     0xb8, 0xb9, 0x00, 0xbb, 0xbc, 0xbd, 0xbe, 0x00, /* 0xb8-0xbf */
50     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xc0-0xc7 */

```



```
51 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xc8-0xcf */
52 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xd7, /* 0xd0-0xd7 */
53 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xd8-0xdf */
54 0xe0, 0x00, 0xe2, 0x00, 0x00, 0x00, 0x00, 0xe7, /* 0xe0-0xe7 */
55 0xe8, 0xe9, 0xea, 0xeb, 0x00, 0x00, 0xee, 0xef, /* 0xe8-0xef */
56 0x00, 0x00, 0x00, 0x00, 0xf4, 0x00, 0x00, 0xf7, /* 0xf0-0xf7 */
57 0x00, 0xf9, 0x00, 0xfb, 0xfc, 0x00, 0x00, 0x00, /* 0xf8-0xff */
58 };
59 static const unsigned char cp1256_page01[72] = {
60 0x00, 0x00, 0x8c, 0x9c, 0x00, 0x00, 0x00, 0x00, /* 0x50-0x57 */
61 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x58-0x5f */
62 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x60-0x67 */
63 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x68-0x6f */
64 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x70-0x77 */
65 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x78-0x7f */
66 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x80-0x87 */
67 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x88-0x8f */
68 0x00, 0x00, 0x83, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x90-0x97 */
69 };
70 static const unsigned char cp1256_page06[208] = {
71 0x00, 0x00, 0x00, 0x00, 0xa1, 0x00, 0x00, 0x00, /* 0x08-0x0f */
72 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x10-0x17 */
73 0x00, 0x00, 0x00, 0xba, 0x00, 0x00, 0x00, 0xbf, /* 0x18-0x1f */
74 0x00, 0xc1, 0xc2, 0xc3, 0xc4, 0xc5, 0xc6, 0xc7, /* 0x20-0x27 */
75 0xc8, 0xc9, 0xca, 0xcb, 0xcc, 0xcd, 0xce, 0xcf, /* 0x28-0x2f */
76 0xd0, 0xd1, 0xd2, 0xd3, 0xd4, 0xd5, 0xd6, 0xd8, /* 0x30-0x37 */
77 0xd9, 0xda, 0xdb, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x38-0x3f */
78 0xdc, 0xdd, 0xde, 0xdf, 0xe1, 0xe3, 0xe4, 0xe5, /* 0x40-0x47 */
79 0xe6, 0xec, 0xed, 0xf0, 0xf1, 0xf2, 0xf3, 0xf5, /* 0x48-0x4f */
80 0xf6, 0xf8, 0xfa, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x50-0x57 */
81 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x58-0x5f */
82 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x60-0x67 */
83 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x68-0x6f */
84 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x70-0x77 */
85 0x00, 0x8a, 0x00, 0x00, 0x00, 0x00, 0x81, 0x00, /* 0x78-0x7f */
86 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x8d, 0x00, /* 0x80-0x87 */
87 0x8f, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x88-0x8f */
88 0x00, 0x9a, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x90-0x97 */
89 0x8e, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x98-0x9f */
90 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xa0-0xa7 */
91 0x00, 0x98, 0x00, 0x00, 0x00, 0x00, 0x00, 0x90, /* 0xa8-0xaf */
92 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xb0-0xb7 */
93 0x00, 0x00, 0x9f, 0x00, 0x00, 0x00, 0xaa, 0x00, /* 0xb8-0xbf */
94 0x00, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xc0-0xc7 */
95 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xc8-0xcf */
96 0x00, 0x00, 0xff, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xd0-0xd7 */
97 };
98 static const unsigned char cp1256_page20[56] = {
99 0x00, 0x00, 0x00, 0x00, 0x9d, 0x9e, 0xfd, 0xfe, /* 0x08-0x0f */
100 0x00, 0x00, 0x00, 0x96, 0x97, 0x00, 0x00, 0x00, /* 0x10-0x17 */
101 0x91, 0x92, 0x82, 0x00, 0x93, 0x94, 0x84, 0x00, /* 0x18-0x1f */
102 0x86, 0x87, 0x95, 0x00, 0x00, 0x00, 0x85, 0x00, /* 0x20-0x27 */
103 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x28-0x2f */
104 0x89, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x30-0x37 */
105 0x00, 0x8b, 0x9b, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x38-0x3f */
106 };
107
108 static int
109 cp1256_wctomb (conv_t conv, unsigned char *r, ucs4_t wc, int n)
110 {
111     unsigned char c = 0;
112     if (wc < 0x0080) {
113         *r = wc;
114         return 1;
115     }
116     else if (wc >= 0x00a0 && wc < 0x0100)
117         c = cp1256_page00[wc-0x00a0];
118     else if (wc >= 0x0150 && wc < 0x0198)
119         c = cp1256_page01[wc-0x0150];
120     else if (wc == 0x02c6)
121         c = 0x88;
122     else if (wc >= 0x0608 && wc < 0x06d8)
123         c = cp1256_page06[wc-0x0608];
124     else if (wc >= 0x2008 && wc < 0x2040)
125         c = cp1256_page20[wc-0x2008];
126     else if (wc == 0x20ac)
127         c = 0x80;
128     else if (wc == 0x2122)
129         c = 0x99;
130     if (c != 0) {
131         *r = c;
132         return 1;
133     }
134     return RET_ILSEQ;
135 }
```

32.207 cp936ext.h

```

1 /*
2 * "$Id$"
3 *
4 * Character encoding support for the Fast Light Tool Kit (FLTK).
5 *
6 * Copyright 1998-2010 by Bill Spitzak and others.
7 *
8 * This library is free software. Distribution and use rights are outlined in
9 * the file "COPYING" which should have been included with this file. If this
10 * file is missing or damaged, see the license at:
11 *
12 *   http://www.fltk.org/COPYING.php
13 *
14 * Please report all bugs and problems on the following page:
15 *
16 *   http://www.fltk.org/str.php
17 */
18
19 #if !defined(WIN32) && !defined(__APPLE__)
20
21 #ifndef CP936
22 #ifdef NEED_TOWC
23 static int
24 cp936ext_mbtowc (conv_t conv, ucs4_t *pwc, const unsigned char *, int n)
25 {
26     return 0;
27 }
28 #endif /* NEED_TOWC */
29
30 #ifdef NEED_TOMB
31 static int
32 cp936ext_wctomb (conv_t conv, unsigned char *r, ucs4_t wc, int n)
33 {
34     return 0;
35 }
36 #endif /* NEED_TOMB */
37
38 #else
39 /*
40 * CP936EXT
41 */
42 #ifdef NEED_TOWC
43
44 static const unsigned short cp936ext_2uni_page81[23766] = {
45     /* 0x81 */
46     0x4e02, 0x4e04, 0x4e05, 0x4e06, 0x4e0f, 0x4e12, 0x4e17, 0x4e1f,
47     0x4e20, 0x4e21, 0x4e23, 0x4e26, 0x4e29, 0x4e2e, 0x4e2f, 0x4e31,
48     0x4e33, 0x4e35, 0x4e37, 0x4e3c, 0x4e40, 0x4e41, 0x4e42, 0x4e44,
49     0x4e46, 0x4e4a, 0x4e51, 0x4e55, 0x4e57, 0x4e5a, 0x4e5b, 0x4e62,
50     0x4e63, 0x4e64, 0x4e65, 0x4e67, 0x4e68, 0x4e6a, 0x4e6b, 0x4e6c,
51     0x4e6d, 0x4e6e, 0x4e6f, 0x4e72, 0x4e74, 0x4e75, 0x4e76, 0x4e77,
52     0x4e78, 0x4e79, 0x4e7a, 0x4e7b, 0x4e7c, 0x4e7d, 0x4e7f, 0x4e80,
53     0x4e81, 0x4e82, 0x4e83, 0x4e84, 0x4e85, 0x4e87, 0x4e8a, 0x4e90,
54     0x4e96, 0x4e97, 0x4e99, 0x4e9c, 0x4e9d, 0x4e9e, 0x4ea3, 0x4eaa,
55     0x4eaf, 0x4eb0, 0x4eb1, 0x4eb4, 0x4eb6, 0x4eb7, 0x4eb8, 0x4eb9,
56     0x4ebc, 0x4ebd, 0x4ebe, 0x4ec8, 0x4ecc, 0x4ecf, 0x4ed0, 0x4ed2,
57     0x4eda, 0x4edb, 0x4edc, 0x4ee0, 0x4ee2, 0x4ee6, 0x4ee7, 0x4ee9,
58     0x4eed, 0x4eee, 0x4eef, 0x4ef1, 0x4ef4, 0x4ef8, 0x4ef9, 0x4efa,
59     0x4efc, 0x4efe, 0x4f00, 0x4f02, 0x4f03, 0x4f04, 0x4f05, 0x4f06,
60     0x4f07, 0x4f08, 0x4f0b, 0x4f0c, 0x4f12, 0x4f13, 0x4f14, 0x4f15,
61     0x4f16, 0x4f1c, 0x4f1d, 0x4f21, 0x4f23, 0x4f28, 0x4f29, 0x4f2c,
62     0x4f2d, 0x4f2e, 0x4f31, 0x4f33, 0x4f35, 0x4f37, 0x4f39, 0x4f3b,
63     0x4f3e, 0x4f3f, 0x4f40, 0x4f41, 0x4f42, 0x4f44, 0x4f45, 0x4f47,
64     0x4f48, 0x4f49, 0x4f4a, 0x4f4b, 0x4f4c, 0x4f52, 0x4f54, 0x4f56,
65     0x4f61, 0x4f62, 0x4f66, 0x4f68, 0x4f6a, 0x4f6b, 0x4f6d, 0x4f6e,
66     0x4f71, 0x4f72, 0x4f75, 0x4f77, 0x4f78, 0x4f79, 0x4f7a, 0x4f7d,
67     0x4f80, 0x4f81, 0x4f82, 0x4f85, 0x4f86, 0x4f87, 0x4f8a, 0x4f8c,
68     0x4f8e, 0x4f90, 0x4f92, 0x4f93, 0x4f95, 0x4f96, 0x4f98, 0x4f99,
69     0x4f9a, 0x4f9c, 0x4f9e, 0x4f9f, 0x4fa1, 0x4fa2,
70     /* 0x82 */
71     0x4fa4, 0x4fab, 0x4fad, 0x4fb0, 0x4fb1, 0x4fb2, 0x4fb3, 0x4fb4,
72     0x4fb6, 0x4fb7, 0x4fb8, 0x4fb9, 0x4fba, 0x4fbb, 0x4fbc, 0x4fbd,
73     0x4fbe, 0x4fc0, 0x4fc1, 0x4fc2, 0x4fc6, 0x4fc7, 0x4fc8, 0x4fc9,
74     0x4fcb, 0x4fcc, 0x4fcd, 0x4fd2, 0x4fd3, 0x4fd4, 0x4fd5, 0x4fd6,
75     0x4fd9, 0x4fdb, 0x4fe0, 0x4fe2, 0x4fe4, 0x4fe5, 0x4fe7, 0x4feb,
76     0x4fec, 0x4ff0, 0x4ff2, 0x4ff4, 0x4ff5, 0x4ff6, 0x4ff7, 0x4ff9,
77     0x4ffb, 0x4ffc, 0x4ffd, 0x4fff, 0x5000, 0x5001, 0x5002, 0x5003,
78     0x5004, 0x5005, 0x5006, 0x5007, 0x5008, 0x5009, 0x500a, 0x500b,
79     0x500e, 0x5010, 0x5011, 0x5013, 0x5015, 0x5016, 0x5017, 0x501b,
80     0x501d, 0x501e, 0x5020, 0x5022, 0x5023, 0x5024, 0x5027, 0x502b,
81     0x502f, 0x5030, 0x5031, 0x5032, 0x5033, 0x5034, 0x5035, 0x5036,
82     0x5037, 0x5038, 0x5039, 0x503b, 0x503d, 0x503f, 0x5040, 0x5041,
83     0x5042, 0x5044, 0x5045, 0x5046, 0x5049, 0x504a, 0x504b, 0x504d,
84     0x5050, 0x5051, 0x5052, 0x5053, 0x5054, 0x5056, 0x5057, 0x5058,
85     0x5059, 0x505b, 0x505d, 0x505e, 0x505f, 0x5060, 0x5061, 0x5062,

```

```
86 0x5063, 0x5064, 0x5066, 0x5067, 0x5068, 0x5069, 0x506a, 0x506b,
87 0x506d, 0x506e, 0x506f, 0x5070, 0x5071, 0x5072, 0x5073, 0x5074,
88 0x5075, 0x5078, 0x5079, 0x507a, 0x507c, 0x507d, 0x5081, 0x5082,
89 0x5083, 0x5084, 0x5086, 0x5087, 0x5089, 0x508a, 0x508b, 0x508c,
90 0x508e, 0x508f, 0x5090, 0x5091, 0x5092, 0x5093, 0x5094, 0x5095,
91 0x5096, 0x5097, 0x5098, 0x5099, 0x509a, 0x509b, 0x509c, 0x509d,
92 0x509e, 0x509f, 0x50a0, 0x50a1, 0x50a2, 0x50a4, 0x50a6, 0x50aa,
93 0x50ab, 0x50ad, 0x50ae, 0x50af, 0x50b0, 0x50b1, 0x50b3, 0x50b4,
94 0x50b5, 0x50b6, 0x50b7, 0x50b8, 0x50b9, 0x50bc,
95 /* 0x83 */
96 0x50bd, 0x50be, 0x50bf, 0x50c0, 0x50c1, 0x50c2, 0x50c3, 0x50c4,
97 0x50c5, 0x50c6, 0x50c7, 0x50c8, 0x50c9, 0x50ca, 0x50cb, 0x50cc,
98 0x50cd, 0x50ce, 0x50d0, 0x50d1, 0x50d2, 0x50d3, 0x50d4, 0x50d5,
99 0x50d7, 0x50d8, 0x50d9, 0x50db, 0x50dc, 0x50dd, 0x50de, 0x50df,
100 0x50e0, 0x50e1, 0x50e2, 0x50e3, 0x50e4, 0x50e5, 0x50e8, 0x50e9,
101 0x50ea, 0x50eb, 0x50ef, 0x50f0, 0x50f1, 0x50f2, 0x50f4, 0x50f6,
102 0x50f7, 0x50f8, 0x50f9, 0x50fa, 0x50fc, 0x50fd, 0x50fe, 0x50ff,
103 0x5100, 0x5101, 0x5102, 0x5103, 0x5104, 0x5105, 0x5108, 0x5109,
104 0x510a, 0x510c, 0x510d, 0x510e, 0x510f, 0x5110, 0x5111, 0x5113,
105 0x5114, 0x5115, 0x5116, 0x5117, 0x5118, 0x5119, 0x511a, 0x511b,
106 0x511c, 0x511d, 0x511e, 0x511f, 0x5120, 0x5122, 0x5123, 0x5124,
107 0x5125, 0x5126, 0x5127, 0x5128, 0x5129, 0x512a, 0x512b, 0x512c,
108 0x512d, 0x512e, 0x512f, 0x5130, 0x5131, 0x5132, 0x5133, 0x5134,
109 0x5135, 0x5136, 0x5137, 0x5138, 0x5139, 0x513a, 0x513b, 0x513c,
110 0x513d, 0x513e, 0x5142, 0x5147, 0x514a, 0x514c, 0x514e, 0x514f,
111 0x5150, 0x5152, 0x5153, 0x5157, 0x5158, 0x5159, 0x515b, 0x515d,
112 0x515e, 0x515f, 0x5160, 0x5161, 0x5163, 0x5164, 0x5166, 0x5167,
113 0x5169, 0x516a, 0x516f, 0x5172, 0x517a, 0x517e, 0x517f, 0x5183,
114 0x5184, 0x5186, 0x5187, 0x518a, 0x518b, 0x518e, 0x518f, 0x5190,
115 0x5191, 0x5193, 0x5194, 0x5198, 0x5199, 0x519d, 0x519e, 0x519f,
116 0x51a1, 0x51a3, 0x51a6, 0x51a7, 0x51a8, 0x51a9, 0x51aa, 0x51ad,
117 0x51ae, 0x51b4, 0x51b8, 0x51b9, 0x51ba, 0x51be, 0x51bf, 0x51c1,
118 0x51c2, 0x51c3, 0x51c5, 0x51c8, 0x51ca, 0x51cd, 0x51ce, 0x51d0,
119 0x51d2, 0x51d3, 0x51d4, 0x51d5, 0x51d6, 0x51d7,
120 /* 0x84 */
121 0x51d8, 0x51d9, 0x51da, 0x51dc, 0x51de, 0x51df, 0x51e2, 0x51e3,
122 0x51e5, 0x51e6, 0x51e7, 0x51e8, 0x51e9, 0x51ea, 0x51ec, 0x51ee,
123 0x51f1, 0x51f2, 0x51f4, 0x51f7, 0x51fe, 0x5204, 0x5205, 0x5209,
124 0x520b, 0x520c, 0x520f, 0x5210, 0x5213, 0x5214, 0x5215, 0x521c,
125 0x521e, 0x521f, 0x5221, 0x5222, 0x5223, 0x5225, 0x5226, 0x5227,
126 0x522a, 0x522c, 0x522f, 0x5231, 0x5232, 0x5234, 0x5235, 0x523c,
127 0x523e, 0x5244, 0x5245, 0x5246, 0x5247, 0x5248, 0x5249, 0x524b,
128 0x524e, 0x524f, 0x5252, 0x5253, 0x5255, 0x5257, 0x5258, 0x5259,
129 0x525a, 0x525b, 0x525d, 0x525f, 0x5260, 0x5262, 0x5263, 0x5264,
130 0x5266, 0x5268, 0x526b, 0x526c, 0x526d, 0x526e, 0x5270, 0x5271,
131 0x5273, 0x5274, 0x5275, 0x5276, 0x5277, 0x5278, 0x5279, 0x527a,
132 0x527b, 0x527c, 0x527e, 0x5280, 0x5283, 0x5284, 0x5285, 0x5286,
133 0x5287, 0x5289, 0x528a, 0x528b, 0x528c, 0x528d, 0x528e, 0x528f,
134 0x5291, 0x5292, 0x5294, 0x5295, 0x5296, 0x5297, 0x5298, 0x5299,
135 0x529a, 0x529c, 0x52a4, 0x52a5, 0x52a6, 0x52a7, 0x52ae, 0x52af,
136 0x52b0, 0x52b4, 0x52b5, 0x52b6, 0x52b7, 0x52b8, 0x52b9, 0x52ba,
137 0x52bb, 0x52bc, 0x52bd, 0x52c0, 0x52c1, 0x52c2, 0x52c4, 0x52c5,
138 0x52c6, 0x52c8, 0x52ca, 0x52cc, 0x52cd, 0x52ce, 0x52cf, 0x52d1,
139 0x52d3, 0x52d4, 0x52d5, 0x52d7, 0x52d9, 0x52da, 0x52db, 0x52dc,
140 0x52dd, 0x52de, 0x52e0, 0x52e1, 0x52e2, 0x52e3, 0x52e5, 0x52e6,
141 0x52e7, 0x52e8, 0x52e9, 0x52ea, 0x52eb, 0x52ec, 0x52ed, 0x52ee,
142 0x52ef, 0x52f1, 0x52f2, 0x52f3, 0x52f4, 0x52f5, 0x52f6, 0x52f7,
143 0x52f8, 0x52fb, 0x52fc, 0x52fd, 0x5301, 0x5302, 0x5303, 0x5304,
144 0x5307, 0x5309, 0x530a, 0x530b, 0x530c, 0x530e,
145 /* 0x85 */
146 0x5311, 0x5312, 0x5313, 0x5314, 0x5318, 0x531b, 0x531c, 0x531e,
147 0x531f, 0x5322, 0x5324, 0x5325, 0x5327, 0x5328, 0x5329, 0x532b,
148 0x532c, 0x532d, 0x532f, 0x5330, 0x5331, 0x5332, 0x5333, 0x5334,
149 0x5335, 0x5336, 0x5337, 0x5338, 0x533c, 0x533d, 0x5340, 0x5342,
150 0x5344, 0x5346, 0x534b, 0x534c, 0x534e, 0x5350, 0x5354, 0x5358,
151 0x5359, 0x535b, 0x535d, 0x5365, 0x5368, 0x536a, 0x536c, 0x536d,
152 0x5372, 0x5376, 0x5379, 0x537b, 0x537c, 0x537d, 0x537e, 0x5380,
153 0x5381, 0x5383, 0x5387, 0x5388, 0x538a, 0x538e, 0x538f, 0x5390,
154 0x5391, 0x5392, 0x5393, 0x5394, 0x5396, 0x5397, 0x5399, 0x539b,
155 0x539c, 0x539e, 0x53a0, 0x53a1, 0x53a4, 0x53a7, 0x53aa, 0x53ab,
156 0x53ac, 0x53ad, 0x53af, 0x53b0, 0x53b1, 0x53b2, 0x53b3, 0x53b4,
157 0x53b5, 0x53b7, 0x53b8, 0x53b9, 0x53ba, 0x53bc, 0x53bd, 0x53be,
158 0x53c0, 0x53c3, 0x53c4, 0x53c5, 0x53c6, 0x53c7, 0x53ce, 0x53cf,
159 0x53d0, 0x53d2, 0x53d3, 0x53d5, 0x53da, 0x53dc, 0x53dd, 0x53de,
160 0x53e1, 0x53e2, 0x53e7, 0x53f4, 0x53fa, 0x53fe, 0x53ff, 0x5400,
161 0x5402, 0x5405, 0x5407, 0x540b, 0x5414, 0x5418, 0x5419, 0x541a,
162 0x541c, 0x5422, 0x5424, 0x542a, 0x5425, 0x542a, 0x5430, 0x5433, 0x5436,
163 0x5437, 0x543a, 0x543d, 0x543f, 0x5441, 0x5442, 0x5444, 0x5445,
164 0x5447, 0x5449, 0x544c, 0x544d, 0x544e, 0x544f, 0x5451, 0x545a,
165 0x545d, 0x545e, 0x545f, 0x5460, 0x5461, 0x5463, 0x5465, 0x5467,
166 0x5469, 0x546a, 0x546b, 0x546c, 0x546d, 0x546e, 0x546f, 0x5470,
167 0x5474, 0x5479, 0x547a, 0x547e, 0x547f, 0x5481, 0x5483, 0x5485,
168 0x5487, 0x5488, 0x5489, 0x548a, 0x548d, 0x5491, 0x5493, 0x5497,
169 0x5498, 0x549c, 0x549e, 0x549f, 0x54a0, 0x54a1,
170 /* 0x86 */
171 0x54a2, 0x54a5, 0x54ae, 0x54b0, 0x54b2, 0x54b5, 0x54b6, 0x54b7,
172 0x54b9, 0x54ba, 0x54bc, 0x54be, 0x54c3, 0x54c5, 0x54ca, 0x54cb,
```

```
173 0x54d6, 0x54d8, 0x54db, 0x54e0, 0x54e1, 0x54e2, 0x54e3, 0x54e4,
174 0x54eb, 0x54ec, 0x54ef, 0x54f0, 0x54f1, 0x54f4, 0x54f5, 0x54f6,
175 0x54f7, 0x54f8, 0x54f9, 0x54fb, 0x54fe, 0x5500, 0x5502, 0x5503,
176 0x5504, 0x5505, 0x5508, 0x550a, 0x550b, 0x550c, 0x550d, 0x550e,
177 0x5512, 0x5513, 0x5515, 0x5516, 0x5517, 0x5518, 0x5519, 0x551a,
178 0x551c, 0x551d, 0x551e, 0x551f, 0x5521, 0x5522, 0x5526, 0x5528,
179 0x5529, 0x552b, 0x552d, 0x5532, 0x5534, 0x5535, 0x5536, 0x5538,
180 0x5539, 0x553a, 0x553b, 0x553d, 0x5540, 0x5542, 0x5545, 0x5547,
181 0x5548, 0x554b, 0x554c, 0x554d, 0x554e, 0x554f, 0x5551, 0x5552,
182 0x5553, 0x5554, 0x5557, 0x5558, 0x5559, 0x555a, 0x555b, 0x555d,
183 0x555e, 0x555f, 0x5560, 0x5562, 0x5563, 0x5568, 0x5569, 0x556b,
184 0x556f, 0x5570, 0x5571, 0x5572, 0x5573, 0x5574, 0x5579, 0x557a,
185 0x557d, 0x557f, 0x5585, 0x5586, 0x558c, 0x558d, 0x558e, 0x5590,
186 0x5592, 0x5593, 0x5595, 0x5596, 0x5597, 0x559a, 0x559b, 0x559e,
187 0x55a0, 0x55a1, 0x55a2, 0x55a3, 0x55a4, 0x55a5, 0x55a6, 0x55a8,
188 0x55a9, 0x55aa, 0x55ab, 0x55ac, 0x55ad, 0x55ae, 0x55af, 0x55b0,
189 0x55b2, 0x55b4, 0x55b6, 0x55b8, 0x55ba, 0x55bc, 0x55bf, 0x55c0,
190 0x55c1, 0x55c2, 0x55c3, 0x55c6, 0x55c7, 0x55c8, 0x55ca, 0x55cb,
191 0x55ce, 0x55cf, 0x55d0, 0x55d5, 0x55d7, 0x55d8, 0x55d9, 0x55da,
192 0x55db, 0x55de, 0x55e0, 0x55e2, 0x55e7, 0x55e9, 0x55ed, 0x55ee,
193 0x55f0, 0x55f1, 0x55f4, 0x55f6, 0x55f8, 0x55f9, 0x55fa, 0x55fb,
194 0x55fc, 0x55ff, 0x5602, 0x5603, 0x5604, 0x5605,
195 /* 0x87 */
196 0x5606, 0x5607, 0x560a, 0x560b, 0x560d, 0x5610, 0x5611, 0x5612,
197 0x5613, 0x5614, 0x5615, 0x5616, 0x5617, 0x5619, 0x561a, 0x561c,
198 0x561d, 0x5620, 0x5621, 0x5622, 0x5625, 0x5626, 0x5628, 0x5629,
199 0x562a, 0x562b, 0x562e, 0x562f, 0x5630, 0x5633, 0x5635, 0x5637,
200 0x5638, 0x563a, 0x563c, 0x563d, 0x563e, 0x5640, 0x5641, 0x5642,
201 0x5643, 0x5644, 0x5645, 0x5646, 0x5647, 0x5648, 0x5649, 0x564a,
202 0x564b, 0x564f, 0x5650, 0x5651, 0x5652, 0x5653, 0x5655, 0x5656,
203 0x565a, 0x565b, 0x565d, 0x565e, 0x565f, 0x5660, 0x5661, 0x5663,
204 0x5665, 0x5666, 0x5667, 0x566d, 0x566e, 0x566f, 0x5670, 0x5672,
205 0x5673, 0x5674, 0x5675, 0x5677, 0x5678, 0x5679, 0x567a, 0x567d,
206 0x567e, 0x567f, 0x5680, 0x5681, 0x5682, 0x5683, 0x5684, 0x5687,
207 0x5688, 0x5689, 0x568a, 0x568b, 0x568c, 0x568d, 0x5690, 0x5691,
208 0x5692, 0x5694, 0x5695, 0x5696, 0x5697, 0x5698, 0x5699, 0x569a,
209 0x569b, 0x569c, 0x569d, 0x569e, 0x569f, 0x56a0, 0x56a1, 0x56a2,
210 0x56a4, 0x56a5, 0x56a6, 0x56a7, 0x56a8, 0x56a9, 0x56aa, 0x56ab,
211 0x56ac, 0x56ad, 0x56ae, 0x56b0, 0x56b1, 0x56b2, 0x56b3, 0x56b4,
212 0x56b5, 0x56b6, 0x56b8, 0x56b9, 0x56ba, 0x56bb, 0x56bd, 0x56be,
213 0x56bf, 0x56c0, 0x56c1, 0x56c2, 0x56c3, 0x56c4, 0x56c5, 0x56c6,
214 0x56c7, 0x56c8, 0x56c9, 0x56cb, 0x56cc, 0x56cd, 0x56ce, 0x56cf,
215 0x56d0, 0x56d1, 0x56d2, 0x56d3, 0x56d5, 0x56d6, 0x56d8, 0x56d9,
216 0x56dc, 0x56e3, 0x56e5, 0x56e6, 0x56e7, 0x56e8, 0x56e9, 0x56ea,
217 0x56ec, 0x56ee, 0x56ef, 0x56f2, 0x56f3, 0x56f6, 0x56f7, 0x56f8,
218 0x56fb, 0x56fc, 0x5700, 0x5701, 0x5702, 0x5705, 0x5707, 0x570b,
219 0x570c, 0x570d, 0x570e, 0x570f, 0x5710, 0x5711,
220 /* 0x88 */
221 0x5712, 0x5713, 0x5714, 0x5715, 0x5716, 0x5717, 0x5718, 0x5719,
222 0x571a, 0x571b, 0x571d, 0x571e, 0x5720, 0x5721, 0x5722, 0x5724,
223 0x5725, 0x5726, 0x5727, 0x572b, 0x5731, 0x5732, 0x5734, 0x5735,
224 0x5736, 0x5737, 0x5738, 0x573c, 0x573d, 0x573f, 0x5741, 0x5743,
225 0x5744, 0x5745, 0x5746, 0x5748, 0x5749, 0x574b, 0x5752, 0x5753,
226 0x5754, 0x5755, 0x5756, 0x5758, 0x5759, 0x5762, 0x5763, 0x5765,
227 0x5767, 0x576c, 0x576e, 0x5770, 0x5771, 0x5772, 0x5774, 0x5775,
228 0x5778, 0x5779, 0x577a, 0x577d, 0x577e, 0x577f, 0x5780, 0x5781,
229 0x5787, 0x5788, 0x5789, 0x578a, 0x578d, 0x578e, 0x578f, 0x5790,
230 0x5791, 0x5794, 0x5795, 0x5796, 0x5797, 0x5798, 0x5799, 0x579a,
231 0x579c, 0x579d, 0x579e, 0x579f, 0x57a5, 0x57a8, 0x57aa, 0x57ac,
232 0x57af, 0x57b0, 0x57b1, 0x57b3, 0x57b5, 0x57b6, 0x57b7, 0x57b9,
233 0x57ba, 0x57bb, 0x57bc, 0x57bd, 0x57be, 0x57bf, 0x57c0, 0x57c1,
234 0x57c4, 0x57c5, 0x57c6, 0x57c7, 0x57c8, 0x57c9, 0x57ca, 0x57cc,
235 0x57cd, 0x57d0, 0x57d1, 0x57d3, 0x57d6, 0x57d7, 0x57db, 0x57dc,
236 0x57de, 0x57e1, 0x57e2, 0x57e3, 0x57e5, 0x57e6, 0x57e7, 0x57e8,
237 0x57e9, 0x57ea, 0x57eb, 0x57ec, 0x57ee, 0x57f0, 0x57f1, 0x57f2,
238 0x57f3, 0x57f5, 0x57f6, 0x57f7, 0x57fb, 0x57fc, 0x57fe, 0x57ff,
239 0x5801, 0x5803, 0x5804, 0x5805, 0x5808, 0x5809, 0x580a, 0x580c,
240 0x580e, 0x580f, 0x5810, 0x5812, 0x5813, 0x5814, 0x5816, 0x5817,
241 0x5818, 0x581a, 0x581b, 0x581c, 0x581d, 0x581f, 0x5822, 0x5823,
242 0x5825, 0x5826, 0x5827, 0x5828, 0x5829, 0x582b, 0x582c, 0x582d,
243 0x582e, 0x582f, 0x5831, 0x5832, 0x5833, 0x5834, 0x5836, 0x5837,
244 0x5838, 0x5839, 0x583a, 0x583b, 0x583c, 0x583d,
245 /* 0x89 */
246 0x583e, 0x583f, 0x5840, 0x5841, 0x5842, 0x5843, 0x5845, 0x5846,
247 0x5847, 0x5848, 0x5849, 0x584a, 0x584b, 0x584e, 0x584f, 0x5850,
248 0x5852, 0x5853, 0x5855, 0x5856, 0x5857, 0x5859, 0x585a, 0x585b,
249 0x585c, 0x585d, 0x585f, 0x5860, 0x5861, 0x5862, 0x5863, 0x5864,
250 0x5866, 0x5867, 0x5868, 0x5869, 0x586a, 0x586d, 0x586e, 0x586f,
251 0x5870, 0x5871, 0x5872, 0x5873, 0x5874, 0x5875, 0x5876, 0x5877,
252 0x5878, 0x5879, 0x587a, 0x587b, 0x587c, 0x587d, 0x587f, 0x5882,
253 0x5884, 0x5886, 0x5887, 0x5888, 0x588a, 0x588b, 0x588c, 0x588d,
254 0x588e, 0x588f, 0x5890, 0x5891, 0x5894, 0x5895, 0x5896, 0x5897,
255 0x5898, 0x589b, 0x589c, 0x589d, 0x58a0, 0x58a1, 0x58a2, 0x58a3,
256 0x58a4, 0x58a5, 0x58a6, 0x58a7, 0x58aa, 0x58ab, 0x58ac, 0x58ad,
257 0x58ae, 0x58af, 0x58b0, 0x58b1, 0x58b2, 0x58b3, 0x58b4, 0x58b5,
258 0x58b6, 0x58b7, 0x58b8, 0x58b9, 0x58ba, 0x58bb, 0x58bd, 0x58be,
259 0x58bf, 0x58c0, 0x58c2, 0x58c3, 0x58c4, 0x58c6, 0x58c7, 0x58c8,
```

```
260 0x58c9, 0x58ca, 0x58cb, 0x58cc, 0x58cd, 0x58ce, 0x58cf, 0x58d0,
261 0x58d2, 0x58d3, 0x58d4, 0x58d6, 0x58d7, 0x58d8, 0x58d9, 0x58da,
262 0x58db, 0x58dc, 0x58dd, 0x58de, 0x58df, 0x58e0, 0x58e1, 0x58e2,
263 0x58e3, 0x58e5, 0x58e6, 0x58e7, 0x58e8, 0x58e9, 0x58ea, 0x58ed,
264 0x58ef, 0x58f1, 0x58f2, 0x58f4, 0x58f5, 0x58f7, 0x58f8, 0x58fa,
265 0x58fb, 0x58fc, 0x58fd, 0x58fe, 0x58ff, 0x5900, 0x5901, 0x5903,
266 0x5905, 0x5906, 0x5908, 0x5909, 0x590a, 0x590b, 0x590c, 0x590e,
267 0x5910, 0x5911, 0x5912, 0x5913, 0x5917, 0x5918, 0x591b, 0x591d,
268 0x591e, 0x5920, 0x5921, 0x5922, 0x5923, 0x5926, 0x5928, 0x592c,
269 0x5930, 0x5932, 0x5933, 0x5935, 0x5936, 0x593b,
270 /* 0x8a */
271 0x593d, 0x593e, 0x593f, 0x5940, 0x5943, 0x5945, 0x5946, 0x594a,
272 0x594c, 0x594d, 0x5950, 0x5952, 0x5953, 0x5959, 0x595b, 0x595c,
273 0x595d, 0x595e, 0x595f, 0x5961, 0x5963, 0x5964, 0x5966, 0x5967,
274 0x5968, 0x5969, 0x596a, 0x596b, 0x596c, 0x596d, 0x596e, 0x596f,
275 0x5970, 0x5971, 0x5972, 0x5975, 0x5977, 0x597a, 0x597b, 0x597c,
276 0x597e, 0x597f, 0x5980, 0x5985, 0x5989, 0x598b, 0x598c, 0x598e,
277 0x598f, 0x5990, 0x5991, 0x5994, 0x5995, 0x5998, 0x599a, 0x599b,
278 0x599c, 0x599d, 0x599f, 0x59a0, 0x59a1, 0x59a2, 0x59a6, 0x59a7,
279 0x59ac, 0x59ad, 0x59b0, 0x59b1, 0x59b3, 0x59b4, 0x59b5, 0x59b6,
280 0x59b7, 0x59b8, 0x59ba, 0x59bc, 0x59bd, 0x59bf, 0x59c0, 0x59c1,
281 0x59c2, 0x59c3, 0x59c4, 0x59c5, 0x59c7, 0x59c8, 0x59c9, 0x59cc,
282 0x59cd, 0x59ce, 0x59cf, 0x59d5, 0x59d6, 0x59d9, 0x59db, 0x59de,
283 0x59df, 0x59e0, 0x59e1, 0x59e2, 0x59e4, 0x59e6, 0x59e7, 0x59e9,
284 0x59ea, 0x59eb, 0x59ed, 0x59ee, 0x59ef, 0x59f0, 0x59f1, 0x59f2,
285 0x59f3, 0x59f4, 0x59f5, 0x59f6, 0x59f7, 0x59f8, 0x59fa, 0x59fc,
286 0x59fd, 0x59fe, 0x5a00, 0x5a02, 0x5a0a, 0x5a0b, 0x5a0d, 0x5a0e,
287 0x5a0f, 0x5a10, 0x5a12, 0x5a14, 0x5a15, 0x5a16, 0x5a17, 0x5a19,
288 0x5a1a, 0x5a1b, 0x5a1d, 0x5a1e, 0x5a21, 0x5a22, 0x5a24, 0x5a26,
289 0x5a27, 0x5a28, 0x5a2a, 0x5a2b, 0x5a2c, 0x5a2d, 0x5a2e, 0x5a2f,
290 0x5a30, 0x5a33, 0x5a35, 0x5a37, 0x5a38, 0x5a39, 0x5a3a, 0x5a3b,
291 0x5a3d, 0x5a3e, 0x5a3f, 0x5a41, 0x5a42, 0x5a43, 0x5a44, 0x5a45,
292 0x5a47, 0x5a48, 0x5a4b, 0x5a4c, 0x5a4d, 0x5a4e, 0x5a4f, 0x5a50,
293 0x5a51, 0x5a52, 0x5a53, 0x5a54, 0x5a56, 0x5a57, 0x5a58, 0x5a59,
294 0x5a5b, 0x5a5c, 0x5a5d, 0x5a5e, 0x5a5f, 0x5a60,
295 /* 0x8b */
296 0x5a61, 0x5a63, 0x5a64, 0x5a65, 0x5a66, 0x5a68, 0x5a69, 0x5a6b,
297 0x5a6c, 0x5a6d, 0x5a6e, 0x5a6f, 0x5a70, 0x5a71, 0x5a72, 0x5a73,
298 0x5a78, 0x5a79, 0x5a7b, 0x5a7c, 0x5a7d, 0x5a7e, 0x5a80, 0x5a81,
299 0x5a82, 0x5a83, 0x5a84, 0x5a85, 0x5a86, 0x5a87, 0x5a88, 0x5a89,
300 0x5a8a, 0x5a8b, 0x5a8c, 0x5a8d, 0x5a8e, 0x5a8f, 0x5a90, 0x5a91,
301 0x5a93, 0x5a94, 0x5a95, 0x5a96, 0x5a97, 0x5a98, 0x5a99, 0x5a9c,
302 0x5a9d, 0x5a9e, 0x5a9f, 0x5aa0, 0x5aa1, 0x5aa2, 0x5aa3, 0x5aa4,
303 0x5aa5, 0x5aa6, 0x5aa7, 0x5aa8, 0x5aa9, 0x5aab, 0x5aac, 0x5aad,
304 0x5aae, 0x5aaf, 0x5ab0, 0x5ab1, 0x5ab4, 0x5ab6, 0x5ab7, 0x5ab9,
305 0x5aba, 0x5abb, 0x5abc, 0x5abd, 0x5abf, 0x5ac0, 0x5ac3, 0x5ac4,
306 0x5ac5, 0x5ac6, 0x5ac7, 0x5ac8, 0x5aca, 0x5acb, 0x5acd, 0x5ace,
307 0x5acf, 0x5ad0, 0x5ad1, 0x5ad3, 0x5ad5, 0x5ad7, 0x5ad9, 0x5ada,
308 0x5adb, 0x5add, 0x5ade, 0x5adf, 0x5ae2, 0x5ae4, 0x5ae5, 0x5ae7,
309 0x5aeb, 0x5aea, 0x5aef, 0x5aed, 0x5aee, 0x5aef, 0x5af0, 0x5af2,
310 0x5af3, 0x5af4, 0x5af5, 0x5af6, 0x5af7, 0x5af8, 0x5af9, 0x5afa,
311 0x5afb, 0x5afc, 0x5afd, 0x5afe, 0x5aff, 0x5b00, 0x5b01, 0x5b02,
312 0x5b03, 0x5b04, 0x5b05, 0x5b06, 0x5b07, 0x5b08, 0x5b0a, 0x5b0b,
313 0x5b0c, 0x5b0d, 0x5b0e, 0x5b0f, 0x5b10, 0x5b11, 0x5b12, 0x5b13,
314 0x5b14, 0x5b15, 0x5b18, 0x5b19, 0x5b1a, 0x5b1b, 0x5b1c, 0x5b1d,
315 0x5b1e, 0x5b1f, 0x5b20, 0x5b21, 0x5b22, 0x5b23, 0x5b24, 0x5b25,
316 0x5b26, 0x5b27, 0x5b28, 0x5b29, 0x5b2a, 0x5b2b, 0x5b2c, 0x5b2d,
317 0x5b2e, 0x5b2f, 0x5b30, 0x5b31, 0x5b33, 0x5b35, 0x5b36, 0x5b38,
318 0x5b39, 0x5b3a, 0x5b3b, 0x5b3c, 0x5b3d, 0x5b3e, 0x5b3f, 0x5b41,
319 0x5b42, 0x5b43, 0x5b44, 0x5b45, 0x5b46, 0x5b47,
320 /* 0x8c */
321 0x5b48, 0x5b49, 0x5b4a, 0x5b4b, 0x5b4c, 0x5b4d, 0x5b4e, 0x5b4f,
322 0x5b52, 0x5b56, 0x5b5e, 0x5b60, 0x5b61, 0x5b67, 0x5b68, 0x5b6b,
323 0x5b6d, 0x5b6e, 0x5b6f, 0x5b72, 0x5b74, 0x5b76, 0x5b77, 0x5b78,
324 0x5b79, 0x5b7b, 0x5b7c, 0x5b7e, 0x5b7f, 0x5b82, 0x5b86, 0x5b8a,
325 0x5b8d, 0x5b8e, 0x5b90, 0x5b91, 0x5b92, 0x5b94, 0x5b96, 0x5b9f,
326 0x5ba7, 0x5ba8, 0x5ba9, 0x5bac, 0x5bad, 0x5bae, 0x5baf, 0x5bb1,
327 0x5bb2, 0x5bb7, 0x5bba, 0x5bbb, 0x5bbc, 0x5bc0, 0x5bc1, 0x5bc3,
328 0x5bc8, 0x5bc9, 0x5bca, 0x5bcb, 0x5bcd, 0x5bce, 0x5bcf, 0x5bd1,
329 0x5bd4, 0x5bd5, 0x5bd6, 0x5bd7, 0x5bd8, 0x5bd9, 0x5bda, 0x5bdb,
330 0x5bdc, 0x5be0, 0x5be2, 0x5be3, 0x5be6, 0x5be7, 0x5be9, 0x5bea,
331 0x5beb, 0x5bec, 0x5bed, 0x5bef, 0x5bf1, 0x5bf2, 0x5bf3, 0x5bf4,
332 0x5bf5, 0x5bf6, 0x5bf7, 0x5bfd, 0x5bfe, 0x5c00, 0x5c02, 0x5c03,
333 0x5c05, 0x5c07, 0x5c08, 0x5c0b, 0x5c0c, 0x5c0d, 0x5c0e, 0x5c10,
334 0x5c12, 0x5c13, 0x5c17, 0x5c19, 0x5c1b, 0x5c1e, 0x5c1f, 0x5c20,
335 0x5c21, 0x5c23, 0x5c26, 0x5c28, 0x5c29, 0x5c2a, 0x5c2b, 0x5c2d,
336 0x5c2e, 0x5c2f, 0x5c30, 0x5c32, 0x5c33, 0x5c35, 0x5c36, 0x5c37,
337 0x5c43, 0x5c44, 0x5c46, 0x5c47, 0x5c4c, 0x5c4d, 0x5c52, 0x5c53,
338 0x5c54, 0x5c56, 0x5c57, 0x5c58, 0x5c5a, 0x5c5b, 0x5c5c, 0x5c5d,
339 0x5c5f, 0x5c62, 0x5c64, 0x5c67, 0x5c68, 0x5c69, 0x5c6a, 0x5c6b,
340 0x5c6c, 0x5c6d, 0x5c70, 0x5c72, 0x5c73, 0x5c74, 0x5c75, 0x5c76,
341 0x5c77, 0x5c78, 0x5c7b, 0x5c7c, 0x5c7d, 0x5c7e, 0x5c80, 0x5c83,
342 0x5c84, 0x5c85, 0x5c86, 0x5c87, 0x5c89, 0x5c8a, 0x5c8b, 0x5c8e,
343 0x5c8f, 0x5c92, 0x5c93, 0x5c95, 0x5c9d, 0x5c9e, 0x5c9f, 0x5ca0,
344 0x5ca1, 0x5ca4, 0x5ca5, 0x5ca6, 0x5ca7, 0x5ca8,
345 /* 0x8d */
346 0x5caa, 0x5cae, 0x5caf, 0x5cb0, 0x5cb2, 0x5cb4, 0x5cb6, 0x5cb9,
```

```
347 0x5cba, 0x5cbb, 0x5cbc, 0x5cbe, 0x5cc0, 0x5cc2, 0x5cc3, 0x5cc5,
348 0x5cc6, 0x5cc7, 0x5cc8, 0x5cc9, 0x5cca, 0x5ccc, 0x5ccd, 0x5cce,
349 0x5ccf, 0x5cd0, 0x5cd1, 0x5cd3, 0x5cd4, 0x5cd5, 0x5cd6, 0x5cd7,
350 0x5cd8, 0x5cda, 0x5cdb, 0x5cdc, 0x5cdd, 0x5cde, 0x5cdf, 0x5ce0,
351 0x5ce2, 0x5ce3, 0x5ce7, 0x5ce9, 0x5ceb, 0x5cec, 0x5cee, 0x5cef,
352 0x5cf1, 0x5cf2, 0x5cf3, 0x5cf4, 0x5cf5, 0x5cf6, 0x5cf7, 0x5cf8,
353 0x5cf9, 0x5cfa, 0x5cfc, 0x5cfd, 0x5cfe, 0x5cff, 0x5d00, 0x5d01,
354 0x5d04, 0x5d05, 0x5d08, 0x5d09, 0x5d0a, 0x5d0b, 0x5d0c, 0x5d0d,
355 0x5d0f, 0x5d10, 0x5d11, 0x5d12, 0x5d13, 0x5d15, 0x5d17, 0x5d18,
356 0x5d19, 0x5d1a, 0x5d1c, 0x5d1d, 0x5d1f, 0x5d20, 0x5d21, 0x5d22,
357 0x5d23, 0x5d25, 0x5d28, 0x5d2a, 0x5d2b, 0x5d2c, 0x5d2f, 0x5d30,
358 0x5d31, 0x5d32, 0x5d33, 0x5d35, 0x5d36, 0x5d37, 0x5d38, 0x5d39,
359 0x5d3a, 0x5d3b, 0x5d3c, 0x5d3f, 0x5d40, 0x5d41, 0x5d42, 0x5d43,
360 0x5d44, 0x5d45, 0x5d46, 0x5d48, 0x5d49, 0x5d4d, 0x5d4e, 0x5d4f,
361 0x5d50, 0x5d51, 0x5d52, 0x5d53, 0x5d54, 0x5d55, 0x5d56, 0x5d57,
362 0x5d59, 0x5d5a, 0x5d5c, 0x5d5e, 0x5d5f, 0x5d60, 0x5d61, 0x5d62,
363 0x5d63, 0x5d64, 0x5d65, 0x5d66, 0x5d67, 0x5d68, 0x5d6a, 0x5d6d,
364 0x5d6e, 0x5d70, 0x5d71, 0x5d72, 0x5d73, 0x5d75, 0x5d76, 0x5d77,
365 0x5d78, 0x5d79, 0x5d7a, 0x5d7b, 0x5d7c, 0x5d7d, 0x5d7e, 0x5d7f,
366 0x5d80, 0x5d81, 0x5d83, 0x5d84, 0x5d85, 0x5d86, 0x5d87, 0x5d88,
367 0x5d89, 0x5d8a, 0x5d8b, 0x5d8c, 0x5d8d, 0x5d8e, 0x5d8f, 0x5d90,
368 0x5d91, 0x5d92, 0x5d93, 0x5d94, 0x5d95, 0x5d96, 0x5d97, 0x5d98,
369 0x5d9a, 0x5d9b, 0x5d9c, 0x5d9e, 0x5d9f, 0x5da0,
370 /* 0x8e */
371 0x5da1, 0x5da2, 0x5da3, 0x5da4, 0x5da5, 0x5da6, 0x5da7, 0x5da8,
372 0x5da9, 0x5daa, 0x5dab, 0x5dac, 0x5dad, 0x5dae, 0x5daf, 0x5db0,
373 0x5db1, 0x5db2, 0x5db3, 0x5db4, 0x5db5, 0x5db6, 0x5db8, 0x5db9,
374 0x5dba, 0x5dbb, 0x5dbc, 0x5dbd, 0x5dbe, 0x5dbf, 0x5dc0, 0x5dc1,
375 0x5dc2, 0x5dc3, 0x5dc4, 0x5dc6, 0x5dc7, 0x5dc8, 0x5dc9, 0x5dca,
376 0x5dcb, 0x5dcc, 0x5dce, 0x5dcf, 0x5dd0, 0x5dd1, 0x5dd2, 0x5dd3,
377 0x5dd4, 0x5dd5, 0x5dd6, 0x5dd7, 0x5dd8, 0x5dd9, 0x5dda, 0x5ddc,
378 0x5ddf, 0x5de0, 0x5de3, 0x5de4, 0x5dea, 0x5dec, 0x5ded, 0x5df0,
379 0x5df5, 0x5df6, 0x5df8, 0x5df9, 0x5dfa, 0x5dfb, 0x5dfc, 0x5dff,
380 0x5e00, 0x5e04, 0x5e07, 0x5e09, 0x5e0a, 0x5e0b, 0x5e0d, 0x5e0e,
381 0x5e12, 0x5e13, 0x5e17, 0x5e1e, 0x5e1f, 0x5e20, 0x5e21, 0x5e22,
382 0x5e23, 0x5e24, 0x5e25, 0x5e28, 0x5e29, 0x5e2a, 0x5e2b, 0x5e2c,
383 0x5e2f, 0x5e30, 0x5e32, 0x5e33, 0x5e34, 0x5e35, 0x5e36, 0x5e39,
384 0x5e3a, 0x5e3e, 0x5e3f, 0x5e40, 0x5e41, 0x5e43, 0x5e46, 0x5e47,
385 0x5e48, 0x5e49, 0x5e4a, 0x5e4b, 0x5e4d, 0x5e4e, 0x5e4f, 0x5e50,
386 0x5e51, 0x5e52, 0x5e53, 0x5e56, 0x5e57, 0x5e58, 0x5e59, 0x5e5a,
387 0x5e5c, 0x5e5d, 0x5e5f, 0x5e60, 0x5e63, 0x5e64, 0x5e65, 0x5e66,
388 0x5e67, 0x5e68, 0x5e69, 0x5e6a, 0x5e6b, 0x5e6c, 0x5e6d, 0x5e6e,
389 0x5e6f, 0x5e70, 0x5e71, 0x5e75, 0x5e77, 0x5e79, 0x5e7e, 0x5e81,
390 0x5e82, 0x5e83, 0x5e85, 0x5e88, 0x5e89, 0x5e8c, 0x5e8d, 0x5e8e,
391 0x5e92, 0x5e98, 0x5e9b, 0x5e9d, 0x5ea1, 0x5ea2, 0x5ea3, 0x5ea4,
392 0x5ea8, 0x5ea9, 0x5eaa, 0x5eab, 0x5eac, 0x5eae, 0x5eaf, 0x5eb0,
393 0x5eb1, 0x5eb2, 0x5eb4, 0x5eba, 0x5ebb, 0x5ebc, 0x5ebd, 0x5ebf,
394 0x5ec0, 0x5ec1, 0x5ec2, 0x5ec3, 0x5ec4, 0x5ec5,
395 /* 0x8f */
396 0x5ec6, 0x5ec7, 0x5ec8, 0x5ecb, 0x5ecc, 0x5ecd, 0x5ece, 0x5ecf,
397 0x5ed0, 0x5ed4, 0x5ed5, 0x5ed7, 0x5ed8, 0x5ed9, 0x5eda, 0x5edc,
398 0x5edd, 0x5ede, 0x5edf, 0x5ee0, 0x5ee1, 0x5ee2, 0x5ee3, 0x5ee4,
399 0x5ee5, 0x5ee6, 0x5ee7, 0x5ee9, 0x5eeb, 0x5eec, 0x5eed, 0x5eee,
400 0x5eef, 0x5ef0, 0x5ef1, 0x5ef2, 0x5ef3, 0x5ef5, 0x5ef8, 0x5ef9,
401 0x5efb, 0x5efc, 0x5efd, 0x5f05, 0x5f06, 0x5f07, 0x5f09, 0x5f0c,
402 0x5f0d, 0x5f10, 0x5f12, 0x5f14, 0x5f16, 0x5f19, 0x5f1a,
403 0x5f1c, 0x5f1d, 0x5f1e, 0x5f21, 0x5f22, 0x5f23, 0x5f24, 0x5f28,
404 0x5f2b, 0x5f2c, 0x5f2e, 0x5f30, 0x5f32, 0x5f33, 0x5f34, 0x5f35,
405 0x5f36, 0x5f37, 0x5f38, 0x5f3b, 0x5f3d, 0x5f3e, 0x5f3f, 0x5f41,
406 0x5f42, 0x5f43, 0x5f44, 0x5f45, 0x5f46, 0x5f47, 0x5f48, 0x5f49,
407 0x5f4a, 0x5f4b, 0x5f4c, 0x5f4d, 0x5f4e, 0x5f4f, 0x5f51, 0x5f54,
408 0x5f59, 0x5f5a, 0x5f5b, 0x5f5c, 0x5f5e, 0x5f5f, 0x5f60, 0x5f63,
409 0x5f65, 0x5f67, 0x5f68, 0x5f6b, 0x5f6e, 0x5f6f, 0x5f72, 0x5f74,
410 0x5f75, 0x5f76, 0x5f78, 0x5f7a, 0x5f7d, 0x5f7e, 0x5f7f, 0x5f83,
411 0x5f86, 0x5f8d, 0x5f8e, 0x5f8f, 0x5f91, 0x5f93, 0x5f94, 0x5f96,
412 0x5f9a, 0x5f9b, 0x5f9d, 0x5f9e, 0x5f9f, 0x5fa0, 0x5fa2, 0x5fa3,
413 0x5fa4, 0x5fa5, 0x5fa6, 0x5fa7, 0x5fa9, 0x5fab, 0x5fac, 0x5faf,
414 0x5fb0, 0x5fb1, 0x5fb2, 0x5fb3, 0x5fb4, 0x5fb6, 0x5fb8, 0x5fb9,
415 0x5fba, 0x5fbb, 0x5fbc, 0x5fbf, 0x5fc0, 0x5fc1, 0x5fc2, 0x5fc7,
416 0x5fc8, 0x5fca, 0x5fcb, 0x5fce, 0x5fd3, 0x5fd4, 0x5fd5, 0x5fda,
417 0x5fdb, 0x5fdc, 0x5fde, 0x5fdf, 0x5fe2, 0x5fe3, 0x5fe5, 0x5fe6,
418 0x5fe8, 0x5fe9, 0x5fec, 0x5fef, 0x5ff0, 0x5ff2, 0x5ff3, 0x5ff4,
419 0x5ff6, 0x5ff7, 0x5ff9, 0x5ffa, 0x5ffc, 0x6007,
420 /* 0x90 */
421 0x6008, 0x6009, 0x600b, 0x600c, 0x6010, 0x6011, 0x6013, 0x6017,
422 0x6018, 0x601a, 0x601e, 0x601f, 0x6022, 0x6023, 0x6024, 0x602c,
423 0x602d, 0x602e, 0x6030, 0x6031, 0x6032, 0x6033, 0x6034, 0x6036,
424 0x6037, 0x6038, 0x6039, 0x603a, 0x603d, 0x603e, 0x6040, 0x6044,
425 0x6045, 0x6046, 0x6047, 0x6048, 0x6049, 0x604a, 0x604c, 0x604e,
426 0x604f, 0x6051, 0x6053, 0x6054, 0x6056, 0x6057, 0x6058, 0x605b,
427 0x605c, 0x605e, 0x605f, 0x6060, 0x6061, 0x6065, 0x6066, 0x606e,
428 0x6071, 0x6072, 0x6074, 0x6075, 0x6077, 0x607e, 0x6080, 0x6081,
429 0x6082, 0x6085, 0x6086, 0x6087, 0x6088, 0x608a, 0x608b, 0x608e,
430 0x608f, 0x6090, 0x6091, 0x6093, 0x6095, 0x6097, 0x6098, 0x6099,
431 0x609c, 0x609e, 0x60a1, 0x60a2, 0x60a4, 0x60a5, 0x60a7, 0x60a9,
432 0x60aa, 0x60ae, 0x60b0, 0x60b3, 0x60b5, 0x60b6, 0x60b7, 0x60b9,
433 0x60ba, 0x60bd, 0x60be, 0x60bf, 0x60c0, 0x60c1, 0x60c2, 0x60c3,
```

```
434 0x60c4, 0x60c7, 0x60c8, 0x60c9, 0x60cc, 0x60cd, 0x60ce, 0x60cf,
435 0x60d0, 0x60d2, 0x60d3, 0x60d4, 0x60d6, 0x60d7, 0x60d9, 0x60db,
436 0x60de, 0x60e1, 0x60e2, 0x60e3, 0x60e4, 0x60e5, 0x60ea, 0x60f1,
437 0x60f2, 0x60f5, 0x60f7, 0x60f8, 0x60fb, 0x60fc, 0x60fd, 0x60fe,
438 0x60ff, 0x6102, 0x6103, 0x6104, 0x6105, 0x6107, 0x610a, 0x610b,
439 0x610c, 0x6110, 0x6111, 0x6111, 0x6112, 0x6113, 0x6114, 0x6116, 0x6117,
440 0x6118, 0x6119, 0x611b, 0x611c, 0x611d, 0x611e, 0x6121, 0x6122,
441 0x6125, 0x6128, 0x6129, 0x612a, 0x612c, 0x612d, 0x612e, 0x612f,
442 0x6130, 0x6131, 0x6132, 0x6132, 0x6133, 0x6134, 0x6135, 0x6136, 0x6137,
443 0x6138, 0x6139, 0x613a, 0x613b, 0x613c, 0x613d, 0x613e, 0x6140,
444 0x6141, 0x6142, 0x6143, 0x6144, 0x6145, 0x6146,
445 /* 0x91 */
446 0x6147, 0x6149, 0x614b, 0x614d, 0x614f, 0x6150, 0x6152, 0x6153,
447 0x6154, 0x6156, 0x6157, 0x6158, 0x6159, 0x615a, 0x615b, 0x615c,
448 0x615e, 0x615f, 0x6160, 0x6161, 0x6163, 0x6164, 0x6165, 0x6166,
449 0x6169, 0x616a, 0x616b, 0x616c, 0x616d, 0x616e, 0x616f, 0x6171,
450 0x6172, 0x6173, 0x6174, 0x6176, 0x6178, 0x6179, 0x617a, 0x617b,
451 0x617c, 0x617d, 0x617e, 0x617e, 0x617f, 0x6180, 0x6181, 0x6182, 0x6183,
452 0x6184, 0x6185, 0x6186, 0x6187, 0x6188, 0x6189, 0x618a, 0x618c,
453 0x618d, 0x618f, 0x6190, 0x6191, 0x6192, 0x6193, 0x6195, 0x6196,
454 0x6197, 0x6198, 0x6199, 0x6199, 0x619a, 0x619b, 0x619c, 0x619e, 0x619f,
455 0x61a0, 0x61a1, 0x61a2, 0x61a3, 0x61a4, 0x61a5, 0x61a6, 0x61aa,
456 0x61ab, 0x61ad, 0x61ae, 0x61af, 0x61b0, 0x61b1, 0x61b2, 0x61b3,
457 0x61b4, 0x61b5, 0x61b6, 0x61b6, 0x61b8, 0x61b9, 0x61ba, 0x61bb, 0x61bc,
458 0x61bd, 0x61bf, 0x61c0, 0x61c1, 0x61c3, 0x61c4, 0x61c5, 0x61c6,
459 0x61c7, 0x61c9, 0x61cc, 0x61cd, 0x61ce, 0x61cf, 0x61d0, 0x61d3,
460 0x61d5, 0x61d6, 0x61d7, 0x61d8, 0x61d9, 0x61da, 0x61db, 0x61dc,
461 0x61dd, 0x61de, 0x61df, 0x61e0, 0x61e1, 0x61e2, 0x61e3, 0x61e4,
462 0x61e5, 0x61e7, 0x61e8, 0x61e9, 0x61ea, 0x61eb, 0x61ec, 0x61ed,
463 0x61ee, 0x61ef, 0x61f0, 0x61f1, 0x61f2, 0x61f3, 0x61f4, 0x61f6,
464 0x61f7, 0x61f8, 0x61f9, 0x61fa, 0x61fb, 0x61fc, 0x61fd, 0x61fe,
465 0x6200, 0x6201, 0x6202, 0x6203, 0x6204, 0x6205, 0x6207, 0x6209,
466 0x6213, 0x6214, 0x6219, 0x6219, 0x621c, 0x621d, 0x621e, 0x6220, 0x6223,
467 0x6226, 0x6227, 0x6228, 0x6229, 0x622b, 0x622d, 0x622f, 0x6230,
468 0x6231, 0x6232, 0x6235, 0x6236, 0x6238, 0x6239, 0x623a, 0x623b,
469 0x623c, 0x6242, 0x6244, 0x6244, 0x6246, 0x624a,
470 /* 0x92 */
471 0x624f, 0x6250, 0x6255, 0x6256, 0x6257, 0x6259, 0x625a, 0x625c,
472 0x625d, 0x625e, 0x625f, 0x6260, 0x6261, 0x6262, 0x6264, 0x6265,
473 0x6268, 0x6271, 0x6272, 0x6274, 0x6275, 0x6277, 0x6278, 0x627a,
474 0x627b, 0x627d, 0x6281, 0x6282, 0x6283, 0x6285, 0x6286, 0x6287,
475 0x6288, 0x628b, 0x628c, 0x628c, 0x628d, 0x628e, 0x628f, 0x6290, 0x6294,
476 0x6299, 0x629c, 0x629d, 0x629e, 0x62a3, 0x62a6, 0x62a7, 0x62a9,
477 0x62aa, 0x62ad, 0x62ae, 0x62af, 0x62b0, 0x62b2, 0x62b3, 0x62b4,
478 0x62b6, 0x62b7, 0x62b8, 0x62ba, 0x62be, 0x62c0, 0x62c1, 0x62c3,
479 0x62cb, 0x62cf, 0x62d1, 0x62d5, 0x62dd, 0x62de, 0x62e0, 0x62e1,
480 0x62e4, 0x62ea, 0x62eb, 0x62f0, 0x62f2, 0x62f5, 0x62f8, 0x62f9,
481 0x62fa, 0x62fb, 0x6300, 0x6303, 0x6304, 0x6305, 0x6306, 0x630a,
482 0x630b, 0x630c, 0x630d, 0x630f, 0x6310, 0x6312, 0x6313, 0x6314,
483 0x6315, 0x6317, 0x6318, 0x6319, 0x631c, 0x6326, 0x6327, 0x6329,
484 0x632c, 0x632d, 0x632e, 0x6330, 0x6331, 0x6333, 0x6334, 0x6335,
485 0x6336, 0x6337, 0x6338, 0x633b, 0x633c, 0x633e, 0x633f, 0x6340,
486 0x6341, 0x6344, 0x6347, 0x6348, 0x634a, 0x6351, 0x6352, 0x6353,
487 0x6354, 0x6356, 0x6357, 0x6358, 0x6359, 0x635a, 0x635b, 0x635c,
488 0x635d, 0x6360, 0x6364, 0x6365, 0x6366, 0x6368, 0x636a, 0x636b,
489 0x636c, 0x636f, 0x6370, 0x6372, 0x6373, 0x6374, 0x6375, 0x6378,
490 0x6379, 0x637c, 0x637d, 0x637e, 0x637f, 0x6381, 0x6383, 0x6384,
491 0x6385, 0x6386, 0x638b, 0x638d, 0x6391, 0x6393, 0x6394, 0x6395,
492 0x6397, 0x6399, 0x639a, 0x639b, 0x639c, 0x639d, 0x639e, 0x639f,
493 0x63a1, 0x63a4, 0x63a6, 0x63ab, 0x63af, 0x63b1, 0x63b2, 0x63b5,
494 0x63b6, 0x63b9, 0x63bb, 0x63bd, 0x63bf, 0x63c0,
495 /* 0x93 */
496 0x63c1, 0x63c2, 0x63c3, 0x63c5, 0x63c7, 0x63c8, 0x63ca, 0x63cb,
497 0x63cc, 0x63d1, 0x63d3, 0x63d4, 0x63d5, 0x63d7, 0x63d8, 0x63d9,
498 0x63da, 0x63db, 0x63dc, 0x63dd, 0x63df, 0x63e2, 0x63e4, 0x63e5,
499 0x63e6, 0x63e7, 0x63e8, 0x63eb, 0x63ec, 0x63ee, 0x63ef, 0x63f0,
500 0x63f1, 0x63f3, 0x63f5, 0x63f7, 0x63f9, 0x63fa, 0x63fb, 0x63fc,
501 0x63fe, 0x6403, 0x6404, 0x6406, 0x6407, 0x6408, 0x6409, 0x640a,
502 0x640d, 0x640e, 0x6411, 0x6412, 0x6415, 0x6416, 0x6417, 0x6418,
503 0x6419, 0x641a, 0x641d, 0x641f, 0x6422, 0x6423, 0x6424, 0x6425,
504 0x6427, 0x6428, 0x6429, 0x642b, 0x642e, 0x642f, 0x6430, 0x6431,
505 0x6432, 0x6433, 0x6435, 0x6436, 0x6437, 0x6438, 0x6439, 0x643b,
506 0x643c, 0x643e, 0x6440, 0x6442, 0x6443, 0x6449, 0x644b, 0x644c,
507 0x644d, 0x644e, 0x644f, 0x6450, 0x6451, 0x6453, 0x6455, 0x6456,
508 0x6457, 0x6459, 0x645a, 0x645b, 0x645c, 0x645d, 0x645f, 0x6460,
509 0x6461, 0x6462, 0x6463, 0x6464, 0x6465, 0x6466, 0x6468, 0x646a,
510 0x646b, 0x646c, 0x646e, 0x646e, 0x6470, 0x6471, 0x6472, 0x6473,
511 0x6474, 0x6475, 0x6476, 0x6477, 0x647b, 0x647c, 0x647d, 0x647e,
512 0x647f, 0x6480, 0x6481, 0x6483, 0x6486, 0x6488, 0x6489, 0x648a,
513 0x648b, 0x648c, 0x648d, 0x648e, 0x648f, 0x6490, 0x6493, 0x6494,
514 0x6497, 0x6498, 0x649a, 0x649b, 0x649c, 0x649d, 0x649f, 0x64a0,
515 0x64a1, 0x64a2, 0x64a3, 0x64a5, 0x64a6, 0x64a7, 0x64a8, 0x64aa,
516 0x64ab, 0x64af, 0x64b1, 0x64b2, 0x64b3, 0x64b4, 0x64b6, 0x64b9,
517 0x64bb, 0x64bd, 0x64be, 0x64bf, 0x64c1, 0x64c3, 0x64c4, 0x64c6,
518 0x64c7, 0x64c8, 0x64c9, 0x64ca, 0x64cb, 0x64cc, 0x64cf, 0x64d1,
519 0x64d3, 0x64d4, 0x64d5, 0x64d6, 0x64d9, 0x64da,
520 /* 0x94 */
```

```
521 0x64db, 0x64dc, 0x64dd, 0x64df, 0x64e0, 0x64e1, 0x64e3, 0x64e5,
522 0x64e7, 0x64e8, 0x64e9, 0x64ea, 0x64eb, 0x64ec, 0x64ed, 0x64ee,
523 0x64ef, 0x64f0, 0x64f1, 0x64f2, 0x64f3, 0x64f4, 0x64f5, 0x64f6,
524 0x64f7, 0x64f8, 0x64f9, 0x64fa, 0x64fb, 0x64fc, 0x64fd, 0x64fe,
525 0x64ff, 0x6501, 0x6502, 0x6503, 0x6504, 0x6505, 0x6506, 0x6507,
526 0x6508, 0x650a, 0x650b, 0x650c, 0x650d, 0x650e, 0x650f, 0x6510,
527 0x6511, 0x6513, 0x6514, 0x6515, 0x6516, 0x6517, 0x6519, 0x651a,
528 0x651b, 0x651c, 0x651d, 0x651e, 0x651f, 0x6520, 0x6521, 0x6522,
529 0x6523, 0x6524, 0x6526, 0x6527, 0x6528, 0x6529, 0x652a, 0x652c,
530 0x652d, 0x6530, 0x6531, 0x6532, 0x6533, 0x6537, 0x653a, 0x653c,
531 0x653d, 0x6540, 0x6541, 0x6542, 0x6543, 0x6544, 0x6546, 0x6547,
532 0x654a, 0x654b, 0x654d, 0x654e, 0x6550, 0x6552, 0x6553, 0x6554,
533 0x6557, 0x6558, 0x655a, 0x655c, 0x655f, 0x6560, 0x6561, 0x6564,
534 0x6565, 0x6567, 0x6568, 0x6569, 0x656a, 0x656d, 0x656e, 0x656f,
535 0x6571, 0x6573, 0x6575, 0x6577, 0x6578, 0x6579, 0x657a, 0x657b,
536 0x657c, 0x657d, 0x657e, 0x657f, 0x6580, 0x6581, 0x6582, 0x6583,
537 0x6584, 0x6585, 0x6586, 0x6588, 0x6589, 0x658a, 0x658d, 0x658e,
538 0x658f, 0x6592, 0x6594, 0x6594, 0x6595, 0x6596, 0x6598, 0x659a, 0x659d,
539 0x659e, 0x65a0, 0x65a2, 0x65a3, 0x65a6, 0x65a8, 0x65aa, 0x65ac,
540 0x65ae, 0x65b1, 0x65b2, 0x65b3, 0x65b4, 0x65b5, 0x65b6, 0x65b7,
541 0x65b8, 0x65ba, 0x65bb, 0x65be, 0x65bf, 0x65c0, 0x65c2, 0x65c7,
542 0x65c8, 0x65c9, 0x65ca, 0x65cd, 0x65d0, 0x65d1, 0x65d3, 0x65d4,
543 0x65d5, 0x65d8, 0x65d9, 0x65da, 0x65db, 0x65dc, 0x65dd, 0x65de,
544 0x65df, 0x65e1, 0x65e3, 0x65e4, 0x65ea, 0x65eb,
545 /* 0x95 */
546 0x65f2, 0x65f3, 0x65f4, 0x65f5, 0x65f8, 0x65f9, 0x65fb, 0x65fc,
547 0x65fd, 0x65fe, 0x65ff, 0x6601, 0x6604, 0x6605, 0x6607, 0x6608,
548 0x6609, 0x660b, 0x660d, 0x6610, 0x6611, 0x6612, 0x6616, 0x6617,
549 0x6618, 0x661a, 0x661b, 0x661c, 0x661e, 0x6621, 0x6622, 0x6623,
550 0x6624, 0x6626, 0x6629, 0x662a, 0x662b, 0x662c, 0x662e, 0x6630,
551 0x6632, 0x6633, 0x6637, 0x6638, 0x6639, 0x663a, 0x663b, 0x663d,
552 0x663f, 0x6640, 0x6642, 0x6644, 0x6645, 0x6646, 0x6647, 0x6648,
553 0x6649, 0x664a, 0x664d, 0x664e, 0x6650, 0x6651, 0x6658, 0x6659,
554 0x665b, 0x665c, 0x665d, 0x665e, 0x6660, 0x6662, 0x6663, 0x6665,
555 0x6667, 0x6669, 0x666a, 0x666b, 0x666c, 0x666d, 0x6671, 0x6672,
556 0x6673, 0x6675, 0x6678, 0x6679, 0x667b, 0x667c, 0x667d, 0x667f,
557 0x6680, 0x6681, 0x6683, 0x6685, 0x6686, 0x6688, 0x6689, 0x668a,
558 0x668b, 0x668d, 0x668e, 0x668f, 0x6690, 0x6692, 0x6693, 0x6694,
559 0x6695, 0x6698, 0x6699, 0x669a, 0x669b, 0x669c, 0x669e, 0x669f,
560 0x66a0, 0x66a1, 0x66a2, 0x66a3, 0x66a4, 0x66a5, 0x66a6, 0x66a9,
561 0x66aa, 0x66ab, 0x66ac, 0x66ad, 0x66af, 0x66b0, 0x66b1, 0x66b2,
562 0x66b3, 0x66b5, 0x66b6, 0x66b7, 0x66b8, 0x66ba, 0x66bb, 0x66bc,
563 0x66bd, 0x66bf, 0x66c0, 0x66c1, 0x66c2, 0x66c3, 0x66c4, 0x66c5,
564 0x66c6, 0x66c7, 0x66c8, 0x66c9, 0x66ca, 0x66cb, 0x66cc, 0x66cd,
565 0x66ce, 0x66cf, 0x66d0, 0x66d1, 0x66d2, 0x66d3, 0x66d4, 0x66d5,
566 0x66d6, 0x66d7, 0x66d8, 0x66da, 0x66de, 0x66df, 0x66e0, 0x66e1,
567 0x66e2, 0x66e3, 0x66e4, 0x66e5, 0x66e7, 0x66e8, 0x66ea, 0x66eb,
568 0x66ec, 0x66ed, 0x66ee, 0x66ef, 0x66f1, 0x66f5, 0x66f6, 0x66f8,
569 0x66fa, 0x66fb, 0x66fd, 0x6701, 0x6702, 0x6703,
570 /* 0x96 */
571 0x6704, 0x6705, 0x6706, 0x6707, 0x670c, 0x670e, 0x670f, 0x6711,
572 0x6712, 0x6713, 0x6716, 0x6718, 0x6719, 0x671a, 0x671c, 0x671e,
573 0x6720, 0x6721, 0x6722, 0x6723, 0x6724, 0x6725, 0x6727, 0x6729,
574 0x672e, 0x6730, 0x6732, 0x6732, 0x6733, 0x6736, 0x6737, 0x6738, 0x6739,
575 0x673b, 0x673c, 0x673e, 0x673f, 0x6741, 0x6744, 0x6745, 0x6747,
576 0x674a, 0x674b, 0x674d, 0x6752, 0x6754, 0x6755, 0x6757, 0x6758,
577 0x6759, 0x675a, 0x675b, 0x675d, 0x675d, 0x6762, 0x6763, 0x6764, 0x6766,
578 0x6767, 0x676b, 0x676c, 0x676e, 0x6771, 0x6774, 0x6776, 0x6778,
579 0x6779, 0x677a, 0x677b, 0x677d, 0x6780, 0x6782, 0x6783, 0x6785,
580 0x6786, 0x6788, 0x678a, 0x678c, 0x678d, 0x678e, 0x678f, 0x6791,
581 0x6792, 0x6793, 0x6794, 0x6796, 0x6799, 0x679b, 0x679f, 0x67a0,
582 0x67a1, 0x67a4, 0x67a6, 0x67a9, 0x67ac, 0x67ae, 0x67b1, 0x67b2,
583 0x67b4, 0x67b9, 0x67ba, 0x67bb, 0x67bc, 0x67bd, 0x67be, 0x67bf,
584 0x67c0, 0x67c2, 0x67c5, 0x67c6, 0x67c7, 0x67c8, 0x67c9, 0x67ca,
585 0x67cb, 0x67cc, 0x67cd, 0x67ce, 0x67d5, 0x67d6, 0x67d7, 0x67db,
586 0x67df, 0x67e1, 0x67e3, 0x67e4, 0x67e6, 0x67e7, 0x67e8, 0x67ea,
587 0x67eb, 0x67ed, 0x67ee, 0x67f2, 0x67f5, 0x67f6, 0x67f7, 0x67f8,
588 0x67f9, 0x67fa, 0x67fb, 0x67fc, 0x67fe, 0x6801, 0x6802, 0x6803,
589 0x6804, 0x6806, 0x680d, 0x6810, 0x6812, 0x6814, 0x6815, 0x6818,
590 0x6819, 0x681a, 0x681b, 0x681c, 0x681e, 0x681f, 0x6820, 0x6822,
591 0x6823, 0x6824, 0x6825, 0x6826, 0x6827, 0x6828, 0x682b, 0x682c,
592 0x682d, 0x682e, 0x682f, 0x6830, 0x6831, 0x6834, 0x6835, 0x6836,
593 0x683a, 0x683b, 0x683f, 0x6847, 0x684b, 0x684d, 0x684f, 0x6852,
594 0x6856, 0x6857, 0x6858, 0x6859, 0x685a, 0x685b,
595 /* 0x97 */
596 0x685c, 0x685d, 0x685e, 0x685f, 0x686a, 0x686c, 0x686d, 0x686e,
597 0x686f, 0x6870, 0x6871, 0x6872, 0x6873, 0x6875, 0x6878, 0x6879,
598 0x687a, 0x687b, 0x687c, 0x687d, 0x687e, 0x687f, 0x6880, 0x6882,
599 0x6884, 0x6887, 0x6888, 0x6889, 0x688a, 0x688b, 0x688c, 0x688d,
600 0x688e, 0x6890, 0x6891, 0x6892, 0x6894, 0x6895, 0x6896, 0x6898,
601 0x6899, 0x689a, 0x689b, 0x689c, 0x689d, 0x689e, 0x689f, 0x68a0,
602 0x68a1, 0x68a3, 0x68a4, 0x68a5, 0x68a9, 0x68aa, 0x68ab, 0x68ac,
603 0x68ae, 0x68b1, 0x68b2, 0x68b4, 0x68b6, 0x68b7, 0x68b8, 0x68b9,
604 0x68ba, 0x68bb, 0x68bc, 0x68bd, 0x68be, 0x68bf, 0x68c1, 0x68c3,
605 0x68c4, 0x68c5, 0x68c6, 0x68c7, 0x68c8, 0x68ca, 0x68cc, 0x68ce,
606 0x68cf, 0x68d0, 0x68d1, 0x68d3, 0x68d4, 0x68d6, 0x68d7, 0x68d9,
607 0x68db, 0x68dc, 0x68dd, 0x68de, 0x68df, 0x68e1, 0x68e2, 0x68e4,
```



```
608 0x68e5, 0x68e6, 0x68e7, 0x68e8, 0x68e9, 0x68ea, 0x68eb, 0x68ec,
609 0x68ed, 0x68ef, 0x68f2, 0x68f3, 0x68f4, 0x68f6, 0x68f7, 0x68f8,
610 0x68fb, 0x68fd, 0x68fe, 0x68ff, 0x6900, 0x6902, 0x6903, 0x6904,
611 0x6906, 0x6907, 0x6908, 0x6909, 0x690a, 0x690c, 0x690f, 0x6911,
612 0x6913, 0x6914, 0x6915, 0x691e, 0x6917, 0x6918, 0x6919, 0x691a,
613 0x691b, 0x691c, 0x691d, 0x691e, 0x6921, 0x6922, 0x6923, 0x6925,
614 0x6926, 0x6927, 0x6928, 0x6929, 0x692a, 0x692b, 0x692c, 0x692e,
615 0x692f, 0x6931, 0x6932, 0x6933, 0x6935, 0x6936, 0x6937, 0x6938,
616 0x693a, 0x693b, 0x693c, 0x693e, 0x6940, 0x6941, 0x6943, 0x6944,
617 0x6945, 0x6946, 0x6947, 0x6948, 0x6949, 0x694a, 0x694b, 0x694c,
618 0x694d, 0x694e, 0x694f, 0x6950, 0x6951, 0x6952, 0x6953, 0x6955,
619 0x6956, 0x6958, 0x6959, 0x695b, 0x695c, 0x695f,
620 /* 0x98 */
621 0x6961, 0x6962, 0x6964, 0x6965, 0x6967, 0x6968, 0x6969, 0x696a,
622 0x696c, 0x696d, 0x696f, 0x6970, 0x6972, 0x6973, 0x6974, 0x6975,
623 0x6976, 0x697a, 0x697b, 0x697d, 0x697e, 0x697f, 0x6981, 0x6983,
624 0x6985, 0x698a, 0x698b, 0x698c, 0x698e, 0x698f, 0x6990, 0x6991,
625 0x6992, 0x6993, 0x6996, 0x6997, 0x6999, 0x699a, 0x699d, 0x699e,
626 0x699f, 0x69a0, 0x69a1, 0x69a2, 0x69a3, 0x69a4, 0x69a5, 0x69a6,
627 0x69a9, 0x69aa, 0x69ac, 0x69ae, 0x69af, 0x69b0, 0x69b2, 0x69b3,
628 0x69b5, 0x69b6, 0x69b8, 0x69b9, 0x69ba, 0x69bc, 0x69bd, 0x69be,
629 0x69bf, 0x69c0, 0x69c2, 0x69c3, 0x69c4, 0x69c5, 0x69c6, 0x69c7,
630 0x69c8, 0x69c9, 0x69cb, 0x69cd, 0x69cf, 0x69d1, 0x69d2, 0x69d3,
631 0x69d5, 0x69d6, 0x69d7, 0x69d8, 0x69d9, 0x69da, 0x69dc, 0x69dd,
632 0x69de, 0x69e1, 0x69e2, 0x69e3, 0x69e4, 0x69e5, 0x69e6, 0x69e7,
633 0x69e8, 0x69e9, 0x69ea, 0x69eb, 0x69ec, 0x69ee, 0x69ef, 0x69f0,
634 0x69f1, 0x69f3, 0x69f4, 0x69f5, 0x69f6, 0x69f7, 0x69f8, 0x69f9,
635 0x69fa, 0x69fb, 0x69fc, 0x69fe, 0x6a00, 0x6a01, 0x6a02, 0x6a03,
636 0x6a04, 0x6a05, 0x6a06, 0x6a07, 0x6a08, 0x6a09, 0x6a0b, 0x6a0c,
637 0x6a0d, 0x6a0e, 0x6a0f, 0x6a10, 0x6a11, 0x6a12, 0x6a13, 0x6a14,
638 0x6a15, 0x6a16, 0x6a19, 0x6a1a, 0x6a1b, 0x6a1c, 0x6a1d, 0x6a1e,
639 0x6a20, 0x6a22, 0x6a23, 0x6a24, 0x6a25, 0x6a26, 0x6a27, 0x6a29,
640 0x6a2b, 0x6a2c, 0x6a2d, 0x6a2e, 0x6a30, 0x6a32, 0x6a33, 0x6a34,
641 0x6a36, 0x6a37, 0x6a38, 0x6a39, 0x6a3a, 0x6a3b, 0x6a3c, 0x6a3f,
642 0x6a40, 0x6a41, 0x6a42, 0x6a43, 0x6a45, 0x6a46, 0x6a48, 0x6a49,
643 0x6a4a, 0x6a4b, 0x6a4c, 0x6a4d, 0x6a4e, 0x6a4f, 0x6a51, 0x6a52,
644 0x6a53, 0x6a54, 0x6a55, 0x6a56, 0x6a57, 0x6a5a,
645 /* 0x99 */
646 0x6a5c, 0x6a5d, 0x6a5e, 0x6a5f, 0x6a60, 0x6a62, 0x6a63, 0x6a64,
647 0x6a66, 0x6a67, 0x6a68, 0x6a69, 0x6a6a, 0x6a6b, 0x6a6c, 0x6a6d,
648 0x6a6e, 0x6a6f, 0x6a70, 0x6a72, 0x6a73, 0x6a74, 0x6a75, 0x6a76,
649 0x6a77, 0x6a78, 0x6a7a, 0x6a7b, 0x6a7d, 0x6a7e, 0x6a7f, 0x6a81,
650 0x6a82, 0x6a83, 0x6a85, 0x6a86, 0x6a87, 0x6a88, 0x6a89, 0x6a8a,
651 0x6a8b, 0x6a8c, 0x6a8d, 0x6a8f, 0x6a92, 0x6a93, 0x6a94, 0x6a95,
652 0x6a96, 0x6a98, 0x6a99, 0x6a9a, 0x6a9b, 0x6a9c, 0x6a9d, 0x6a9e,
653 0x6a9f, 0x6aa1, 0x6aa2, 0x6aa3, 0x6aa4, 0x6aa5, 0x6aa6, 0x6aa7,
654 0x6aa8, 0x6aaa, 0x6aad, 0x6aae, 0x6aaf, 0x6ab0, 0x6ab1, 0x6ab2,
655 0x6ab3, 0x6ab4, 0x6ab5, 0x6ab6, 0x6ab7, 0x6ab8, 0x6ab9, 0x6aba,
656 0x6abb, 0x6abc, 0x6abd, 0x6abe, 0x6abf, 0x6ac0, 0x6ac1, 0x6ac2,
657 0x6ac3, 0x6ac4, 0x6ac5, 0x6ac6, 0x6ac7, 0x6ac8, 0x6ac9, 0x6aca,
658 0x6acb, 0x6acc, 0x6acd, 0x6ace, 0x6acf, 0x6ad0, 0x6ad1, 0x6ad2,
659 0x6ad3, 0x6ad4, 0x6ad5, 0x6ad6, 0x6ad7, 0x6ad8, 0x6ad9, 0x6ada,
660 0x6adb, 0x6adc, 0x6add, 0x6ade, 0x6adf, 0x6ae0, 0x6ae1, 0x6ae2,
661 0x6ae3, 0x6ae4, 0x6ae5, 0x6ae6, 0x6ae7, 0x6ae8, 0x6ae9, 0x6aea,
662 0x6aeb, 0x6aec, 0x6aed, 0x6aee, 0x6aef, 0x6af0, 0x6af1, 0x6af2,
663 0x6af3, 0x6af4, 0x6af5, 0x6af6, 0x6af7, 0x6af8, 0x6af9, 0x6afa,
664 0x6afb, 0x6afc, 0x6afd, 0x6afe, 0x6aff, 0x6b00, 0x6b01, 0x6b02,
665 0x6b03, 0x6b04, 0x6b05, 0x6b06, 0x6b07, 0x6b08, 0x6b09, 0x6b0a,
666 0x6b0b, 0x6b0c, 0x6b0d, 0x6b0e, 0x6b0f, 0x6b10, 0x6b11, 0x6b12,
667 0x6b13, 0x6b14, 0x6b15, 0x6b16, 0x6b17, 0x6b18, 0x6b19, 0x6b1a,
668 0x6b1b, 0x6b1c, 0x6b1d, 0x6b1e, 0x6b1f, 0x6b25, 0x6b26, 0x6b28,
669 0x6b29, 0x6b2a, 0x6b2b, 0x6b2c, 0x6b2d, 0x6b2e,
670 /* 0x9a */
671 0x6b2f, 0x6b30, 0x6b31, 0x6b33, 0x6b34, 0x6b35, 0x6b36, 0x6b38,
672 0x6b3b, 0x6b3c, 0x6b3d, 0x6b3f, 0x6b40, 0x6b41, 0x6b42, 0x6b44,
673 0x6b45, 0x6b48, 0x6b4a, 0x6b4b, 0x6b4d, 0x6b4e, 0x6b4f, 0x6b50,
674 0x6b51, 0x6b52, 0x6b53, 0x6b54, 0x6b55, 0x6b56, 0x6b57, 0x6b58,
675 0x6b5a, 0x6b5b, 0x6b5c, 0x6b5d, 0x6b5e, 0x6b5f, 0x6b60, 0x6b61,
676 0x6b68, 0x6b69, 0x6b6b, 0x6b6c, 0x6b6d, 0x6b6e, 0x6b6f, 0x6b70,
677 0x6b71, 0x6b72, 0x6b73, 0x6b74, 0x6b75, 0x6b76, 0x6b77, 0x6b78,
678 0x6b7a, 0x6b7d, 0x6b7e, 0x6b7f, 0x6b80, 0x6b85, 0x6b88, 0x6b8c,
679 0x6b8e, 0x6b8f, 0x6b90, 0x6b91, 0x6b94, 0x6b95, 0x6b97, 0x6b98,
680 0x6b99, 0x6b9c, 0x6b9d, 0x6b9e, 0x6b9f, 0x6ba0, 0x6ba2, 0x6ba3,
681 0x6ba4, 0x6ba5, 0x6ba6, 0x6ba7, 0x6ba8, 0x6ba9, 0x6bab, 0x6bac,
682 0x6bad, 0x6bae, 0x6baf, 0x6bb0, 0x6bb1, 0x6bb2, 0x6bb6, 0x6bb8,
683 0x6bb9, 0x6bba, 0x6bbb, 0x6bbc, 0x6bbd, 0x6bbe, 0x6bc0, 0x6bc3,
684 0x6bc4, 0x6bc6, 0x6bc7, 0x6bc8, 0x6bc9, 0x6bca, 0x6bcc, 0x6bce,
685 0x6bd0, 0x6bd1, 0x6bd8, 0x6bda, 0x6bdc, 0x6bdd, 0x6bde, 0x6bdf,
686 0x6be0, 0x6be2, 0x6be3, 0x6be4, 0x6be5, 0x6be6, 0x6be7, 0x6be8,
687 0x6be9, 0x6bec, 0x6bed, 0x6bee, 0x6bf0, 0x6bf1, 0x6bf2, 0x6bf4,
688 0x6bf6, 0x6bf7, 0x6bf8, 0x6bfa, 0x6bfb, 0x6bfc, 0x6bfe, 0x6bff,
689 0x6c00, 0x6c01, 0x6c02, 0x6c03, 0x6c04, 0x6c08, 0x6c09, 0x6c0a,
690 0x6c0b, 0x6c0c, 0x6c0e, 0x6c0f, 0x6c10, 0x6c11, 0x6c1c, 0x6c1e,
691 0x6c20, 0x6c23, 0x6c25, 0x6c2b, 0x6c2c, 0x6c2d, 0x6c31, 0x6c33,
692 0x6c36, 0x6c37, 0x6c39, 0x6c3a, 0x6c3b, 0x6c3c, 0x6c3e, 0x6c3f,
693 0x6c43, 0x6c44, 0x6c45, 0x6c48, 0x6c4b, 0x6c4c, 0x6c4d, 0x6c4e,
694 0x6c4f, 0x6c51, 0x6c52, 0x6c53, 0x6c56, 0x6c58,
```

```
695 /* 0x9b */
696 0x6c59, 0x6c5a, 0x6c62, 0x6c63, 0x6c65, 0x6c66, 0x6c67, 0x6c6b,
697 0x6c6c, 0x6c6d, 0x6c6e, 0x6c6f, 0x6c71, 0x6c73, 0x6c75, 0x6c77,
698 0x6c78, 0x6c7a, 0x6c7b, 0x6c7c, 0x6c7f, 0x6c80, 0x6c84, 0x6c87,
699 0x6c8a, 0x6c8b, 0x6c8d, 0x6c8e, 0x6c91, 0x6c92, 0x6c95, 0x6c96,
700 0x6c97, 0x6c98, 0x6c9a, 0x6c9c, 0x6c9d, 0x6c9e, 0x6ca0, 0x6ca2,
701 0x6ca8, 0x6cac, 0x6caf, 0x6cb0, 0x6cb4, 0x6cb5, 0x6cb6, 0x6cb7,
702 0x6cba, 0x6cc0, 0x6cc1, 0x6cc2, 0x6cc3, 0x6cc6, 0x6cc7, 0x6cc8,
703 0x6ccb, 0x6ccd, 0x6cce, 0x6ccf, 0x6cd1, 0x6cd2, 0x6cd8, 0x6cd9,
704 0x6cda, 0x6cdc, 0x6cdd, 0x6cdf, 0x6ce4, 0x6ce6, 0x6ce7, 0x6ce9,
705 0x6cec, 0x6ced, 0x6cf2, 0x6cf4, 0x6cf9, 0x6cfe, 0x6d00, 0x6d02,
706 0x6d03, 0x6d05, 0x6d06, 0x6d08, 0x6d09, 0x6d0a, 0x6d0d, 0x6d0f,
707 0x6d10, 0x6d11, 0x6d13, 0x6d14, 0x6d15, 0x6d16, 0x6d18, 0x6d1c,
708 0x6d1d, 0x6d1f, 0x6d20, 0x6d21, 0x6d22, 0x6d23, 0x6d24, 0x6d26,
709 0x6d28, 0x6d29, 0x6d2c, 0x6d2d, 0x6d2f, 0x6d30, 0x6d34, 0x6d36,
710 0x6d37, 0x6d38, 0x6d3a, 0x6d3f, 0x6d40, 0x6d42, 0x6d44, 0x6d49,
711 0x6d4c, 0x6d50, 0x6d55, 0x6d56, 0x6d57, 0x6d58, 0x6d5b, 0x6d5d,
712 0x6d5f, 0x6d61, 0x6d62, 0x6d64, 0x6d65, 0x6d67, 0x6d68, 0x6d6b,
713 0x6d6c, 0x6d6d, 0x6d70, 0x6d71, 0x6d72, 0x6d73, 0x6d75, 0x6d76,
714 0x6d79, 0x6d7a, 0x6d7b, 0x6d7d, 0x6d7e, 0x6d7f, 0x6d80, 0x6d81,
715 0x6d83, 0x6d84, 0x6d86, 0x6d87, 0x6d88a, 0x6d88b, 0x6d8d, 0x6d8f,
716 0x6d90, 0x6d92, 0x6d96, 0x6d97, 0x6d98, 0x6d99, 0x6d9a, 0x6d9c,
717 0x6da2, 0x6da5, 0x6dac, 0x6dad, 0x6db0, 0x6db1, 0x6db3, 0x6db4,
718 0x6db6, 0x6db7, 0x6db9, 0x6dba, 0x6dbb, 0x6dbc, 0x6dbd, 0x6dbe,
719 0x6dc1, 0x6dc2, 0x6dc3, 0x6dc8, 0x6dc9, 0x6dca,
720 /* 0x9c */
721 0x6dcd, 0x6dce, 0x6dcf, 0x6dd0, 0x6dd2, 0x6dd3, 0x6dd4, 0x6dd5,
722 0x6dd7, 0x6dda, 0x6ddb, 0x6ddc, 0x6ddf, 0x6de2, 0x6de3, 0x6de5,
723 0x6de7, 0x6de8, 0x6de9, 0x6dea, 0x6ded, 0x6def, 0x6df0, 0x6df2,
724 0x6df4, 0x6df5, 0x6df6, 0x6df8, 0x6dfa, 0x6dfd, 0x6dfe, 0x6dff,
725 0x6e00, 0x6e01, 0x6e02, 0x6e03, 0x6e04, 0x6e06, 0x6e07, 0x6e08,
726 0x6e09, 0x6e0b, 0x6e0f, 0x6e12, 0x6e13, 0x6e15, 0x6e18, 0x6e19,
727 0x6e1b, 0x6e1c, 0x6e1e, 0x6e1f, 0x6e22, 0x6e26, 0x6e27, 0x6e28,
728 0x6e2a, 0x6e2c, 0x6e2e, 0x6e30, 0x6e31, 0x6e33, 0x6e35, 0x6e36,
729 0x6e37, 0x6e39, 0x6e3b, 0x6e3c, 0x6e3d, 0x6e3e, 0x6e3f, 0x6e40,
730 0x6e41, 0x6e42, 0x6e45, 0x6e46, 0x6e47, 0x6e48, 0x6e49, 0x6e4a,
731 0x6e4b, 0x6e4c, 0x6e4f, 0x6e50, 0x6e51, 0x6e52, 0x6e55, 0x6e57,
732 0x6e59, 0x6e5a, 0x6e5c, 0x6e5d, 0x6e5e, 0x6e60, 0x6e61, 0x6e62,
733 0x6e63, 0x6e64, 0x6e65, 0x6e66, 0x6e67, 0x6e68, 0x6e69, 0x6e6a,
734 0x6e6c, 0x6e6d, 0x6e6f, 0x6e70, 0x6e71, 0x6e72, 0x6e73, 0x6e74,
735 0x6e75, 0x6e76, 0x6e77, 0x6e78, 0x6e79, 0x6e7a, 0x6e7b, 0x6e7c,
736 0x6e7d, 0x6e80, 0x6e81, 0x6e82, 0x6e84, 0x6e87, 0x6e88, 0x6e8a,
737 0x6e8b, 0x6e8c, 0x6e8d, 0x6e8e, 0x6e91, 0x6e92, 0x6e93, 0x6e94,
738 0x6e95, 0x6e96, 0x6e97, 0x6e99, 0x6e9a, 0x6e9b, 0x6e9d, 0x6e9e,
739 0x6ea0, 0x6ea1, 0x6ea3, 0x6ea4, 0x6ea6, 0x6ea8, 0x6ea9, 0x6eab,
740 0x6eac, 0x6ead, 0x6eae, 0x6eb0, 0x6eb3, 0x6eb5, 0x6eb8, 0x6eb9,
741 0x6ebc, 0x6ebe, 0x6ebf, 0x6ec0, 0x6ec3, 0x6ec4, 0x6ec5, 0x6ec6,
742 0x6ec8, 0x6ec9, 0x6eca, 0x6ecc, 0x6ecd, 0x6ece, 0x6ed0, 0x6ed2,
743 0x6ed6, 0x6ed8, 0x6ed9, 0x6edb, 0x6edc, 0x6edd, 0x6ee3, 0x6ee7,
744 0x6eea, 0x6eeb, 0x6eec, 0x6eed, 0x6eee, 0x6eef,
745 /* 0x9d */
746 0x6ef0, 0x6ef1, 0x6ef2, 0x6ef3, 0x6ef5, 0x6ef6, 0x6ef7, 0x6ef8,
747 0x6efa, 0x6efb, 0x6efc, 0x6efd, 0x6efe, 0x6eff, 0x6f00, 0x6f01,
748 0x6f03, 0x6f04, 0x6f05, 0x6f07, 0x6f08, 0x6f0a, 0x6f0b, 0x6f0c,
749 0x6f0d, 0x6f0e, 0x6f10, 0x6f11, 0x6f12, 0x6f16, 0x6f17, 0x6f18,
750 0x6f19, 0x6f1a, 0x6f1b, 0x6f1c, 0x6f1d, 0x6f1e, 0x6f1f, 0x6f21,
751 0x6f22, 0x6f23, 0x6f25, 0x6f26, 0x6f27, 0x6f28, 0x6f2c, 0x6f2e,
752 0x6f30, 0x6f32, 0x6f34, 0x6f35, 0x6f37, 0x6f38, 0x6f39, 0x6f3a,
753 0x6f3b, 0x6f3c, 0x6f3d, 0x6f3f, 0x6f40, 0x6f41, 0x6f42, 0x6f43,
754 0x6f44, 0x6f45, 0x6f48, 0x6f49, 0x6f4a, 0x6f4c, 0x6f4e, 0x6f4f,
755 0x6f50, 0x6f51, 0x6f52, 0x6f53, 0x6f54, 0x6f55, 0x6f56, 0x6f57,
756 0x6f59, 0x6f5a, 0x6f5b, 0x6f5d, 0x6f5f, 0x6f60, 0x6f61, 0x6f63,
757 0x6f64, 0x6f65, 0x6f67, 0x6f68, 0x6f69, 0x6f6a, 0x6f6b, 0x6f6c,
758 0x6f6f, 0x6f70, 0x6f71, 0x6f73, 0x6f75, 0x6f76, 0x6f77, 0x6f79,
759 0x6f7b, 0x6f7d, 0x6f7e, 0x6f7f, 0x6f80, 0x6f81, 0x6f82, 0x6f83,
760 0x6f85, 0x6f86, 0x6f87, 0x6f8a, 0x6f8b, 0x6f8e, 0x6f90, 0x6f91,
761 0x6f92, 0x6f93, 0x6f94, 0x6f95, 0x6f96, 0x6f97, 0x6f98, 0x6f99,
762 0x6f9a, 0x6f9b, 0x6f9d, 0x6f9e, 0x6f9f, 0x6fa0, 0x6fa2, 0x6fa3,
763 0x6fa4, 0x6fa5, 0x6fa6, 0x6fa8, 0x6fa9, 0x6faa, 0x6fab, 0x6fac,
764 0x6fad, 0x6fae, 0x6faf, 0x6fb0, 0x6fb1, 0x6fb2, 0x6fb4, 0x6fb5,
765 0x6fb7, 0x6fb8, 0x6fba, 0x6fbb, 0x6fbc, 0x6fbd, 0x6fbe, 0x6fbf,
766 0x6fc1, 0x6fc3, 0x6fc4, 0x6fc5, 0x6fc6, 0x6fc7, 0x6fc8, 0x6fca,
767 0x6fcb, 0x6fcc, 0x6fcd, 0x6fce, 0x6fcf, 0x6fd0, 0x6fd3, 0x6fd4,
768 0x6fd5, 0x6fd6, 0x6fd7, 0x6fd8, 0x6fd9, 0x6fda, 0x6fdb, 0x6fdc,
769 0x6fdd, 0x6fdf, 0x6fe2, 0x6fe3, 0x6fe4, 0x6fe5,
770 /* 0x9e */
771 0x6fe6, 0x6fe7, 0x6fe8, 0x6fe9, 0x6fea, 0x6feb, 0x6fec, 0x6fed,
772 0x6fef0, 0x6fef1, 0x6fef2, 0x6fef3, 0x6fef4, 0x6fef5, 0x6fef6, 0x6fef7,
773 0x6fef8, 0x6fef9, 0x6fefa, 0x6fefb, 0x6fefc, 0x6fefd, 0x6fefe, 0x6fef,
774 0x7000, 0x7001, 0x7002, 0x7003, 0x7004, 0x7005, 0x7006, 0x7007,
775 0x7008, 0x7009, 0x700a, 0x700b, 0x700c, 0x700d, 0x700e, 0x700f,
776 0x7010, 0x7012, 0x7013, 0x7014, 0x7015, 0x7016, 0x7017, 0x7018,
777 0x7019, 0x701c, 0x701d, 0x701e, 0x701f, 0x7020, 0x7021, 0x7022,
778 0x7024, 0x7025, 0x7026, 0x7027, 0x7028, 0x7029, 0x702a, 0x702b,
779 0x702c, 0x702d, 0x702e, 0x702f, 0x7030, 0x7031, 0x7032, 0x7033,
780 0x7034, 0x7036, 0x7037, 0x7038, 0x703a, 0x703b, 0x703c, 0x703d,
781 0x703e, 0x703f, 0x7040, 0x7041, 0x7042, 0x7043, 0x7044, 0x7045,
```

```
782 0x7046, 0x7047, 0x7048, 0x7049, 0x704a, 0x704b, 0x704d, 0x704e,
783 0x7050, 0x7051, 0x7052, 0x7053, 0x7054, 0x7055, 0x7056, 0x7057,
784 0x7058, 0x7059, 0x705a, 0x705b, 0x705c, 0x705d, 0x705f, 0x7060,
785 0x7061, 0x7062, 0x7063, 0x7064, 0x7065, 0x7066, 0x7067, 0x7068,
786 0x7069, 0x706a, 0x706e, 0x7071, 0x7072, 0x7073, 0x7074, 0x7077,
787 0x7079, 0x707a, 0x707b, 0x707d, 0x7081, 0x7082, 0x7083, 0x7084,
788 0x7086, 0x7087, 0x7088, 0x708b, 0x708c, 0x708d, 0x708f, 0x7090,
789 0x7091, 0x7093, 0x7097, 0x7098, 0x709a, 0x709b, 0x709e, 0x709f,
790 0x70a0, 0x70a1, 0x70a2, 0x70a3, 0x70a4, 0x70a5, 0x70a6, 0x70a7,
791 0x70a8, 0x70a9, 0x70aa, 0x70b0, 0x70b2, 0x70b4, 0x70b5, 0x70b6,
792 0x70ba, 0x70be, 0x70bf, 0x70c4, 0x70c5, 0x70c6, 0x70c7, 0x70c9,
793 0x70cb, 0x70cc, 0x70cd, 0x70ce, 0x70cf, 0x70d0, 0x70d1, 0x70d2,
794 0x70d3, 0x70d4, 0x70d5, 0x70d6, 0x70d7, 0x70da,
795 /* 0x9f */
796 0x70dc, 0x70dd, 0x70de, 0x70e0, 0x70e1, 0x70e2, 0x70e3, 0x70e5,
797 0x70ea, 0x70ee, 0x70f0, 0x70f1, 0x70f2, 0x70f3, 0x70f4, 0x70f5,
798 0x70f6, 0x70f8, 0x70fa, 0x70fb, 0x70fc, 0x70fe, 0x70ff, 0x7100,
799 0x7101, 0x7102, 0x7103, 0x7104, 0x7105, 0x7106, 0x7107, 0x7108,
800 0x710b, 0x710c, 0x710d, 0x710e, 0x710f, 0x7111, 0x7112, 0x7114,
801 0x7117, 0x711b, 0x711c, 0x711d, 0x711e, 0x711f, 0x7120, 0x7121,
802 0x7122, 0x7123, 0x7124, 0x7125, 0x7127, 0x7128, 0x7129, 0x712a,
803 0x712b, 0x712c, 0x712d, 0x712e, 0x7132, 0x7133, 0x7134, 0x7135,
804 0x7137, 0x7138, 0x7139, 0x713a, 0x713b, 0x713c, 0x713d, 0x713e,
805 0x713f, 0x7140, 0x7141, 0x7142, 0x7143, 0x7144, 0x7146, 0x7147,
806 0x7148, 0x7149, 0x714b, 0x714d, 0x714f, 0x7150, 0x7151, 0x7152,
807 0x7153, 0x7154, 0x7155, 0x7156, 0x7157, 0x7158, 0x7159, 0x715a,
808 0x715b, 0x715d, 0x715f, 0x7160, 0x7161, 0x7162, 0x7163, 0x7165,
809 0x7169, 0x716a, 0x716b, 0x716c, 0x716d, 0x716f, 0x7170, 0x7171,
810 0x7174, 0x7175, 0x7176, 0x7177, 0x7179, 0x717b, 0x717c, 0x717e,
811 0x717f, 0x7180, 0x7181, 0x7182, 0x7183, 0x7185, 0x7186, 0x7187,
812 0x7188, 0x7189, 0x718b, 0x718c, 0x718d, 0x718e, 0x7190, 0x7191,
813 0x7192, 0x7193, 0x7195, 0x7196, 0x7197, 0x719a, 0x719b, 0x719c,
814 0x719d, 0x719e, 0x71a1, 0x71a2, 0x71a3, 0x71a4, 0x71a5, 0x71a6,
815 0x71a7, 0x71a9, 0x71aa, 0x71ab, 0x71ad, 0x71ae, 0x71af, 0x71b0,
816 0x71b1, 0x71b2, 0x71b4, 0x71b6, 0x71b7, 0x71b8, 0x71ba, 0x71bb,
817 0x71bc, 0x71bd, 0x71be, 0x71bf, 0x71c0, 0x71c1, 0x71c2, 0x71c4,
818 0x71c5, 0x71c6, 0x71c7, 0x71c8, 0x71c9, 0x71ca, 0x71cb, 0x71cc,
819 0x71cd, 0x71cf, 0x71d0, 0x71d1, 0x71d2, 0x71d3,
820 /* 0xa0 */
821 0x71d6, 0x71d7, 0x71d8, 0x71d9, 0x71da, 0x71db, 0x71dc, 0x71dd,
822 0x71de, 0x71df, 0x71e1, 0x71e2, 0x71e3, 0x71e4, 0x71e6, 0x71e8,
823 0x71e9, 0x71ea, 0x71eb, 0x71ec, 0x71ed, 0x71ef, 0x71f0, 0x71f1,
824 0x71f2, 0x71f3, 0x71f4, 0x71f5, 0x71f6, 0x71f7, 0x71f8, 0x71fa,
825 0x71fb, 0x71fc, 0x71fd, 0x71fe, 0x71ff, 0x7200, 0x7201, 0x7202,
826 0x7203, 0x7204, 0x7205, 0x7207, 0x7208, 0x7209, 0x720a, 0x720b,
827 0x720c, 0x720d, 0x720e, 0x720f, 0x7210, 0x7211, 0x7212, 0x7213,
828 0x7214, 0x7215, 0x7216, 0x7217, 0x7218, 0x7219, 0x721a, 0x721b,
829 0x721c, 0x721e, 0x721f, 0x7220, 0x7221, 0x7222, 0x7223, 0x7224,
830 0x7225, 0x7226, 0x7227, 0x7229, 0x722b, 0x722d, 0x722e, 0x722f,
831 0x7232, 0x7233, 0x7234, 0x723a, 0x723c, 0x723e, 0x7240, 0x7241,
832 0x7242, 0x7243, 0x7244, 0x7245, 0x7246, 0x7249, 0x724a, 0x724b,
833 0x724e, 0x724f, 0x7250, 0x7251, 0x7253, 0x7254, 0x7255, 0x7257,
834 0x7258, 0x725a, 0x725c, 0x725e, 0x7260, 0x7263, 0x7264, 0x7265,
835 0x7268, 0x726a, 0x726b, 0x726c, 0x726d, 0x7270, 0x7271, 0x7273,
836 0x7274, 0x7276, 0x7277, 0x7278, 0x727b, 0x727c, 0x727d, 0x7282,
837 0x7283, 0x7285, 0x7286, 0x7287, 0x7288, 0x7289, 0x728c, 0x728e,
838 0x7290, 0x7291, 0x7293, 0x7294, 0x7295, 0x7296, 0x7297, 0x7298,
839 0x7299, 0x729a, 0x729b, 0x729c, 0x729d, 0x729e, 0x72a0, 0x72a1,
840 0x72a2, 0x72a3, 0x72a4, 0x72a5, 0x72a6, 0x72a7, 0x72a8, 0x72a9,
841 0x72aa, 0x72ab, 0x72ae, 0x72b1, 0x72b2, 0x72b3, 0x72b5, 0x72ba,
842 0x72bb, 0x72bc, 0x72bd, 0x72be, 0x72bf, 0x72c0, 0x72c5, 0x72c6,
843 0x72c7, 0x72c9, 0x72ca, 0x72cb, 0x72cc, 0x72cf, 0x72d1, 0x72d3,
844 0x72d4, 0x72d5, 0x72d6, 0x72d8, 0x72da, 0x72db,
845 /* 0xa1 */
846 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
847 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
848 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
849 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
850 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
851 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
852 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
853 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
854 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
855 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
856 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
857 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
858 0x3000, 0x3001, 0x3002, 0x00b7, 0x02c9, 0x02c7, 0x00a8, 0x3003,
859 0x3005, 0x2014, 0xff5e, 0x2016, 0x2026, 0x2018, 0x2019, 0x201c,
860 0x201d, 0x3014, 0x3015, 0x3008, 0x3009, 0x300a, 0x300b, 0x300c,
861 0x300d, 0x300e, 0x300f, 0x3016, 0x3017, 0x3010, 0x3011, 0x00b1,
862 0x00d7, 0x00f7, 0x2236, 0x2227, 0x2228, 0x2211, 0x220f, 0x222a,
863 0x2229, 0x2208, 0x2237, 0x221a, 0x22a5, 0x2225, 0x2220, 0x2312,
864 0x2299, 0x222b, 0x222e, 0x2261, 0x224c, 0x2248, 0x223d, 0x221d,
865 0x2260, 0x226e, 0x226f, 0x2264, 0x2265, 0x221e, 0x2235, 0x2234,
866 0x2642, 0x2640, 0x00b0, 0x2032, 0x2033, 0x2103, 0xff04, 0x00a4,
867 0xffe0, 0xffe1, 0x2030, 0x00a7, 0x2116, 0x2606, 0x2605, 0x25cb,
868 0x25cf, 0x25ce, 0x25c7, 0x25c6, 0x25a1, 0x25a0, 0x25b3, 0x25b2,
```

```

869 0x203b, 0x2192, 0x2190, 0x2191, 0x2193, 0x3013,
870 /* 0xa2 */
871 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
872 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
873 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
874 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
875 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
876 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
877 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
878 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
879 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
880 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
881 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
882 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
883 0x2170, 0x2171, 0x2172, 0x2173, 0x2174, 0x2175, 0x2176, 0x2177,
884 0x2178, 0x2179, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
885 0x2488, 0x2489, 0x248a, 0x248b, 0x248c, 0x248d, 0x248e, 0x248f,
886 0x2490, 0x2491, 0x2492, 0x2493, 0x2494, 0x2495, 0x2496, 0x2497,
887 0x2498, 0x2499, 0x249a, 0x249b, 0x2474, 0x2475, 0x2476, 0x2477,
888 0x2478, 0x2479, 0x247a, 0x247b, 0x247c, 0x247d, 0x247e, 0x247f,
889 0x2480, 0x2481, 0x2482, 0x2483, 0x2484, 0x2485, 0x2486, 0x2487,
890 0x2460, 0x2461, 0x2462, 0x2463, 0x2464, 0x2465, 0x2466, 0x2467,
891 0x2468, 0x2469, 0xffff, 0xffff, 0x3220, 0x3221, 0x3222, 0x3223,
892 0x3224, 0x3225, 0x3226, 0x3227, 0x3228, 0x3229, 0xffff, 0xffff,
893 0x2160, 0x2161, 0x2162, 0x2163, 0x2164, 0x2165, 0x2166, 0x2167,
894 0x2168, 0x2169, 0x216a, 0x216b, 0xffff, 0xffff,
895 /* 0xa3 */
896 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
897 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
898 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
899 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
900 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
901 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
902 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
903 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
904 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
905 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
906 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
907 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
908 0xff01, 0xff02, 0xff03, 0xffe5, 0xff05, 0xff06, 0xff07, 0xff08,
909 0xff09, 0xff0a, 0xff0b, 0xff0c, 0xff0d, 0xff0e, 0xff0f, 0xff10,
910 0xff11, 0xff12, 0xff13, 0xff14, 0xff15, 0xff16, 0xff17, 0xff18,
911 0xff19, 0xff1a, 0xff1b, 0xff1c, 0xff1d, 0xff1e, 0xff1f, 0xff20,
912 0xff21, 0xff22, 0xff23, 0xff24, 0xff25, 0xff26, 0xff27, 0xff28,
913 0xff29, 0xff2a, 0xff2b, 0xff2c, 0xff2d, 0xff2e, 0xff2f, 0xff30,
914 0xff31, 0xff32, 0xff33, 0xff34, 0xff35, 0xff36, 0xff37, 0xff38,
915 0xff39, 0xff3a, 0xff3b, 0xff3c, 0xff3d, 0xff3e, 0xff3f, 0xff40,
916 0xff41, 0xff42, 0xff43, 0xff44, 0xff45, 0xff46, 0xff47, 0xff48,
917 0xff49, 0xff4a, 0xff4b, 0xff4c, 0xff4d, 0xff4e, 0xff4f, 0xff50,
918 0xff51, 0xff52, 0xff53, 0xff54, 0xff55, 0xff56, 0xff57, 0xff58,
919 0xff59, 0xff5a, 0xff5b, 0xff5c, 0xff5d, 0xffe3,
920 /* 0xa4 */
921 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
922 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
923 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
924 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
925 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
926 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
927 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
928 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
929 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
930 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
931 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
932 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
933 0x3041, 0x3042, 0x3043, 0x3044, 0x3045, 0x3046, 0x3047, 0x3048,
934 0x3049, 0x304a, 0x304b, 0x304c, 0x304d, 0x304e, 0x304f, 0x3050,
935 0x3051, 0x3052, 0x3053, 0x3054, 0x3055, 0x3056, 0x3057, 0x3058,
936 0x3059, 0x305a, 0x305b, 0x305c, 0x305d, 0x305e, 0x305f, 0x3060,
937 0x3061, 0x3062, 0x3063, 0x3064, 0x3065, 0x3066, 0x3067, 0x3068,
938 0x3069, 0x306a, 0x306b, 0x306c, 0x306d, 0x306e, 0x306f, 0x3070,
939 0x3071, 0x3072, 0x3073, 0x3074, 0x3075, 0x3076, 0x3077, 0x3078,
940 0x3079, 0x307a, 0x307b, 0x307c, 0x307d, 0x307e, 0x307f, 0x3080,
941 0x3081, 0x3082, 0x3083, 0x3084, 0x3085, 0x3086, 0x3087, 0x3088,
942 0x3089, 0x308a, 0x308b, 0x308c, 0x308d, 0x308e, 0x308f, 0x3090,
943 0x3091, 0x3092, 0x3093, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
944 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
945 /* 0xa5 */
946 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
947 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
948 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
949 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
950 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
951 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
952 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
953 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
954 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
955 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,

```

```
956 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
957 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
958 0x30a1, 0x30a2, 0x30a3, 0x30a4, 0x30a5, 0x30a6, 0x30a7, 0x30a8,
959 0x30a9, 0x30aa, 0x30ab, 0x30ac, 0x30ad, 0x30ae, 0x30af, 0x30b0,
960 0x30b1, 0x30b2, 0x30b3, 0x30b4, 0x30b5, 0x30b6, 0x30b7, 0x30b8,
961 0x30b9, 0x30ba, 0x30bb, 0x30bc, 0x30bd, 0x30be, 0x30bf, 0x30c0,
962 0x30c1, 0x30c2, 0x30c3, 0x30c4, 0x30c5, 0x30c6, 0x30c7, 0x30c8,
963 0x30c9, 0x30ca, 0x30cb, 0x30cc, 0x30cd, 0x30ce, 0x30cf, 0x30d0,
964 0x30d1, 0x30d2, 0x30d3, 0x30d4, 0x30d5, 0x30d6, 0x30d7, 0x30d8,
965 0x30d9, 0x30da, 0x30db, 0x30dc, 0x30dd, 0x30de, 0x30df, 0x30e0,
966 0x30e1, 0x30e2, 0x30e3, 0x30e4, 0x30e5, 0x30e6, 0x30e7, 0x30e8,
967 0x30e9, 0x30ea, 0x30eb, 0x30ec, 0x30ed, 0x30ee, 0x30ef, 0x30f0,
968 0x30f1, 0x30f2, 0x30f3, 0x30f4, 0x30f5, 0x30f6, 0xffff, 0xffff,
969 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
970 /* 0xa6 */
971 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
972 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
973 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
974 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
975 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
976 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
977 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
978 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
979 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
980 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
981 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
982 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
983 0x0391, 0x0392, 0x0393, 0x0394, 0x0395, 0x0396, 0x0397, 0x0398,
984 0x0399, 0x039a, 0x039b, 0x039c, 0x039d, 0x039e, 0x039f, 0x03a0,
985 0x03a1, 0x03a3, 0x03a4, 0x03a5, 0x03a6, 0x03a7, 0x03a8, 0x03a9,
986 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
987 0x03b1, 0x03b2, 0x03b3, 0x03b4, 0x03b5, 0x03b6, 0x03b7, 0x03b8,
988 0x03b9, 0x03ba, 0x03bb, 0x03bc, 0x03bd, 0x03be, 0x03bf, 0x03c0,
989 0x03c1, 0x03c3, 0x03c4, 0x03c5, 0x03c6, 0x03c7, 0x03c8, 0x03c9,
990 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xfe35,
991 0xfe36, 0xfe39, 0xfe3a, 0xfe3f, 0xfe40, 0xfe3d, 0xfe3e, 0xfe41,
992 0xfe42, 0xfe43, 0xfe44, 0xffff, 0xffff, 0xfe3b, 0xfe3c, 0xfe37,
993 0xfe38, 0xfe31, 0xffff, 0xfe33, 0xfe34, 0xffff, 0xffff, 0xffff,
994 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
995 /* 0xa7 */
996 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
997 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
998 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
999 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
1000 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
1001 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
1002 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
1003 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
1004 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
1005 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
1006 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
1007 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
1008 0x0410, 0x0411, 0x0412, 0x0413, 0x0414, 0x0415, 0x0401, 0x0416,
1009 0x0417, 0x0418, 0x0419, 0x041a, 0x041b, 0x041c, 0x041d, 0x041e,
1010 0x041f, 0x0420, 0x0421, 0x0422, 0x0423, 0x0424, 0x0425, 0x0426,
1011 0x0427, 0x0428, 0x0429, 0x042a, 0x042b, 0x042c, 0x042d, 0x042e,
1012 0x042f, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
1013 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
1014 0x0430, 0x0431, 0x0432, 0x0433, 0x0434, 0x0435, 0x0451, 0x0436,
1015 0x0437, 0x0438, 0x0439, 0x043a, 0x043b, 0x043c, 0x043d, 0x043e,
1016 0x043f, 0x0440, 0x0441, 0x0442, 0x0443, 0x0444, 0x0445, 0x0446,
1017 0x0447, 0x0448, 0x0449, 0x044a, 0x044b, 0x044c, 0x044d, 0x044e,
1018 0x044f, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
1019 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
1020 /* 0xa8 */
1021 0x02ca, 0x02cb, 0x02d9, 0x2013, 0x2015, 0x2025, 0x2035, 0x2105,
1022 0x2109, 0x2196, 0x2197, 0x2198, 0x2199, 0x2215, 0x221f, 0x2223,
1023 0x2252, 0x2266, 0x2267, 0x22bf, 0x2550, 0x2551, 0x2552, 0x2553,
1024 0x2554, 0x2555, 0x2556, 0x2557, 0x2558, 0x2559, 0x255a, 0x255b,
1025 0x255c, 0x255d, 0x255e, 0x255f, 0x2560, 0x2561, 0x2562, 0x2563,
1026 0x2564, 0x2565, 0x2566, 0x2567, 0x2568, 0x2569, 0x256a, 0x256b,
1027 0x256c, 0x256d, 0x256e, 0x256f, 0x2570, 0x2571, 0x2572, 0x2573,
1028 0x2581, 0x2582, 0x2583, 0x2584, 0x2585, 0x2586, 0x2587, 0x2588,
1029 0x2589, 0x258a, 0x258b, 0x258c, 0x258d, 0x258e, 0x258f, 0x2593,
1030 0x2594, 0x2595, 0x25bc, 0x25bd, 0x25e2, 0x25e3, 0x25e4, 0x25e5,
1031 0x2609, 0x2295, 0x3012, 0x301d, 0x301e, 0xffff, 0xffff, 0xffff,
1032 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
1033 0x0101, 0x00e1, 0x01ce, 0x00e0, 0x0113, 0x00e9, 0x011b, 0x00e8,
1034 0x012b, 0x00ed, 0x01d0, 0x00ec, 0x014d, 0x00f3, 0x01d2, 0x00f2,
1035 0x016b, 0x00fa, 0x01d4, 0x01d6, 0x01d8, 0x01da, 0x01dc,
1036 0x00fc, 0x00ea, 0x0251, 0xffff, 0x0144, 0x0148, 0xffff, 0x0261,
1037 0xffff, 0xffff, 0xffff, 0xffff, 0x3105, 0x3106, 0x3107, 0x3108,
1038 0x3109, 0x310a, 0x310b, 0x310c, 0x310d, 0x310e, 0x310f, 0x3110,
1039 0x3111, 0x3112, 0x3113, 0x3114, 0x3115, 0x3116, 0x3117, 0x3118,
1040 0x3119, 0x311a, 0x311b, 0x311c, 0x311d, 0x311e, 0x311f, 0x3120,
1041 0x3121, 0x3122, 0x3123, 0x3124, 0x3125, 0x3126, 0x3127, 0x3128,
1042 0x3129, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
```

```

1043 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
1044 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
1045 /* 0xa9 */
1046 0x3021, 0x3022, 0x3023, 0x3024, 0x3025, 0x3026, 0x3027, 0x3028,
1047 0x3029, 0x32a3, 0x338e, 0x338f, 0x339c, 0x339d, 0x339e, 0x33a1,
1048 0x33c4, 0x33ce, 0x33d1, 0x33d2, 0x33d5, 0xfe30, 0xfe2, 0xfe4,
1049 0xffffd, 0x2121, 0x3231, 0xffffd, 0x2010, 0xffffd, 0xffffd, 0xffffd,
1050 0x30fc, 0x309b, 0x309c, 0x30fd, 0x30fe, 0x3006, 0x309d, 0x309e,
1051 0xfe49, 0xfe4a, 0xfe4b, 0xfe4c, 0xfe4d, 0xfe4e, 0xfe4f, 0xfe50,
1052 0xfe51, 0xfe52, 0xfe54, 0xfe55, 0xfe56, 0xfe57, 0xfe59, 0xfe5a,
1053 0xfe5b, 0xfe5c, 0xfe5d, 0xfe5e, 0xfe5f, 0xfe60, 0xfe61, 0xfe62,
1054 0xfe63, 0xfe64, 0xfe65, 0xfe66, 0xfe68, 0xfe69, 0xfe6a, 0xfe6b,
1055 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
1056 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0x3007, 0xffffd, 0xffffd,
1057 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
1058 0xffffd, 0xffffd, 0xffffd, 0x2500, 0x2501, 0x2502, 0x2503, 0x2504,
1059 0x2505, 0x2506, 0x2507, 0x2508, 0x2509, 0x250a, 0x250b, 0x250c,
1060 0x250d, 0x250e, 0x250f, 0x2510, 0x2511, 0x2512, 0x2513, 0x2514,
1061 0x2515, 0x2516, 0x2517, 0x2518, 0x2519, 0x251a, 0x251b, 0x251c,
1062 0x251d, 0x251e, 0x251f, 0x2520, 0x2521, 0x2522, 0x2523, 0x2524,
1063 0x2525, 0x2526, 0x2527, 0x2528, 0x2529, 0x252a, 0x252b, 0x252c,
1064 0x252d, 0x252e, 0x252f, 0x2530, 0x2531, 0x2532, 0x2533, 0x2534,
1065 0x2535, 0x2536, 0x2537, 0x2538, 0x2539, 0x253a, 0x253b, 0x253c,
1066 0x253d, 0x253e, 0x253f, 0x2540, 0x2541, 0x2542, 0x2543, 0x2544,
1067 0x2545, 0x2546, 0x2547, 0x2548, 0x2549, 0x254a, 0x254b, 0xffffd,
1068 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
1069 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
1070 /* 0xaa */
1071 0x72dc, 0x72dd, 0x72df, 0x72e2, 0x72e3, 0x72e4, 0x72e5, 0x72e6,
1072 0x72e7, 0x72ea, 0x72eb, 0x72f5, 0x72f6, 0x72f9, 0x72fd, 0x72fe,
1073 0x72ff, 0x7300, 0x7302, 0x7304, 0x7305, 0x7306, 0x7307, 0x7308,
1074 0x7309, 0x730b, 0x730c, 0x730d, 0x730f, 0x7310, 0x7311, 0x7312,
1075 0x7314, 0x7318, 0x7319, 0x731a, 0x731f, 0x7320, 0x7323, 0x7324,
1076 0x7326, 0x7327, 0x7328, 0x732d, 0x732f, 0x7330, 0x7332, 0x7333,
1077 0x7335, 0x7336, 0x733a, 0x733b, 0x733c, 0x733d, 0x7340, 0x7341,
1078 0x7342, 0x7343, 0x7344, 0x7345, 0x7346, 0x7347, 0x7348, 0x7349,
1079 0x734a, 0x734b, 0x734c, 0x734e, 0x734f, 0x7351, 0x7353, 0x7354,
1080 0x7355, 0x7356, 0x7358, 0x7359, 0x735a, 0x735b, 0x735c, 0x735d,
1081 0x735e, 0x735f, 0x7361, 0x7362, 0x7363, 0x7364, 0x7365, 0x7366,
1082 0x7367, 0x7368, 0x7369, 0x736a, 0x736b, 0x736e, 0x7370, 0x7371,
1083 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
1084 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
1085 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
1086 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
1087 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
1088 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
1089 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
1090 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
1091 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
1092 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
1093 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
1094 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
1095 /* 0xab */
1096 0x7372, 0x7373, 0x7374, 0x7375, 0x7376, 0x7377, 0x7378, 0x7379,
1097 0x737a, 0x737b, 0x737c, 0x737d, 0x737f, 0x7380, 0x7381, 0x7382,
1098 0x7383, 0x7385, 0x7386, 0x7388, 0x738a, 0x738c, 0x738d, 0x738f,
1099 0x7390, 0x7392, 0x7393, 0x7394, 0x7395, 0x7397, 0x7398, 0x7399,
1100 0x739a, 0x739c, 0x739d, 0x739e, 0x73a0, 0x73a1, 0x73a3, 0x73a4,
1101 0x73a5, 0x73a6, 0x73a7, 0x73a8, 0x73aa, 0x73ac, 0x73ad, 0x73b1,
1102 0x73b4, 0x73b5, 0x73b6, 0x73b8, 0x73b9, 0x73bc, 0x73bd, 0x73be,
1103 0x73bf, 0x73c1, 0x73c3, 0x73c4, 0x73c5, 0x73c6, 0x73c7, 0x73cb,
1104 0x73cc, 0x73ce, 0x73d2, 0x73d3, 0x73d4, 0x73d5, 0x73d6, 0x73d7,
1105 0x73d8, 0x73da, 0x73db, 0x73dc, 0x73dd, 0x73df, 0x73e1, 0x73e2,
1106 0x73e3, 0x73e4, 0x73e6, 0x73e8, 0x73ea, 0x73eb, 0x73ec, 0x73ee,
1107 0x73ef, 0x73f0, 0x73f1, 0x73f3, 0x73f4, 0x73f5, 0x73f6, 0x73f7,
1108 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
1109 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
1110 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
1111 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
1112 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
1113 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
1114 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
1115 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
1116 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
1117 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
1118 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
1119 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
1120 /* 0xac */
1121 0x73f8, 0x73f9, 0x73fa, 0x73fb, 0x73fc, 0x73fd, 0x73fe, 0x73ff,
1122 0x7400, 0x7401, 0x7402, 0x7404, 0x7407, 0x7408, 0x740b, 0x740c,
1123 0x740d, 0x740e, 0x7411, 0x7412, 0x7413, 0x7414, 0x7415, 0x7416,
1124 0x7417, 0x7418, 0x7419, 0x741c, 0x741d, 0x741e, 0x741f, 0x7420,
1125 0x7421, 0x7423, 0x7424, 0x7427, 0x7429, 0x742b, 0x742d, 0x742f,
1126 0x7431, 0x7432, 0x7437, 0x7438, 0x7439, 0x743a, 0x743b, 0x743d,
1127 0x743e, 0x743f, 0x7440, 0x7442, 0x7443, 0x7444, 0x7445, 0x7446,
1128 0x7447, 0x7448, 0x7449, 0x744a, 0x744b, 0x744c, 0x744d, 0x744e,
1129 0x744f, 0x7450, 0x7451, 0x7452, 0x7453, 0x7454, 0x7456, 0x7458,

```

```
1130 0x745d, 0x7460, 0x7461, 0x7462, 0x7463, 0x7464, 0x7465, 0x7466,
1131 0x7467, 0x7468, 0x7469, 0x746a, 0x746b, 0x746c, 0x746e, 0x746e, 0x746e,
1132 0x7471, 0x7472, 0x7473, 0x7473, 0x7474, 0x7475, 0x7478, 0x7479, 0x747a,
1133 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
1134 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
1135 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
1136 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
1137 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
1138 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
1139 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
1140 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
1141 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
1142 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
1143 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
1144 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
1145 /* 0xad */
1146 0x747b, 0x747c, 0x747d, 0x747f, 0x7482, 0x7484, 0x7485, 0x7486,
1147 0x7488, 0x7489, 0x748a, 0x748a, 0x748c, 0x748d, 0x748f, 0x7491, 0x7492,
1148 0x7493, 0x7494, 0x7495, 0x7496, 0x7497, 0x7498, 0x7499, 0x749a,
1149 0x749b, 0x749d, 0x749f, 0x74a0, 0x74a1, 0x74a2, 0x74a3, 0x74a4,
1150 0x74a5, 0x74a6, 0x74aa, 0x74aa, 0x74ab, 0x74ac, 0x74ad, 0x74ae, 0x74af,
1151 0x74b0, 0x74b1, 0x74b2, 0x74b3, 0x74b4, 0x74b5, 0x74b6, 0x74b7,
1152 0x74b8, 0x74b9, 0x74bb, 0x74bc, 0x74bd, 0x74be, 0x74bf, 0x74c0,
1153 0x74c1, 0x74c2, 0x74c3, 0x74c3, 0x74c4, 0x74c5, 0x74c6, 0x74c7, 0x74c8,
1154 0x74c9, 0x74ca, 0x74cb, 0x74cc, 0x74cd, 0x74ce, 0x74cf, 0x74d0,
1155 0x74d1, 0x74d3, 0x74d4, 0x74d5, 0x74d6, 0x74d7, 0x74d8, 0x74d9,
1156 0x74da, 0x74db, 0x74dd, 0x74dd, 0x74df, 0x74e1, 0x74e5, 0x74e7, 0x74e8,
1157 0x74e9, 0x74ea, 0x74eb, 0x74ec, 0x74ed, 0x74f0, 0x74f1, 0x74f2,
1158 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
1159 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
1160 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
1161 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
1162 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
1163 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
1164 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
1165 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
1166 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
1167 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
1168 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
1169 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
1170 /* 0xae */
1171 0x74f3, 0x74f5, 0x74f8, 0x74f9, 0x74fa, 0x74fb, 0x74fc, 0x74fd,
1172 0x74fe, 0x7500, 0x7501, 0x7502, 0x7503, 0x7505, 0x7506, 0x7507,
1173 0x7508, 0x7509, 0x750a, 0x750b, 0x750c, 0x750e, 0x7510, 0x7512,
1174 0x7514, 0x7515, 0x7516, 0x7516, 0x7517, 0x751b, 0x751d, 0x751e, 0x7520,
1175 0x7521, 0x7522, 0x7523, 0x7524, 0x7526, 0x7527, 0x752a, 0x752e,
1176 0x7534, 0x7536, 0x7539, 0x753c, 0x753d, 0x753f, 0x7541, 0x7542,
1177 0x7543, 0x7544, 0x7546, 0x7546, 0x7547, 0x7549, 0x754a, 0x754d, 0x7550,
1178 0x7551, 0x7552, 0x7553, 0x7555, 0x7556, 0x7557, 0x7558, 0x755d,
1179 0x755e, 0x755f, 0x7560, 0x7561, 0x7562, 0x7563, 0x7564, 0x7567,
1180 0x7568, 0x7569, 0x756b, 0x756c, 0x756d, 0x756e, 0x756f, 0x7570,
1181 0x7571, 0x7573, 0x7575, 0x7576, 0x7577, 0x757a, 0x757b, 0x757c,
1182 0x757d, 0x757e, 0x7580, 0x7581, 0x7582, 0x7584, 0x7585, 0x7587,
1183 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
1184 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
1185 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
1186 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
1187 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
1188 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
1189 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
1190 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
1191 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
1192 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
1193 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
1194 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
1195 /* 0xaf */
1196 0x7588, 0x7589, 0x758a, 0x758c, 0x758d, 0x758e, 0x7590, 0x7593,
1197 0x7595, 0x7598, 0x759b, 0x759c, 0x759e, 0x75a2, 0x75a6, 0x75a7,
1198 0x75a8, 0x75a9, 0x75aa, 0x75ad, 0x75b6, 0x75b7, 0x75ba, 0x75bb,
1199 0x75bf, 0x75c0, 0x75c1, 0x75c6, 0x75cb, 0x75cc, 0x75ce, 0x75cf,
1200 0x75d0, 0x75d1, 0x75d3, 0x75d7, 0x75d9, 0x75da, 0x75dc, 0x75dd,
1201 0x75df, 0x75e0, 0x75e1, 0x75e5, 0x75e9, 0x75ec, 0x75ed, 0x75ee,
1202 0x75ef, 0x75f2, 0x75f3, 0x75f5, 0x75f6, 0x75f7, 0x75f8, 0x75fa,
1203 0x75fb, 0x75fd, 0x75fe, 0x75fe, 0x7602, 0x7604, 0x7607, 0x7608,
1204 0x7609, 0x760b, 0x760d, 0x760e, 0x760f, 0x7611, 0x7612, 0x7613,
1205 0x7614, 0x7616, 0x761a, 0x761c, 0x761d, 0x761e, 0x7621, 0x7623,
1206 0x7627, 0x7628, 0x762c, 0x762c, 0x762e, 0x762f, 0x7631, 0x7632, 0x7636,
1207 0x7637, 0x7639, 0x763a, 0x763b, 0x763d, 0x7641, 0x7642, 0x7644,
1208 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
1209 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
1210 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
1211 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
1212 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
1213 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
1214 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
1215 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
1216 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
```

```

1217 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
1218 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
1219 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
1220 /* 0xb0 */
1221 0x7645, 0x7646, 0x7647, 0x7648, 0x7649, 0x764a, 0x764b, 0x764e,
1222 0x764f, 0x7650, 0x7651, 0x7652, 0x7653, 0x7655, 0x7657, 0x7658,
1223 0x7659, 0x765a, 0x765b, 0x765d, 0x765f, 0x7660, 0x7661, 0x7662,
1224 0x7664, 0x7665, 0x7666, 0x7667, 0x7668, 0x7669, 0x766a, 0x766c,
1225 0x766d, 0x766e, 0x7670, 0x7671, 0x7672, 0x7673, 0x7674, 0x7675,
1226 0x7676, 0x7677, 0x7679, 0x767a, 0x767c, 0x767f, 0x7680, 0x7681,
1227 0x7683, 0x7685, 0x7689, 0x768a, 0x768c, 0x768d, 0x768f, 0x7690,
1228 0x7692, 0x7694, 0x7695, 0x7695, 0x7697, 0x7698, 0x769a, 0x769c,
1229 0x769d, 0x769e, 0x769f, 0x76a0, 0x76a1, 0x76a2, 0x76a3, 0x76a5,
1230 0x76a6, 0x76a7, 0x76a8, 0x76a9, 0x76aa, 0x76ab, 0x76ac, 0x76ad,
1231 0x76af, 0x76b0, 0x76b3, 0x76b5, 0x76b6, 0x76b7, 0x76b8, 0x76b9,
1232 0x76ba, 0x76bb, 0x76bc, 0x76bd, 0x76be, 0x76c0, 0x76c1, 0x76c3,
1233 0x554a, 0x963f, 0x57c3, 0x6328, 0x54ce, 0x5509, 0x54c0, 0x7691,
1234 0x764c, 0x853c, 0x77ee, 0x827e, 0x788d, 0x7231, 0x9698, 0x978d,
1235 0x6c28, 0x5b89, 0x4ffa, 0x6309, 0x6697, 0x5cb8, 0x80fa, 0x6848,
1236 0x80ae, 0x6602, 0x76ce, 0x51f9, 0x6556, 0x71ac, 0x7ff1, 0x8884,
1237 0x50b2, 0x5965, 0x61ca, 0x6fb3, 0x82ad, 0x634c, 0x6252, 0x53ed,
1238 0x5427, 0x7b06, 0x516b, 0x75a4, 0x5df4, 0x62d4, 0x8dcb, 0x9776,
1239 0x628a, 0x8019, 0x575d, 0x9738, 0x7f62, 0x7238, 0x767d, 0x67c1,
1240 0x767e, 0x6446, 0x4f70, 0x8d25, 0x62dc, 0x7a17, 0x6591, 0x73ed,
1241 0x642c, 0x6273, 0x822c, 0x9881, 0x677f, 0x7248, 0x626e, 0x62cc,
1242 0x4f34, 0x74e3, 0x534a, 0x529e, 0x7eca, 0x90a6, 0x5e2e, 0x6886,
1243 0x699c, 0x8180, 0x7ed1, 0x68d2, 0x78c5, 0x868c, 0x9551, 0x508d,
1244 0x8c24, 0x82de, 0x80de, 0x5305, 0x8912, 0x5265,
1245 /* 0xb1 */
1246 0x76c4, 0x76c7, 0x76c9, 0x76cb, 0x76cc, 0x76d3, 0x76d5, 0x76d9,
1247 0x76da, 0x76dc, 0x76dd, 0x76de, 0x76e0, 0x76e1, 0x76e2, 0x76e3,
1248 0x76e4, 0x76e6, 0x76e7, 0x76e8, 0x76e9, 0x76ea, 0x76eb, 0x76ec,
1249 0x76ed, 0x76f0, 0x76f3, 0x76f5, 0x76f6, 0x76f7, 0x76fa, 0x76fb,
1250 0x76fd, 0x76ff, 0x7700, 0x7702, 0x7703, 0x7705, 0x7706, 0x770a,
1251 0x770c, 0x770e, 0x770f, 0x7710, 0x7711, 0x7712, 0x7713, 0x7714,
1252 0x7715, 0x7716, 0x7717, 0x7718, 0x771b, 0x771c, 0x771d, 0x771e,
1253 0x7721, 0x7723, 0x7724, 0x7725, 0x7727, 0x772a, 0x772b, 0x772c,
1254 0x772e, 0x7730, 0x7731, 0x7732, 0x7733, 0x7734, 0x7739, 0x773b,
1255 0x773d, 0x773e, 0x773f, 0x7742, 0x7744, 0x7745, 0x7746, 0x7748,
1256 0x7749, 0x774a, 0x774b, 0x774c, 0x774d, 0x774e, 0x774f, 0x7752,
1257 0x7753, 0x7754, 0x7755, 0x7756, 0x7757, 0x7758, 0x7759, 0x775c,
1258 0x8584, 0x96f9, 0x4fdd, 0x5821, 0x9971, 0x5b9d, 0x62b1, 0x62a5,
1259 0x66b4, 0x8c79, 0x9c8d, 0x7206, 0x676f, 0x7891, 0x60b2, 0x5351,
1260 0x5317, 0x8f88, 0x80cc, 0x8d1d, 0x94a1, 0x500d, 0x72c8, 0x5907,
1261 0x60eb, 0x7119, 0x88ab, 0x5954, 0x82ef, 0x672c, 0x7b28, 0x5d29,
1262 0x7ef7, 0x752d, 0x6cf5, 0x8e66, 0x8ff8, 0x903c, 0x9f3b, 0x6bd4,
1263 0x9119, 0x7b14, 0x5f7c, 0x78a7, 0x84d6, 0x853d, 0x6bd5, 0x6bd9,
1264 0x6bd6, 0x5e01, 0x5e87, 0x75f9, 0x95ed, 0x655d, 0x5f0a, 0x5f5c,
1265 0x8f9f, 0x58c1, 0x81c2, 0x907f, 0x965b, 0x97ad, 0x8fb9, 0x7f16,
1266 0x8d2c, 0x6241, 0x4fbf, 0x53d8, 0x535e, 0x8fa8, 0x8fa9, 0x8fab,
1267 0x904d, 0x6807, 0x5f6a, 0x8198, 0x8868, 0x9cd6, 0x618b, 0x522b,
1268 0x762a, 0x5f6c, 0x658c, 0x6fd2, 0x6ee8, 0x5bbe, 0x6448, 0x5175,
1269 0x51b0, 0x67c4, 0x4e19, 0x79c9, 0x997c, 0x70b3,
1270 /* 0xb2 */
1271 0x775d, 0x775e, 0x775f, 0x7760, 0x7764, 0x7767, 0x7769, 0x776a,
1272 0x776d, 0x776e, 0x776f, 0x7770, 0x7771, 0x7772, 0x7773, 0x7774,
1273 0x7775, 0x7776, 0x7777, 0x7778, 0x777a, 0x777b, 0x777c, 0x7781,
1274 0x7782, 0x7783, 0x7786, 0x7787, 0x7788, 0x7789, 0x778a, 0x778b,
1275 0x778f, 0x7790, 0x7793, 0x7794, 0x7795, 0x7796, 0x7797, 0x7798,
1276 0x7799, 0x779a, 0x779b, 0x779c, 0x779d, 0x779e, 0x77a1, 0x77a3,
1277 0x77a4, 0x77a6, 0x77a8, 0x77ab, 0x77ad, 0x77ae, 0x77af, 0x77b1,
1278 0x77b2, 0x77b4, 0x77b6, 0x77b7, 0x77b8, 0x77b9, 0x77ba, 0x77bc,
1279 0x77be, 0x77c0, 0x77c1, 0x77c2, 0x77c3, 0x77c4, 0x77c5, 0x77c6,
1280 0x77c7, 0x77c8, 0x77c9, 0x77ca, 0x77cb, 0x77cc, 0x77ce, 0x77cf,
1281 0x77d0, 0x77d1, 0x77d2, 0x77d3, 0x77d4, 0x77d5, 0x77de, 0x77d8,
1282 0x77d9, 0x77da, 0x77dd, 0x77de, 0x77df, 0x77e0, 0x77e1, 0x77e4,
1283 0x75c5, 0x5e76, 0x73bb, 0x83e0, 0x64ad, 0x62e8, 0x94b5, 0x6ce2,
1284 0x535a, 0x52c3, 0x640f, 0x94c2, 0x7b94, 0x4f2f, 0x5e1b, 0x8236,
1285 0x8116, 0x818a, 0x6e24, 0x6cca, 0x9a73, 0x6355, 0x535c, 0x54fa,
1286 0x8865, 0x57e0, 0x4e0d, 0x5e03, 0x6b65, 0x7c3f, 0x90e8, 0x6016,
1287 0x64e6, 0x731c, 0x88c1, 0x6750, 0x624d, 0x8d22, 0x776c, 0x8e29,
1288 0x91c7, 0x5f69, 0x83dc, 0x8521, 0x9910, 0x53c2, 0x8695, 0x6b8b,
1289 0x60ed, 0x60e8, 0x707f, 0x82cd, 0x8231, 0x4ed3, 0x6ca7, 0x85cf,
1290 0x64cd, 0x7cd9, 0x69fd, 0x66f9, 0x8349, 0x5395, 0x7b56, 0x4fa7,
1291 0x518c, 0x6d4b, 0x5c42, 0x8e6d, 0x63d2, 0x53c9, 0x832c, 0x8336,
1292 0x67e5, 0x78b4, 0x643d, 0x5bdf, 0x5c94, 0x5dee, 0x8be7, 0x62c6,
1293 0x67f4, 0x8c7a, 0x6400, 0x63ba, 0x8749, 0x998b, 0x8c17, 0x7f20,
1294 0x94f2, 0x4ea7, 0x9610, 0x98a4, 0x660c, 0x7316,
1295 /* 0xb3 */
1296 0x77e6, 0x77e8, 0x77ea, 0x77ef, 0x77f0, 0x77f1, 0x77f2, 0x77f4,
1297 0x77f5, 0x77f7, 0x77f9, 0x77fa, 0x77fb, 0x77fc, 0x7803, 0x7804,
1298 0x7805, 0x7806, 0x7807, 0x7808, 0x780a, 0x780b, 0x780e, 0x780f,
1299 0x7810, 0x7813, 0x7815, 0x7815, 0x7819, 0x781b, 0x781e, 0x7820, 0x7821,
1300 0x7822, 0x7824, 0x7828, 0x782a, 0x782b, 0x782e, 0x782f, 0x7831,
1301 0x7832, 0x7833, 0x7835, 0x7836, 0x783d, 0x783f, 0x7841, 0x7842,
1302 0x7843, 0x7844, 0x7846, 0x7846, 0x7848, 0x7849, 0x784a, 0x784d,
1303 0x784f, 0x7851, 0x7853, 0x7854, 0x7858, 0x7859, 0x785a, 0x785b,

```



```
1304 0x785c, 0x785e, 0x785f, 0x7860, 0x7861, 0x7862, 0x7863, 0x7864,
1305 0x7865, 0x7866, 0x7867, 0x7868, 0x7869, 0x786f, 0x7870, 0x7871,
1306 0x7872, 0x7873, 0x7874, 0x7875, 0x7876, 0x7878, 0x7879, 0x787a,
1307 0x787b, 0x787d, 0x787e, 0x787f, 0x7880, 0x7881, 0x7882, 0x7883,
1308 0x573a, 0x5c1d, 0x5e38, 0x957f, 0x507f, 0x80a0, 0x5382, 0x655e,
1309 0x7545, 0x5531, 0x5021, 0x8d85, 0x6284, 0x949e, 0x671d, 0x5632,
1310 0x6f6e, 0x5de2, 0x5435, 0x7092, 0x8f66, 0x626f, 0x64a4, 0x63a3,
1311 0x5f7b, 0x6f88, 0x90f4, 0x81e3, 0x8fb0, 0x5c18, 0x6668, 0x5ff1,
1312 0x6c89, 0x9648, 0x8d81, 0x886c, 0x6491, 0x79f0, 0x57ce, 0x6a59,
1313 0x6210, 0x5448, 0x4e58, 0x7a0b, 0x60e9, 0x6f84, 0x8bda, 0x627f,
1314 0x901e, 0x9a8b, 0x79e4, 0x5403, 0x75f4, 0x6301, 0x5319, 0x6c60,
1315 0x8fdf, 0x5f1b, 0x9a70, 0x803b, 0x9f7f, 0x4f88, 0x5c3a, 0x8d64,
1316 0x7fc5, 0x65a5, 0x70bd, 0x5145, 0x51b2, 0x866b, 0x5d07, 0x5ba0,
1317 0x62bd, 0x916c, 0x7574, 0x8e0c, 0x7a20, 0x6101, 0x7b79, 0x4ec7,
1318 0x7ef8, 0x7785, 0x4e11, 0x81ed, 0x521d, 0x51fa, 0x6a71, 0x53a8,
1319 0x8e87, 0x9504, 0x96cf, 0x6ec1, 0x9664, 0x695a,
1320 /* 0xb4 */
1321 0x7884, 0x7885, 0x7886, 0x7888, 0x788a, 0x788b, 0x788f, 0x7890,
1322 0x7892, 0x7894, 0x7895, 0x7896, 0x7899, 0x789d, 0x789e, 0x78a0,
1323 0x78a2, 0x78a4, 0x78a6, 0x78a8, 0x78a9, 0x78aa, 0x78ab, 0x78ac,
1324 0x78ad, 0x78ae, 0x78af, 0x78b5, 0x78b6, 0x78b7, 0x78b8, 0x78ba,
1325 0x78bb, 0x78bc, 0x78bd, 0x78bf, 0x78c0, 0x78c2, 0x78c3, 0x78c4,
1326 0x78c6, 0x78c7, 0x78c8, 0x78cc, 0x78cd, 0x78ce, 0x78cf, 0x78d1,
1327 0x78d2, 0x78d3, 0x78d6, 0x78d7, 0x78d8, 0x78da, 0x78db, 0x78dc,
1328 0x78dd, 0x78de, 0x78df, 0x78e0, 0x78e1, 0x78e2, 0x78e3, 0x78e4,
1329 0x78e5, 0x78e6, 0x78e7, 0x78e9, 0x78ea, 0x78eb, 0x78ed, 0x78ee,
1330 0x78ef, 0x78f0, 0x78f1, 0x78f3, 0x78f5, 0x78f6, 0x78f8, 0x78f9,
1331 0x78fb, 0x78fc, 0x78fd, 0x78fe, 0x78ff, 0x7900, 0x7902, 0x7903,
1332 0x7904, 0x7906, 0x7907, 0x7908, 0x7909, 0x790a, 0x790b, 0x790c,
1333 0x7840, 0x50a8, 0x77d7, 0x6410, 0x89e6, 0x5904, 0x63e3, 0x5ddd,
1334 0x7a7f, 0x693d, 0x4f20, 0x8239, 0x5598, 0x4e32, 0x75ae, 0x7a97,
1335 0x5e62, 0x5e8a, 0x95ef, 0x521b, 0x5439, 0x708a, 0x6376, 0x9524,
1336 0x5782, 0x6625, 0x693f, 0x9187, 0x5507, 0x6df3, 0x7eaf, 0x8822,
1337 0x6233, 0x7ef0, 0x75b5, 0x8328, 0x78c1, 0x96cc, 0x8f9e, 0x6148,
1338 0x74f7, 0x8bcd, 0x6b64, 0x523a, 0x8d50, 0x6b21, 0x806a, 0x8471,
1339 0x56f1, 0x5306, 0x4ece, 0x4e1b, 0x51d1, 0x7c97, 0x918b, 0x7c07,
1340 0x4fc3, 0x8e7f, 0x7be1, 0x7a9c, 0x6467, 0x5d14, 0x50ac, 0x8106,
1341 0x7601, 0x7cb9, 0x6dec, 0x7fe0, 0x6751, 0x5b58, 0x5bf8, 0x78cb,
1342 0x64ae, 0x6413, 0x63aa, 0x632b, 0x9519, 0x642d, 0x8fbc, 0x7b54,
1343 0x7629, 0x6253, 0x5927, 0x5446, 0x6b79, 0x50a3, 0x6234, 0x5e26,
1344 0x6b86, 0x4ee3, 0x8d37, 0x888b, 0x5f85, 0x902e,
1345 /* 0xb5 */
1346 0x790d, 0x790e, 0x790f, 0x7910, 0x7911, 0x7912, 0x7914, 0x7915,
1347 0x7916, 0x7917, 0x7918, 0x7919, 0x791a, 0x791b, 0x791c, 0x791d,
1348 0x791f, 0x7920, 0x7921, 0x7922, 0x7923, 0x7925, 0x7926, 0x7927,
1349 0x7928, 0x7929, 0x792a, 0x792b, 0x792c, 0x792d, 0x792e, 0x792f,
1350 0x7930, 0x7931, 0x7932, 0x7933, 0x7935, 0x7936, 0x7937, 0x7938,
1351 0x7939, 0x793d, 0x793f, 0x7942, 0x7943, 0x7944, 0x7945, 0x7947,
1352 0x794a, 0x794b, 0x794c, 0x794d, 0x794e, 0x794f, 0x7950, 0x7951,
1353 0x7952, 0x7954, 0x7955, 0x7958, 0x7959, 0x7961, 0x7963, 0x7964,
1354 0x7966, 0x7969, 0x796a, 0x796b, 0x796c, 0x796e, 0x7970, 0x7971,
1355 0x7972, 0x7973, 0x7974, 0x7975, 0x7976, 0x7979, 0x797b, 0x797c,
1356 0x797d, 0x797e, 0x797f, 0x7982, 0x7983, 0x7986, 0x7987, 0x7988,
1357 0x7989, 0x798b, 0x798c, 0x798d, 0x798e, 0x7990, 0x7991, 0x7992,
1358 0x6020, 0x803d, 0x62c5, 0x4e39, 0x5355, 0x90f8, 0x63b8, 0x80c6,
1359 0x65e6, 0x6c2e, 0x4f46, 0x60ee, 0x6de1, 0x8bde, 0x5f39, 0x86cb,
1360 0x5f53, 0x6321, 0x515a, 0x8361, 0x6863, 0x5200, 0x6363, 0x8e48,
1361 0x5012, 0x5c9b, 0x7977, 0x5bfc, 0x5230, 0x7a3b, 0x60bc, 0x9053,
1362 0x76d7, 0x5fb7, 0x5f97, 0x7684, 0x8e6c, 0x706f, 0x767b, 0x7b49,
1363 0x77aa, 0x51f3, 0x9093, 0x5824, 0x4f4e, 0x6ef4, 0x8fea, 0x654c,
1364 0x7b1b, 0x72c4, 0x6da4, 0x7fdf, 0x5ae1, 0x62b5, 0x5e95, 0x5730,
1365 0x8482, 0x7b2c, 0x5e1d, 0x5f1f, 0x9012, 0x7f14, 0x98a0, 0x6382,
1366 0x6ec7, 0x7898, 0x70b9, 0x5178, 0x975b, 0x57ab, 0x7535, 0x4f43,
1367 0x7538, 0x5e97, 0x60e6, 0x5960, 0x6dc0, 0x6bbf, 0x7889, 0x53fc,
1368 0x96d5, 0x51cb, 0x5201, 0x6389, 0x540a, 0x9493, 0x8c03, 0x8dcc,
1369 0x7239, 0x789f, 0x8776, 0x8fed, 0x8c0d, 0x53e0,
1370 /* 0xb6 */
1371 0x7993, 0x7994, 0x7995, 0x7996, 0x7997, 0x7998, 0x7999, 0x799b,
1372 0x799c, 0x799d, 0x799e, 0x799f, 0x79a0, 0x79a1, 0x79a2, 0x79a3,
1373 0x79a4, 0x79a5, 0x79a6, 0x79a8, 0x79a9, 0x79aa, 0x79ab, 0x79ac,
1374 0x79ad, 0x79ae, 0x79af, 0x79b0, 0x79b1, 0x79b2, 0x79b4, 0x79b5,
1375 0x79b6, 0x79b7, 0x79b8, 0x79bc, 0x79bf, 0x79c2, 0x79c4, 0x79c5,
1376 0x79c7, 0x79c8, 0x79ca, 0x79cc, 0x79ce, 0x79cf, 0x79d0, 0x79d3,
1377 0x79d4, 0x79d6, 0x79d7, 0x79d9, 0x79da, 0x79db, 0x79dc, 0x79dd,
1378 0x79de, 0x79e0, 0x79e1, 0x79e2, 0x79e5, 0x79e8, 0x79ea, 0x79ec,
1379 0x79ee, 0x79f1, 0x79f2, 0x79f3, 0x79f4, 0x79f5, 0x79f6, 0x79f7,
1380 0x79f9, 0x79fa, 0x79fc, 0x79fd, 0x79fe, 0x79ff, 0x7a01, 0x7a04, 0x7a05,
1381 0x7a07, 0x7a08, 0x7a09, 0x7a0a, 0x7a0c, 0x7a0f, 0x7a10, 0x7a11,
1382 0x7a12, 0x7a13, 0x7a15, 0x7a16, 0x7a18, 0x7a19, 0x7a1b, 0x7a1c,
1383 0x4e01, 0x76ef, 0x53ee, 0x9489, 0x9876, 0x9f0e, 0x952d, 0x5b9a,
1384 0x8ba2, 0x4e22, 0x4e1c, 0x51ac, 0x8463, 0x61c2, 0x52a8, 0x680b,
1385 0x4f97, 0x606b, 0x51bb, 0x6d1e, 0x515c, 0x6296, 0x6597, 0x9661,
1386 0x8c46, 0x9017, 0x75d8, 0x90fd, 0x7763, 0x6bd2, 0x728a, 0x72ec,
1387 0x8bfb, 0x5835, 0x7779, 0x8d4c, 0x675c, 0x9540, 0x809a, 0x5ea6,
1388 0x6e21, 0x5992, 0x7aef, 0x77ed, 0x953b, 0x6bb5, 0x65ad, 0x7f0e,
1389 0x5806, 0x5151, 0x961f, 0x5bf9, 0x58a9, 0x5428, 0x8e72, 0x6566,
1390 0x987f, 0x56e4, 0x949d, 0x76fe, 0x9041, 0x6387, 0x54c6, 0x591a,
```

```
1391 0x593a, 0x579b, 0x8eb2, 0x6735, 0x8dfa, 0x8235, 0x5241, 0x60f0,
1392 0x5815, 0x86fe, 0x5ce8, 0x9e45, 0x4fc4, 0x989d, 0x8bb9, 0x5a25,
1393 0x6076, 0x5384, 0x627c, 0x904f, 0x9102, 0x997f, 0x6069, 0x800c,
1394 0x513f, 0x8033, 0x5c14, 0x9975, 0x6d31, 0x4e8c,
1395 /* 0xb7 */
1396 0x7a1d, 0x7a1f, 0x7a21, 0x7a22, 0x7a24, 0x7a25, 0x7a26, 0x7a27,
1397 0x7a28, 0x7a29, 0x7a2a, 0x7a2b, 0x7a2c, 0x7a2d, 0x7a2e, 0x7a2f,
1398 0x7a30, 0x7a31, 0x7a32, 0x7a34, 0x7a35, 0x7a36, 0x7a38, 0x7a3a,
1399 0x7a3e, 0x7a40, 0x7a41, 0x7a42, 0x7a43, 0x7a44, 0x7a45, 0x7a47,
1400 0x7a48, 0x7a49, 0x7a4a, 0x7a4b, 0x7a4c, 0x7a4d, 0x7a4e, 0x7a4f,
1401 0x7a50, 0x7a52, 0x7a53, 0x7a54, 0x7a55, 0x7a56, 0x7a58, 0x7a59,
1402 0x7a5a, 0x7a5b, 0x7a5c, 0x7a5d, 0x7a5e, 0x7a5f, 0x7a60, 0x7a61,
1403 0x7a62, 0x7a63, 0x7a64, 0x7a65, 0x7a66, 0x7a67, 0x7a68, 0x7a69,
1404 0x7a6a, 0x7a6b, 0x7a6c, 0x7a6d, 0x7a6e, 0x7a6f, 0x7a71, 0x7a72,
1405 0x7a73, 0x7a75, 0x7a7b, 0x7a7c, 0x7a7d, 0x7a7e, 0x7a82, 0x7a85,
1406 0x7a87, 0x7a89, 0x7a8a, 0x7a8b, 0x7a8c, 0x7a8e, 0x7a8f, 0x7a90,
1407 0x7a93, 0x7a94, 0x7a99, 0x7a9a, 0x7a9b, 0x7a9e, 0x7aa1, 0x7aa2,
1408 0x8d30, 0x53d1, 0x7f5a, 0x7b4f, 0x4f10, 0x4e4f, 0x9600, 0x6cd5,
1409 0x73d0, 0x85e9, 0x5e06, 0x756a, 0x7ffb, 0x6a0a, 0x77fe, 0x9492,
1410 0x7e41, 0x51e1, 0x70e6, 0x53cd, 0x8fd4, 0x8303, 0x8d29, 0x72af,
1411 0x996d, 0x6cdb, 0x574a, 0x82b3, 0x65b9, 0x80aa, 0x623f, 0x9632,
1412 0x59a8, 0x4eff, 0x8bbf, 0x7eba, 0x653e, 0x83f2, 0x975e, 0x5561,
1413 0x98de, 0x80a5, 0x532a, 0x8bfd, 0x5420, 0x80ba, 0x5e9f, 0x6cb8,
1414 0x8d39, 0x82ac, 0x915a, 0x5429, 0x6c1b, 0x5206, 0x7eb7, 0x575f,
1415 0x711a, 0x6c7e, 0x7c89, 0x594b, 0x4efd, 0x5fff, 0x6124, 0x7caa,
1416 0x4e30, 0x5c01, 0x67ab, 0x8702, 0x5cf0, 0x950b, 0x98ce, 0x75af,
1417 0x70fd, 0x9022, 0x51af, 0x7f1d, 0x8bbd, 0x5949, 0x51e4, 0x4f5b,
1418 0x5426, 0x592b, 0x6577, 0x80a4, 0x5b75, 0x6276, 0x62c2, 0x8f90,
1419 0x5e45, 0x6c1f, 0x7b26, 0x4f0f, 0x4fd8, 0x670d,
1420 /* 0xb8 */
1421 0x7aa3, 0x7aa4, 0x7aa7, 0x7aa9, 0x7aaa, 0x7aab, 0x7aae, 0x7aaf,
1422 0x7ab0, 0x7ab1, 0x7ab2, 0x7ab4, 0x7ab5, 0x7ab6, 0x7ab7, 0x7ab8,
1423 0x7ab9, 0x7aba, 0x7abb, 0x7abc, 0x7abd, 0x7abe, 0x7ac0, 0x7acl,
1424 0x7ac2, 0x7ac3, 0x7ac4, 0x7ac5, 0x7ac6, 0x7ac7, 0x7ac8, 0x7ac9,
1425 0x7aca, 0x7acc, 0x7acd, 0x7ace, 0x7acf, 0x7ad0, 0x7ad1, 0x7ad2,
1426 0x7ad3, 0x7ad4, 0x7ad5, 0x7ad7, 0x7ad8, 0x7ada, 0x7adb, 0x7adc,
1427 0x7add, 0x7ae1, 0x7ae2, 0x7ae4, 0x7ae7, 0x7ae8, 0x7ae9, 0x7aea,
1428 0x7aeb, 0x7aec, 0x7aee, 0x7af0, 0x7af1, 0x7af2, 0x7af3, 0x7af4,
1429 0x7af5, 0x7af6, 0x7af7, 0x7af8, 0x7afb, 0x7afc, 0x7afe, 0x7b00,
1430 0x7b01, 0x7b02, 0x7b05, 0x7b07, 0x7b09, 0x7b0c, 0x7b0d, 0x7b0e,
1431 0x7b10, 0x7b12, 0x7b13, 0x7b16, 0x7b17, 0x7b18, 0x7b1a, 0x7b1c,
1432 0x7b1d, 0x7b1f, 0x7b21, 0x7b22, 0x7b23, 0x7b27, 0x7b29, 0x7b2d,
1433 0x6d6e, 0x6daa, 0x798f, 0x88b1, 0x5f17, 0x752b, 0x629a, 0x8f85,
1434 0x4fef, 0x91dc, 0x65a7, 0x812f, 0x8151, 0x5e9c, 0x8150, 0x8d74,
1435 0x526f, 0x8986, 0x8d4b, 0x590d, 0x5085, 0x4ed8, 0x961c, 0x7236,
1436 0x8179, 0x8d1f, 0x5bcc, 0x8ba3, 0x9644, 0x5987, 0x7f1a, 0x5490,
1437 0x5676, 0x560e, 0x8be5, 0x6539, 0x6982, 0x9499, 0x76d6, 0x6e89,
1438 0x5e72, 0x7518, 0x6746, 0x67d1, 0x7aff, 0x809d, 0x8d76, 0x611f,
1439 0x79c6, 0x6562, 0x8d63, 0x5188, 0x521a, 0x94a2, 0x7f38, 0x809b,
1440 0x7eb2, 0x5c97, 0x6e2f, 0x6760, 0x7bd9, 0x768b, 0x9ad8, 0x818f,
1441 0x7f94, 0x7cd5, 0x641e, 0x9550, 0x7a3f, 0x544a, 0x54e5, 0x6b4c,
1442 0x6401, 0x6208, 0x9e3d, 0x80f3, 0x7599, 0x5272, 0x9769, 0x845b,
1443 0x683c, 0x86e4, 0x9601, 0x9694, 0x94ec, 0x4e2a, 0x5404, 0x7ed9,
1444 0x6839, 0x8ddf, 0x8015, 0x66f4, 0x5e9a, 0x7fb9,
1445 /* 0xb9 */
1446 0x7b2f, 0x7b30, 0x7b32, 0x7b34, 0x7b35, 0x7b36, 0x7b37, 0x7b39,
1447 0x7b3b, 0x7b3d, 0x7b3f, 0x7b40, 0x7b41, 0x7b42, 0x7b43, 0x7b44,
1448 0x7b46, 0x7b48, 0x7b4a, 0x7b4d, 0x7b4e, 0x7b53, 0x7b55, 0x7b57,
1449 0x7b59, 0x7b5c, 0x7b5e, 0x7b5f, 0x7b61, 0x7b63, 0x7b64, 0x7b65,
1450 0x7b66, 0x7b67, 0x7b68, 0x7b69, 0x7b6a, 0x7b6b, 0x7b6c, 0x7b6d,
1451 0x7b6f, 0x7b70, 0x7b73, 0x7b74, 0x7b76, 0x7b78, 0x7b7a, 0x7b7c,
1452 0x7b7d, 0x7b7f, 0x7b81, 0x7b82, 0x7b83, 0x7b84, 0x7b86, 0x7b87,
1453 0x7b88, 0x7b89, 0x7b8a, 0x7b8b, 0x7b8c, 0x7b8e, 0x7b8f, 0x7b91,
1454 0x7b92, 0x7b93, 0x7b96, 0x7b98, 0x7b99, 0x7b9a, 0x7b9b, 0x7b9e,
1455 0x7b9f, 0x7ba0, 0x7ba3, 0x7ba4, 0x7ba5, 0x7bae, 0x7baf, 0x7bb0,
1456 0x7bb2, 0x7bb3, 0x7bb5, 0x7bb6, 0x7bb7, 0x7bb9, 0x7bba, 0x7bbb,
1457 0x7bbc, 0x7bbd, 0x7bbe, 0x7bbf, 0x7bc0, 0x7bc2, 0x7bc3, 0x7bc4,
1458 0x57c2, 0x803f, 0x6897, 0x5de5, 0x653b, 0x529f, 0x606d, 0x9f9a,
1459 0x4f9b, 0x8eac, 0x516c, 0x5bab, 0x5f13, 0x5de9, 0x6c5e, 0x62f1,
1460 0x8d21, 0x5171, 0x94a9, 0x52fe, 0x6c9f, 0x82df, 0x72d7, 0x57a2,
1461 0x6784, 0x8d2d, 0x591f, 0x8f9c, 0x83c7, 0x5495, 0x7b8d, 0x4f30,
1462 0x6cbd, 0x5b64, 0x59d1, 0x9f13, 0x53e4, 0x86ca, 0x9aa8, 0x8c37,
1463 0x80a1, 0x6545, 0x987e, 0x56fa, 0x96c7, 0x522e, 0x74dc, 0x5250,
1464 0x5be1, 0x6302, 0x8902, 0x4e56, 0x62d0, 0x602a, 0x68fa, 0x5173,
1465 0x5b98, 0x51a0, 0x89c2, 0x7ba1, 0x9986, 0x7f50, 0x60ef, 0x704c,
1466 0x8d2f, 0x5149, 0x5e7f, 0x901b, 0x7470, 0x89c4, 0x572d, 0x7845,
1467 0x5f52, 0x9f9f, 0x95fa, 0x8f68, 0x9b3c, 0x8be1, 0x7678, 0x6842,
1468 0x67dc, 0x8dea, 0x8d35, 0x523d, 0x8f8a, 0x6eda, 0x68cd, 0x9505,
1469 0x90ed, 0x56fd, 0x679c, 0x88f9, 0x8fc7, 0x54c8,
1470 /* 0xba */
1471 0x7bc5, 0x7bc8, 0x7bc9, 0x7bca, 0x7bcb, 0x7bcd, 0x7bce, 0x7bcf,
1472 0x7bd0, 0x7bd2, 0x7bd4, 0x7bd5, 0x7bd6, 0x7bd7, 0x7bd8, 0x7bdb,
1473 0x7bdc, 0x7bde, 0x7bdf, 0x7be0, 0x7be2, 0x7be3, 0x7be4, 0x7be7,
1474 0x7be8, 0x7be9, 0x7beb, 0x7bec, 0x7bed, 0x7bef, 0x7bf0, 0x7bf2,
1475 0x7bf3, 0x7bf4, 0x7bf5, 0x7bf6, 0x7bf8, 0x7bf9, 0x7bfa, 0x7bfb,
1476 0x7bfd, 0x7bff, 0x7c00, 0x7c01, 0x7c02, 0x7c03, 0x7c04, 0x7c05,
1477 0x7c06, 0x7c08, 0x7c09, 0x7c0a, 0x7c0d, 0x7c0e, 0x7c10, 0x7c11,
```

```
1478 0x7c12, 0x7c13, 0x7c14, 0x7c15, 0x7c17, 0x7c18, 0x7c19, 0x7c1a,
1479 0x7c1b, 0x7c1c, 0x7c1d, 0x7c1e, 0x7c20, 0x7c21, 0x7c22, 0x7c23,
1480 0x7c24, 0x7c25, 0x7c28, 0x7c28, 0x7c29, 0x7c2b, 0x7c2c, 0x7c2d, 0x7c2e,
1481 0x7c2f, 0x7c30, 0x7c31, 0x7c32, 0x7c33, 0x7c34, 0x7c35, 0x7c36,
1482 0x7c37, 0x7c39, 0x7c3a, 0x7c3b, 0x7c3c, 0x7c3d, 0x7c3e, 0x7c42,
1483 0x9ab8, 0x5b69, 0x6d77, 0x6c26, 0x4ea5, 0x5bb3, 0x9a87, 0x9163,
1484 0x61a8, 0x90af, 0x97e9, 0x542b, 0x6db5, 0x5bd2, 0x51fd, 0x558a,
1485 0x7f55, 0x7ff0, 0x64bc, 0x634d, 0x65f1, 0x61be, 0x608d, 0x710a,
1486 0x6c57, 0x6c49, 0x592f, 0x676d, 0x822a, 0x58d5, 0x568e, 0x8c6a,
1487 0x6beb, 0x90dd, 0x597d, 0x8017, 0x53f7, 0x6d69, 0x5475, 0x559d,
1488 0x8377, 0x83cf, 0x6838, 0x79be, 0x548c, 0x4f55, 0x5408, 0x76d2,
1489 0x8c89, 0x9602, 0x6cb3, 0x6db8, 0x8d6b, 0x8910, 0x9e64, 0x8d3a,
1490 0x563f, 0x9ed1, 0x75d5, 0x5f88, 0x72e0, 0x6068, 0x54fc, 0x4ea8,
1491 0x6a2a, 0x8861, 0x6052, 0x8f70, 0x54c4, 0x70d8, 0x8679, 0x9e3f,
1492 0x6d2a, 0x5b8f, 0x5f18, 0x7ea2, 0x5589, 0x4faf, 0x7334, 0x543c,
1493 0x539a, 0x5019, 0x540e, 0x547c, 0x4e4e, 0x5ffd, 0x745a, 0x58f6,
1494 0x846b, 0x80e1, 0x8774, 0x72d0, 0x7cca, 0x6e56,
1495 /* 0xbb */
1496 0x7c43, 0x7c44, 0x7c45, 0x7c46, 0x7c47, 0x7c48, 0x7c49, 0x7c4a,
1497 0x7c4b, 0x7c4c, 0x7c4e, 0x7c4f, 0x7c50, 0x7c51, 0x7c52, 0x7c53,
1498 0x7c54, 0x7c55, 0x7c56, 0x7c57, 0x7c58, 0x7c59, 0x7c5a, 0x7c5b,
1499 0x7c5c, 0x7c5d, 0x7c5e, 0x7c5f, 0x7c60, 0x7c61, 0x7c62, 0x7c63,
1500 0x7c64, 0x7c65, 0x7c66, 0x7c67, 0x7c68, 0x7c69, 0x7c6a, 0x7c6b,
1501 0x7c6c, 0x7c6d, 0x7c6e, 0x7c6f, 0x7c70, 0x7c71, 0x7c72, 0x7c75,
1502 0x7c76, 0x7c77, 0x7c78, 0x7c79, 0x7c7a, 0x7c7e, 0x7c7f, 0x7c80,
1503 0x7c81, 0x7c82, 0x7c83, 0x7c84, 0x7c85, 0x7c86, 0x7c87, 0x7c88,
1504 0x7c8a, 0x7c8b, 0x7c8c, 0x7c8d, 0x7c8e, 0x7c8f, 0x7c90, 0x7c93,
1505 0x7c94, 0x7c96, 0x7c99, 0x7c9a, 0x7c9b, 0x7ca0, 0x7ca1, 0x7ca3,
1506 0x7ca6, 0x7ca7, 0x7ca8, 0x7ca9, 0x7cab, 0x7cac, 0x7cad, 0x7caf,
1507 0x7cb0, 0x7cb4, 0x7cb5, 0x7cb6, 0x7cb7, 0x7cb8, 0x7cba, 0x7cbb,
1508 0x5f27, 0x864e, 0x552c, 0x62a4, 0x4e92, 0x6caa, 0x6237, 0x82b1,
1509 0x54d7, 0x534e, 0x733e, 0x6ed1, 0x753b, 0x5212, 0x5316, 0x8bdd,
1510 0x69d0, 0x5f8a, 0x6000, 0x6dee, 0x574f, 0x6b22, 0x73af, 0x6853,
1511 0x8fd8, 0x7f13, 0x6362, 0x60a3, 0x5524, 0x75ea, 0x8c62, 0x7115,
1512 0x6da3, 0x5ba6, 0x5e7b, 0x8352, 0x614c, 0x9ec4, 0x78fa, 0x8757,
1513 0x7c27, 0x7687, 0x51f0, 0x60f6, 0x714c, 0x6643, 0x5e4c, 0x604d,
1514 0x8c0e, 0x7070, 0x6325, 0x8f89, 0x5fbd, 0x6062, 0x86d4, 0x56de,
1515 0x6bc1, 0x6094, 0x6167, 0x5349, 0x60e0, 0x6666, 0x8d3f, 0x79fd,
1516 0x4f1a, 0x70e9, 0x6c47, 0x8bb3, 0x8bf2, 0x7ed8, 0x8364, 0x660f,
1517 0x5a5a, 0x9b42, 0x6d51, 0x6df7, 0x8c41, 0x6d3b, 0x4f19, 0x706b,
1518 0x83b7, 0x6216, 0x60d1, 0x970d, 0x8d27, 0x7978, 0x51fb, 0x573e,
1519 0x57fa, 0x673a, 0x7578, 0x7a3d, 0x79ef, 0x7b95,
1520 /* 0xbc */
1521 0x7cbf, 0x7cc0, 0x7cc2, 0x7cc3, 0x7cc4, 0x7cc6, 0x7cc9, 0x7ccb,
1522 0x7cce, 0x7ccf, 0x7cd0, 0x7cd1, 0x7cd2, 0x7cd3, 0x7cd4, 0x7cd8,
1523 0x7cda, 0x7cdb, 0x7cdd, 0x7cde, 0x7ce1, 0x7ce2, 0x7ce3, 0x7ce4,
1524 0x7ce5, 0x7ce6, 0x7ce7, 0x7ce9, 0x7cea, 0x7ceb, 0x7cec, 0x7ced,
1525 0x7cee, 0x7cf0, 0x7cf1, 0x7cf2, 0x7cf3, 0x7cf4, 0x7cf5, 0x7cf6,
1526 0x7cf7, 0x7cf9, 0x7cfa, 0x7cfc, 0x7cfd, 0x7cfe, 0x7cff, 0x7d00,
1527 0x7d01, 0x7d02, 0x7d03, 0x7d04, 0x7d05, 0x7d06, 0x7d07, 0x7d08,
1528 0x7d09, 0x7d0b, 0x7d0c, 0x7d0d, 0x7d0e, 0x7d0f, 0x7d10, 0x7d11,
1529 0x7d12, 0x7d13, 0x7d14, 0x7d15, 0x7d16, 0x7d17, 0x7d18, 0x7d19,
1530 0x7d1a, 0x7d1b, 0x7d1c, 0x7d1d, 0x7d1e, 0x7d1f, 0x7d21, 0x7d23,
1531 0x7d24, 0x7d25, 0x7d26, 0x7d28, 0x7d29, 0x7d2a, 0x7d2c, 0x7d2d,
1532 0x7d2e, 0x7d30, 0x7d31, 0x7d32, 0x7d33, 0x7d34, 0x7d35, 0x7d36,
1533 0x808c, 0x9965, 0x8ff9, 0x6fc0, 0x8ba5, 0x9e21, 0x59ec, 0x7ee9,
1534 0x7f09, 0x5409, 0x6781, 0x68d8, 0x8f91, 0x7c4d, 0x96c6, 0x53ca,
1535 0x6025, 0x75be, 0x6c72, 0x5373, 0x5ac9, 0x7ea7, 0x6324, 0x51e0,
1536 0x810a, 0x5df1, 0x84df, 0x6280, 0x5180, 0x5b63, 0x4f0e, 0x796d,
1537 0x5242, 0x60b8, 0x6d4e, 0x5bc4, 0x5bc2, 0x8ba1, 0x8bb0, 0x65e2,
1538 0x5fcc, 0x9645, 0x5993, 0x7ee7, 0x7eaa, 0x5609, 0x67b7, 0x5939,
1539 0x4f73, 0x5bb6, 0x52a0, 0x835a, 0x988a, 0x8d3e, 0x7532, 0x94be,
1540 0x5047, 0x7a3c, 0x4ef7, 0x67b6, 0x9a7e, 0x5ac1, 0x6b7c, 0x76d1,
1541 0x575a, 0x5c16, 0x7b3a, 0x95f4, 0x714e, 0x517c, 0x80a9, 0x8270,
1542 0x5978, 0x7f04, 0x8327, 0x68c0, 0x67ec, 0x78b1, 0x7877, 0x62e3,
1543 0x6361, 0x7b80, 0x4fed, 0x526a, 0x51cf, 0x8350, 0x69db, 0x9274,
1544 0x8df5, 0x8d31, 0x89c1, 0x952e, 0x7bad, 0x4ef6,
1545 /* 0xbd */
1546 0x7d37, 0x7d38, 0x7d39, 0x7d3a, 0x7d3b, 0x7d3c, 0x7d3d, 0x7d3e,
1547 0x7d3f, 0x7d40, 0x7d41, 0x7d42, 0x7d43, 0x7d44, 0x7d45, 0x7d46,
1548 0x7d47, 0x7d48, 0x7d49, 0x7d4a, 0x7d4b, 0x7d4c, 0x7d4d, 0x7d4e,
1549 0x7d4f, 0x7d50, 0x7d51, 0x7d52, 0x7d53, 0x7d54, 0x7d55, 0x7d56,
1550 0x7d57, 0x7d58, 0x7d59, 0x7d5a, 0x7d5b, 0x7d5c, 0x7d5d, 0x7d5e,
1551 0x7d5f, 0x7d60, 0x7d61, 0x7d62, 0x7d63, 0x7d64, 0x7d65, 0x7d66,
1552 0x7d67, 0x7d68, 0x7d69, 0x7d6a, 0x7d6b, 0x7d6c, 0x7d6d, 0x7d6e,
1553 0x7d70, 0x7d71, 0x7d72, 0x7d73, 0x7d74, 0x7d75, 0x7d76, 0x7d78,
1554 0x7d79, 0x7d7a, 0x7d7b, 0x7d7c, 0x7d7d, 0x7d7e, 0x7d7f, 0x7d80,
1555 0x7d81, 0x7d82, 0x7d83, 0x7d84, 0x7d85, 0x7d86, 0x7d87, 0x7d88,
1556 0x7d89, 0x7d8a, 0x7d8b, 0x7d8c, 0x7d8d, 0x7d8e, 0x7d8f, 0x7d90,
1557 0x7d91, 0x7d92, 0x7d93, 0x7d94, 0x7d95, 0x7d96, 0x7d97, 0x7d98,
1558 0x5065, 0x8230, 0x5251, 0x996f, 0x6e10, 0x6e85, 0x6da7, 0x5efa,
1559 0x50f5, 0x59dc, 0x5c06, 0x6d46, 0x6c5f, 0x7586, 0x848b, 0x6868,
1560 0x5956, 0x8bb2, 0x5320, 0x9171, 0x964d, 0x8549, 0x6912, 0x7901,
1561 0x7126, 0x80f6, 0x4ea4, 0x90ca, 0x6d47, 0x9a84, 0x5a07, 0x56bc,
1562 0x6405, 0x94f0, 0x77eb, 0x4fa5, 0x811a, 0x72e1, 0x89d2, 0x997a,
1563 0x7f34, 0x7ede, 0x527f, 0x6559, 0x9175, 0x8f7f, 0x8f83, 0x53eb,
1564 0x7a96, 0x63ed, 0x63a5, 0x7686, 0x79f8, 0x8857, 0x9636, 0x622a,
```

```
1565 0x52ab, 0x8282, 0x6854, 0x6770, 0x6377, 0x776b, 0x7aed, 0x6d01,
1566 0x7ed3, 0x89e3, 0x59d0, 0x6212, 0x85c9, 0x82a5, 0x754c, 0x501f,
1567 0x4ecb, 0x75a5, 0x8bbeb, 0x5c4a, 0x5dfe, 0x7b4b, 0x65a4, 0x91d1,
1568 0x4eca, 0x6d25, 0x895f, 0x7d27, 0x9526, 0x4ec5, 0x8c28, 0x8fdb,
1569 0x9773, 0x664b, 0x7981, 0x8fd1, 0x70ec, 0x6d78,
1570 /* 0xbe */
1571 0x7d99, 0x7d9a, 0x7d9b, 0x7d9c, 0x7d9d, 0x7d9e, 0x7d9f, 0x7da0,
1572 0x7da1, 0x7da2, 0x7da3, 0x7da4, 0x7da5, 0x7da7, 0x7da8, 0x7da9,
1573 0x7daa, 0x7dab, 0x7dac, 0x7dad, 0x7daf, 0x7db0, 0x7db1, 0x7db2,
1574 0x7db3, 0x7db4, 0x7db5, 0x7db6, 0x7db7, 0x7db8, 0x7db9, 0x7dba,
1575 0x7dbb, 0x7dbc, 0x7dbd, 0x7dbe, 0x7dbf, 0x7dc0, 0x7dc1, 0x7dc2,
1576 0x7dc3, 0x7dc4, 0x7dc5, 0x7dc6, 0x7dc7, 0x7dc8, 0x7dc9, 0x7dca,
1577 0x7dcb, 0x7dcc, 0x7dcd, 0x7dce, 0x7dcf, 0x7dd0, 0x7dd1, 0x7dd2,
1578 0x7dd3, 0x7dd4, 0x7dd5, 0x7dd6, 0x7dd7, 0x7dd8, 0x7dd9, 0x7dda,
1579 0x7ddb, 0x7ddc, 0x7ddd, 0x7dde, 0x7ddf, 0x7de0, 0x7de1, 0x7de2,
1580 0x7de3, 0x7de4, 0x7de5, 0x7de6, 0x7de7, 0x7de8, 0x7de9, 0x7dea,
1581 0x7deb, 0x7dec, 0x7ded, 0x7dee, 0x7def, 0x7df0, 0x7df1, 0x7df2,
1582 0x7df3, 0x7df4, 0x7df5, 0x7df6, 0x7df7, 0x7df8, 0x7df9, 0x7dfa,
1583 0x5c3d, 0x52b2, 0x8346, 0x5162, 0x830e, 0x775b, 0x6676, 0x9cb8,
1584 0x4eac, 0x60ca, 0x7cbe, 0x7cb3, 0x7ecf, 0x4e95, 0x8b66, 0x666f,
1585 0x9888, 0x9759, 0x5883, 0x656c, 0x955c, 0x5f84, 0x75c9, 0x9756,
1586 0x7adf, 0x7ade, 0x51c0, 0x70af, 0x7a98, 0x63ea, 0x7a76, 0x7ea0,
1587 0x7396, 0x97ed, 0x4e45, 0x7078, 0x4e5d, 0x9152, 0x53a9, 0x6551,
1588 0x65e7, 0x81fc, 0x8205, 0x548e, 0x5c31, 0x759a, 0x97a0, 0x62d8,
1589 0x72d9, 0x75bd, 0x5c45, 0x9a79, 0x83ca, 0x5c40, 0x5480, 0x77e9,
1590 0x4e3e, 0x6cae, 0x805a, 0x62d2, 0x636e, 0x5de8, 0x5177, 0x8ddd,
1591 0x8e1e, 0x952f, 0x4ff1, 0x53e5, 0x60e7, 0x70ac, 0x5267, 0x6350,
1592 0x9e43, 0x5a1f, 0x5026, 0x7737, 0x5377, 0x7ee2, 0x6485, 0x652b,
1593 0x6289, 0x6398, 0x5014, 0x7235, 0x89c9, 0x51b3, 0x8bc0, 0x7edd,
1594 0x5747, 0x83cc, 0x94a7, 0x519b, 0x541b, 0x5cfb,
1595 /* 0xbf */
1596 0x7dfb, 0x7dfc, 0x7dfd, 0x7dfe, 0x7dff, 0x7e00, 0x7e01, 0x7e02,
1597 0x7e03, 0x7e04, 0x7e05, 0x7e06, 0x7e07, 0x7e08, 0x7e09, 0x7e0a,
1598 0x7e0b, 0x7e0c, 0x7e0d, 0x7e0e, 0x7e0f, 0x7e10, 0x7e11, 0x7e12,
1599 0x7e13, 0x7e14, 0x7e15, 0x7e16, 0x7e17, 0x7e18, 0x7e19, 0x7e1a,
1600 0x7e1b, 0x7e1c, 0x7e1d, 0x7e1e, 0x7e1f, 0x7e20, 0x7e21, 0x7e22,
1601 0x7e23, 0x7e24, 0x7e25, 0x7e26, 0x7e27, 0x7e28, 0x7e29, 0x7e2a,
1602 0x7e2b, 0x7e2c, 0x7e2d, 0x7e2e, 0x7e2f, 0x7e30, 0x7e31, 0x7e32,
1603 0x7e33, 0x7e34, 0x7e35, 0x7e36, 0x7e37, 0x7e38, 0x7e39, 0x7e3a,
1604 0x7e3c, 0x7e3d, 0x7e3e, 0x7e3f, 0x7e40, 0x7e42, 0x7e43, 0x7e44,
1605 0x7e45, 0x7e46, 0x7e48, 0x7e49, 0x7e4a, 0x7e4b, 0x7e4c, 0x7e4d,
1606 0x7e4e, 0x7e4f, 0x7e50, 0x7e51, 0x7e52, 0x7e53, 0x7e54, 0x7e55,
1607 0x7e56, 0x7e57, 0x7e58, 0x7e59, 0x7e5a, 0x7e5b, 0x7e5c, 0x7e5d,
1608 0x4fca, 0x7ae3, 0x6d5a, 0x90e1, 0x9a8f, 0x5580, 0x5496, 0x5361,
1609 0x54af, 0x5f00, 0x63e9, 0x6977, 0x51ef, 0x6168, 0x520a, 0x582a,
1610 0x52d8, 0x574e, 0x780d, 0x770b, 0x5eb7, 0x6177, 0x7ce0, 0x625b,
1611 0x6297, 0x4ea2, 0x7095, 0x8003, 0x62f7, 0x70e4, 0x9760, 0x5777,
1612 0x82db, 0x67ef, 0x68f5, 0x78d5, 0x9897, 0x79d1, 0x58f3, 0x54b3,
1613 0x53ef, 0x6e34, 0x514b, 0x523b, 0x5ba2, 0x8bfe, 0x80af, 0x5543,
1614 0x57a6, 0x6073, 0x5751, 0x542d, 0x7a7a, 0x6050, 0x5b54, 0x63a7,
1615 0x62a0, 0x53e3, 0x6263, 0x5bc7, 0x67af, 0x54ed, 0x7a9f, 0x82e6,
1616 0x9177, 0x5e93, 0x88e4, 0x5938, 0x57ae, 0x630e, 0x8de8, 0x80ef,
1617 0x5757, 0x7b77, 0x4fa9, 0x5feb, 0x5bbd, 0x6b3e, 0x5321, 0x7b50,
1618 0x72c2, 0x6846, 0x77ff, 0x7736, 0x65f7, 0x51b5, 0x4e8f, 0x76d4,
1619 0x5cbf, 0x7aa5, 0x8475, 0x594e, 0x9b41, 0x5080,
1620 /* 0xc0 */
1621 0x7e5e, 0x7e5f, 0x7e60, 0x7e61, 0x7e62, 0x7e63, 0x7e64, 0x7e65,
1622 0x7e66, 0x7e67, 0x7e68, 0x7e69, 0x7e6a, 0x7e6b, 0x7e6c, 0x7e6d,
1623 0x7e6e, 0x7e6f, 0x7e70, 0x7e71, 0x7e72, 0x7e73, 0x7e74, 0x7e75,
1624 0x7e76, 0x7e77, 0x7e78, 0x7e79, 0x7e7a, 0x7e7b, 0x7e7c, 0x7e7d,
1625 0x7e7e, 0x7e7f, 0x7e80, 0x7e81, 0x7e83, 0x7e84, 0x7e85, 0x7e86,
1626 0x7e87, 0x7e88, 0x7e89, 0x7e8a, 0x7e8b, 0x7e8c, 0x7e8d, 0x7e8e,
1627 0x7e8f, 0x7e90, 0x7e91, 0x7e92, 0x7e93, 0x7e94, 0x7e95, 0x7e96,
1628 0x7e97, 0x7e98, 0x7e99, 0x7e9a, 0x7e9c, 0x7e9d, 0x7e9e, 0x7e9f,
1629 0x7eb4, 0x7ebb, 0x7ebc, 0x7ed6, 0x7ee4, 0x7eec, 0x7ef9, 0x7f0a,
1630 0x7f10, 0x7f1e, 0x7f37, 0x7f39, 0x7f3b, 0x7f3c, 0x7f3d, 0x7f3e,
1631 0x7f3f, 0x7f40, 0x7f41, 0x7f43, 0x7f46, 0x7f47, 0x7f48, 0x7f49,
1632 0x7f4a, 0x7f4b, 0x7f4c, 0x7f4d, 0x7f4e, 0x7f4f, 0x7f52, 0x7f53,
1633 0x9988, 0x6127, 0x6e83, 0x5764, 0x6606, 0x6346, 0x56f0, 0x62ec,
1634 0x6269, 0x5ed3, 0x9614, 0x5783, 0x62c9, 0x5587, 0x8721, 0x814a,
1635 0x8fa3, 0x5566, 0x83b1, 0x6765, 0x8d56, 0x84dd, 0x5a6a, 0x680f,
1636 0x62e6, 0x7bee, 0x9611, 0x5170, 0x6f9c, 0x8c30, 0x63fd, 0x89c8,
1637 0x61d2, 0x7f06, 0x70c2, 0x6ee5, 0x7405, 0x6994, 0x72fc, 0x5eca,
1638 0x90ce, 0x6717, 0x6d6a, 0x635e, 0x52b3, 0x7262, 0x8001, 0x4f6c,
1639 0x59e5, 0x916a, 0x70d9, 0x6d9d, 0x52d2, 0x4e50, 0x96f7, 0x956d,
1640 0x857e, 0x78ca, 0x7d2f, 0x5121, 0x5792, 0x64c2, 0x808b, 0x7c7b,
1641 0x6cea, 0x68f1, 0x695e, 0x51b7, 0x5398, 0x68a8, 0x7281, 0x9ece,
1642 0x7bf1, 0x72f8, 0x79bb, 0x6f13, 0x7406, 0x674e, 0x91cc, 0x9ca4,
1643 0x793c, 0x8389, 0x8354, 0x540f, 0x6817, 0x4e3d, 0x5389, 0x52b1,
1644 0x783e, 0x5386, 0x5229, 0x5088, 0x4f8b, 0x4fd0,
1645 /* 0xc1 */
1646 0x7f56, 0x7f59, 0x7f5b, 0x7f5c, 0x7f5d, 0x7f5e, 0x7f60, 0x7f63,
1647 0x7f64, 0x7f65, 0x7f66, 0x7f67, 0x7f6b, 0x7f6c, 0x7f6d, 0x7f6f,
1648 0x7f70, 0x7f73, 0x7f75, 0x7f76, 0x7f77, 0x7f78, 0x7f7a, 0x7f7b,
1649 0x7f7c, 0x7f7d, 0x7f7f, 0x7f80, 0x7f82, 0x7f83, 0x7f84, 0x7f85,
1650 0x7f86, 0x7f87, 0x7f88, 0x7f89, 0x7f8b, 0x7f8d, 0x7f8f, 0x7f90,
1651 0x7f91, 0x7f92, 0x7f93, 0x7f95, 0x7f96, 0x7f97, 0x7f98, 0x7f99,
```

```
1652 0x7f9b, 0x7f9c, 0x7fa0, 0x7fa2, 0x7fa3, 0x7fa5, 0x7fa6, 0x7fa8,
1653 0x7fa9, 0x7faa, 0x7fab, 0x7fac, 0x7fad, 0x7fae, 0x7fb1, 0x7fb3,
1654 0x7fb4, 0x7fb5, 0x7fb6, 0x7fb7, 0x7fba, 0x7fbb, 0x7fbe, 0x7fc0,
1655 0x7fc2, 0x7fc3, 0x7fc4, 0x7fc6, 0x7fc7, 0x7fc8, 0x7fc9, 0x7fcb,
1656 0x7fcd, 0x7fcf, 0x7fd0, 0x7fd1, 0x7fd2, 0x7fd3, 0x7fd6, 0x7fd7,
1657 0x7fd9, 0x7fda, 0x7fdb, 0x7fdc, 0x7fdd, 0x7fde, 0x7fe2, 0x7fe3,
1658 0x75e2, 0x7acb, 0x7c92, 0x6ca5, 0x96b6, 0x529b, 0x7483, 0x54e9,
1659 0x4fe9, 0x8054, 0x83b2, 0x8fde, 0x9570, 0x5ec9, 0x601c, 0x6d9f,
1660 0x5e18, 0x655b, 0x8138, 0x94fe, 0x604b, 0x70bc, 0x7ec3, 0x7cae,
1661 0x51c9, 0x6881, 0x7cb1, 0x826f, 0x4e24, 0x8f86, 0x91cf, 0x667e,
1662 0x4eae, 0x8c05, 0x64a9, 0x804a, 0x50da, 0x7597, 0x71ce, 0x5be5,
1663 0x8fbd, 0x6f66, 0x4e86, 0x6482, 0x9563, 0x5ed6, 0x6599, 0x5217,
1664 0x88c2, 0x70c8, 0x52a3, 0x730e, 0x7433, 0x6797, 0x78f7, 0x9716,
1665 0x4e34, 0x90bb, 0x9cde, 0x6dcf, 0x51db, 0x8d41, 0x541d, 0x62ce,
1666 0x73b2, 0x83f1, 0x96f6, 0x9f84, 0x94c3, 0x4f36, 0x7f9a, 0x51cc,
1667 0x7075, 0x9675, 0x5cad, 0x9886, 0x53e6, 0x4ee4, 0x6e9c, 0x7409,
1668 0x69b4, 0x786b, 0x998f, 0x7559, 0x5218, 0x7624, 0x6d41, 0x67f3,
1669 0x516d, 0x9f99, 0x804b, 0x5499, 0x7b3c, 0x7abf,
1670 /* 0xc2 */
1671 0x7fe4, 0x7fe7, 0x7fe8, 0x7fea, 0x7feb, 0x7fec, 0x7fed, 0x7fef,
1672 0x7ff2, 0x7ff4, 0x7ff5, 0x7ff6, 0x7ff7, 0x7ff8, 0x7ff9, 0x7ffa,
1673 0x7ffd, 0x7ffe, 0x7fff, 0x8002, 0x8007, 0x8008, 0x8009, 0x800a,
1674 0x800e, 0x800f, 0x8011, 0x8013, 0x801a, 0x801b, 0x801d, 0x801e,
1675 0x801f, 0x8021, 0x8023, 0x8024, 0x802b, 0x802c, 0x802d, 0x802e,
1676 0x802f, 0x8030, 0x8032, 0x8034, 0x8039, 0x803a, 0x803c, 0x803e,
1677 0x8040, 0x8041, 0x8044, 0x8045, 0x8047, 0x8048, 0x8049, 0x804e,
1678 0x804f, 0x8050, 0x8051, 0x8052, 0x8053, 0x8054, 0x8055, 0x8056, 0x8057, 0x8059,
1679 0x805b, 0x805c, 0x805d, 0x805e, 0x805f, 0x8060, 0x8061, 0x8062,
1680 0x8063, 0x8064, 0x8065, 0x8066, 0x8067, 0x8068, 0x8069, 0x806a,
1681 0x806d, 0x806e, 0x806f, 0x8070, 0x8072, 0x8073, 0x8074, 0x8075,
1682 0x8076, 0x8077, 0x8078, 0x8079, 0x807a, 0x807b, 0x807c, 0x807d,
1683 0x9686, 0x5784, 0x62e2, 0x9647, 0x697c, 0x5a04, 0x6402, 0x7bd3,
1684 0x6f0f, 0x964b, 0x82a6, 0x5362, 0x9885, 0x5e90, 0x7089, 0x63b3,
1685 0x5364, 0x864f, 0x9c81, 0x9e93, 0x788c, 0x9732, 0x8def, 0x8d42,
1686 0x9e7f, 0x6f5e, 0x7984, 0x5f55, 0x9646, 0x622e, 0x9a74, 0x5415,
1687 0x94dd, 0x4fa3, 0x65c5, 0x5c65, 0x5c61, 0x7f15, 0x8651, 0x6c2f,
1688 0x5f8b, 0x7387, 0x6ee4, 0x7eff, 0x5ce6, 0x631b, 0x5b6a, 0x6ee6,
1689 0x5375, 0x4e71, 0x63a0, 0x7565, 0x62a1, 0x8f6e, 0x4f26, 0x4ed1,
1690 0x6ca6, 0x7eb6, 0x8bba, 0x841d, 0x87ba, 0x7f57, 0x903b, 0x9523,
1691 0x7ba9, 0x9aa1, 0x88f8, 0x843d, 0x6dlb, 0x9a86, 0x7edc, 0x5988,
1692 0x9ebb, 0x739b, 0x7801, 0x8682, 0x9a6c, 0x9a82, 0x561b, 0x5417,
1693 0x57cb, 0x4e70, 0x9ea6, 0x5356, 0x8fc8, 0x8109, 0x7792, 0x9992,
1694 0x86ee, 0x6ee1, 0x8513, 0x66fc, 0x6162, 0x6f2b,
1695 /* 0xc3 */
1696 0x807e, 0x8081, 0x8082, 0x8085, 0x8088, 0x808a, 0x808d, 0x808e,
1697 0x808f, 0x8090, 0x8091, 0x8092, 0x8094, 0x8095, 0x8097, 0x8099,
1698 0x809e, 0x80a3, 0x80a6, 0x80a7, 0x80a8, 0x80ac, 0x80b0, 0x80b3,
1699 0x80b5, 0x80b6, 0x80b8, 0x80b9, 0x80bb, 0x80c5, 0x80c7, 0x80c8,
1700 0x80c9, 0x80ca, 0x80cb, 0x80cf, 0x80d0, 0x80d1, 0x80d2, 0x80d3,
1701 0x80d4, 0x80d5, 0x80d8, 0x80df, 0x80e0, 0x80e2, 0x80e3, 0x80e6,
1702 0x80ee, 0x80f5, 0x80f7, 0x80f9, 0x80fb, 0x80fe, 0x80ff, 0x8100,
1703 0x8101, 0x8103, 0x8104, 0x8105, 0x8107, 0x8108, 0x810b, 0x810c,
1704 0x8115, 0x8117, 0x8119, 0x811b, 0x811c, 0x811d, 0x811f, 0x8120,
1705 0x8121, 0x8122, 0x8123, 0x8124, 0x8125, 0x8126, 0x8127, 0x8128,
1706 0x8129, 0x812a, 0x812b, 0x812d, 0x812e, 0x8130, 0x8133, 0x8134,
1707 0x8135, 0x8137, 0x8139, 0x813a, 0x813b, 0x813c, 0x813d, 0x813f,
1708 0x8c29, 0x8292, 0x832b, 0x76f2, 0x6c13, 0x5fd9, 0x83bd, 0x732b,
1709 0x8305, 0x951a, 0x6bdb, 0x77db, 0x94c6, 0x536f, 0x8302, 0x5192,
1710 0x5e3d, 0x8c8c, 0x8d38, 0x4e48, 0x73ab, 0x679a, 0x6885, 0x9176,
1711 0x9709, 0x7164, 0x6ca1, 0x7709, 0x5a92, 0x9541, 0x6bcf, 0x7f8e,
1712 0x6627, 0x5bd0, 0x59b9, 0x5a9a, 0x95e8, 0x95f7, 0x4eec, 0x840c,
1713 0x8499, 0x6aac, 0x76df, 0x9530, 0x731b, 0x68a6, 0x5b5f, 0x772f,
1714 0x919a, 0x9761, 0x7cdc, 0x8ff7, 0x8c1c, 0x5f25, 0x7c73, 0x79d8,
1715 0x89c5, 0x6ccc, 0x871c, 0x5bc6, 0x5e42, 0x68c9, 0x7720, 0x7ef5,
1716 0x5195, 0x514d, 0x52c9, 0x5a29, 0x7f05, 0x9762, 0x82d7, 0x63cf,
1717 0x7784, 0x85d0, 0x79d2, 0x6e3a, 0x5e99, 0x5999, 0x8511, 0x706d,
1718 0x6c11, 0x62bf, 0x76bf, 0x654f, 0x60af, 0x95fd, 0x660e, 0x879f,
1719 0x9e23, 0x94ed, 0x540d, 0x547d, 0x8c2c, 0x6478,
1720 /* 0xc4 */
1721 0x8140, 0x8141, 0x8142, 0x8143, 0x8144, 0x8145, 0x8147, 0x8149,
1722 0x814d, 0x814e, 0x814f, 0x8152, 0x8156, 0x8157, 0x8158, 0x815b,
1723 0x815c, 0x815d, 0x815e, 0x815f, 0x8161, 0x8162, 0x8163, 0x8164,
1724 0x8166, 0x8168, 0x816a, 0x816b, 0x816c, 0x816f, 0x8172, 0x8173,
1725 0x8175, 0x8176, 0x8177, 0x8178, 0x8181, 0x8183, 0x8184, 0x8185,
1726 0x8186, 0x8187, 0x8189, 0x818b, 0x818c, 0x818d, 0x818e, 0x8190,
1727 0x8192, 0x8193, 0x8194, 0x8195, 0x8196, 0x8197, 0x8199, 0x819a,
1728 0x819e, 0x819f, 0x81a0, 0x81a1, 0x81a2, 0x81a4, 0x81a5, 0x81a7,
1729 0x81a9, 0x81ab, 0x81ac, 0x81ad, 0x81ae, 0x81af, 0x81b0, 0x81b1,
1730 0x81b2, 0x81b4, 0x81b5, 0x81b6, 0x81b7, 0x81b8, 0x81b9, 0x81bc,
1731 0x81bd, 0x81be, 0x81bf, 0x81c4, 0x81c5, 0x81c7, 0x81c8, 0x81c9,
1732 0x81cb, 0x81cd, 0x81ce, 0x81cf, 0x81d0, 0x81d1, 0x81d2, 0x81d3,
1733 0x6479, 0x8611, 0x6a21, 0x819c, 0x78e8, 0x6469, 0x9b54, 0x62b9,
1734 0x672b, 0x83ab, 0x58a8, 0x9ed8, 0x6cab, 0x6f20, 0x5bde, 0x964c,
1735 0x8c0b, 0x725f, 0x67d0, 0x62c7, 0x7261, 0x4ea9, 0x59c6, 0x6bcd,
1736 0x5893, 0x66ae, 0x5e55, 0x52df, 0x6155, 0x6728, 0x76ee, 0x7766,
1737 0x7267, 0x7a46, 0x62ff, 0x54ea, 0x5450, 0x94a0, 0x90a3, 0x5a1c,
1738 0x7eb3, 0x6c16, 0x4e43, 0x5976, 0x8010, 0x5948, 0x5357, 0x7537,
```

```
1739 0x96be, 0x56ca, 0x6320, 0x8111, 0x607c, 0x95f9, 0x6dd6, 0x5462,
1740 0x9981, 0x5185, 0x5ae9, 0x80fd, 0x59ae, 0x9713, 0x502a, 0x6ce5,
1741 0x5c3c, 0x62df, 0x4f60, 0x533f, 0x817b, 0x9006, 0x6eba, 0x852b,
1742 0x62c8, 0x5e74, 0x78be, 0x64b5, 0x637b, 0x5ff5, 0x5a18, 0x917f,
1743 0x9e1f, 0x5c3f, 0x634f, 0x8042, 0x5b7d, 0x556e, 0x954a, 0x954d,
1744 0x6d85, 0x60a8, 0x67e0, 0x72de, 0x51dd, 0x5b81,
1745 /* 0xc5 */
1746 0x81d4, 0x81d5, 0x81d6, 0x81d7, 0x81d8, 0x81d9, 0x81da, 0x81db,
1747 0x81dc, 0x81dd, 0x81de, 0x81df, 0x81e0, 0x81e1, 0x81e2, 0x81e4,
1748 0x81e5, 0x81e6, 0x81e8, 0x81e9, 0x81eb, 0x81ee, 0x81ef, 0x81f0,
1749 0x81f1, 0x81f2, 0x81f5, 0x81f6, 0x81f7, 0x81f8, 0x81f9, 0x81fa,
1750 0x81fd, 0x81ff, 0x8203, 0x8207, 0x8208, 0x8209, 0x820a, 0x820b,
1751 0x820e, 0x820f, 0x8211, 0x8213, 0x8215, 0x8216, 0x8217, 0x8218,
1752 0x8219, 0x821a, 0x821d, 0x8220, 0x8224, 0x8225, 0x8226, 0x8227,
1753 0x8229, 0x822e, 0x8232, 0x823a, 0x823c, 0x823d, 0x823f, 0x8240,
1754 0x8241, 0x8242, 0x8243, 0x8245, 0x8246, 0x8248, 0x824a, 0x824c,
1755 0x824d, 0x824e, 0x8250, 0x8251, 0x8252, 0x8253, 0x8254, 0x8255,
1756 0x8256, 0x8257, 0x8259, 0x825b, 0x825c, 0x825d, 0x825e, 0x8260,
1757 0x8261, 0x8262, 0x8263, 0x8264, 0x8265, 0x8266, 0x8267, 0x8269,
1758 0x62e7, 0x6cde, 0x725b, 0x626d, 0x94ae, 0x7ebd, 0x8113, 0x6d53,
1759 0x519c, 0x5f04, 0x5974, 0x52aa, 0x6012, 0x5973, 0x6696, 0x8650,
1760 0x759f, 0x632a, 0x61e6, 0x7cef, 0x8bfa, 0x54e6, 0x6b27, 0x9e25,
1761 0x6bb4, 0x85d5, 0x5455, 0x5076, 0x6ca4, 0x556a, 0x8db4, 0x722c,
1762 0x5e15, 0x6015, 0x7436, 0x62cd, 0x6392, 0x724c, 0x5f98, 0x6e43,
1763 0x6d3e, 0x6500, 0x6f58, 0x76d8, 0x78d0, 0x76fc, 0x7554, 0x5224,
1764 0x53db, 0x4e53, 0x5e9e, 0x65c1, 0x802a, 0x80d6, 0x629b, 0x5486,
1765 0x5228, 0x70ae, 0x888d, 0x8dd1, 0x6ce1, 0x5478, 0x80da, 0x57f9,
1766 0x88f4, 0x8d54, 0x966a, 0x914d, 0x4f69, 0x6c9b, 0x55b7, 0x76c6,
1767 0x7830, 0x62a8, 0x70f9, 0x6f8e, 0x5f6d, 0x84ec, 0x68da, 0x787c,
1768 0x7bf7, 0x81a8, 0x670b, 0x9e4f, 0x6367, 0x78b0, 0x576f, 0x7812,
1769 0x9739, 0x6279, 0x62ab, 0x5288, 0x7435, 0x6bd7,
1770 /* 0xc6 */
1771 0x826a, 0x826b, 0x826c, 0x826d, 0x8271, 0x8275, 0x8276, 0x8277,
1772 0x8278, 0x827b, 0x827c, 0x8280, 0x8281, 0x8283, 0x8285, 0x8286,
1773 0x8287, 0x8289, 0x828c, 0x8290, 0x8293, 0x8294, 0x8295, 0x8296,
1774 0x829a, 0x829b, 0x829e, 0x82a0, 0x82a2, 0x82a3, 0x82a7, 0x82b2,
1775 0x82b5, 0x82b6, 0x82ba, 0x82bb, 0x82bc, 0x82bf, 0x82c0, 0x82c2,
1776 0x82c3, 0x82c5, 0x82c6, 0x82c9, 0x82d0, 0x82d6, 0x82d9, 0x82da,
1777 0x82dd, 0x82e2, 0x82e7, 0x82e8, 0x82e9, 0x82ea, 0x82ec, 0x82ed,
1778 0x82ee, 0x82f0, 0x82f2, 0x82f3, 0x82f5, 0x82f6, 0x82f8, 0x82fa,
1779 0x82fc, 0x82fd, 0x82fe, 0x82ff, 0x8300, 0x830a, 0x830b, 0x830d,
1780 0x8310, 0x8312, 0x8313, 0x8316, 0x8318, 0x8319, 0x831d, 0x831e,
1781 0x831f, 0x8320, 0x8321, 0x8322, 0x8323, 0x8324, 0x8325, 0x8326,
1782 0x8329, 0x832a, 0x832e, 0x8330, 0x8332, 0x8337, 0x833b, 0x833d,
1783 0x5564, 0x813e, 0x75b2, 0x76ae, 0x5339, 0x75de, 0x50fb, 0x5c41,
1784 0x8b6c, 0x7bc7, 0x504f, 0x7247, 0x9a97, 0x98d8, 0x6f02, 0x74e2,
1785 0x7968, 0x6487, 0x77a5, 0x62fc, 0x9891, 0x8d2b, 0x54c1, 0x8058,
1786 0x4e52, 0x576a, 0x82f9, 0x840d, 0x5e73, 0x51ed, 0x74f6, 0x8bc4,
1787 0x5c4f, 0x5761, 0x6cfc, 0x9887, 0x5a46, 0x7834, 0x9b44, 0x8feb,
1788 0x7c95, 0x5256, 0x6251, 0x94fa, 0x4ec6, 0x8386, 0x8461, 0x83e9,
1789 0x84b2, 0x57d4, 0x6734, 0x5703, 0x666e, 0x6d66, 0x8c31, 0x66dd,
1790 0x7011, 0x671f, 0x6b3a, 0x6816, 0x621a, 0x59bb, 0x4e03, 0x51c4,
1791 0x6f06, 0x67d2, 0x6c8f, 0x5176, 0x68cb, 0x5947, 0x6b67, 0x7566,
1792 0x5d0e, 0x8110, 0x9f50, 0x65d7, 0x7948, 0x7941, 0x9a91, 0x8d77,
1793 0x5c82, 0x4e5e, 0x4f01, 0x542f, 0x5951, 0x780c, 0x5668, 0x6c14,
1794 0x8fc4, 0x5f03, 0x6c7d, 0x6ce3, 0x8bab, 0x6390,
1795 /* 0xc7 */
1796 0x833e, 0x833f, 0x8341, 0x8342, 0x8344, 0x8345, 0x8348, 0x834a,
1797 0x834b, 0x834c, 0x834d, 0x834e, 0x8353, 0x8355, 0x8356, 0x8357,
1798 0x8358, 0x8359, 0x835d, 0x8362, 0x8370, 0x8371, 0x8372, 0x8373,
1799 0x8374, 0x8375, 0x8376, 0x8379, 0x837a, 0x837e, 0x837f, 0x8380,
1800 0x8381, 0x8382, 0x8383, 0x8384, 0x8387, 0x8388, 0x838a, 0x838b,
1801 0x838c, 0x838d, 0x838f, 0x8390, 0x8391, 0x8394, 0x8395, 0x8396,
1802 0x8397, 0x8399, 0x839a, 0x839d, 0x839f, 0x83a1, 0x83a2, 0x83a3,
1803 0x83a4, 0x83a5, 0x83a6, 0x83a7, 0x83ac, 0x83ad, 0x83ae, 0x83af,
1804 0x83b5, 0x83bb, 0x83be, 0x83bf, 0x83c2, 0x83c3, 0x83c4, 0x83c6,
1805 0x83c8, 0x83c9, 0x83cb, 0x83cd, 0x83ce, 0x83d0, 0x83d1, 0x83d2,
1806 0x83d3, 0x83d5, 0x83d7, 0x83d9, 0x83da, 0x83db, 0x83de, 0x83e2,
1807 0x83e3, 0x83e4, 0x83e6, 0x83e7, 0x83e8, 0x83eb, 0x83ec, 0x83ed,
1808 0x6070, 0x6d3d, 0x7275, 0x6266, 0x948e, 0x94c5, 0x5343, 0x8fc1,
1809 0x7b7e, 0x4edf, 0x8c26, 0x4e7e, 0x9ed4, 0x94b1, 0x94b3, 0x524d,
1810 0x6f5c, 0x9063, 0x6d45, 0x8c34, 0x5811, 0x5d4c, 0x6b20, 0x6b49,
1811 0x67aa, 0x545b, 0x8154, 0x7f8c, 0x5899, 0x8537, 0x5f3a, 0x62a2,
1812 0x6a47, 0x9539, 0x6572, 0x6084, 0x6865, 0x77a7, 0x4e54, 0x4fa8,
1813 0x5de7, 0x9798, 0x64ac, 0x7fd8, 0x5ced, 0x4fcf, 0x7a8d, 0x5207,
1814 0x8304, 0x4e14, 0x602f, 0x7a83, 0x94a6, 0x4fb5, 0x4eb2, 0x79e6,
1815 0x7434, 0x52e4, 0x82b9, 0x64d2, 0x79bd, 0x5bdd, 0x6c81, 0x9752,
1816 0x8f7b, 0x6c22, 0x503e, 0x537f, 0x6e05, 0x64ce, 0x6674, 0x6c30,
1817 0x60c5, 0x9877, 0x8bf7, 0x5e86, 0x743c, 0x7a77, 0x79cb, 0x4e18,
1818 0x90b1, 0x7403, 0x6c42, 0x56da, 0x914b, 0x6cc5, 0x8d8b, 0x533a,
1819 0x86c6, 0x66f2, 0x8eaf, 0x5c48, 0x9a71, 0x6e20,
1820 /* 0xc8 */
1821 0x83ee, 0x83ef, 0x83f3, 0x83f4, 0x83f5, 0x83f6, 0x83f7, 0x83fa,
1822 0x83fb, 0x83fc, 0x83fe, 0x83ff, 0x8400, 0x8402, 0x8405, 0x8407,
1823 0x8408, 0x8409, 0x840a, 0x8410, 0x8412, 0x8413, 0x8414, 0x8415,
1824 0x8416, 0x8417, 0x8419, 0x841a, 0x841b, 0x841e, 0x841f, 0x8420,
1825 0x8421, 0x8422, 0x8423, 0x8429, 0x842a, 0x842b, 0x842c, 0x842d,
```

```
1826 0x842e, 0x842f, 0x8430, 0x8432, 0x8433, 0x8434, 0x8435, 0x8436,
1827 0x8437, 0x8439, 0x843a, 0x843b, 0x843e, 0x843f, 0x8440, 0x8441,
1828 0x8442, 0x8443, 0x8444, 0x8445, 0x8447, 0x8448, 0x8449, 0x844a,
1829 0x844b, 0x844c, 0x844d, 0x844e, 0x844f, 0x8450, 0x8452, 0x8453,
1830 0x8454, 0x8455, 0x8456, 0x8458, 0x845d, 0x845e, 0x845f, 0x8460,
1831 0x8462, 0x8464, 0x8465, 0x8466, 0x8467, 0x8468, 0x846a, 0x846e,
1832 0x846f, 0x8470, 0x8472, 0x8474, 0x8477, 0x8479, 0x847b, 0x847c,
1833 0x53d6, 0x5a36, 0x9f8b, 0x8da3, 0x53bb, 0x5708, 0x98a7, 0x6743,
1834 0x919b, 0x6cc9, 0x5168, 0x75ca, 0x62f3, 0x72ac, 0x5238, 0x529d,
1835 0x7f3a, 0x7094, 0x7638, 0x5374, 0x9e4a, 0x69b7, 0x786e, 0x96c0,
1836 0x88d9, 0x7fa4, 0x7136, 0x71c3, 0x5189, 0x67d3, 0x74e4, 0x58e4,
1837 0x6518, 0x56b7, 0x56b8, 0x8ba9, 0x9976, 0x6270, 0x7ed5, 0x60f9, 0x70ed,
1838 0x58ec, 0x4ec1, 0x4eba, 0x5fcd, 0x97e7, 0x4efb, 0x8ba4, 0x5203,
1839 0x598a, 0x7eab, 0x6254, 0x4ecd, 0x65e5, 0x620e, 0x8338, 0x84c9,
1840 0x8363, 0x878d, 0x7194, 0x6eb6, 0x5bb9, 0x7ed2, 0x5197, 0x63c9,
1841 0x67d4, 0x8089, 0x8339, 0x8815, 0x5112, 0x5b7a, 0x5982, 0x8fb1,
1842 0x4e73, 0x6c5d, 0x5165, 0x8925, 0x8f6f, 0x962e, 0x854a, 0x745e,
1843 0x9510, 0x95f0, 0x6da6, 0x82e5, 0x5f31, 0x6492, 0x6d12, 0x8428,
1844 0x816e, 0x9cc3, 0x585e, 0x8d5b, 0x4e09, 0x53c1,
1845 /* 0xc9 */
1846 0x847d, 0x847e, 0x847f, 0x8480, 0x8481, 0x8483, 0x8484, 0x8485,
1847 0x8486, 0x848a, 0x848d, 0x848f, 0x8490, 0x8491, 0x8492, 0x8493,
1848 0x8494, 0x8495, 0x8496, 0x8498, 0x849a, 0x849b, 0x849d, 0x849e,
1849 0x849f, 0x84a0, 0x84a2, 0x84a3, 0x84a4, 0x84a5, 0x84a6, 0x84a7,
1850 0x84a8, 0x84a9, 0x84aa, 0x84ab, 0x84ac, 0x84ad, 0x84ae, 0x84b0,
1851 0x84b1, 0x84b3, 0x84b5, 0x84b6, 0x84b7, 0x84bb, 0x84bc, 0x84be,
1852 0x84c0, 0x84c2, 0x84c3, 0x84c5, 0x84c6, 0x84c7, 0x84c8, 0x84cb,
1853 0x84cc, 0x84ce, 0x84cf, 0x84d2, 0x84d4, 0x84d5, 0x84d7, 0x84d8,
1854 0x84d9, 0x84da, 0x84db, 0x84dc, 0x84de, 0x84e1, 0x84e2, 0x84e4,
1855 0x84e7, 0x84e8, 0x84e9, 0x84ea, 0x84eb, 0x84ed, 0x84ee, 0x84ef,
1856 0x84f1, 0x84f2, 0x84f3, 0x84f4, 0x84f5, 0x84f6, 0x84f7, 0x84f8,
1857 0x84f9, 0x84fa, 0x84fb, 0x84fd, 0x84fe, 0x8500, 0x8501, 0x8502,
1858 0x4f1e, 0x6563, 0x6851, 0x55d3, 0x4e27, 0x6414, 0x9a9a, 0x626b,
1859 0x5ac2, 0x745f, 0x8272, 0x6da9, 0x68ee, 0x50e7, 0x838e, 0x7802,
1860 0x6740, 0x5239, 0x6c99, 0x7eb1, 0x50bb, 0x5565, 0x715e, 0x7b5b,
1861 0x6652, 0x73ca, 0x82eb, 0x6749, 0x5c71, 0x5220, 0x717d, 0x886b,
1862 0x95ea, 0x9655, 0x64c5, 0x8d61, 0x81b3, 0x5584, 0x6c55, 0x6247,
1863 0x7f2e, 0x5892, 0x4f24, 0x5546, 0x8d4f, 0x664c, 0x4e0a, 0x5c1a,
1864 0x88f3, 0x68a2, 0x634e, 0x7a0d, 0x70e7, 0x828d, 0x52fa, 0x97f6,
1865 0x5c11, 0x54e8, 0x90b5, 0x7ecd, 0x5962, 0x8d4a, 0x86c7, 0x820c,
1866 0x820d, 0x8d66, 0x6444, 0x5c04, 0x6151, 0x6d89, 0x793c, 0x8bbe,
1867 0x7837, 0x7533, 0x547b, 0x4f38, 0x8eab, 0x6df1, 0x5a20, 0x7ec5,
1868 0x795e, 0x6c88, 0x5ba1, 0x5a76, 0x751a, 0x80be, 0x614e, 0x6e17,
1869 0x58f0, 0x751f, 0x7525, 0x7272, 0x5347, 0x7ef3,
1870 /* 0xca */
1871 0x8503, 0x8504, 0x8505, 0x8506, 0x8507, 0x8508, 0x8509, 0x850a,
1872 0x850b, 0x850d, 0x850e, 0x850f, 0x8510, 0x8512, 0x8514, 0x8515,
1873 0x8516, 0x8518, 0x8519, 0x851b, 0x851c, 0x851d, 0x851e, 0x8520,
1874 0x8522, 0x8523, 0x8524, 0x8525, 0x8526, 0x8527, 0x8528, 0x8529,
1875 0x852a, 0x852d, 0x852e, 0x852f, 0x8530, 0x8531, 0x8532, 0x8533,
1876 0x8534, 0x8535, 0x8536, 0x853e, 0x853f, 0x8540, 0x8541, 0x8542,
1877 0x8544, 0x8545, 0x8546, 0x8547, 0x854b, 0x854c, 0x854d, 0x854e,
1878 0x854f, 0x8550, 0x8551, 0x8552, 0x8553, 0x8554, 0x8555, 0x8557,
1879 0x8558, 0x855a, 0x855b, 0x855c, 0x855d, 0x855f, 0x8560, 0x8561,
1880 0x8562, 0x8563, 0x8565, 0x8566, 0x8567, 0x8569, 0x856a, 0x856b,
1881 0x856c, 0x856d, 0x856e, 0x856f, 0x8570, 0x8571, 0x8573, 0x8575,
1882 0x8576, 0x8577, 0x8578, 0x857c, 0x857d, 0x857f, 0x8580, 0x8581,
1883 0x7701, 0x76db, 0x5269, 0x80dc, 0x5723, 0x5e08, 0x5931, 0x72ee,
1884 0x65bd, 0x6e7f, 0x8bd7, 0x5c38, 0x8671, 0x5341, 0x77f3, 0x62fe,
1885 0x65f6, 0x4ec0, 0x98df, 0x8680, 0x5b9e, 0x8bc6, 0x53f2, 0x77e2,
1886 0x4f7f, 0x5c4e, 0x9a76, 0x59cb, 0x5f0f, 0x793a, 0x58eb, 0x4e16,
1887 0x67ff, 0x4e8b, 0x62ed, 0x8a93, 0x901d, 0x52bf, 0x662f, 0x55dc,
1888 0x566c, 0x9002, 0x4ed5, 0x4f8d, 0x91ca, 0x9970, 0x6c0f, 0x5e02,
1889 0x6043, 0x5ba4, 0x89c6, 0x8bd5, 0x6536, 0x624b, 0x9996, 0x5b88,
1890 0x5bff, 0x6388, 0x552e, 0x53d7, 0x7626, 0x517d, 0x852c, 0x67a2,
1891 0x68b3, 0x6b8a, 0x6292, 0x8f93, 0x53d4, 0x8212, 0x6dd1, 0x758f,
1892 0x4e66, 0x8d4e, 0x5b70, 0x719f, 0x85af, 0x6691, 0x66d9, 0x7f72,
1893 0x8700, 0x9ecd, 0x9f20, 0x5c5e, 0x672f, 0x8ff0, 0x6811, 0x675f,
1894 0x620d, 0x7ad6, 0x5885, 0x5eb6, 0x6570, 0x6f31,
1895 /* 0xcb */
1896 0x8582, 0x8583, 0x8586, 0x8588, 0x8589, 0x858a, 0x858b, 0x858c,
1897 0x858d, 0x858e, 0x8590, 0x8591, 0x8592, 0x8593, 0x8594, 0x8595,
1898 0x8596, 0x8597, 0x8598, 0x8599, 0x859a, 0x859d, 0x859e, 0x859f,
1899 0x85a0, 0x85a1, 0x85a2, 0x85a3, 0x85a5, 0x85a6, 0x85a7, 0x85a9,
1900 0x85ab, 0x85ac, 0x85ad, 0x85b1, 0x85b2, 0x85b3, 0x85b4, 0x85b5,
1901 0x85b6, 0x85b8, 0x85ba, 0x85bb, 0x85bc, 0x85bd, 0x85be, 0x85bf,
1902 0x85c0, 0x85c2, 0x85c3, 0x85c4, 0x85c5, 0x85c6, 0x85c7, 0x85c8,
1903 0x85ca, 0x85cb, 0x85cc, 0x85cd, 0x85ce, 0x85d1, 0x85d2, 0x85d4,
1904 0x85d6, 0x85d7, 0x85d8, 0x85d9, 0x85da, 0x85db, 0x85dd, 0x85de,
1905 0x85df, 0x85e0, 0x85e1, 0x85e2, 0x85e3, 0x85e5, 0x85e6, 0x85e7,
1906 0x85e8, 0x85ea, 0x85eb, 0x85ec, 0x85ed, 0x85ee, 0x85ef, 0x85f0,
1907 0x85f1, 0x85f2, 0x85f3, 0x85f4, 0x85f5, 0x85f6, 0x85f7, 0x85f8,
1908 0x6055, 0x5237, 0x800d, 0x6454, 0x8870, 0x7529, 0x5e05, 0x6813,
1909 0x62f4, 0x971c, 0x53cc, 0x723d, 0x8c01, 0x6c34, 0x7761, 0x7a0e,
1910 0x542e, 0x77ac, 0x987a, 0x821c, 0x8bf4, 0x7855, 0x6714, 0x70c1,
1911 0x65af, 0x6495, 0x5636, 0x601d, 0x79c1, 0x53f8, 0x4e1d, 0x6b7b,
1912 0x8086, 0x5bfa, 0x55e3, 0x56db, 0x4f3a, 0x4f3c, 0x9972, 0x5df3,
```

```
1913 0x677e, 0x8038, 0x6002, 0x9882, 0x9001, 0x5b8b, 0x8bbc, 0x8bf5,
1914 0x641c, 0x8258, 0x64de, 0x55fd, 0x82cf, 0x9165, 0x4fd7, 0x7d20,
1915 0x901f, 0x7c9f, 0x50f3, 0x5851, 0x6eaf, 0x5bbf, 0x8bc9, 0x8083,
1916 0x9178, 0x849c, 0x7b97, 0x867d, 0x968b, 0x968f, 0x7ee5, 0x9ad3,
1917 0x788e, 0x5c81, 0x7a57, 0x9042, 0x96a7, 0x795f, 0x5b59, 0x635f,
1918 0x7b0b, 0x84d1, 0x68ad, 0x5506, 0x7f29, 0x7410, 0x7d22, 0x9501,
1919 0x6240, 0x584c, 0x4ed6, 0x5b83, 0x5979, 0x5854,
1920 /* 0xcc */
1921 0x85f9, 0x85fa, 0x85fc, 0x85fd, 0x85fe, 0x8600, 0x8601, 0x8602,
1922 0x8603, 0x8604, 0x8606, 0x8607, 0x8608, 0x8609, 0x860a, 0x860b,
1923 0x860c, 0x860d, 0x860e, 0x860f, 0x8610, 0x8612, 0x8613, 0x8614,
1924 0x8615, 0x8617, 0x8618, 0x8619, 0x861a, 0x861b, 0x861c, 0x861d,
1925 0x861e, 0x861f, 0x8620, 0x8621, 0x8622, 0x8623, 0x8624, 0x8625,
1926 0x8626, 0x8628, 0x862a, 0x862b, 0x862c, 0x862d, 0x862e, 0x862f,
1927 0x8630, 0x8631, 0x8632, 0x8633, 0x8634, 0x8635, 0x8636, 0x8637,
1928 0x8639, 0x863a, 0x863b, 0x863d, 0x863e, 0x863f, 0x8640, 0x8641,
1929 0x8642, 0x8643, 0x8644, 0x8645, 0x8646, 0x8647, 0x8648, 0x8649,
1930 0x864a, 0x864b, 0x864c, 0x864d, 0x8652, 0x8653, 0x8655, 0x8656,
1931 0x8658, 0x8659, 0x865b, 0x865c, 0x865d, 0x865f, 0x8660, 0x8661,
1932 0x8663, 0x8664, 0x8665, 0x8666, 0x8667, 0x8668, 0x8669, 0x866a,
1933 0x736d, 0x8617, 0x8618, 0x8e4b, 0x8e0f, 0x80ce, 0x82d4, 0x62ac, 0x53f0,
1934 0x6cf0, 0x915e, 0x592a, 0x6001, 0x6c70, 0x574d, 0x644a, 0x8d2a,
1935 0x762b, 0x6ee9, 0x575b, 0x6a80, 0x75f0, 0x6f6d, 0x8c2d, 0x8c08,
1936 0x5766, 0x6bef, 0x8892, 0x78b3, 0x63a2, 0x53f9, 0x70ad, 0x6c64,
1937 0x5858, 0x642a, 0x5802, 0x68e0, 0x819b, 0x5510, 0x7cd6, 0x5018,
1938 0x8eba, 0x6dcc, 0x8d9f, 0x70eb, 0x638f, 0x6d9b, 0x6ed4, 0x7ee6,
1939 0x8404, 0x6843, 0x9003, 0x6dd8, 0x9676, 0x8ba8, 0x5957, 0x7279,
1940 0x85e4, 0x817e, 0x75bc, 0x8a8a, 0x68af, 0x5254, 0x8e22, 0x9511,
1941 0x63d0, 0x8989, 0x8e44, 0x557c, 0x4f53, 0x66ff, 0x568f, 0x60d5,
1942 0x6d95, 0x5243, 0x5c49, 0x5929, 0x6dfb, 0x586b, 0x7530, 0x751c,
1943 0x606c, 0x8214, 0x8146, 0x6311, 0x6761, 0x8fe2, 0x773a, 0x8df3,
1944 0x8d34, 0x94c1, 0x5e16, 0x5385, 0x542c, 0x70c3,
1945 /* 0xcd */
1946 0x866d, 0x866f, 0x8670, 0x8672, 0x8673, 0x8674, 0x8675, 0x8676,
1947 0x8677, 0x8678, 0x8683, 0x8684, 0x8685, 0x8686, 0x8687, 0x8688,
1948 0x8689, 0x868e, 0x868f, 0x8690, 0x8691, 0x8692, 0x8694, 0x8696,
1949 0x8697, 0x8698, 0x8699, 0x869a, 0x869b, 0x869e, 0x869f, 0x86a0,
1950 0x86a1, 0x86a2, 0x86a5, 0x86a6, 0x86ab, 0x86ad, 0x86ae, 0x86b2,
1951 0x86b3, 0x86b7, 0x86b8, 0x86b9, 0x86bb, 0x86bc, 0x86bd, 0x86be,
1952 0x86bf, 0x86c1, 0x86c2, 0x86c3, 0x86c5, 0x86c8, 0x86cc, 0x86cd,
1953 0x86d2, 0x86d3, 0x86d5, 0x86d6, 0x86d7, 0x86da, 0x86dc, 0x86dd,
1954 0x86e0, 0x86e1, 0x86e2, 0x86e3, 0x86e5, 0x86e6, 0x86e7, 0x86e8,
1955 0x86ea, 0x86eb, 0x86ec, 0x86ef, 0x86f5, 0x86f6, 0x86f7, 0x86fa,
1956 0x86fb, 0x86fc, 0x86fd, 0x86ff, 0x8701, 0x8704, 0x8705, 0x8706,
1957 0x870b, 0x870c, 0x870e, 0x870f, 0x8710, 0x8711, 0x8714, 0x8716,
1958 0x6c40, 0x5ef7, 0x505c, 0x4ead, 0x5ead, 0x633a, 0x8247, 0x901a,
1959 0x6850, 0x916e, 0x77b3, 0x540c, 0x94dc, 0x5f64, 0x7ae5, 0x6876,
1960 0x6345, 0x7b52, 0x7edf, 0x75db, 0x5077, 0x6295, 0x5934, 0x900f,
1961 0x51f8, 0x79c3, 0x7a81, 0x56fe, 0x5f92, 0x9014, 0x6d82, 0x5c60,
1962 0x571f, 0x5410, 0x5154, 0x6e4d, 0x56e2, 0x63a8, 0x9893, 0x817f,
1963 0x8715, 0x892a, 0x9000, 0x541e, 0x5c6f, 0x81c0, 0x62d6, 0x6258,
1964 0x8131, 0x9e35, 0x9640, 0x9a6e, 0x9a7c, 0x692d, 0x59a5, 0x62d3,
1965 0x553e, 0x6316, 0x54c7, 0x86d9, 0x6d3c, 0x5a03, 0x74e6, 0x889c,
1966 0x6b6a, 0x5916, 0x8c4c, 0x5f2f, 0x6e7e, 0x73a9, 0x987d, 0x4e38,
1967 0x70f7, 0x5b8c, 0x7897, 0x633d, 0x665a, 0x7696, 0x60cb, 0x5b9b,
1968 0x5a49, 0x4e07, 0x8155, 0x6c6a, 0x738b, 0x4ea1, 0x6789, 0x7f51,
1969 0x5f80, 0x65fa, 0x671b, 0x5fd8, 0x5984, 0x5a01,
1970 /* 0xce */
1971 0x8719, 0x871b, 0x871d, 0x871f, 0x8720, 0x8724, 0x8726, 0x8727,
1972 0x8728, 0x872a, 0x872b, 0x872c, 0x872d, 0x872f, 0x8730, 0x8732,
1973 0x8733, 0x8735, 0x8736, 0x8738, 0x8739, 0x873a, 0x873c, 0x873d,
1974 0x8740, 0x8741, 0x8742, 0x8743, 0x8744, 0x8745, 0x8746, 0x874a,
1975 0x874b, 0x874d, 0x874e, 0x8750, 0x8751, 0x8752, 0x8754, 0x8755,
1976 0x8756, 0x8758, 0x875a, 0x875b, 0x875c, 0x875d, 0x875e, 0x875f,
1977 0x8761, 0x8762, 0x8766, 0x8767, 0x8768, 0x8769, 0x876a, 0x876b,
1978 0x876c, 0x876d, 0x876f, 0x8771, 0x8772, 0x8773, 0x8775, 0x8777,
1979 0x8778, 0x8779, 0x877a, 0x877f, 0x8780, 0x8781, 0x8784, 0x8786,
1980 0x8787, 0x8789, 0x878a, 0x878c, 0x878e, 0x878f, 0x8790, 0x8791,
1981 0x8792, 0x8794, 0x8795, 0x8796, 0x8798, 0x8799, 0x879a, 0x879b,
1982 0x879c, 0x879d, 0x879e, 0x87a0, 0x87a1, 0x87a2, 0x87a3, 0x87a4,
1983 0x5dcd, 0x5fae, 0x5371, 0x97e6, 0x8fdd, 0x6845, 0x56f4, 0x552f,
1984 0x60df, 0x4e3a, 0x6f4d, 0x7ef4, 0x82c7, 0x840e, 0x59d4, 0x4f1f,
1985 0x4f2a, 0x5c3e, 0x7eac, 0x672a, 0x851a, 0x5473, 0x754f, 0x80c3,
1986 0x5582, 0x9b4f, 0x4f4d, 0x6e2d, 0x8c13, 0x5c09, 0x6170, 0x536b,
1987 0x761f, 0x6e29, 0x868a, 0x6587, 0x95fb, 0x7eb9, 0x543b, 0x7a33,
1988 0x7d0a, 0x95ee, 0x55e1, 0x7fc1, 0x74ee, 0x631d, 0x8717, 0x6da1,
1989 0x7a9d, 0x6211, 0x65a1, 0x5367, 0x63e1, 0x6c83, 0x5deb, 0x545c,
1990 0x94a8, 0x4e4c, 0x6c61, 0x8bec, 0x5c4b, 0x65e0, 0x829c, 0x68a7,
1991 0x543e, 0x5434, 0x6bcb, 0x6b66, 0x4e94, 0x6342, 0x5348, 0x821e,
1992 0x4f0d, 0x4fae, 0x575e, 0x620a, 0x96fe, 0x6664, 0x7269, 0x52ff,
1993 0x52a1, 0x609f, 0x8bef, 0x6614, 0x7199, 0x6790, 0x897f, 0x7852,
1994 0x77fd, 0x6670, 0x563b, 0x5438, 0x9521, 0x727a,
1995 /* 0xcf */
1996 0x87a5, 0x87a6, 0x87a7, 0x87a9, 0x87aa, 0x87ae, 0x87b0, 0x87b1,
1997 0x87b2, 0x87b4, 0x87b6, 0x87b7, 0x87b8, 0x87b9, 0x87bb, 0x87bc,
1998 0x87be, 0x87bf, 0x87c1, 0x87c2, 0x87c3, 0x87c4, 0x87c5, 0x87c7,
1999 0x87c8, 0x87c9, 0x87cc, 0x87cd, 0x87ce, 0x87cf, 0x87d0, 0x87d4,
```



```
2000 0x87d5, 0x87d6, 0x87d7, 0x87d8, 0x87d9, 0x87da, 0x87dc, 0x87dd,
2001 0x87de, 0x87df, 0x87e1, 0x87e2, 0x87e3, 0x87e4, 0x87e6, 0x87e7,
2002 0x87e8, 0x87e9, 0x87eb, 0x87ec, 0x87ed, 0x87ef, 0x87f0, 0x87f1,
2003 0x87f2, 0x87f3, 0x87f4, 0x87f5, 0x87f6, 0x87f7, 0x87f8, 0x87fa,
2004 0x87fb, 0x87fc, 0x87fd, 0x87ff, 0x8800, 0x8801, 0x8802, 0x8804,
2005 0x8805, 0x8806, 0x8807, 0x8808, 0x8809, 0x880b, 0x880c, 0x880d,
2006 0x880e, 0x880f, 0x8810, 0x8811, 0x8812, 0x8814, 0x8817, 0x8818,
2007 0x8819, 0x881a, 0x881c, 0x881d, 0x881e, 0x881f, 0x8820, 0x8823,
2008 0x7a00, 0x606f, 0x5e0c, 0x6089, 0x819d, 0x5915, 0x60dc, 0x7184,
2009 0x70ef, 0x6eaa, 0x6c50, 0x7280, 0x6a84, 0x88ad, 0x5e2d, 0x4e60,
2010 0x5ab3, 0x559c, 0x94e3, 0x6d17, 0x7cfc, 0x9699, 0x620f, 0x7ec6,
2011 0x778e, 0x867e, 0x5323, 0x971e, 0x8f96, 0x6687, 0x5ce1, 0x4fa0,
2012 0x72ed, 0x4e0b, 0x53a6, 0x590f, 0x5413, 0x6380, 0x9528, 0x5148,
2013 0x4ed9, 0x9c9c, 0x7ea4, 0x54b8, 0x8d24, 0x8854, 0x8237, 0x95f2,
2014 0x6d8e, 0x5f26, 0x5acc, 0x663e, 0x9669, 0x73b0, 0x732e, 0x53bf,
2015 0x817a, 0x9985, 0x7fa1, 0x5baa, 0x9677, 0x9650, 0x7ebf, 0x76f8,
2016 0x53a2, 0x9576, 0x9999, 0x7bb1, 0x8944, 0x6e58, 0x4e61, 0x7fd4,
2017 0x7965, 0x8be6, 0x60f3, 0x54cd, 0x4eab, 0x9879, 0x5df7, 0x6a61,
2018 0x50cf, 0x5411, 0x8c61, 0x8427, 0x785d, 0x9704, 0x524a, 0x54ee,
2019 0x56a3, 0x9500, 0x6d88, 0x5bb5, 0x6dc6, 0x6653,
2020 /* 0xd0 */
2021 0x8824, 0x8825, 0x8826, 0x8827, 0x8828, 0x8829, 0x882a, 0x882b,
2022 0x882c, 0x882d, 0x882e, 0x882f, 0x8830, 0x8831, 0x8833, 0x8834,
2023 0x8835, 0x8836, 0x8837, 0x8838, 0x883a, 0x883b, 0x883d, 0x883e,
2024 0x883f, 0x8841, 0x8842, 0x8843, 0x8846, 0x8847, 0x8848, 0x8849,
2025 0x884a, 0x884b, 0x884e, 0x884f, 0x8850, 0x8851, 0x8852, 0x8853,
2026 0x8855, 0x8856, 0x8858, 0x885a, 0x885b, 0x885c, 0x885d, 0x885e,
2027 0x885f, 0x8860, 0x8866, 0x8867, 0x886a, 0x886d, 0x886f, 0x8871,
2028 0x8873, 0x8874, 0x8875, 0x8876, 0x8878, 0x8879, 0x887a, 0x887b,
2029 0x887c, 0x8880, 0x8883, 0x8886, 0x8887, 0x8889, 0x888a, 0x888c,
2030 0x888e, 0x888f, 0x8890, 0x8891, 0x8893, 0x8894, 0x8895, 0x8897,
2031 0x8898, 0x8899, 0x889a, 0x889b, 0x889d, 0x889e, 0x889f, 0x88aa,
2032 0x88a1, 0x88a3, 0x88a5, 0x88a6, 0x88a7, 0x88a8, 0x88a9, 0x88aa,
2033 0x5c0f, 0x5b5d, 0x6821, 0x8096, 0x5578, 0x7b11, 0x6548, 0x6954,
2034 0x4e9b, 0x6b47, 0x874e, 0x978b, 0x534f, 0x631f, 0x643a, 0x90aa,
2035 0x659c, 0x80c1, 0x8c10, 0x5199, 0x68b0, 0x5378, 0x87f9, 0x61c8,
2036 0x6cc4, 0x6cfb, 0x8c22, 0x5c51, 0x85aa, 0x82af, 0x950c, 0x6b23,
2037 0x8f9b, 0x65b0, 0x5ffb, 0x5fc3, 0x4fe1, 0x8845, 0x661f, 0x8165,
2038 0x7329, 0x60fa, 0x5174, 0x5211, 0x578b, 0x5f62, 0x90a2, 0x884c,
2039 0x9192, 0x5e78, 0x674f, 0x6027, 0x59d3, 0x5144, 0x51f6, 0x80f8,
2040 0x5308, 0x6c79, 0x96c4, 0x718a, 0x4f11, 0x4fee, 0x7f9e, 0x673d,
2041 0x55c5, 0x9508, 0x79c0, 0x8896, 0x7ee3, 0x589f, 0x620c, 0x9700,
2042 0x865a, 0x5618, 0x987b, 0x5f90, 0x8bb8, 0x84c4, 0x9157, 0x53d9,
2043 0x65ed, 0x5e8f, 0x755c, 0x6064, 0x7d6e, 0x5a7f, 0x7eea, 0x7eed,
2044 0x8f69, 0x55a7, 0x5ba3, 0x60ac, 0x65cb, 0x7384,
2045 /* 0xd1 */
2046 0x88ac, 0x88ae, 0x88af, 0x88b0, 0x88b2, 0x88b3, 0x88b4, 0x88b5,
2047 0x88b6, 0x88b8, 0x88b9, 0x88ba, 0x88bb, 0x88bd, 0x88be, 0x88bf,
2048 0x88c0, 0x88c3, 0x88c4, 0x88c7, 0x88c8, 0x88ca, 0x88cb, 0x88cc,
2049 0x88cd, 0x88cf, 0x88d0, 0x88d1, 0x88d3, 0x88d6, 0x88d7, 0x88da,
2050 0x88db, 0x88dc, 0x88dd, 0x88de, 0x88e0, 0x88e1, 0x88e6, 0x88e7,
2051 0x88e9, 0x88ea, 0x88eb, 0x88ec, 0x88ed, 0x88ee, 0x88ef, 0x88f2,
2052 0x88f5, 0x88f6, 0x88f7, 0x88fa, 0x88fb, 0x88fd, 0x88ff, 0x8900,
2053 0x8901, 0x8903, 0x8904, 0x8905, 0x8906, 0x8907, 0x8908, 0x8909,
2054 0x890b, 0x890c, 0x890d, 0x890e, 0x890f, 0x8911, 0x8914, 0x8915,
2055 0x8916, 0x8917, 0x8918, 0x891c, 0x891d, 0x891e, 0x891f, 0x8920,
2056 0x8922, 0x8923, 0x8924, 0x8926, 0x8927, 0x8928, 0x8929, 0x892c,
2057 0x892d, 0x892e, 0x892f, 0x8931, 0x8932, 0x8933, 0x8935, 0x8937,
2058 0x9009, 0x7663, 0x7729, 0x7eda, 0x9774, 0x859b, 0x5b66, 0x7a74,
2059 0x96ea, 0x8840, 0x52cb, 0x718f, 0x5faa, 0x65ec, 0x8be2, 0x5bfb,
2060 0x9a6f, 0x5de1, 0x6b89, 0x6c5b, 0x8bad, 0x8baf, 0x900a, 0x8fc5,
2061 0x538b, 0x62bc, 0x9e26, 0x9e2d, 0x5440, 0x4e2b, 0x82bd, 0x7259,
2062 0x869c, 0x5d16, 0x8859, 0x6daf, 0x96c5, 0x54d1, 0x4e9a, 0x8bb6,
2063 0x7109, 0x54bd, 0x9609, 0x70df, 0x6df9, 0x76d0, 0x4e25, 0x7814,
2064 0x8712, 0x5ca9, 0x5ef6, 0x8a00, 0x989c, 0x960e, 0x708e, 0x6cbf,
2065 0x5944, 0x63a9, 0x773c, 0x884d, 0x6f14, 0x8273, 0x5830, 0x71d5,
2066 0x538c, 0x781a, 0x96c1, 0x5501, 0x5f66, 0x7130, 0x5bb4, 0x8c1a,
2067 0x9a8c, 0x6b83, 0x592e, 0x9e2f, 0x79e7, 0x6768, 0x626c, 0x4f6f,
2068 0x75a1, 0x7f8a, 0x6d0b, 0x9633, 0x6c27, 0x4ef0, 0x75d2, 0x517b,
2069 0x6837, 0x6f3e, 0x9080, 0x8170, 0x5996, 0x7476,
2070 /* 0xd2 */
2071 0x8938, 0x8939, 0x893a, 0x893b, 0x893c, 0x893d, 0x893e, 0x893f,
2072 0x8940, 0x8942, 0x8943, 0x8945, 0x8946, 0x8947, 0x8948, 0x8949,
2073 0x894a, 0x894b, 0x894c, 0x894d, 0x894e, 0x894f, 0x8950, 0x8951,
2074 0x8952, 0x8953, 0x8954, 0x8955, 0x8956, 0x8957, 0x8958, 0x8959,
2075 0x895a, 0x895b, 0x895c, 0x895d, 0x8960, 0x8961, 0x8962, 0x8963,
2076 0x8964, 0x8965, 0x8967, 0x8968, 0x8969, 0x896a, 0x896b, 0x896c,
2077 0x896d, 0x896e, 0x896f, 0x8970, 0x8971, 0x8972, 0x8973, 0x8974,
2078 0x8975, 0x8976, 0x8977, 0x8978, 0x8979, 0x897a, 0x897c, 0x897d,
2079 0x897e, 0x8980, 0x8982, 0x8984, 0x8985, 0x8987, 0x8988, 0x8989,
2080 0x898a, 0x898b, 0x898c, 0x898d, 0x898e, 0x898f, 0x8990, 0x8991,
2081 0x8992, 0x8993, 0x8994, 0x8995, 0x8996, 0x8997, 0x8998, 0x8999,
2082 0x899a, 0x899b, 0x899c, 0x899d, 0x899e, 0x899f, 0x89a0, 0x89a1,
2083 0x6447, 0x5c27, 0x9065, 0x7a91, 0x8c23, 0x59da, 0x54ac, 0x8200,
2084 0x836f, 0x8981, 0x8000, 0x6930, 0x564e, 0x8036, 0x7237, 0x91ce,
2085 0x51b6, 0x4e5f, 0x9875, 0x6396, 0x4e1a, 0x53f6, 0x66f3, 0x814b,
2086 0x591c, 0x6db2, 0x4e00, 0x58f9, 0x533b, 0x63d6, 0x94f1, 0x4f9d,
```

```

2087 0x4f0a, 0x8863, 0x9890, 0x5937, 0x9057, 0x79fb, 0x4eea, 0x80f0,
2088 0x7591, 0x6c82, 0x5b9c, 0x59e8, 0x5f5d, 0x6905, 0x8681, 0x501a,
2089 0x5df2, 0x4e59, 0x77e3, 0x4ee5, 0x827a, 0x6291, 0x6613, 0x9091,
2090 0x5c79, 0x4ebf, 0x5f79, 0x81c6, 0x9038, 0x8084, 0x75ab, 0x4ea6,
2091 0x88d4, 0x610f, 0x6bc5, 0x5fc6, 0x4e49, 0x76ca, 0x6ea2, 0x8be3,
2092 0x8bae, 0x8c0a, 0x8bd1, 0x5f02, 0x7ffc, 0x7fcc, 0x7ece, 0x8335,
2093 0x836b, 0x56e0, 0x6bb7, 0x97f3, 0x9634, 0x59fb, 0x541f, 0x94f6,
2094 0x6deb, 0x5bc5, 0x996e, 0x5c39, 0x5f15, 0x9690,
2095 /* 0xd3 */
2096 0x89a2, 0x89a3, 0x89a4, 0x89a5, 0x89a6, 0x89a7, 0x89a8, 0x89a9,
2097 0x89aa, 0x89ab, 0x89ac, 0x89ad, 0x89ae, 0x89af, 0x89b0, 0x89b1,
2098 0x89b2, 0x89b3, 0x89b4, 0x89b5, 0x89b6, 0x89b7, 0x89b8, 0x89b9,
2099 0x89ba, 0x89bb, 0x89bc, 0x89bd, 0x89be, 0x89bf, 0x89c0, 0x89c3,
2100 0x89cd, 0x89d3, 0x89d4, 0x89d5, 0x89d7, 0x89d8, 0x89d9, 0x89db,
2101 0x89dd, 0x89df, 0x89e0, 0x89e1, 0x89e2, 0x89e4, 0x89e7, 0x89e8,
2102 0x89e9, 0x89ea, 0x89ec, 0x89ed, 0x89ee, 0x89f0, 0x89f1, 0x89f2,
2103 0x89f4, 0x89f5, 0x89f6, 0x89f7, 0x89f8, 0x89f9, 0x89fa, 0x89fb,
2104 0x89fc, 0x89fd, 0x89fe, 0x89ff, 0x8a01, 0x8a02, 0x8a03, 0x8a04,
2105 0x8a05, 0x8a06, 0x8a08, 0x8a09, 0x8a0a, 0x8a0b, 0x8a0c, 0x8a0d,
2106 0x8a0e, 0x8a0f, 0x8a10, 0x8a11, 0x8a12, 0x8a13, 0x8a14, 0x8a15,
2107 0x8a16, 0x8a17, 0x8a18, 0x8a19, 0x8a1a, 0x8a1b, 0x8a1c, 0x8a1d,
2108 0x5370, 0x82f1, 0x6a31, 0x5a74, 0x9e70, 0x5e94, 0x7f28, 0x83b9,
2109 0x8424, 0x8425, 0x8367, 0x8747, 0x8fce, 0x8d62, 0x76c8, 0x5f71,
2110 0x9896, 0x896c, 0x6620, 0x54df, 0x62e5, 0x4f63, 0x81c3, 0x75c8,
2111 0x5eb8, 0x96cd, 0x8e0a, 0x86f9, 0x548f, 0x6cf3, 0x6d8c, 0x6c38,
2112 0x607f, 0x52c7, 0x7528, 0x5e7d, 0x4f18, 0x60a0, 0x5fe7, 0x5c24,
2113 0x7531, 0x90ae, 0x94c0, 0x72b9, 0x6cb9, 0x6e38, 0x9149, 0x6709,
2114 0x53cb, 0x53f3, 0x4f51, 0x91c9, 0x8bf1, 0x53c8, 0x5e7c, 0x8fc2,
2115 0x6de4, 0x4e8e, 0x76c2, 0x6986, 0x865e, 0x611a, 0x8206, 0x4f59,
2116 0x4fde, 0x903e, 0x9c7c, 0x6109, 0x6e1d, 0x6e14, 0x9685, 0x4e88,
2117 0x5a31, 0x96e8, 0x4e0e, 0x5c7f, 0x79b9, 0x5b87, 0x8bed, 0x7fbd,
2118 0x7389, 0x57df, 0x828b, 0x90c1, 0x5401, 0x9047, 0x55bb, 0x5cea,
2119 0x5fa1, 0x6108, 0x6b32, 0x72f1, 0x80b2, 0x8a89,
2120 /* 0xd4 */
2121 0x8a1e, 0x8a1f, 0x8a20, 0x8a21, 0x8a22, 0x8a23, 0x8a24, 0x8a25,
2122 0x8a26, 0x8a27, 0x8a28, 0x8a29, 0x8a2a, 0x8a2b, 0x8a2c, 0x8a2d,
2123 0x8a2e, 0x8a2f, 0x8a30, 0x8a31, 0x8a32, 0x8a33, 0x8a34, 0x8a35,
2124 0x8a36, 0x8a37, 0x8a38, 0x8a39, 0x8a3a, 0x8a3b, 0x8a3c, 0x8a3d,
2125 0x8a3f, 0x8a40, 0x8a41, 0x8a42, 0x8a43, 0x8a44, 0x8a45, 0x8a46,
2126 0x8a47, 0x8a49, 0x8a4a, 0x8a4b, 0x8a4c, 0x8a4d, 0x8a4e, 0x8a4f,
2127 0x8a50, 0x8a51, 0x8a52, 0x8a53, 0x8a54, 0x8a55, 0x8a56, 0x8a57,
2128 0x8a58, 0x8a59, 0x8a5a, 0x8a5b, 0x8a5c, 0x8a5d, 0x8a5e, 0x8a5f,
2129 0x8a60, 0x8a61, 0x8a62, 0x8a63, 0x8a64, 0x8a65, 0x8a66, 0x8a67,
2130 0x8a68, 0x8a69, 0x8a6a, 0x8a6b, 0x8a6c, 0x8a6d, 0x8a6e, 0x8a6f,
2131 0x8a70, 0x8a71, 0x8a72, 0x8a73, 0x8a74, 0x8a75, 0x8a76, 0x8a77,
2132 0x8a78, 0x8a7a, 0x8a7b, 0x8a7c, 0x8a7d, 0x8a7e, 0x8a7f, 0x8a80,
2133 0x6d74, 0x5bd3, 0x88d5, 0x9884, 0x8c6b, 0x9a6d, 0x9e33, 0x6e0a,
2134 0x51a4, 0x5143, 0x57a3, 0x8881, 0x539f, 0x63f4, 0x8f95, 0x56ed,
2135 0x5458, 0x5706, 0x733f, 0x6e90, 0x7f18, 0x8fdc, 0x82d1, 0x613f,
2136 0x6028, 0x9662, 0x66f0, 0x7ea6, 0x8d8a, 0x8dc3, 0x94a5, 0x5cb3,
2137 0x7ca4, 0x6708, 0x60a6, 0x9605, 0x8018, 0x4e91, 0x90e7, 0x5300,
2138 0x9668, 0x5141, 0x8fd0, 0x8574, 0x915d, 0x6655, 0x97f5, 0x5b55,
2139 0x531d, 0x7838, 0x6742, 0x683d, 0x54c9, 0x707e, 0x5bb0, 0x8f7d,
2140 0x518d, 0x5728, 0x54b1, 0x6512, 0x6682, 0x8d5e, 0x8d43, 0x810f,
2141 0x846c, 0x906d, 0x7cdf, 0x51ff, 0x85fb, 0x67a3, 0x65e9, 0x6fa1,
2142 0x86a4, 0x8e81, 0x566a, 0x9020, 0x7682, 0x7076, 0x71e5, 0x8d23,
2143 0x62e9, 0x5219, 0x6cfd, 0x8d3c, 0x600e, 0x589e, 0x618e, 0x66fe,
2144 0x8d60, 0x624e, 0x55b3, 0x6e23, 0x672d, 0x8f67,
2145 /* 0xd5 */
2146 0x8a81, 0x8a82, 0x8a83, 0x8a84, 0x8a85, 0x8a86, 0x8a87, 0x8a88,
2147 0x8a8b, 0x8a8c, 0x8a8d, 0x8a8e, 0x8a8f, 0x8a90, 0x8a91, 0x8a92,
2148 0x8a94, 0x8a95, 0x8a96, 0x8a97, 0x8a98, 0x8a99, 0x8a9a, 0x8a9b,
2149 0x8a9c, 0x8a9d, 0x8a9e, 0x8a9f, 0x8aa0, 0x8aa1, 0x8aa2, 0x8aa3,
2150 0x8aa4, 0x8aa5, 0x8aa6, 0x8aa7, 0x8aa8, 0x8aa9, 0x8aaa, 0x8aab,
2151 0x8aac, 0x8aad, 0x8aae, 0x8aaf, 0x8ab0, 0x8ab1, 0x8ab2, 0x8ab3,
2152 0x8ab4, 0x8ab5, 0x8ab6, 0x8ab7, 0x8ab8, 0x8ab9, 0x8aba, 0x8abb,
2153 0x8abc, 0x8abd, 0x8abe, 0x8abf, 0x8ac0, 0x8ac1, 0x8ac2, 0x8ac3,
2154 0x8ac4, 0x8ac5, 0x8ac6, 0x8ac7, 0x8ac8, 0x8ac9, 0x8aca, 0x8acb,
2155 0x8acc, 0x8acd, 0x8ace, 0x8acf, 0x8ad0, 0x8ad1, 0x8ad2, 0x8ad3,
2156 0x8ad4, 0x8ad5, 0x8ad6, 0x8ad7, 0x8ad8, 0x8ad9, 0x8ada, 0x8adb,
2157 0x8adc, 0x8add, 0x8ade, 0x8adf, 0x8ae0, 0x8ae1, 0x8ae2, 0x8ae3,
2158 0x94e1, 0x95f8, 0x7728, 0x6805, 0x69a8, 0x548b, 0x4e4d, 0x70b8,
2159 0x8bc8, 0x6458, 0x658b, 0x5b85, 0x7a84, 0x503a, 0x5be8, 0x77bb,
2160 0x6be1, 0x8a79, 0x7c98, 0x6cbe, 0x76cf, 0x65a9, 0x8f97, 0x5d2d,
2161 0x5c55, 0x8638, 0x6808, 0x5360, 0x6218, 0x7ad9, 0x6e5b, 0x7efd,
2162 0x6a1f, 0x7ae0, 0x5f70, 0x6f33, 0x5f20, 0x638c, 0x6da8, 0x6756,
2163 0x4e08, 0x5e10, 0x8d26, 0x4ed7, 0x80c0, 0x7634, 0x969c, 0x62db,
2164 0x662d, 0x627e, 0x6cbc, 0x8d75, 0x7167, 0x7f69, 0x5146, 0x8087,
2165 0x53ec, 0x906e, 0x6298, 0x54f2, 0x86f0, 0x8f99, 0x8005, 0x9517,
2166 0x8517, 0x8fd9, 0x6d59, 0x73cd, 0x659f, 0x771f, 0x7504, 0x7827,
2167 0x81fb, 0x8dle, 0x9488, 0x4fa6, 0x6795, 0x75b9, 0x8bca, 0x9707,
2168 0x632f, 0x9547, 0x9635, 0x84b8, 0x6323, 0x7741, 0x5f81, 0x72f0,
2169 0x4e89, 0x6014, 0x6574, 0x62ef, 0x6b63, 0x653f,
2170 /* 0xd6 */
2171 0x8ae4, 0x8ae5, 0x8ae6, 0x8ae7, 0x8ae8, 0x8ae9, 0x8aea, 0x8aeb,
2172 0x8aec, 0x8aed, 0x8aee, 0x8aef, 0x8af0, 0x8af1, 0x8af2, 0x8af3,
2173 0x8af4, 0x8af5, 0x8af6, 0x8af7, 0x8af8, 0x8af9, 0x8afa, 0x8afb,

```

```
2174 0x8afc, 0x8afd, 0x8afe, 0x8aff, 0x8b00, 0x8b01, 0x8b02, 0x8b03,
2175 0x8b04, 0x8b05, 0x8b06, 0x8b08, 0x8b09, 0x8b0a, 0x8b0b, 0x8b0c,
2176 0x8b0d, 0x8b0e, 0x8b0f, 0x8b10, 0x8b11, 0x8b12, 0x8b13, 0x8b14,
2177 0x8b15, 0x8b16, 0x8b17, 0x8b18, 0x8b19, 0x8b1a, 0x8b1b, 0x8b1c,
2178 0x8b1d, 0x8b1e, 0x8b1f, 0x8b20, 0x8b21, 0x8b22, 0x8b23, 0x8b24,
2179 0x8b25, 0x8b27, 0x8b28, 0x8b29, 0x8b2a, 0x8b2b, 0x8b2c, 0x8b2d,
2180 0x8b2e, 0x8b2f, 0x8b30, 0x8b31, 0x8b32, 0x8b33, 0x8b34, 0x8b35,
2181 0x8b36, 0x8b37, 0x8b38, 0x8b39, 0x8b3a, 0x8b3b, 0x8b3c, 0x8b3d,
2182 0x8b3e, 0x8b3f, 0x8b40, 0x8b41, 0x8b42, 0x8b43, 0x8b44, 0x8b45,
2183 0x5e27, 0x75c7, 0x90d1, 0x8bc1, 0x829d, 0x679d, 0x652f, 0x5431,
2184 0x8718, 0x77e5, 0x80a2, 0x8102, 0x6c41, 0x4e4b, 0x7ec7, 0x804c,
2185 0x76f4, 0x690d, 0x6b96, 0x6267, 0x503c, 0x4f84, 0x5740, 0x6307,
2186 0x6b62, 0x8dbe, 0x53ea, 0x65e8, 0x7eb8, 0x5fd7, 0x631a, 0x63b7,
2187 0x81f3, 0x81f4, 0x7f6e, 0x5e1c, 0x5cd9, 0x5236, 0x667a, 0x79e9,
2188 0x7a1a, 0x8d28, 0x7099, 0x75d4, 0x6ede, 0x6cbb, 0x7a92, 0x4e2d,
2189 0x76c5, 0x5fe0, 0x949f, 0x8877, 0x7ec8, 0x79cd, 0x80bf, 0x91cd,
2190 0x4ef2, 0x4f17, 0x821f, 0x5468, 0x5dde, 0x6d32, 0x8bcc, 0x7ca5,
2191 0x8f74, 0x8098, 0x5e1a, 0x5492, 0x76b1, 0x5b99, 0x663c, 0x9aa4,
2192 0x73e0, 0x682a, 0x86db, 0x6731, 0x732a, 0x8bf8, 0x8bdb, 0x9010,
2193 0x7af9, 0x70db, 0x716e, 0x62c4, 0x77a9, 0x5631, 0x4e3b, 0x8457,
2194 0x67f1, 0x52a9, 0x86c0, 0x8d2e, 0x94f8, 0x7b51,
2195 /* 0xd7 */
2196 0x8b46, 0x8b47, 0x8b48, 0x8b49, 0x8b4a, 0x8b4b, 0x8b4c, 0x8b4d,
2197 0x8b4e, 0x8b4f, 0x8b50, 0x8b51, 0x8b52, 0x8b53, 0x8b54, 0x8b55,
2198 0x8b56, 0x8b57, 0x8b58, 0x8b59, 0x8b5a, 0x8b5b, 0x8b5c, 0x8b5d,
2199 0x8b5e, 0x8b5f, 0x8b60, 0x8b61, 0x8b62, 0x8b63, 0x8b64, 0x8b65,
2200 0x8b67, 0x8b68, 0x8b69, 0x8b6a, 0x8b6b, 0x8b6d, 0x8b6e, 0x8b6f,
2201 0x8b70, 0x8b71, 0x8b72, 0x8b73, 0x8b74, 0x8b75, 0x8b76, 0x8b77,
2202 0x8b78, 0x8b79, 0x8b7a, 0x8b7b, 0x8b7c, 0x8b7d, 0x8b7e, 0x8b7f,
2203 0x8b80, 0x8b81, 0x8b82, 0x8b83, 0x8b84, 0x8b85, 0x8b86, 0x8b87,
2204 0x8b88, 0x8b89, 0x8b8a, 0x8b8b, 0x8b8c, 0x8b8d, 0x8b8e, 0x8b8f,
2205 0x8b90, 0x8b91, 0x8b92, 0x8b93, 0x8b94, 0x8b95, 0x8b96, 0x8b97,
2206 0x8b98, 0x8b99, 0x8b9a, 0x8b9b, 0x8b9c, 0x8b9d, 0x8b9e, 0x8b9f,
2207 0x8bac, 0x8bb1, 0x8bbb, 0x8bc7, 0x8bd0, 0x8bea, 0x8c09, 0x8c1e,
2208 0x4f4f, 0x6ce8, 0x795d, 0x9a7b, 0x6293, 0x722a, 0x62fd, 0x4e13,
2209 0x7816, 0x8f6c, 0x64b0, 0x8d5a, 0x7bc6, 0x6869, 0x5e84, 0x88c5,
2210 0x5986, 0x649e, 0x58ee, 0x72b6, 0x690e, 0x9525, 0x8ffd, 0x8d58,
2211 0x5760, 0x7f00, 0x8c06, 0x51c6, 0x6349, 0x62d9, 0x5353, 0x684c,
2212 0x7422, 0x8301, 0x914c, 0x5544, 0x7740, 0x707c, 0x6d4a, 0x5179,
2213 0x54a8, 0x8d44, 0x59ff, 0x6ecb, 0x6dc4, 0x5b5c, 0x7d2b, 0x4ed4,
2214 0x7c7d, 0x6ed3, 0x5b50, 0x81ea, 0x6e0d, 0x5b57, 0x9b03, 0x68d5,
2215 0x8e2a, 0x5b97, 0x7efc, 0x603b, 0x7eb5, 0x90b9, 0x8d70, 0x594f,
2216 0x63cd, 0x79df, 0x8db3, 0x5352, 0x65cf, 0x7956, 0x8bc5, 0x963b,
2217 0x7ec4, 0x94bb, 0x7e82, 0x5634, 0x9189, 0x6700, 0x7f6a, 0x5c0a,
2218 0x9075, 0x6628, 0x5de6, 0x4f50, 0x67de, 0x505a, 0x4f5c, 0x5750,
2219 0x5ea7, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
2220 /* 0xd8 */
2221 0x8c38, 0x8c39, 0x8c3a, 0x8c3b, 0x8c3c, 0x8c3d, 0x8c3e, 0x8c3f,
2222 0x8c40, 0x8c42, 0x8c43, 0x8c44, 0x8c45, 0x8c48, 0x8c4a, 0x8c4b,
2223 0x8c4d, 0x8c4e, 0x8c4f, 0x8c50, 0x8c51, 0x8c52, 0x8c53, 0x8c54,
2224 0x8c56, 0x8c57, 0x8c58, 0x8c59, 0x8c5b, 0x8c5c, 0x8c5d, 0x8c5e,
2225 0x8c5f, 0x8c60, 0x8c63, 0x8c64, 0x8c65, 0x8c66, 0x8c67, 0x8c68,
2226 0x8c69, 0x8c6c, 0x8c6d, 0x8c6e, 0x8c6f, 0x8c70, 0x8c71, 0x8c72,
2227 0x8c74, 0x8c75, 0x8c76, 0x8c77, 0x8c7b, 0x8c7c, 0x8c7d, 0x8c7e,
2228 0x8c7f, 0x8c80, 0x8c81, 0x8c83, 0x8c84, 0x8c86, 0x8c87, 0x8c88,
2229 0x8c8b, 0x8c8d, 0x8c8e, 0x8c8f, 0x8c90, 0x8c91, 0x8c92, 0x8c93,
2230 0x8c95, 0x8c96, 0x8c97, 0x8c99, 0x8c9a, 0x8c9b, 0x8c9c, 0x8c9d,
2231 0x8c9e, 0x8c9f, 0x8ca0, 0x8ca1, 0x8ca2, 0x8ca3, 0x8ca4, 0x8ca5,
2232 0x8ca6, 0x8ca7, 0x8ca8, 0x8ca9, 0x8caa, 0x8cab, 0x8cac, 0x8cad,
2233 0x4e8d, 0x4e0c, 0x5140, 0x4e10, 0x5eff, 0x5345, 0x4e15, 0x4e98,
2234 0x4e1e, 0x9b32, 0x5b6c, 0x5669, 0x4e28, 0x79ba, 0x4e3f, 0x5315,
2235 0x4e47, 0x592d, 0x723b, 0x536e, 0x6c10, 0x56df, 0x80e4, 0x9997,
2236 0x6bd3, 0x777e, 0x9f17, 0x4e36, 0x4e9f, 0x9f10, 0x4e5c, 0x4e69,
2237 0x4e93, 0x8288, 0x5b5b, 0x556c, 0x560f, 0x4ec4, 0x538d, 0x539d,
2238 0x53a3, 0x53a5, 0x53ae, 0x9765, 0x8d5d, 0x531a, 0x53f5, 0x5326,
2239 0x532e, 0x533e, 0x8d5c, 0x5366, 0x5363, 0x5202, 0x5208, 0x520e,
2240 0x522d, 0x5233, 0x523f, 0x5240, 0x524c, 0x525e, 0x5261, 0x525c,
2241 0x84af, 0x527d, 0x5282, 0x5281, 0x5290, 0x5293, 0x5182, 0x7f54,
2242 0x4ebb, 0x4ec3, 0x4ec9, 0x4ec2, 0x4ee8, 0x4ee1, 0x4eeb, 0x4ede,
2243 0x4f1b, 0x4ef3, 0x4f22, 0x4f64, 0x4ef5, 0x4f25, 0x4f27, 0x4f09,
2244 0x4f2b, 0x4f5e, 0x4f67, 0x6538, 0x4f5a, 0x4f5d,
2245 /* 0xd9 */
2246 0x8cae, 0x8caf, 0x8cb0, 0x8cb1, 0x8cb2, 0x8cb3, 0x8cb4, 0x8cb5,
2247 0x8cb6, 0x8cb7, 0x8cb8, 0x8cb9, 0x8cba, 0x8cbb, 0x8cbc, 0x8cbd,
2248 0x8cbe, 0x8cbf, 0x8cc0, 0x8cc1, 0x8cc2, 0x8cc3, 0x8cc4, 0x8cc5,
2249 0x8cc6, 0x8cc7, 0x8cc8, 0x8cc9, 0x8cca, 0x8ccb, 0x8ccc, 0x8ccd,
2250 0x8cce, 0x8ccf, 0x8cd0, 0x8cd1, 0x8cd2, 0x8cd3, 0x8cd4, 0x8cd5,
2251 0x8cd6, 0x8cd7, 0x8cd8, 0x8cd9, 0x8cda, 0x8cdb, 0x8cdc, 0x8cdd,
2252 0x8cde, 0x8cdf, 0x8ce0, 0x8ce1, 0x8ce2, 0x8ce3, 0x8ce4, 0x8ce5,
2253 0x8ce6, 0x8ce7, 0x8ce8, 0x8ce9, 0x8cea, 0x8ceb, 0x8cec, 0x8ced,
2254 0x8cee, 0x8cef, 0x8cf0, 0x8cf1, 0x8cf2, 0x8cf3, 0x8cf4, 0x8cf5,
2255 0x8cf6, 0x8cf7, 0x8cf8, 0x8cf9, 0x8cfa, 0x8cfb, 0x8cfc, 0x8cfd,
2256 0x8cfe, 0x8cff, 0x8d00, 0x8d01, 0x8d02, 0x8d03, 0x8d04, 0x8d05,
2257 0x8d06, 0x8d07, 0x8d08, 0x8d09, 0x8d0a, 0x8d0b, 0x8d0c, 0x8d0d,
2258 0x4f5f, 0x4f57, 0x4f32, 0x4f3d, 0x4f76, 0x4f74, 0x4f91, 0x4f89,
2259 0x4f83, 0x4f8f, 0x4f7e, 0x4f7b, 0x4faa, 0x4f7c, 0x4fac, 0x4f94,
2260 0x4fe6, 0x4fe8, 0x4fea, 0x4fc5, 0x4fda, 0x4fe3, 0x4fdc, 0x4fd1,
```

```
2261 0x4fdf, 0x4ff8, 0x5029, 0x504c, 0x4ff3, 0x502c, 0x500f, 0x502e,
2262 0x502d, 0x4ffe, 0x501c, 0x500c, 0x5025, 0x5028, 0x507e, 0x5043,
2263 0x5055, 0x5048, 0x504e, 0x504e, 0x506c, 0x507b, 0x50a5, 0x50a7, 0x50a9,
2264 0x50ba, 0x50d6, 0x5106, 0x50ed, 0x50ec, 0x50e6, 0x50ee, 0x5107,
2265 0x510b, 0x4edd, 0x6c3d, 0x4f58, 0x4f65, 0x4fce, 0x9fa0, 0x6c46,
2266 0x7c74, 0x516e, 0x5dfd, 0x9ec9, 0x9998, 0x5181, 0x5914, 0x5f19,
2267 0x530d, 0x8a07, 0x5310, 0x51eb, 0x5919, 0x5155, 0x4ea0, 0x5156,
2268 0x4eb3, 0x886e, 0x88a4, 0x4eb5, 0x8114, 0x88d2, 0x7980, 0x5b34,
2269 0x8803, 0x7fb8, 0x51ab, 0x51b1, 0x51bd, 0x51bc,
2270 /* 0xda */
2271 0x8d0e, 0x8d0f, 0x8d10, 0x8d11, 0x8d12, 0x8d13, 0x8d14, 0x8d15,
2272 0x8d16, 0x8d17, 0x8d18, 0x8d19, 0x8d1a, 0x8d1b, 0x8d1c, 0x8d20,
2273 0x8d51, 0x8d52, 0x8d57, 0x8d5f, 0x8d65, 0x8d68, 0x8d69, 0x8d6a,
2274 0x8d6c, 0x8d6e, 0x8d6f, 0x8d71, 0x8d72, 0x8d78, 0x8d79, 0x8d7a,
2275 0x8d7b, 0x8d7c, 0x8d7d, 0x8d7e, 0x8d7f, 0x8d80, 0x8d82, 0x8d83,
2276 0x8d86, 0x8d87, 0x8d88, 0x8d89, 0x8d8c, 0x8d8d, 0x8d8e, 0x8d8f,
2277 0x8d90, 0x8d92, 0x8d93, 0x8d95, 0x8d96, 0x8d97, 0x8d98, 0x8d99,
2278 0x8d9a, 0x8d9b, 0x8d9c, 0x8d9c, 0x8d9d, 0x8d9e, 0x8da0, 0x8da1, 0x8da2,
2279 0x8da4, 0x8da5, 0x8da6, 0x8da7, 0x8da8, 0x8da9, 0x8daa, 0x8dab,
2280 0x8dac, 0x8dad, 0x8dae, 0x8daf, 0x8db0, 0x8db2, 0x8db6, 0x8db7,
2281 0x8db9, 0x8dbb, 0x8dbd, 0x8dbd, 0x8dc0, 0x8dc1, 0x8dc2, 0x8dc5, 0x8dc7,
2282 0x8dc8, 0x8dc9, 0x8dca, 0x8dcd, 0x8dd0, 0x8dd2, 0x8dd3, 0x8dd4,
2283 0x51c7, 0x5196, 0x51a2, 0x51a5, 0x8baa, 0x8ba6, 0x8ba7, 0x8baa,
2284 0x8bb4, 0x8bb5, 0x8bb7, 0x8bb7, 0x8bc2, 0x8bc3, 0x8bcb, 0x8bcf, 0x8bce,
2285 0x8bd2, 0x8bd3, 0x8bd4, 0x8bd6, 0x8bd8, 0x8bd9, 0x8bdc, 0x8bdf,
2286 0x8be0, 0x8be4, 0x8be8, 0x8be9, 0x8bee, 0x8bf0, 0x8bf3, 0x8bf6,
2287 0x8bf9, 0x8bfc, 0x8bff, 0x8c00, 0x8c02, 0x8c04, 0x8c07, 0x8c0c,
2288 0x8c0f, 0x8c11, 0x8c12, 0x8c14, 0x8c15, 0x8c16, 0x8c19, 0x8c1b,
2289 0x8c18, 0x8c1d, 0x8c1f, 0x8c20, 0x8c21, 0x8c25, 0x8c27, 0x8c2a,
2290 0x8c2b, 0x8c2c, 0x8c2f, 0x8c32, 0x8c33, 0x8c35, 0x8c36, 0x5369,
2291 0x537a, 0x961d, 0x9622, 0x9621, 0x9631, 0x962a, 0x963d, 0x963c,
2292 0x9642, 0x9649, 0x9654, 0x965f, 0x9667, 0x966c, 0x9672, 0x9674,
2293 0x9688, 0x968d, 0x9697, 0x96b0, 0x9097, 0x909b, 0x909d, 0x9099,
2294 0x90ac, 0x90a1, 0x90b4, 0x90b3, 0x90b6, 0x90ba,
2295 /* 0xdb */
2296 0x8dd5, 0x8dd8, 0x8dd9, 0x8ddc, 0x8de0, 0x8de1, 0x8de2, 0x8de5,
2297 0x8de6, 0x8de7, 0x8de9, 0x8ded, 0x8dee, 0x8df0, 0x8df1, 0x8df2,
2298 0x8df4, 0x8df6, 0x8dfc, 0x8dfe, 0x8dff, 0x8e00, 0x8e01, 0x8e02,
2299 0x8e03, 0x8e04, 0x8e06, 0x8e07, 0x8e08, 0x8e0b, 0x8e0d, 0x8e0e,
2300 0x8e10, 0x8e11, 0x8e12, 0x8e13, 0x8e15, 0x8e16, 0x8e17, 0x8e18,
2301 0x8e19, 0x8e1a, 0x8e1b, 0x8e1c, 0x8e20, 0x8e21, 0x8e24, 0x8e25,
2302 0x8e26, 0x8e27, 0x8e28, 0x8e2b, 0x8e2d, 0x8e30, 0x8e32, 0x8e33,
2303 0x8e34, 0x8e36, 0x8e37, 0x8e38, 0x8e3b, 0x8e3c, 0x8e3e, 0x8e3f,
2304 0x8e43, 0x8e45, 0x8e46, 0x8e4c, 0x8e4d, 0x8e4e, 0x8e4f, 0x8e50,
2305 0x8e53, 0x8e54, 0x8e55, 0x8e56, 0x8e57, 0x8e58, 0x8e5a, 0x8e5b,
2306 0x8e5c, 0x8e5d, 0x8e5e, 0x8e5f, 0x8e60, 0x8e61, 0x8e62, 0x8e63,
2307 0x8e64, 0x8e65, 0x8e67, 0x8e68, 0x8e6a, 0x8e6b, 0x8e6e, 0x8e71,
2308 0x90b8, 0x90b0, 0x90cf, 0x90c5, 0x90be, 0x90d0, 0x90c4, 0x90c7,
2309 0x90d3, 0x90e6, 0x90e2, 0x90dc, 0x90d7, 0x90db, 0x90eb, 0x90ef,
2310 0x90fe, 0x9104, 0x9122, 0x911e, 0x9123, 0x9131, 0x912f, 0x9139,
2311 0x9143, 0x9146, 0x520d, 0x5942, 0x52a2, 0x52ac, 0x52ad, 0x52be,
2312 0x54ff, 0x52d0, 0x52d6, 0x52f0, 0x53df, 0x71ee, 0x77cd, 0x5ef4,
2313 0x51f5, 0x51fc, 0x9b2f, 0x53b6, 0x5f01, 0x755a, 0x5def, 0x574c,
2314 0x57a9, 0x57a1, 0x587e, 0x587e, 0x58bc, 0x58c5, 0x58d1, 0x5729, 0x572c,
2315 0x572a, 0x5733, 0x5739, 0x572e, 0x572f, 0x575c, 0x573b, 0x5742,
2316 0x5769, 0x5785, 0x576b, 0x5786, 0x577c, 0x577b, 0x5768, 0x576d,
2317 0x5776, 0x5773, 0x57ad, 0x57a4, 0x578c, 0x57b2, 0x57cf, 0x57a7,
2318 0x57b4, 0x5793, 0x57a0, 0x57d5, 0x57d8, 0x57da, 0x57d9, 0x57d2,
2319 0x57b8, 0x57f4, 0x57ef, 0x57f8, 0x57e4, 0x57dd,
2320 /* 0xdc */
2321 0x8e73, 0x8e75, 0x8e77, 0x8e78, 0x8e79, 0x8e7a, 0x8e7b, 0x8e7d,
2322 0x8e7e, 0x8e80, 0x8e82, 0x8e83, 0x8e84, 0x8e86, 0x8e88, 0x8e89,
2323 0x8e8a, 0x8e8b, 0x8e8c, 0x8e8d, 0x8e8e, 0x8e91, 0x8e92, 0x8e93,
2324 0x8e95, 0x8e96, 0x8e97, 0x8e98, 0x8e99, 0x8e9a, 0x8e9b, 0x8e9d,
2325 0x8e9f, 0x8ea0, 0x8ea1, 0x8ea2, 0x8ea3, 0x8ea4, 0x8ea5, 0x8ea6,
2326 0x8ea7, 0x8ea8, 0x8ea9, 0x8eaa, 0x8ead, 0x8eae, 0x8eb0, 0x8eb1,
2327 0x8eb3, 0x8eb4, 0x8eb5, 0x8eb6, 0x8eb7, 0x8eb8, 0x8eb9, 0x8ebb,
2328 0x8ebc, 0x8ebd, 0x8ebe, 0x8ebf, 0x8ec0, 0x8ec1, 0x8ec2, 0x8ec3,
2329 0x8ec4, 0x8ec5, 0x8ec6, 0x8ec7, 0x8ec8, 0x8ec9, 0x8eca, 0x8ech,
2330 0x8ecc, 0x8ecd, 0x8ecf, 0x8ed0, 0x8ed1, 0x8ed2, 0x8ed3, 0x8ed4,
2331 0x8ed5, 0x8ed6, 0x8ed7, 0x8ed8, 0x8ed9, 0x8eda, 0x8edb, 0x8edc,
2332 0x8edd, 0x8ede, 0x8edf, 0x8ee0, 0x8ee1, 0x8ee2, 0x8ee3, 0x8ee4,
2333 0x580b, 0x580d, 0x57fd, 0x57ed, 0x5800, 0x581e, 0x5819, 0x5844,
2334 0x5820, 0x5865, 0x586c, 0x5881, 0x5889, 0x589a, 0x5880, 0x99a8,
2335 0x9f19, 0x61ff, 0x8279, 0x827d, 0x827f, 0x828f, 0x828a, 0x82a8,
2336 0x8284, 0x828e, 0x8291, 0x8297, 0x8299, 0x82ab, 0x82b8, 0x82be,
2337 0x82b0, 0x82c8, 0x82ca, 0x82e3, 0x8298, 0x82b7, 0x82ae, 0x82cb,
2338 0x82cc, 0x82c1, 0x82a9, 0x82b4, 0x82a1, 0x82aa, 0x829f, 0x82c4,
2339 0x82ce, 0x82a4, 0x82e1, 0x8309, 0x82f7, 0x82e4, 0x830f, 0x8307,
2340 0x82dc, 0x82f4, 0x82d2, 0x82d8, 0x830c, 0x82fb, 0x82d3, 0x8311,
2341 0x831a, 0x8306, 0x8314, 0x8315, 0x82e0, 0x82d5, 0x831c, 0x8351,
2342 0x835b, 0x835c, 0x8308, 0x8392, 0x833c, 0x8334, 0x8331, 0x839b,
2343 0x835e, 0x832f, 0x834f, 0x8347, 0x8343, 0x835f, 0x8340, 0x8317,
2344 0x8360, 0x832d, 0x833a, 0x8333, 0x8366, 0x8365,
2345 /* 0xdd */
2346 0x8ee5, 0x8ee6, 0x8ee7, 0x8ee8, 0x8ee9, 0x8eea, 0x8eeb, 0x8eec,
2347 0x8eed, 0x8eee, 0x8eef, 0x8ef0, 0x8ef1, 0x8ef2, 0x8ef3, 0x8ef4,
```

```

2348 0x8ef5, 0x8ef6, 0x8ef7, 0x8ef8, 0x8ef9, 0x8efa, 0x8efb, 0x8efc,
2349 0x8efd, 0x8efe, 0x8eff, 0x8f00, 0x8f01, 0x8f02, 0x8f03, 0x8f04,
2350 0x8f05, 0x8f06, 0x8f07, 0x8f08, 0x8f09, 0x8f0a, 0x8f0b, 0x8f0c,
2351 0x8f0d, 0x8f0e, 0x8f0f, 0x8f10, 0x8f11, 0x8f12, 0x8f13, 0x8f14,
2352 0x8f15, 0x8f16, 0x8f17, 0x8f18, 0x8f19, 0x8f1a, 0x8f1b, 0x8f1c,
2353 0x8f1d, 0x8f1e, 0x8f1f, 0x8f20, 0x8f21, 0x8f22, 0x8f23, 0x8f24,
2354 0x8f25, 0x8f26, 0x8f27, 0x8f28, 0x8f29, 0x8f2a, 0x8f2b, 0x8f2c,
2355 0x8f2d, 0x8f2e, 0x8f2f, 0x8f30, 0x8f31, 0x8f32, 0x8f33, 0x8f34,
2356 0x8f35, 0x8f36, 0x8f37, 0x8f38, 0x8f39, 0x8f3a, 0x8f3b, 0x8f3c,
2357 0x8f3d, 0x8f3e, 0x8f3f, 0x8f40, 0x8f41, 0x8f42, 0x8f43, 0x8f44,
2358 0x8368, 0x831b, 0x8369, 0x836c, 0x836a, 0x836d, 0x836e, 0x83b0,
2359 0x8378, 0x83b3, 0x83b4, 0x83a0, 0x83aa, 0x8393, 0x839c, 0x8385,
2360 0x837c, 0x83b6, 0x83a9, 0x837d, 0x83b8, 0x837b, 0x8398, 0x839e,
2361 0x83a8, 0x83ba, 0x83bc, 0x83c1, 0x8401, 0x83e5, 0x83d8, 0x8507,
2362 0x8418, 0x840b, 0x83dd, 0x83fd, 0x83d6, 0x841c, 0x8438, 0x8411,
2363 0x8406, 0x83d4, 0x83df, 0x840f, 0x8403, 0x83f8, 0x83f9, 0x83ea,
2364 0x83c5, 0x83c0, 0x8426, 0x83f0, 0x83e1, 0x845c, 0x8451, 0x845a,
2365 0x8459, 0x8473, 0x8487, 0x8488, 0x847a, 0x8489, 0x8478, 0x843c,
2366 0x8446, 0x8469, 0x8476, 0x848c, 0x848e, 0x8431, 0x846d, 0x84c1,
2367 0x84cd, 0x84d0, 0x84e6, 0x84bd, 0x84d3, 0x84ca, 0x84bf, 0x84ba,
2368 0x84e0, 0x84a1, 0x84b9, 0x84b4, 0x8497, 0x84e5, 0x84e3, 0x850c,
2369 0x750d, 0x8538, 0x84f0, 0x8539, 0x851f, 0x853a,
2370 /* 0xde */
2371 0x8f45, 0x8f46, 0x8f47, 0x8f48, 0x8f49, 0x8f4a, 0x8f4b, 0x8f4c,
2372 0x8f4d, 0x8f4e, 0x8f4f, 0x8f50, 0x8f51, 0x8f52, 0x8f53, 0x8f54,
2373 0x8f55, 0x8f56, 0x8f57, 0x8f58, 0x8f59, 0x8f5a, 0x8f5b, 0x8f5c,
2374 0x8f5d, 0x8f5e, 0x8f5f, 0x8f60, 0x8f61, 0x8f62, 0x8f63, 0x8f64,
2375 0x8f65, 0x8f6a, 0x8f80, 0x8f8c, 0x8f92, 0x8f9d, 0x8fa0, 0x8fal,
2376 0x8fa2, 0x8fa4, 0x8fa5, 0x8fa6, 0x8fa7, 0x8faa, 0x8fac, 0x8fad,
2377 0x8fae, 0x8faf, 0x8fb2, 0x8fb3, 0x8fb4, 0x8fb5, 0x8fb7, 0x8fb8,
2378 0x8fba, 0x8fbb, 0x8fbc, 0x8fbf, 0x8fc0, 0x8fc3, 0x8fc6, 0x8fc9,
2379 0x8fca, 0x8fcb, 0x8fcc, 0x8fcd, 0x8fcf, 0x8fd2, 0x8fd6, 0x8fd7,
2380 0x8fda, 0x8fe0, 0x8fe1, 0x8fe3, 0x8fe7, 0x8fec, 0x8fef, 0x8ff1,
2381 0x8ff2, 0x8ff4, 0x8ff5, 0x8ff6, 0x8ffa, 0x8ffb, 0x8ffc, 0x8ffe,
2382 0x8fff, 0x9007, 0x9008, 0x900c, 0x900e, 0x9013, 0x9015, 0x9018,
2383 0x8556, 0x853b, 0x84ff, 0x84fc, 0x8559, 0x8548, 0x8568, 0x8564,
2384 0x855e, 0x857a, 0x77a2, 0x8543, 0x8572, 0x857b, 0x85a4, 0x85a8,
2385 0x8587, 0x858f, 0x8579, 0x85ae, 0x859c, 0x8585, 0x85b9, 0x85b7,
2386 0x85b0, 0x85d3, 0x85c1, 0x85dc, 0x85ff, 0x8627, 0x8605, 0x8629,
2387 0x8616, 0x863c, 0x5efe, 0x5f08, 0x593c, 0x5941, 0x8037, 0x5955,
2388 0x595a, 0x5958, 0x530f, 0x5c22, 0x5c25, 0x5c2c, 0x5c34, 0x624c,
2389 0x626a, 0x629f, 0x62bb, 0x62ca, 0x62da, 0x62d7, 0x62ee, 0x6322,
2390 0x62f6, 0x6339, 0x634b, 0x6343, 0x63ad, 0x63f6, 0x6371, 0x637a,
2391 0x638e, 0x63b4, 0x636d, 0x63ac, 0x638a, 0x6369, 0x63ae, 0x63bc,
2392 0x63f2, 0x63f8, 0x63e0, 0x63ff, 0x63c4, 0x63de, 0x63ce, 0x6452,
2393 0x63c6, 0x63be, 0x6445, 0x6441, 0x640b, 0x641b, 0x6420, 0x640c,
2394 0x6426, 0x6421, 0x645e, 0x6484, 0x646d, 0x6496,
2395 /* 0xdf */
2396 0x9019, 0x901c, 0x9023, 0x9024, 0x9025, 0x9027, 0x9028, 0x9029,
2397 0x902a, 0x902b, 0x902c, 0x9030, 0x9031, 0x9032, 0x9033, 0x9034,
2398 0x9037, 0x9039, 0x903a, 0x903d, 0x903f, 0x9040, 0x9043, 0x9045,
2399 0x9046, 0x9048, 0x9049, 0x904a, 0x904b, 0x904c, 0x904e, 0x9054,
2400 0x9055, 0x9056, 0x9059, 0x905a, 0x905c, 0x905d, 0x905e, 0x905f,
2401 0x9060, 0x9061, 0x9064, 0x9066, 0x9067, 0x9069, 0x906a, 0x906b,
2402 0x906c, 0x906f, 0x9070, 0x9071, 0x9072, 0x9073, 0x9076, 0x9077,
2403 0x9078, 0x9079, 0x907a, 0x907b, 0x907c, 0x907e, 0x9081, 0x9084,
2404 0x9085, 0x9086, 0x9087, 0x9089, 0x908a, 0x908c, 0x908d, 0x908e,
2405 0x908f, 0x9090, 0x9092, 0x9094, 0x9096, 0x9098, 0x909a, 0x909c,
2406 0x909e, 0x909f, 0x90a0, 0x90a4, 0x90a5, 0x90a7, 0x90a8, 0x90a9,
2407 0x90ab, 0x90ad, 0x90b2, 0x90b7, 0x90bc, 0x90bd, 0x90bf, 0x90c0,
2408 0x647a, 0x64b7, 0x64b8, 0x6499, 0x64ba, 0x64c0, 0x64d0, 0x64d7,
2409 0x64e4, 0x64e2, 0x6509, 0x6525, 0x652e, 0x5f0b, 0x5fd2, 0x7519,
2410 0x5f11, 0x535f, 0x53f1, 0x53fd, 0x53e9, 0x53e8, 0x53fb, 0x5412,
2411 0x5416, 0x5406, 0x544b, 0x5452, 0x5453, 0x5454, 0x5456, 0x5443,
2412 0x5421, 0x5457, 0x5459, 0x5423, 0x5432, 0x5482, 0x5494, 0x5477,
2413 0x5471, 0x5464, 0x549a, 0x549b, 0x5484, 0x5476, 0x5466, 0x549d,
2414 0x54d0, 0x54ad, 0x54c2, 0x54b4, 0x54d2, 0x54a7, 0x54a6, 0x54d3,
2415 0x54d4, 0x5472, 0x54a3, 0x54d5, 0x54bb, 0x54bf, 0x54cc, 0x54d9,
2416 0x54da, 0x54dc, 0x54a9, 0x54aa, 0x54a4, 0x54dd, 0x54cf, 0x54de,
2417 0x551b, 0x54e7, 0x5520, 0x54fd, 0x5514, 0x54f3, 0x5522, 0x5523,
2418 0x550f, 0x5511, 0x5527, 0x552a, 0x5567, 0x558f, 0x55b5, 0x5549,
2419 0x556d, 0x5541, 0x5555, 0x553f, 0x5550, 0x553c,
2420 /* 0xe0 */
2421 0x90c2, 0x90c3, 0x90c6, 0x90c8, 0x90c9, 0x90cb, 0x90cc, 0x90cd,
2422 0x90d2, 0x90d4, 0x90d5, 0x90d6, 0x90d8, 0x90d9, 0x90da, 0x90de,
2423 0x90df, 0x90e0, 0x90e3, 0x90e4, 0x90e5, 0x90e9, 0x90ea, 0x90ec,
2424 0x90ee, 0x90f0, 0x90f1, 0x90f2, 0x90f3, 0x90f5, 0x90f6, 0x90f7,
2425 0x90f9, 0x90fa, 0x90fb, 0x90fc, 0x90ff, 0x9100, 0x9101, 0x9103,
2426 0x9105, 0x9106, 0x9107, 0x9108, 0x9109, 0x910a, 0x910b, 0x910c,
2427 0x910d, 0x910e, 0x910f, 0x9110, 0x9111, 0x9112, 0x9113, 0x9114,
2428 0x9115, 0x9116, 0x9117, 0x9118, 0x911a, 0x911b, 0x911c, 0x911d,
2429 0x911f, 0x9120, 0x9121, 0x9124, 0x9125, 0x9126, 0x9127, 0x9128,
2430 0x9129, 0x912a, 0x912b, 0x912c, 0x912d, 0x912e, 0x9130, 0x9132,
2431 0x9133, 0x9134, 0x9135, 0x9136, 0x9137, 0x9138, 0x913a, 0x913b,
2432 0x913c, 0x913d, 0x913e, 0x913f, 0x9140, 0x9141, 0x9142, 0x9144,
2433 0x5537, 0x5556, 0x5575, 0x5576, 0x5577, 0x5533, 0x5530, 0x555c,
2434 0x558b, 0x55d2, 0x5583, 0x55b1, 0x55b9, 0x5588, 0x5581, 0x559f,

```

```
2435 0x557e, 0x55d6, 0x5591, 0x557b, 0x55df, 0x55bd, 0x55be, 0x5594,
2436 0x5599, 0x55ea, 0x55f7, 0x55c9, 0x561f, 0x55d1, 0x55eb, 0x55ec,
2437 0x55d4, 0x55e6, 0x55dd, 0x55c4, 0x55ef, 0x55e5, 0x55f2, 0x55f3,
2438 0x55cc, 0x55cd, 0x55e8, 0x55f5, 0x55e4, 0x8f94, 0x561e, 0x5608,
2439 0x560c, 0x5601, 0x5624, 0x5623, 0x55fe, 0x5600, 0x5627, 0x562d,
2440 0x5658, 0x5639, 0x5657, 0x562c, 0x564d, 0x5662, 0x5659, 0x565c,
2441 0x564c, 0x5654, 0x5686, 0x5664, 0x5671, 0x566b, 0x567b, 0x567c,
2442 0x5685, 0x5693, 0x56af, 0x56d4, 0x56d7, 0x56dd, 0x56e1, 0x56f5,
2443 0x56eb, 0x56f9, 0x56ff, 0x5704, 0x570a, 0x5709, 0x571c, 0x5e0f,
2444 0x5e19, 0x5e14, 0x5e11, 0x5e31, 0x5e3b, 0x5e3c,
2445 /* 0xe1 */
2446 0x9145, 0x9147, 0x9148, 0x9151, 0x9153, 0x9154, 0x9155, 0x9156,
2447 0x9158, 0x9159, 0x915b, 0x915c, 0x915f, 0x9160, 0x9166, 0x9167,
2448 0x9168, 0x916b, 0x916d, 0x9173, 0x917a, 0x917b, 0x917c, 0x9180,
2449 0x9181, 0x9182, 0x9183, 0x9184, 0x9186, 0x9188, 0x918a, 0x918e,
2450 0x918f, 0x9193, 0x9194, 0x9195, 0x9196, 0x9197, 0x9198, 0x9199,
2451 0x919c, 0x919d, 0x919e, 0x919f, 0x91a0, 0x91a1, 0x91a4, 0x91a5,
2452 0x91a6, 0x91a7, 0x91a8, 0x91a9, 0x91ab, 0x91ac, 0x91b0, 0x91b1,
2453 0x91b2, 0x91b3, 0x91b6, 0x91b7, 0x91b8, 0x91b9, 0x91bb, 0x91bc,
2454 0x91bd, 0x91be, 0x91bf, 0x91c0, 0x91c1, 0x91c2, 0x91c3, 0x91c4,
2455 0x91c5, 0x91c6, 0x91c8, 0x91cb, 0x91d0, 0x91d2, 0x91d3, 0x91d4,
2456 0x91d5, 0x91d6, 0x91d7, 0x91d8, 0x91d9, 0x91da, 0x91db, 0x91dd,
2457 0x91de, 0x91df, 0x91e0, 0x91e1, 0x91e2, 0x91e3, 0x91e4, 0x91e5,
2458 0x5e37, 0x5e44, 0x5e54, 0x5e5b, 0x5e5e, 0x5e61, 0x5e8c, 0x5c7a,
2459 0x5c8d, 0x5c90, 0x5c96, 0x5c88, 0x5c98, 0x5c99, 0x5c91, 0x5c9a,
2460 0x5c9c, 0x5cb5, 0x5ca2, 0x5cbd, 0x5cac, 0x5cab, 0x5cb1, 0x5ca3,
2461 0x5cc1, 0x5cb7, 0x5cc4, 0x5cd2, 0x5ce4, 0x5ccb, 0x5ce5, 0x5d02,
2462 0x5d03, 0x5d27, 0x5d26, 0x5d2e, 0x5d24, 0x5d1e, 0x5d06, 0x5d1b,
2463 0x5d58, 0x5d3e, 0x5d34, 0x5d3d, 0x5d6c, 0x5d5b, 0x5d6f, 0x5d5d,
2464 0x5d6b, 0x5d4b, 0x5d4a, 0x5d69, 0x5d74, 0x5d82, 0x5d99, 0x5d9d,
2465 0x8c73, 0x5db7, 0x5dc5, 0x5f73, 0x5f77, 0x5f82, 0x5f87, 0x5f89,
2466 0x5f8c, 0x5f95, 0x5f99, 0x5f9c, 0x5fa8, 0x5fad, 0x5fb5, 0x5fbc,
2467 0x8862, 0x5f61, 0x72ad, 0x72b0, 0x72b4, 0x72b7, 0x72b8, 0x72c3,
2468 0x72c1, 0x72ce, 0x72cd, 0x72d2, 0x72e8, 0x72ef, 0x72e9, 0x72f2,
2469 0x72f4, 0x72f7, 0x7301, 0x72f3, 0x7303, 0x72fa,
2470 /* 0xe2 */
2471 0x91e6, 0x91e7, 0x91e8, 0x91e9, 0x91ea, 0x91eb, 0x91ec, 0x91ed,
2472 0x91ee, 0x91ef, 0x91f0, 0x91f1, 0x91f2, 0x91f3, 0x91f4, 0x91f5,
2473 0x91f6, 0x91f7, 0x91f8, 0x91f9, 0x91fa, 0x91fb, 0x91fc, 0x91fd,
2474 0x91fe, 0x91ff, 0x9200, 0x9201, 0x9202, 0x9203, 0x9204, 0x9205,
2475 0x9206, 0x9207, 0x9208, 0x9209, 0x920a, 0x920b, 0x920c, 0x920d,
2476 0x920e, 0x920f, 0x9210, 0x9211, 0x9212, 0x9213, 0x9214, 0x9215,
2477 0x9216, 0x9217, 0x9218, 0x9219, 0x921a, 0x921b, 0x921c, 0x921d,
2478 0x921e, 0x921f, 0x9220, 0x9221, 0x9222, 0x9223, 0x9224, 0x9225,
2479 0x9226, 0x9227, 0x9228, 0x9229, 0x922a, 0x922b, 0x922c, 0x922d,
2480 0x922e, 0x922f, 0x9230, 0x9231, 0x9232, 0x9233, 0x9234, 0x9235,
2481 0x9236, 0x9237, 0x9238, 0x9239, 0x923a, 0x923b, 0x923c, 0x923d,
2482 0x923e, 0x923f, 0x9240, 0x9241, 0x9242, 0x9243, 0x9244, 0x9245,
2483 0x72fb, 0x7317, 0x7313, 0x7321, 0x730a, 0x731e, 0x731d, 0x7315,
2484 0x7322, 0x7339, 0x7325, 0x732c, 0x7338, 0x7331, 0x7350, 0x734d,
2485 0x7357, 0x7360, 0x736c, 0x736f, 0x737e, 0x821b, 0x5925, 0x98e7,
2486 0x5924, 0x5902, 0x9963, 0x9967, 0x9968, 0x9969, 0x996a, 0x996b,
2487 0x996c, 0x9974, 0x9977, 0x997d, 0x9980, 0x9984, 0x9987, 0x998a,
2488 0x998d, 0x9990, 0x9991, 0x9993, 0x9994, 0x9995, 0x5e80, 0x5e91,
2489 0x5e8b, 0x5e96, 0x5ea5, 0x5ea0, 0x5eb9, 0x5eb5, 0x5ebe, 0x5eb3,
2490 0x8d53, 0x5ed2, 0x5ed1, 0x5edb, 0x5ee8, 0x5eea, 0x81ba, 0x5fc4,
2491 0x5fc9, 0x5fd6, 0x5fcf, 0x6003, 0x5fee, 0x6004, 0x5fe1, 0x5fe4,
2492 0x5ffe, 0x6005, 0x6006, 0x5fea, 0x5fed, 0x5ff8, 0x6019, 0x6035,
2493 0x6026, 0x601b, 0x600f, 0x600d, 0x6029, 0x602b, 0x600a, 0x603f,
2494 0x6021, 0x6078, 0x6079, 0x607b, 0x607a,
2495 /* 0xe3 */
2496 0x9246, 0x9247, 0x9248, 0x9249, 0x924a, 0x924b, 0x924c, 0x924d,
2497 0x924e, 0x924f, 0x9250, 0x9251, 0x9252, 0x9253, 0x9254, 0x9255,
2498 0x9256, 0x9257, 0x9258, 0x9259, 0x925a, 0x925b, 0x925c, 0x925d,
2499 0x925e, 0x925f, 0x9260, 0x9261, 0x9262, 0x9263, 0x9264, 0x9265,
2500 0x9266, 0x9267, 0x9268, 0x9269, 0x926a, 0x926b, 0x926c, 0x926d,
2501 0x926e, 0x926f, 0x9270, 0x9271, 0x9272, 0x9273, 0x9275, 0x9276,
2502 0x9277, 0x9278, 0x9279, 0x927a, 0x927b, 0x927c, 0x927d, 0x927e,
2503 0x927f, 0x9280, 0x9281, 0x9282, 0x9283, 0x9284, 0x9285, 0x9286,
2504 0x9287, 0x9288, 0x9289, 0x928a, 0x928b, 0x928c, 0x928d, 0x928f,
2505 0x9290, 0x9291, 0x9292, 0x9293, 0x9294, 0x9295, 0x9296, 0x9297,
2506 0x9298, 0x9299, 0x929a, 0x929b, 0x929c, 0x929d, 0x929e, 0x929f,
2507 0x92a0, 0x92a1, 0x92a2, 0x92a3, 0x92a4, 0x92a5, 0x92a6, 0x92a7,
2508 0x606a, 0x607d, 0x6096, 0x609a, 0x60ad, 0x609d, 0x6083, 0x6092,
2509 0x608c, 0x609b, 0x60ec, 0x60bb, 0x60b1, 0x60dd, 0x60d8, 0x60c6,
2510 0x60da, 0x60b4, 0x6120, 0x6126, 0x6115, 0x6123, 0x60f4, 0x6100,
2511 0x610e, 0x612b, 0x614a, 0x6175, 0x61ac, 0x6194, 0x61a7, 0x61b7,
2512 0x61d4, 0x61f5, 0x5fdd, 0x96b3, 0x95e9, 0x95eb, 0x95f1, 0x95f3,
2513 0x95f5, 0x95f6, 0x95fc, 0x95fe, 0x9603, 0x9604, 0x9606, 0x9608,
2514 0x960a, 0x960b, 0x960c, 0x960d, 0x960f, 0x9612, 0x9615, 0x9616,
2515 0x9617, 0x9619, 0x961a, 0x4e2c, 0x723f, 0x6215, 0x6c35, 0x6c54,
2516 0x6c5c, 0x6c4a, 0x6ca3, 0x6c85, 0x6c90, 0x6c94, 0x6c8c, 0x6c68,
2517 0x6c69, 0x6c74, 0x6c76, 0x6c86, 0x6ca9, 0x6cd0, 0x6cd4, 0x6cad,
2518 0x6cf7, 0x6cf8, 0x6cf1, 0x6cd7, 0x6cb2, 0x6ce0, 0x6cd6, 0x6cfa,
2519 0x6ceb, 0x6cee, 0x6cb1, 0x6cd3, 0x6cef, 0x6cfe,
2520 /* 0xe4 */
2521 0x92a8, 0x92a9, 0x92aa, 0x92ab, 0x92ac, 0x92ad, 0x92af, 0x92b0,
```

```
2522 0x92b1, 0x92b2, 0x92b3, 0x92b4, 0x92b5, 0x92b6, 0x92b7, 0x92b8,
2523 0x92b9, 0x92ba, 0x92bb, 0x92bc, 0x92bd, 0x92be, 0x92bf, 0x92c0,
2524 0x92c1, 0x92c2, 0x92c3, 0x92c4, 0x92c5, 0x92c6, 0x92c7, 0x92c9,
2525 0x92ca, 0x92cb, 0x92cc, 0x92cd, 0x92ce, 0x92cf, 0x92d0, 0x92d1,
2526 0x92d2, 0x92d3, 0x92d4, 0x92d5, 0x92d6, 0x92d7, 0x92d8, 0x92d9,
2527 0x92da, 0x92db, 0x92dc, 0x92dd, 0x92de, 0x92df, 0x92e0, 0x92e1,
2528 0x92e2, 0x92e3, 0x92e4, 0x92e5, 0x92e6, 0x92e7, 0x92e8, 0x92e9,
2529 0x92ea, 0x92eb, 0x92ec, 0x92ed, 0x92ee, 0x92ef, 0x92f0, 0x92f1,
2530 0x92f2, 0x92f3, 0x92f4, 0x92f5, 0x92f6, 0x92f7, 0x92f8, 0x92f9,
2531 0x92fa, 0x92fb, 0x92fc, 0x92fd, 0x92fe, 0x92ff, 0x9300, 0x9301,
2532 0x9302, 0x9303, 0x9304, 0x9305, 0x9306, 0x9307, 0x9308, 0x9309,
2533 0x6d39, 0x6d27, 0x6d0c, 0x6d43, 0x6d48, 0x6d07, 0x6d04, 0x6d19,
2534 0x6d0e, 0x6d2b, 0x6d4d, 0x6d2e, 0x6d35, 0x6d1a, 0x6d4f, 0x6d52,
2535 0x6d54, 0x6d33, 0x6d91, 0x6d6f, 0x6d9e, 0x6da0, 0x6d5e, 0x6d93,
2536 0x6d94, 0x6d5c, 0x6d60, 0x6d7c, 0x6d63, 0x6e1a, 0x6dc7, 0x6dc5,
2537 0x6dde, 0x6e0e, 0x6dbf, 0x6de0, 0x6e11, 0x6de6, 0x6ddd, 0x6dd9,
2538 0x6e16, 0x6dab, 0x6e0c, 0x6dae, 0x6e2b, 0x6e6e, 0x6e4e, 0x6e6b,
2539 0x6eb2, 0x6e5f, 0x6e86, 0x6e53, 0x6e54, 0x6e32, 0x6e25, 0x6e44,
2540 0x6edf, 0x6eb1, 0x6e98, 0x6ee0, 0x6f2d, 0x6ee2, 0x6ea5, 0x6ea7,
2541 0x6ebd, 0x6ebb, 0x6eb7, 0x6ed7, 0x6eb4, 0x6ecf, 0x6e8f, 0x6ec2,
2542 0x6e9f, 0x6f62, 0x6f46, 0x6f46, 0x6f47, 0x6f24, 0x6f15, 0x6ef9, 0x6f2f,
2543 0x6f36, 0x6f4b, 0x6f74, 0x6f2a, 0x6f09, 0x6f29, 0x6f89, 0x6f8d,
2544 0x6f8c, 0x6f78, 0x6f72, 0x6f7c, 0x6f7a, 0x6fd1,
2545 /* 0xe5 */
2546 0x930a, 0x930b, 0x930c, 0x930d, 0x930e, 0x930f, 0x9310, 0x9311,
2547 0x9312, 0x9313, 0x9314, 0x9315, 0x9316, 0x9317, 0x9318, 0x9319,
2548 0x931a, 0x931b, 0x931c, 0x931d, 0x931e, 0x931f, 0x9320, 0x9321,
2549 0x9322, 0x9323, 0x9324, 0x9325, 0x9326, 0x9327, 0x9328, 0x9329,
2550 0x932a, 0x932b, 0x932c, 0x932d, 0x932e, 0x932f, 0x9330, 0x9331,
2551 0x9332, 0x9333, 0x9334, 0x9335, 0x9336, 0x9337, 0x9338, 0x9339,
2552 0x933a, 0x933b, 0x933c, 0x933d, 0x933e, 0x933f, 0x9340, 0x9341, 0x9342,
2553 0x9343, 0x9344, 0x9345, 0x9346, 0x9347, 0x9348, 0x9349, 0x934a,
2554 0x934b, 0x934c, 0x934d, 0x934e, 0x934f, 0x9350, 0x9351, 0x9352,
2555 0x9353, 0x9354, 0x9355, 0x9356, 0x9357, 0x9358, 0x9359, 0x935a,
2556 0x935b, 0x935c, 0x935d, 0x935e, 0x935f, 0x9360, 0x9361, 0x9362,
2557 0x9363, 0x9364, 0x9365, 0x9366, 0x9367, 0x9368, 0x9369, 0x936a,
2558 0x6fc9, 0x6fa7, 0x6fb9, 0x6fb6, 0x6fc2, 0x6fe1, 0x6fee, 0x6fde,
2559 0x6fe0, 0x6fef, 0x701a, 0x7023, 0x701b, 0x7039, 0x7035, 0x704f,
2560 0x705e, 0x5b80, 0x5b84, 0x5b95, 0x5b93, 0x5ba5, 0x5bb8, 0x752f,
2561 0x9a9e, 0x6434, 0x5be4, 0x5bee, 0x8930, 0x5bf0, 0x8e47, 0x8b07,
2562 0x8fb6, 0x8fd3, 0x8fd5, 0x8fe5, 0x8fee, 0x8fe4, 0x8fe9, 0x8fe6,
2563 0x8ff3, 0x8fe8, 0x9005, 0x9004, 0x900b, 0x9026, 0x9011, 0x900d,
2564 0x9016, 0x9021, 0x9035, 0x9036, 0x902d, 0x902f, 0x9044, 0x9051,
2565 0x9052, 0x9050, 0x9068, 0x9058, 0x9062, 0x905b, 0x66b9, 0x9074,
2566 0x907d, 0x9082, 0x9088, 0x9083, 0x908b, 0x5f50, 0x5f57, 0x5f56,
2567 0x5f58, 0x5c3b, 0x54ab, 0x5c50, 0x5c59, 0x5b71, 0x5c63, 0x5c66,
2568 0x7fbc, 0x5f2a, 0x5f29, 0x5f2d, 0x8274, 0x5f3c, 0x9b3b, 0x5c6e,
2569 0x5981, 0x5983, 0x598d, 0x59a9, 0x59aa, 0x59a3,
2570 /* 0xe6 */
2571 0x936c, 0x936d, 0x936e, 0x936f, 0x9370, 0x9371, 0x9372, 0x9373,
2572 0x9374, 0x9375, 0x9376, 0x9377, 0x9378, 0x9379, 0x937a, 0x937b,
2573 0x937c, 0x937d, 0x937e, 0x937f, 0x9380, 0x9381, 0x9382, 0x9383,
2574 0x9384, 0x9385, 0x9386, 0x9387, 0x9388, 0x9389, 0x938a, 0x938b,
2575 0x938c, 0x938d, 0x938e, 0x938f, 0x9390, 0x9391, 0x9392, 0x9393, 0x9394,
2576 0x9395, 0x9396, 0x9397, 0x9398, 0x9399, 0x939a, 0x939b, 0x939c,
2577 0x939d, 0x939e, 0x939f, 0x93a0, 0x93a1, 0x93a2, 0x93a3, 0x93a4,
2578 0x93a5, 0x93a6, 0x93a7, 0x93a8, 0x93a9, 0x93aa, 0x93ab, 0x93ac,
2579 0x93ad, 0x93ae, 0x93af, 0x93b0, 0x93b1, 0x93b2, 0x93b3, 0x93b4,
2580 0x93b5, 0x93b6, 0x93b7, 0x93b8, 0x93b9, 0x93ba, 0x93bb, 0x93bc,
2581 0x93bd, 0x93be, 0x93bf, 0x93c0, 0x93c1, 0x93c2, 0x93c3, 0x93c4,
2582 0x93c5, 0x93c6, 0x93c7, 0x93c8, 0x93c9, 0x93cb, 0x93cc, 0x93cd,
2583 0x5997, 0x59ca, 0x59ab, 0x599e, 0x59a4, 0x59d2, 0x59b2, 0x59af,
2584 0x59d7, 0x59be, 0x5a05, 0x5a06, 0x59dd, 0x5a08, 0x59e3, 0x59d8,
2585 0x59f9, 0x5a0c, 0x5a09, 0x5a32, 0x5a34, 0x5a11, 0x5a23, 0x5a13,
2586 0x5a40, 0x5a67, 0x5a4a, 0x5a55, 0x5a3c, 0x5a62, 0x5a75, 0x80ec,
2587 0x5aaa, 0x5a9b, 0x5a77, 0x5a7a, 0x5abe, 0x5aeb, 0x5ab2, 0x5ad2,
2588 0x5ad4, 0x5ab8, 0x5ae0, 0x5ae3, 0x5af1, 0x5ad6, 0x5ae6, 0x5ad8,
2589 0x5adc, 0x5b09, 0x5b17, 0x5b16, 0x5b32, 0x5b37, 0x5b40, 0x5c15,
2590 0x5c1c, 0x5b5a, 0x5b65, 0x5b73, 0x5b51, 0x5b53, 0x5b62, 0x9a75,
2591 0x9a77, 0x9a78, 0x9a7a, 0x9a7f, 0x9a7d, 0x9a80, 0x9a81, 0x9a85,
2592 0x9a88, 0x9a8a, 0x9a90, 0x9a92, 0x9a93, 0x9a96, 0x9a98, 0x9a9b,
2593 0x9a9c, 0x9a9d, 0x9a9f, 0x9aa0, 0x9aa2, 0x9aa3, 0x9aa5, 0x9aa7,
2594 0x7e9f, 0x7ea1, 0x7ea3, 0x7ea5, 0x7ea8, 0x7ea9,
2595 /* 0xe7 */
2596 0x93ce, 0x93cf, 0x93d0, 0x93d1, 0x93d2, 0x93d3, 0x93d4, 0x93d5,
2597 0x93d7, 0x93d8, 0x93d9, 0x93da, 0x93db, 0x93dc, 0x93dd, 0x93de,
2598 0x93df, 0x93e0, 0x93e1, 0x93e2, 0x93e3, 0x93e4, 0x93e5, 0x93e6,
2599 0x93e7, 0x93e8, 0x93e9, 0x93ea, 0x93eb, 0x93ec, 0x93ed, 0x93ee,
2600 0x93ef, 0x93f0, 0x93f1, 0x93f2, 0x93f3, 0x93f4, 0x93f5, 0x93f6,
2601 0x93f7, 0x93f8, 0x93f9, 0x93fa, 0x93fb, 0x93fc, 0x93fd, 0x93fe,
2602 0x93ff, 0x9400, 0x9401, 0x9402, 0x9403, 0x9404, 0x9405, 0x9406,
2603 0x9407, 0x9408, 0x9409, 0x940a, 0x940b, 0x940c, 0x940d, 0x940e,
2604 0x940f, 0x9410, 0x9411, 0x9412, 0x9413, 0x9414, 0x9415, 0x9416,
2605 0x9417, 0x9418, 0x9419, 0x941a, 0x941b, 0x941c, 0x941d, 0x941e,
2606 0x941f, 0x9420, 0x9421, 0x9422, 0x9423, 0x9424, 0x9425, 0x9426,
2607 0x9427, 0x9428, 0x9429, 0x942a, 0x942b, 0x942c, 0x942d, 0x942e,
2608 0x7ead, 0x7eb0, 0x7ebe, 0x7ec0, 0x7ec1, 0x7ec2, 0x7ec9, 0x7ecb,
```

```

2609 0x7ecc, 0x7ed0, 0x7ed4, 0x7ed7, 0x7edb, 0x7ee0, 0x7ee1, 0x7ee8,
2610 0x7eeb, 0x7eee, 0x7eef, 0x7ef1, 0x7ef2, 0x7ef0d, 0x7ef6, 0x7efa,
2611 0x7efb, 0x7efe, 0x7ef01, 0x7ef02, 0x7ef03, 0x7ef07, 0x7ef08, 0x7f0b,
2612 0x7f0c, 0x7f0f, 0x7f11, 0x7f12, 0x7f17, 0x7f19, 0x7f1c, 0x7f1b,
2613 0x7f1f, 0x7f21, 0x7f22, 0x7f23, 0x7f24, 0x7f25, 0x7f26, 0x7f27,
2614 0x7f2a, 0x7f2b, 0x7f2c, 0x7f2d, 0x7f2f, 0x7f30, 0x7f31, 0x7f32,
2615 0x7f33, 0x7f35, 0x5e7a, 0x757f, 0x5ddb, 0x753e, 0x9095, 0x738e,
2616 0x7391, 0x73ae, 0x73a2, 0x739f, 0x73cf, 0x73c2, 0x73d1, 0x73b7,
2617 0x73b3, 0x73c0, 0x73c9, 0x73c8, 0x73e5, 0x73d9, 0x987c, 0x740a,
2618 0x73e9, 0x73e7, 0x73de, 0x73ba, 0x73f2, 0x740f, 0x742a, 0x745b,
2619 0x7426, 0x7425, 0x7428, 0x7430, 0x742e, 0x742c,
2620 /* 0xe8 */
2621 0x942f, 0x9430, 0x9431, 0x9432, 0x9433, 0x9434, 0x9435, 0x9436,
2622 0x9437, 0x9438, 0x9439, 0x943a, 0x943b, 0x943c, 0x943d, 0x943e,
2623 0x9440, 0x9441, 0x9442, 0x9443, 0x9444, 0x9445, 0x9446, 0x9447,
2624 0x9448, 0x9449, 0x944a, 0x944b, 0x944c, 0x944d, 0x944e, 0x944f,
2625 0x9450, 0x9451, 0x9452, 0x9453, 0x9454, 0x9455, 0x9456, 0x9457,
2626 0x9458, 0x9459, 0x945a, 0x945b, 0x945c, 0x945d, 0x945e, 0x945f,
2627 0x9460, 0x9461, 0x9462, 0x9463, 0x9464, 0x9465, 0x9466, 0x9467,
2628 0x9468, 0x9469, 0x946a, 0x946c, 0x946d, 0x946e, 0x946f, 0x9470,
2629 0x9471, 0x9472, 0x9473, 0x9474, 0x9475, 0x9476, 0x9477, 0x9478,
2630 0x9479, 0x947a, 0x947b, 0x947c, 0x947d, 0x947e, 0x947f, 0x9480,
2631 0x9481, 0x9482, 0x9483, 0x9484, 0x9491, 0x9496, 0x9498, 0x94c7,
2632 0x94cf, 0x94d3, 0x94d4, 0x94da, 0x94e6, 0x94fb, 0x951c, 0x9520,
2633 0x741b, 0x741a, 0x7441, 0x745c, 0x7457, 0x7455, 0x7459, 0x7477,
2634 0x746d, 0x747e, 0x749c, 0x748e, 0x7480, 0x7481, 0x7487, 0x748b,
2635 0x749e, 0x74a8, 0x74a9, 0x74a7, 0x7490, 0x74a7, 0x74d2, 0x74ba, 0x97ea,
2636 0x97eb, 0x97ec, 0x674c, 0x6753, 0x675e, 0x6748, 0x6769, 0x67a5,
2637 0x6787, 0x676a, 0x6773, 0x6798, 0x67a7, 0x6775, 0x67a8, 0x679e,
2638 0x67ad, 0x678b, 0x6777, 0x677c, 0x67f0, 0x6809, 0x67d8, 0x680a,
2639 0x67e9, 0x67b0, 0x680c, 0x67d9, 0x67b5, 0x67da, 0x67b3, 0x67dd,
2640 0x6800, 0x67c3, 0x67b8, 0x67e2, 0x680e, 0x67c1, 0x67fd, 0x6832,
2641 0x6833, 0x6860, 0x6861, 0x684e, 0x6862, 0x6844, 0x6864, 0x6883,
2642 0x681d, 0x6855, 0x6866, 0x6841, 0x6867, 0x6840, 0x683e, 0x684a,
2643 0x6849, 0x6829, 0x68b5, 0x688f, 0x6874, 0x6877, 0x6893, 0x686b,
2644 0x68c2, 0x696e, 0x68fc, 0x691f, 0x6920, 0x68f9,
2645 /* 0xe9 */
2646 0x9527, 0x9533, 0x953d, 0x9543, 0x9548, 0x954b, 0x9555, 0x955a,
2647 0x9560, 0x956e, 0x9574, 0x9575, 0x9577, 0x9578, 0x9579, 0x957a,
2648 0x957b, 0x957c, 0x957d, 0x957e, 0x9580, 0x9581, 0x9582, 0x9583,
2649 0x9584, 0x9585, 0x9586, 0x9587, 0x9588, 0x9589, 0x958a, 0x958b,
2650 0x958c, 0x958d, 0x958e, 0x958f, 0x9590, 0x9591, 0x9592, 0x9593,
2651 0x9594, 0x9595, 0x9596, 0x9597, 0x9598, 0x9599, 0x959a, 0x959b,
2652 0x959c, 0x959d, 0x959e, 0x959f, 0x95a0, 0x95a1, 0x95a2, 0x95a3,
2653 0x95a4, 0x95a5, 0x95a6, 0x95a7, 0x95a8, 0x95a9, 0x95aa, 0x95ab,
2654 0x95ac, 0x95ad, 0x95ae, 0x95af, 0x95b0, 0x95b1, 0x95b2, 0x95b3,
2655 0x95b4, 0x95b5, 0x95b6, 0x95b7, 0x95b8, 0x95b9, 0x95ba, 0x95bb,
2656 0x95bc, 0x95bd, 0x95be, 0x95bf, 0x95c0, 0x95c1, 0x95c2, 0x95c3,
2657 0x95c4, 0x95c5, 0x95c6, 0x95c7, 0x95c8, 0x95c9, 0x95ca, 0x95cb,
2658 0x6924, 0x68f0, 0x690b, 0x6901, 0x6957, 0x68e3, 0x6910, 0x6971,
2659 0x6939, 0x6960, 0x6942, 0x695d, 0x6984, 0x696b, 0x6980, 0x6998,
2660 0x6978, 0x6934, 0x69cc, 0x6987, 0x6988, 0x69ce, 0x6989, 0x6966,
2661 0x6963, 0x6979, 0x699b, 0x69a7, 0x69bb, 0x69ab, 0x69ad, 0x69d4,
2662 0x69b1, 0x69c1, 0x69ca, 0x69df, 0x6995, 0x69e0, 0x698d, 0x69ff,
2663 0x6a2f, 0x69ed, 0x6a17, 0x6a18, 0x6a65, 0x69f2, 0x6a44, 0x6a3e,
2664 0x6aa0, 0x6aa5, 0x6aa5b, 0x6aa35, 0x6aa8e, 0x6a79, 0x6a3d, 0x6a28,
2665 0x6aa8, 0x6aa7c, 0x6aa91, 0x6aa90, 0x6aa9, 0x6aa97, 0x6aab, 0x7337,
2666 0x7352, 0x6b81, 0x6b82, 0x6b87, 0x6b84, 0x6b92, 0x6b93, 0x6b8d,
2667 0x6b9a, 0x6b9b, 0x6ba1, 0x6baa, 0x8f6b, 0x8f6d, 0x8f71, 0x8f72,
2668 0x8f73, 0x8f75, 0x8f76, 0x8f78, 0x8f77, 0x8f79, 0x8f7a, 0x8f7c,
2669 0x8f7e, 0x8f81, 0x8f82, 0x8f84, 0x8f87, 0x8f8b,
2670 /* 0xea */
2671 0x95cc, 0x95cd, 0x95ce, 0x95cf, 0x95d0, 0x95d1, 0x95d2, 0x95d3,
2672 0x95d4, 0x95d5, 0x95d6, 0x95d7, 0x95d8, 0x95d9, 0x95da, 0x95db,
2673 0x95dc, 0x95dd, 0x95de, 0x95df, 0x95e0, 0x95e1, 0x95e2, 0x95e3,
2674 0x95e4, 0x95e5, 0x95e6, 0x95e7, 0x95ec, 0x95ff, 0x9607, 0x9613,
2675 0x9618, 0x961b, 0x961e, 0x9620, 0x9623, 0x9624, 0x9625, 0x9626,
2676 0x9627, 0x9628, 0x9629, 0x962b, 0x962c, 0x962d, 0x962f, 0x9630,
2677 0x9637, 0x9638, 0x9639, 0x963a, 0x963e, 0x9641, 0x9643, 0x964a,
2678 0x964e, 0x964f, 0x9651, 0x9652, 0x9653, 0x9656, 0x9657, 0x9658,
2679 0x9659, 0x965a, 0x965c, 0x965d, 0x965e, 0x9660, 0x9663, 0x9665,
2680 0x9666, 0x966b, 0x966d, 0x966e, 0x966f, 0x9670, 0x9671, 0x9673,
2681 0x9678, 0x9679, 0x967a, 0x967b, 0x967c, 0x967d, 0x967e, 0x967f,
2682 0x9680, 0x9681, 0x9682, 0x9683, 0x9684, 0x9687, 0x9689, 0x968a,
2683 0x8f8d, 0x8f8e, 0x8f8f, 0x8f98, 0x8f9a, 0x8ece, 0x620b, 0x6217,
2684 0x621b, 0x621f, 0x6222, 0x6221, 0x6225, 0x6224, 0x622c, 0x81e7,
2685 0x74ef, 0x74f4, 0x74ff, 0x750f, 0x7511, 0x7513, 0x6534, 0x65ee,
2686 0x65ef, 0x65f0, 0x660a, 0x6619, 0x6772, 0x6603, 0x6615, 0x6600,
2687 0x7085, 0x66f7, 0x661d, 0x6634, 0x6631, 0x6636, 0x6635, 0x8006,
2688 0x665f, 0x6654, 0x6641, 0x664f, 0x6656, 0x6661, 0x6657, 0x6677,
2689 0x6684, 0x668c, 0x66a7, 0x669d, 0x66be, 0x66db, 0x66dc, 0x66e6,
2690 0x66e9, 0x8d32, 0x8d33, 0x8d36, 0x8d3b, 0x8d3d, 0x8d40, 0x8d45,
2691 0x8d46, 0x8d48, 0x8d49, 0x8d47, 0x8d4d, 0x8d55, 0x8d59, 0x89c7,
2692 0x89ca, 0x89cb, 0x89cc, 0x89ce, 0x89cf, 0x89d0, 0x89d1, 0x726e,
2693 0x729f, 0x725d, 0x7266, 0x726f, 0x727e, 0x727f, 0x7284, 0x728b,
2694 0x728d, 0x728f, 0x7292, 0x6308, 0x6332, 0x63b0,
2695 /* 0xeb */

```



```
2696 0x968c, 0x968e, 0x9691, 0x9692, 0x9693, 0x9695, 0x9696, 0x969a,
2697 0x969b, 0x969d, 0x969e, 0x969f, 0x96a0, 0x96a1, 0x96a2, 0x96a3,
2698 0x96a4, 0x96a5, 0x96a6, 0x96a8, 0x96a9, 0x96aa, 0x96ab, 0x96ac,
2699 0x96ad, 0x96ae, 0x96af, 0x96b1, 0x96b2, 0x96b4, 0x96b5, 0x96b7,
2700 0x96b8, 0x96ba, 0x96bb, 0x96bf, 0x96c2, 0x96c3, 0x96c8, 0x96ca,
2701 0x96cb, 0x96d0, 0x96d1, 0x96d3, 0x96d4, 0x96d6, 0x96d7, 0x96d8,
2702 0x96d9, 0x96da, 0x96db, 0x96dc, 0x96dd, 0x96de, 0x96df, 0x96e1,
2703 0x96e2, 0x96e3, 0x96e4, 0x96e5, 0x96e6, 0x96e7, 0x96eb, 0x96ec,
2704 0x96ed, 0x96ee, 0x96f0, 0x96f1, 0x96f2, 0x96f4, 0x96f5, 0x96f8,
2705 0x96fa, 0x96fb, 0x96fc, 0x96fd, 0x96ff, 0x9702, 0x9703, 0x9705,
2706 0x970a, 0x970b, 0x970c, 0x9710, 0x9711, 0x9712, 0x9714, 0x9715,
2707 0x9717, 0x9718, 0x9719, 0x971a, 0x971b, 0x971d, 0x971f, 0x9720,
2708 0x643f, 0x64d8, 0x8004, 0x6bea, 0x6bf3, 0x6bfd, 0x6bf5, 0x6bf9,
2709 0x6c05, 0x6c07, 0x6c06, 0x6c0d, 0x6c15, 0x6c18, 0x6c19, 0x6c1a,
2710 0x6c21, 0x6c29, 0x6c24, 0x6c2a, 0x6c32, 0x6535, 0x6555, 0x656b,
2711 0x724d, 0x7252, 0x7256, 0x7230, 0x8662, 0x5216, 0x809f, 0x809c,
2712 0x8093, 0x80bc, 0x670a, 0x80bd, 0x80b1, 0x80ab, 0x80ad, 0x80b4,
2713 0x80b7, 0x80e7, 0x80e8, 0x80e9, 0x80ea, 0x80db, 0x80c2, 0x80c4,
2714 0x80d9, 0x80cd, 0x80d7, 0x6710, 0x80dd, 0x80eb, 0x80f1, 0x80f4,
2715 0x80ed, 0x810d, 0x810e, 0x80f2, 0x80fc, 0x6715, 0x8112, 0x8c5a,
2716 0x8136, 0x811e, 0x8118, 0x812c, 0x8118, 0x8132, 0x8148, 0x814c, 0x8153,
2717 0x8174, 0x8159, 0x815a, 0x8171, 0x8160, 0x8169, 0x817c, 0x817d,
2718 0x816d, 0x8167, 0x584d, 0x5ab5, 0x8188, 0x8182, 0x8191, 0x6ed5,
2719 0x81a3, 0x81aa, 0x81cc, 0x6726, 0x81ca, 0x81bb,
2720 /* 0xec */
2721 0x9721, 0x9722, 0x9723, 0x9724, 0x9725, 0x9726, 0x9727, 0x9728,
2722 0x9729, 0x972b, 0x972c, 0x972e, 0x972f, 0x9731, 0x9733, 0x9734,
2723 0x9735, 0x9736, 0x9737, 0x973a, 0x973b, 0x973c, 0x973d, 0x973f,
2724 0x9740, 0x9741, 0x9742, 0x9743, 0x9744, 0x9745, 0x9746, 0x9747,
2725 0x9748, 0x9749, 0x974a, 0x974b, 0x974c, 0x974d, 0x974e, 0x974f,
2726 0x9750, 0x9751, 0x9754, 0x9755, 0x9757, 0x9758, 0x975a, 0x975c,
2727 0x975d, 0x975f, 0x9763, 0x9764, 0x9766, 0x9767, 0x9768, 0x976a,
2728 0x976b, 0x976c, 0x976d, 0x976e, 0x976f, 0x9770, 0x9771, 0x9772,
2729 0x9775, 0x9777, 0x9778, 0x9779, 0x977a, 0x977b, 0x977d, 0x977e,
2730 0x977f, 0x9780, 0x9781, 0x9782, 0x9783, 0x9784, 0x9786, 0x9787,
2731 0x9788, 0x9789, 0x978a, 0x978c, 0x978e, 0x978f, 0x9790, 0x9793,
2732 0x9795, 0x9796, 0x9797, 0x9799, 0x979a, 0x979b, 0x979c, 0x979d,
2733 0x81c1, 0x81a6, 0x6b24, 0x6b37, 0x6b39, 0x6b43, 0x6b46, 0x6b59,
2734 0x98d1, 0x98d2, 0x98d3, 0x98d5, 0x98d9, 0x98da, 0x6bb3, 0x5f40,
2735 0x6bc2, 0x89f3, 0x6590, 0x9f51, 0x6593, 0x65bc, 0x65c6, 0x65c4,
2736 0x65c3, 0x65cc, 0x65ce, 0x65d2, 0x65d6, 0x7080, 0x709c, 0x709e,
2737 0x709d, 0x70bb, 0x70c0, 0x70b7, 0x70ab, 0x70b1, 0x70e8, 0x70ca,
2738 0x7110, 0x7113, 0x7116, 0x712f, 0x7131, 0x7173, 0x715c, 0x7168,
2739 0x7145, 0x7172, 0x714a, 0x7178, 0x717a, 0x7198, 0x71b3, 0x71b5,
2740 0x71a8, 0x71a0, 0x71e0, 0x71d4, 0x71e7, 0x71f9, 0x721d, 0x7228,
2741 0x706c, 0x7118, 0x7166, 0x71b9, 0x623e, 0x623d, 0x6243, 0x6248,
2742 0x6249, 0x793b, 0x7940, 0x7946, 0x7949, 0x795b, 0x795c, 0x7953,
2743 0x795a, 0x7962, 0x7957, 0x7960, 0x796f, 0x7967, 0x797a, 0x7985,
2744 0x798a, 0x799a, 0x79a7, 0x79b3, 0x5fd1, 0x5fd0,
2745 /* 0xed */
2746 0x979e, 0x979f, 0x97a1, 0x97a2, 0x97a4, 0x97a5, 0x97a6, 0x97a7,
2747 0x97a8, 0x97a9, 0x97aa, 0x97ac, 0x97ae, 0x97b0, 0x97b1, 0x97b3,
2748 0x97b5, 0x97b6, 0x97b7, 0x97b8, 0x97b9, 0x97ba, 0x97bb, 0x97bc,
2749 0x97bd, 0x97be, 0x97bf, 0x97c0, 0x97c1, 0x97c2, 0x97c3, 0x97c4,
2750 0x97c5, 0x97c6, 0x97c7, 0x97c8, 0x97c9, 0x97ca, 0x97cb, 0x97cc,
2751 0x97cd, 0x97ce, 0x97cf, 0x97d0, 0x97d1, 0x97d2, 0x97d3, 0x97d4,
2752 0x97d5, 0x97d6, 0x97d7, 0x97d8, 0x97d9, 0x97da, 0x97db, 0x97dc,
2753 0x97dd, 0x97de, 0x97df, 0x97e0, 0x97e1, 0x97e2, 0x97e3, 0x97e4,
2754 0x97e5, 0x97e8, 0x97ee, 0x97ef, 0x97f0, 0x97f1, 0x97f2, 0x97f4,
2755 0x97f7, 0x97f8, 0x97f9, 0x97fa, 0x97fb, 0x97fc, 0x97fd, 0x97fe,
2756 0x97ff, 0x9800, 0x9801, 0x9802, 0x9803, 0x9804, 0x9805, 0x9806,
2757 0x9807, 0x9808, 0x9809, 0x980a, 0x980b, 0x980c, 0x980d, 0x980e,
2758 0x603c, 0x605d, 0x605a, 0x6067, 0x6041, 0x6059, 0x6063, 0x60ab,
2759 0x6106, 0x610d, 0x615d, 0x61a9, 0x619d, 0x61cb, 0x61d1, 0x6206,
2760 0x8080, 0x807f, 0x6c93, 0x6cf6, 0x6dfc, 0x77f6, 0x77f8, 0x7800,
2761 0x7809, 0x7817, 0x7818, 0x7811, 0x65ab, 0x782d, 0x781c, 0x781d,
2762 0x7839, 0x783a, 0x783b, 0x781f, 0x783c, 0x7825, 0x782c, 0x7823,
2763 0x7829, 0x784e, 0x786d, 0x7856, 0x7857, 0x7826, 0x7850, 0x7847,
2764 0x784c, 0x786a, 0x789b, 0x7893, 0x789a, 0x7887, 0x789c, 0x78a1,
2765 0x78a3, 0x78b2, 0x78b9, 0x78a5, 0x78d4, 0x78d9, 0x78c9, 0x78ec,
2766 0x78f2, 0x7905, 0x78f4, 0x7913, 0x7924, 0x791e, 0x7934, 0x9f9b,
2767 0x9ef9, 0x9efb, 0x9efc, 0x76f1, 0x7704, 0x770d, 0x76f9, 0x7707,
2768 0x7708, 0x771a, 0x7722, 0x7719, 0x772d, 0x7726, 0x7735, 0x7738,
2769 0x7750, 0x7751, 0x7747, 0x7743, 0x775a, 0x7768,
2770 /* 0xee */
2771 0x980f, 0x9810, 0x9811, 0x9812, 0x9813, 0x9814, 0x9815, 0x9816,
2772 0x9817, 0x9818, 0x9819, 0x981a, 0x981b, 0x981c, 0x981d, 0x981e,
2773 0x981f, 0x9820, 0x9821, 0x9822, 0x9823, 0x9824, 0x9825, 0x9826,
2774 0x9827, 0x9828, 0x9829, 0x982a, 0x982b, 0x982c, 0x982d, 0x982e,
2775 0x982f, 0x9830, 0x9831, 0x9832, 0x9833, 0x9834, 0x9835, 0x9836,
2776 0x9837, 0x9838, 0x9839, 0x983a, 0x983b, 0x983c, 0x983d, 0x983e,
2777 0x983f, 0x9840, 0x9841, 0x9842, 0x9843, 0x9844, 0x9845, 0x9846,
2778 0x9847, 0x9848, 0x9849, 0x984a, 0x984b, 0x984c, 0x984d, 0x984e,
2779 0x984f, 0x9850, 0x9851, 0x9852, 0x9853, 0x9854, 0x9855, 0x9856,
2780 0x9857, 0x9858, 0x9859, 0x985a, 0x985b, 0x985c, 0x985d, 0x985e,
2781 0x985f, 0x9860, 0x9861, 0x9862, 0x9863, 0x9864, 0x9865, 0x9866,
2782 0x9867, 0x9868, 0x9869, 0x986a, 0x986b, 0x986c, 0x986d, 0x986e,
```

```
2783 0x7762, 0x7765, 0x777f, 0x778d, 0x777d, 0x7780, 0x778c, 0x7791,
2784 0x779f, 0x77a0, 0x77b0, 0x77b5, 0x77bd, 0x753a, 0x7540, 0x754e,
2785 0x754b, 0x7548, 0x755b, 0x755b, 0x7572, 0x7579, 0x7583, 0x75f8, 0x7f61,
2786 0x7f5f, 0x8a48, 0x7f68, 0x7f74, 0x7f71, 0x7f79, 0x7f81, 0x7f7e,
2787 0x76cd, 0x76e5, 0x8832, 0x9485, 0x9486, 0x9487, 0x948b, 0x948a,
2788 0x948c, 0x948d, 0x948f, 0x9490, 0x9494, 0x9497, 0x9495, 0x949a,
2789 0x949b, 0x949c, 0x94a3, 0x94a4, 0x94ab, 0x94aa, 0x94ad, 0x94ac,
2790 0x94af, 0x94b0, 0x94b2, 0x94b4, 0x94b6, 0x94b7, 0x94b8, 0x94b9,
2791 0x94ba, 0x94bc, 0x94bd, 0x94bf, 0x94c4, 0x94c8, 0x94c9, 0x94ca,
2792 0x94cb, 0x94cc, 0x94cd, 0x94ce, 0x94d0, 0x94d1, 0x94d2, 0x94d5,
2793 0x94d6, 0x94d7, 0x94d9, 0x94d8, 0x94db, 0x94de, 0x94df, 0x94e0,
2794 0x94e2, 0x94e4, 0x94e5, 0x94e7, 0x94e8, 0x94ea,
2795 /* 0xef */
2796 0x986f, 0x9870, 0x9871, 0x9872, 0x9873, 0x9874, 0x988b, 0x988e,
2797 0x9892, 0x9895, 0x9899, 0x98a3, 0x98a8, 0x98a9, 0x98aa, 0x98ab,
2798 0x98ac, 0x98ad, 0x98ae, 0x98af, 0x98b0, 0x98b1, 0x98b2, 0x98b3,
2799 0x98b4, 0x98b5, 0x98b6, 0x98b7, 0x98b8, 0x98b9, 0x98ba, 0x98bb,
2800 0x98bc, 0x98bd, 0x98be, 0x98bf, 0x98c0, 0x98c1, 0x98c2, 0x98c3,
2801 0x98c4, 0x98c5, 0x98c6, 0x98c7, 0x98c8, 0x98c9, 0x98ca, 0x98cb,
2802 0x98cc, 0x98cd, 0x98cf, 0x98d0, 0x98d4, 0x98d6, 0x98d7, 0x98db,
2803 0x98dc, 0x98dd, 0x98de, 0x98e0, 0x98e1, 0x98e2, 0x98e3, 0x98e4, 0x98e5,
2804 0x98e6, 0x98e9, 0x98ea, 0x98eb, 0x98ec, 0x98ed, 0x98ee, 0x98ef,
2805 0x98f0, 0x98f1, 0x98f2, 0x98f3, 0x98f4, 0x98f5, 0x98f6, 0x98f7,
2806 0x98f8, 0x98fd, 0x98fa, 0x98fb, 0x98fc, 0x98fd, 0x98fe, 0x98ff,
2807 0x9900, 0x9901, 0x9902, 0x9903, 0x9904, 0x9905, 0x9906, 0x9907,
2808 0x94e9, 0x94eb, 0x94ee, 0x94ef, 0x94f3, 0x94f4, 0x94f5, 0x94f7,
2809 0x94f9, 0x94fc, 0x94fd, 0x94ff, 0x9503, 0x9502, 0x9506, 0x9507,
2810 0x9509, 0x950a, 0x950d, 0x950e, 0x950f, 0x9512, 0x9513, 0x9514,
2811 0x9515, 0x9516, 0x9518, 0x951b, 0x951d, 0x951e, 0x951f, 0x9522,
2812 0x952a, 0x952b, 0x9529, 0x9529, 0x952c, 0x9531, 0x9532, 0x9534, 0x9536,
2813 0x9537, 0x9538, 0x953c, 0x953e, 0x953f, 0x9542, 0x9535, 0x9544,
2814 0x9545, 0x9546, 0x9549, 0x954c, 0x954e, 0x954f, 0x9552, 0x9553,
2815 0x9554, 0x9556, 0x9557, 0x9558, 0x9559, 0x955b, 0x955e, 0x955f,
2816 0x955d, 0x9561, 0x9562, 0x9564, 0x9565, 0x9566, 0x9567, 0x9568,
2817 0x9569, 0x956a, 0x956b, 0x956c, 0x956f, 0x9571, 0x9572, 0x9573,
2818 0x953a, 0x77e7, 0x77ec, 0x96c9, 0x79d5, 0x79ed, 0x79e3, 0x79eb,
2819 0x7a06, 0x5d47, 0x7a03, 0x7a02, 0x7a1e, 0x7a14,
2820 /* 0xf0 */
2821 0x9908, 0x9909, 0x990a, 0x990b, 0x990c, 0x990e, 0x990f, 0x9911,
2822 0x9912, 0x9913, 0x9914, 0x9915, 0x9916, 0x9917, 0x9918, 0x9919,
2823 0x991a, 0x991b, 0x991c, 0x991d, 0x991e, 0x991f, 0x9920, 0x9921,
2824 0x9922, 0x9923, 0x9924, 0x9925, 0x9926, 0x9927, 0x9928, 0x9929,
2825 0x992a, 0x992b, 0x992c, 0x992d, 0x992f, 0x9930, 0x9931, 0x9932,
2826 0x9933, 0x9934, 0x9935, 0x9936, 0x9937, 0x9938, 0x9939, 0x993a,
2827 0x993b, 0x993c, 0x993d, 0x993e, 0x993f, 0x9940, 0x9941, 0x9942,
2828 0x9943, 0x9944, 0x9945, 0x9946, 0x9947, 0x9948, 0x9949, 0x994a,
2829 0x994b, 0x994c, 0x994d, 0x994e, 0x994f, 0x9950, 0x9951, 0x9952,
2830 0x9953, 0x9956, 0x9957, 0x9958, 0x9959, 0x995a, 0x995b, 0x995c,
2831 0x995d, 0x995e, 0x995f, 0x9960, 0x9961, 0x9962, 0x9964, 0x9966,
2832 0x9973, 0x9978, 0x9979, 0x997b, 0x997e, 0x9982, 0x9983, 0x9989,
2833 0x7a39, 0x7a37, 0x7a51, 0x9ecf, 0x99a5, 0x7a70, 0x7688, 0x768e,
2834 0x7693, 0x7699, 0x76a4, 0x74de, 0x74e0, 0x752c, 0x9e20, 0x9e22,
2835 0x9e28, 0x9e29, 0x9e2a, 0x9e2b, 0x9e2c, 0x9e32, 0x9e31, 0x9e36,
2836 0x9e38, 0x9e37, 0x9e39, 0x9e3a, 0x9e3e, 0x9e41, 0x9e42, 0x9e44,
2837 0x9e46, 0x9e47, 0x9e48, 0x9e49, 0x9e4b, 0x9e4c, 0x9e4e, 0x9e51,
2838 0x9e55, 0x9e57, 0x9e5a, 0x9e5b, 0x9e5c, 0x9e5e, 0x9e63, 0x9e66,
2839 0x9e67, 0x9e68, 0x9e69, 0x9e6a, 0x9e6b, 0x9e6c, 0x9e71, 0x9e6d,
2840 0x9e73, 0x7592, 0x7594, 0x7596, 0x75a0, 0x759d, 0x75ac, 0x75a3,
2841 0x75b3, 0x75b4, 0x75b8, 0x75c4, 0x75b1, 0x75b0, 0x75c3, 0x75c2,
2842 0x75d6, 0x75cd, 0x75e3, 0x75e8, 0x75e6, 0x75e4, 0x75eb, 0x75e7,
2843 0x7603, 0x75f1, 0x75fc, 0x75ff, 0x7610, 0x7600, 0x7605, 0x760c,
2844 0x7617, 0x760a, 0x7625, 0x7618, 0x7615, 0x7619,
2845 /* 0xf1 */
2846 0x998c, 0x998e, 0x999a, 0x999b, 0x999c, 0x999d, 0x999e, 0x999f,
2847 0x99a0, 0x99a1, 0x99a2, 0x99a3, 0x99a4, 0x99a6, 0x99a7, 0x99a9,
2848 0x99aa, 0x99ab, 0x99ac, 0x99ad, 0x99ae, 0x99af, 0x99b0, 0x99b1,
2849 0x99b2, 0x99b3, 0x99b4, 0x99b5, 0x99b6, 0x99b7, 0x99b8, 0x99b9,
2850 0x99ba, 0x99bb, 0x99bc, 0x99bd, 0x99be, 0x99bf, 0x99c0, 0x99c1,
2851 0x99c2, 0x99c3, 0x99c4, 0x99c5, 0x99c6, 0x99c7, 0x99c8, 0x99c9,
2852 0x99ca, 0x99cb, 0x99cc, 0x99cd, 0x99ce, 0x99cf, 0x99d0, 0x99d1,
2853 0x99d2, 0x99d3, 0x99d4, 0x99d5, 0x99d6, 0x99d7, 0x99d8, 0x99d9,
2854 0x99da, 0x99db, 0x99dc, 0x99dd, 0x99de, 0x99df, 0x99e0, 0x99e1,
2855 0x99e2, 0x99e3, 0x99e4, 0x99e5, 0x99e6, 0x99e7, 0x99e8, 0x99e9,
2856 0x99ea, 0x99eb, 0x99ec, 0x99ed, 0x99ee, 0x99ef, 0x99f0, 0x99f1,
2857 0x99f2, 0x99f3, 0x99f4, 0x99f5, 0x99f6, 0x99f7, 0x99f8, 0x99f9,
2858 0x761b, 0x763c, 0x7622, 0x7620, 0x7640, 0x762d, 0x7630, 0x763f,
2859 0x7635, 0x7643, 0x763e, 0x7633, 0x764d, 0x765e, 0x7654, 0x765c,
2860 0x7656, 0x766b, 0x766f, 0x7fca, 0x7ae6, 0x7a78, 0x7a79, 0x7a80,
2861 0x7a86, 0x7a88, 0x7a95, 0x7aa6, 0x7aa0, 0x7aac, 0x7aa8, 0x7aad,
2862 0x7ab3, 0x8864, 0x8869, 0x8872, 0x887d, 0x887f, 0x8882, 0x88a2,
2863 0x88c6, 0x88b7, 0x88bc, 0x88c9, 0x88e2, 0x88ce, 0x88e3, 0x88e5,
2864 0x88f1, 0x891a, 0x88fc, 0x88e8, 0x88fe, 0x88f0, 0x8921, 0x8919,
2865 0x8913, 0x891b, 0x890a, 0x8934, 0x892b, 0x8936, 0x8941, 0x8966,
2866 0x897b, 0x758b, 0x80e5, 0x76b2, 0x76b4, 0x77dc, 0x8012, 0x8014,
2867 0x8016, 0x801c, 0x8020, 0x8022, 0x8025, 0x8026, 0x8027, 0x8029,
2868 0x8028, 0x8031, 0x800b, 0x8035, 0x8043, 0x8046, 0x804d, 0x8052,
2869 0x8069, 0x8071, 0x8983, 0x9878, 0x9880, 0x9883,
```

```
2870 /* 0xf2 */
2871 0x99fa, 0x99fb, 0x99fc, 0x99fd, 0x99fe, 0x99ff, 0x9a00, 0x9a01,
2872 0x9a02, 0x9a03, 0x9a04, 0x9a05, 0x9a06, 0x9a07, 0x9a08, 0x9a09,
2873 0x9a0a, 0x9a0b, 0x9a0c, 0x9a0d, 0x9a0e, 0x9a0f, 0x9a10, 0x9a11,
2874 0x9a12, 0x9a13, 0x9a14, 0x9a15, 0x9a16, 0x9a17, 0x9a18, 0x9a19,
2875 0x9a1a, 0x9a1b, 0x9a1c, 0x9a1d, 0x9a1e, 0x9a1f, 0x9a20, 0x9a21,
2876 0x9a22, 0x9a23, 0x9a24, 0x9a25, 0x9a26, 0x9a27, 0x9a28, 0x9a29,
2877 0x9a2a, 0x9a2b, 0x9a2c, 0x9a2d, 0x9a2e, 0x9a2f, 0x9a30, 0x9a31,
2878 0x9a32, 0x9a33, 0x9a34, 0x9a35, 0x9a36, 0x9a37, 0x9a38, 0x9a39,
2879 0x9a3a, 0x9a3b, 0x9a3c, 0x9a3d, 0x9a3e, 0x9a3f, 0x9a40, 0x9a41,
2880 0x9a42, 0x9a43, 0x9a44, 0x9a45, 0x9a46, 0x9a47, 0x9a48, 0x9a49,
2881 0x9a4a, 0x9a4b, 0x9a4c, 0x9a4d, 0x9a4e, 0x9a4f, 0x9a50, 0x9a51,
2882 0x9a52, 0x9a53, 0x9a54, 0x9a55, 0x9a56, 0x9a57, 0x9a58, 0x9a59,
2883 0x9889, 0x988c, 0x988d, 0x988f, 0x9894, 0x989a, 0x989b, 0x989e,
2884 0x989f, 0x98a1, 0x98a2, 0x98a5, 0x98a6, 0x864d, 0x8654, 0x866c,
2885 0x866e, 0x867f, 0x867a, 0x867c, 0x867b, 0x86a8, 0x868d, 0x868b,
2886 0x86ac, 0x869d, 0x86a7, 0x86a3, 0x86aa, 0x8693, 0x86a9, 0x86b6,
2887 0x86c4, 0x86b5, 0x86ce, 0x86b0, 0x86ba, 0x86b1, 0x86af, 0x86c9,
2888 0x86cf, 0x86b4, 0x86e9, 0x86f1, 0x86f2, 0x86ed, 0x86f3, 0x86d0,
2889 0x8713, 0x86de, 0x86f4, 0x86df, 0x86d8, 0x86d1, 0x8703, 0x8707,
2890 0x86f8, 0x8708, 0x870a, 0x870d, 0x8709, 0x8723, 0x873b, 0x871e,
2891 0x8725, 0x872e, 0x871a, 0x873e, 0x8748, 0x8734, 0x8731, 0x8729,
2892 0x8737, 0x873f, 0x8782, 0x8722, 0x877d, 0x877e, 0x877b, 0x8760,
2893 0x8770, 0x874c, 0x876e, 0x878b, 0x8753, 0x8763, 0x877c, 0x8764,
2894 0x8759, 0x8765, 0x8793, 0x87af, 0x87a8, 0x87d2,
2895 /* 0xf3 */
2896 0x9a5a, 0x9a5b, 0x9a5c, 0x9a5d, 0x9a5e, 0x9a5f, 0x9a60, 0x9a61,
2897 0x9a62, 0x9a63, 0x9a64, 0x9a65, 0x9a66, 0x9a67, 0x9a68, 0x9a69,
2898 0x9a6a, 0x9a6b, 0x9a72, 0x9a83, 0x9a89, 0x9a8d, 0x9a8e, 0x9a94,
2899 0x9a95, 0x9a99, 0x9aa6, 0x9aa9, 0x9aaa, 0x9aab, 0x9aac, 0x9aad,
2900 0x9aae, 0x9aaf, 0x9ab2, 0x9ab3, 0x9ab4, 0x9ab5, 0x9ab9, 0x9abb,
2901 0x9abd, 0x9abe, 0x9abf, 0x9ac3, 0x9ac4, 0x9ac6, 0x9ac7, 0x9ac8,
2902 0x9ac9, 0x9aca, 0x9acd, 0x9ace, 0x9acf, 0x9ad0, 0x9ad2, 0x9ad4,
2903 0x9ad5, 0x9ad6, 0x9ad7, 0x9ad9, 0x9ada, 0x9adb, 0x9adc, 0x9add,
2904 0x9ade, 0x9ae0, 0x9ae2, 0x9ae3, 0x9ae4, 0x9ae5, 0x9ae7, 0x9ae8,
2905 0x9ae9, 0x9aea, 0x9aeb, 0x9aec, 0x9aee, 0x9af0, 0x9af1, 0x9af2, 0x9af3,
2906 0x9af4, 0x9af5, 0x9af6, 0x9af7, 0x9af8, 0x9afa, 0x9afc, 0x9afd,
2907 0x9afe, 0x9aff, 0x9b00, 0x9b01, 0x9b02, 0x9b04, 0x9b05, 0x9b06,
2908 0x87c6, 0x8788, 0x8785, 0x87ad, 0x8797, 0x8783, 0x87ab, 0x87e5,
2909 0x87ac, 0x87b5, 0x87b3, 0x87cb, 0x87d3, 0x87bd, 0x87d1, 0x87c0,
2910 0x87ca, 0x87db, 0x87ea, 0x87e0, 0x87ee, 0x8816, 0x8813, 0x87fe,
2911 0x880a, 0x881b, 0x8821, 0x8839, 0x883c, 0x7f36, 0x7f42, 0x7f44,
2912 0x7f45, 0x8210, 0x7afa, 0x7afd, 0x7b08, 0x7b03, 0x7b04, 0x7b15,
2913 0x7b0a, 0x7b2b, 0x7b0f, 0x7b47, 0x7b38, 0x7b2a, 0x7b19, 0x7b2e,
2914 0x7b31, 0x7b20, 0x7b25, 0x7b24, 0x7b33, 0x7b3e, 0x7b1e, 0x7b58,
2915 0x7b5a, 0x7b45, 0x7b75, 0x7b4c, 0x7b5d, 0x7b60, 0x7b6e, 0x7b7b,
2916 0x7b62, 0x7b72, 0x7b71, 0x7b90, 0x7ba6, 0x7ba7, 0x7bb8, 0x7bac,
2917 0x7b9d, 0x7ba8, 0x7b85, 0x7baa, 0x7b9c, 0x7ba2, 0x7bab, 0x7bb4,
2918 0x7bd1, 0x7bc1, 0x7bcc, 0x7bdd, 0x7bda, 0x7be5, 0x7be6, 0x7bea,
2919 0x7c0c, 0x7bfe, 0x7bfc, 0x7c0f, 0x7c16, 0x7c0b,
2920 /* 0xf4 */
2921 0x9b07, 0x9b09, 0x9b0a, 0x9b0b, 0x9b0c, 0x9b0d, 0x9b0e, 0x9b10,
2922 0x9b11, 0x9b12, 0x9b14, 0x9b15, 0x9b16, 0x9b17, 0x9b18, 0x9b19,
2923 0x9b1a, 0x9b1b, 0x9b1c, 0x9b1d, 0x9b1e, 0x9b20, 0x9b21, 0x9b22,
2924 0x9b24, 0x9b25, 0x9b26, 0x9b27, 0x9b28, 0x9b29, 0x9b2a, 0x9b2b,
2925 0x9b2c, 0x9b2d, 0x9b2e, 0x9b30, 0x9b31, 0x9b33, 0x9b34, 0x9b35,
2926 0x9b36, 0x9b37, 0x9b38, 0x9b39, 0x9b3a, 0x9b3d, 0x9b3e, 0x9b3f,
2927 0x9b40, 0x9b46, 0x9b4a, 0x9b4b, 0x9b4c, 0x9b4e, 0x9b50, 0x9b52,
2928 0x9b53, 0x9b55, 0x9b56, 0x9b57, 0x9b58, 0x9b59, 0x9b5a, 0x9b5b,
2929 0x9b5c, 0x9b5d, 0x9b5e, 0x9b5f, 0x9b60, 0x9b61, 0x9b62, 0x9b63,
2930 0x9b64, 0x9b65, 0x9b66, 0x9b67, 0x9b68, 0x9b69, 0x9b6a, 0x9b6b,
2931 0x9b6c, 0x9b6d, 0x9b6e, 0x9b6f, 0x9b70, 0x9b71, 0x9b72, 0x9b73,
2932 0x9b74, 0x9b75, 0x9b76, 0x9b77, 0x9b78, 0x9b79, 0x9b7a, 0x9b7b,
2933 0x7c1f, 0x7c2a, 0x7c26, 0x7c38, 0x7c41, 0x7c40, 0x81fe, 0x8201,
2934 0x8202, 0x8204, 0x81ec, 0x8844, 0x8221, 0x8222, 0x8223, 0x822d,
2935 0x822f, 0x8228, 0x822b, 0x8238, 0x823b, 0x8233, 0x8234, 0x823e,
2936 0x8244, 0x8249, 0x824b, 0x824f, 0x825a, 0x825f, 0x8268, 0x887e,
2937 0x8885, 0x8888, 0x88d8, 0x88df, 0x895e, 0x7f9d, 0x7fa7,
2938 0x7faf, 0x7fb0, 0x7fb2, 0x7c7c, 0x6549, 0x7c91, 0x7c9d, 0x7c9c,
2939 0x7c9e, 0x7ca2, 0x7cb2, 0x7cbc, 0x7cbd, 0x7cc1, 0x7cc7, 0x7ccc,
2940 0x7ccd, 0x7cc8, 0x7cc5, 0x7cd7, 0x7ce8, 0x826e, 0x66a8, 0x7fbf,
2941 0x7fce, 0x7fd5, 0x7fe5, 0x7fe1, 0x7fe6, 0x7fe9, 0x7fee, 0x7ff3,
2942 0x7cf8, 0x7d77, 0x7da6, 0x7dae, 0x7e47, 0x7e9b, 0x9eb8, 0x9eb4,
2943 0x8d73, 0x8d84, 0x8d94, 0x8d91, 0x8db1, 0x8d67, 0x8d6d, 0x8c47,
2944 0x8c49, 0x914a, 0x9150, 0x914e, 0x914f, 0x9164,
2945 /* 0xf5 */
2946 0x9b7c, 0x9b7d, 0x9b7e, 0x9b7f, 0x9b80, 0x9b81, 0x9b82, 0x9b83,
2947 0x9b84, 0x9b85, 0x9b86, 0x9b87, 0x9b88, 0x9b89, 0x9b8a, 0x9b8b,
2948 0x9b8c, 0x9b8d, 0x9b8e, 0x9b8f, 0x9b90, 0x9b91, 0x9b92, 0x9b93,
2949 0x9b94, 0x9b95, 0x9b96, 0x9b97, 0x9b98, 0x9b99, 0x9b9a, 0x9b9b,
2950 0x9b9c, 0x9b9d, 0x9b9e, 0x9b9f, 0x9ba0, 0x9ba1, 0x9ba2, 0x9ba3,
2951 0x9ba4, 0x9ba5, 0x9ba6, 0x9ba7, 0x9ba8, 0x9ba9, 0x9baa, 0x9bab,
2952 0x9bac, 0x9bad, 0x9bae, 0x9baf, 0x9bb0, 0x9bb1, 0x9bb2, 0x9bb3,
2953 0x9bb4, 0x9bb5, 0x9bb6, 0x9bb7, 0x9bb8, 0x9bb9, 0x9bba, 0x9bbb,
2954 0x9bbc, 0x9bbd, 0x9bbe, 0x9bbf, 0x9bc0, 0x9bc1, 0x9bc2, 0x9bc3,
2955 0x9bc4, 0x9bc5, 0x9bc6, 0x9bc7, 0x9bc8, 0x9bc9, 0x9bca, 0x9bcb,
2956 0x9bcc, 0x9bcd, 0x9bce, 0x9bcf, 0x9bd0, 0x9bd1, 0x9bd2, 0x9bd3,
```

```
2957 0x9bd4, 0x9bd5, 0x9bd6, 0x9bd7, 0x9bd8, 0x9bd9, 0x9bda, 0x9bdb,
2958 0x9162, 0x9161, 0x9170, 0x9169, 0x916f, 0x917d, 0x917e, 0x9172,
2959 0x9174, 0x9179, 0x918c, 0x918c, 0x9185, 0x9190, 0x918d, 0x9191, 0x91a2,
2960 0x91a3, 0x91aa, 0x91ad, 0x91ae, 0x91af, 0x91b5, 0x91b4, 0x91ba,
2961 0x8c55, 0x9e7e, 0x8db8, 0x8deb, 0x8e05, 0x8e59, 0x8e69, 0x8db5,
2962 0x8dbf, 0x8dbc, 0x8dbb, 0x8dba, 0x8dc4, 0x8dd6, 0x8dd7, 0x8dda, 0x8dde,
2963 0x8dce, 0x8dcf, 0x8ddb, 0x8dc6, 0x8dec, 0x8df7, 0x8df8, 0x8de3,
2964 0x8df9, 0x8dfb, 0x8de4, 0x8e09, 0x8dfd, 0x8e14, 0x8e1d, 0x8e1f,
2965 0x8e2c, 0x8e2e, 0x8e23, 0x8e2f, 0x8e3a, 0x8e40, 0x8e39, 0x8e35,
2966 0x8e3d, 0x8e31, 0x8e49, 0x8e41, 0x8e42, 0x8e51, 0x8e52, 0x8e4a,
2967 0x8e70, 0x8e76, 0x8e7c, 0x8e6f, 0x8e74, 0x8e85, 0x8e8f, 0x8e94,
2968 0x8e90, 0x8e9c, 0x8e9e, 0x8c78, 0x8c82, 0x8c8a, 0x8c85, 0x8c98,
2969 0x8c94, 0x659b, 0x89d6, 0x89de, 0x89da, 0x89dc,
2970 /* 0xf6 */
2971 0x9bdc, 0x9bdd, 0x9bde, 0x9bdf, 0x9be0, 0x9be1, 0x9be2, 0x9be3,
2972 0x9be4, 0x9be5, 0x9be6, 0x9be7, 0x9be8, 0x9be9, 0x9bea, 0x9beb,
2973 0x9bec, 0x9bed, 0x9bee, 0x9bef, 0x9bf0, 0x9bf1, 0x9bf2, 0x9bf3,
2974 0x9bf4, 0x9bf5, 0x9bf6, 0x9bf7, 0x9bf8, 0x9bf9, 0x9bfa, 0x9bfb,
2975 0x9bfc, 0x9bfd, 0x9bfe, 0x9bff, 0x9c00, 0x9c01, 0x9c02, 0x9c03,
2976 0x9c04, 0x9c05, 0x9c06, 0x9c07, 0x9c08, 0x9c09, 0x9c0a, 0x9c0b,
2977 0x9c0c, 0x9c0d, 0x9c0e, 0x9c0f, 0x9c10, 0x9c11, 0x9c12, 0x9c13,
2978 0x9c14, 0x9c15, 0x9c16, 0x9c17, 0x9c18, 0x9c19, 0x9c1a, 0x9c1b,
2979 0x9c1c, 0x9c1d, 0x9c1e, 0x9c1f, 0x9c20, 0x9c21, 0x9c22, 0x9c23,
2980 0x9c24, 0x9c25, 0x9c26, 0x9c27, 0x9c28, 0x9c29, 0x9c2a, 0x9c2b,
2981 0x9c2c, 0x9c2d, 0x9c2e, 0x9c2f, 0x9c30, 0x9c31, 0x9c32, 0x9c33,
2982 0x9c34, 0x9c35, 0x9c36, 0x9c37, 0x9c38, 0x9c39, 0x9c3a, 0x9c3b,
2983 0x89e5, 0x89eb, 0x89ef, 0x8a3e, 0x8b26, 0x9753, 0x96e9, 0x96f3,
2984 0x96ef, 0x9706, 0x9701, 0x9708, 0x970f, 0x970e, 0x972a, 0x972d,
2985 0x9730, 0x973e, 0x97f8, 0x97f3, 0x97f5, 0x97f6, 0x97f7, 0x97f8,
2986 0x9789, 0x978a, 0x978c, 0x978d, 0x978e, 0x978f, 0x9790, 0x9791,
2987 0x96bd, 0x96ce, 0x96d2, 0x77bf, 0x96e0, 0x928e, 0x92ae, 0x92c8,
2988 0x933e, 0x936a, 0x93ca, 0x938f, 0x943e, 0x946b, 0x9c7f, 0x9c82,
2989 0x9c85, 0x9c8d, 0x9c87, 0x9c88, 0x7a23, 0x9c8b, 0x9c8e, 0x9c90,
2990 0x9c91, 0x9c92, 0x9c94, 0x9c95, 0x9c9a, 0x9c9b, 0x9c9e, 0x9c9f,
2991 0x9ca0, 0x9ca1, 0x9ca2, 0x9ca3, 0x9ca5, 0x9ca6, 0x9ca7, 0x9ca8,
2992 0x9cab, 0x9cab, 0x9cad, 0x9cae, 0x9cb0, 0x9cb1, 0x9cb2, 0x9cb3,
2993 0x9cb4, 0x9cb5, 0x9cb6, 0x9cb7, 0x9cba, 0x9cbb, 0x9cbc, 0x9cbd,
2994 0x9cc4, 0x9cc5, 0x9cc6, 0x9cc7, 0x9cca, 0x9ccb,
2995 /* 0xf7 */
2996 0x9c3c, 0x9c3d, 0x9c3e, 0x9c3f, 0x9c40, 0x9c41, 0x9c42, 0x9c43,
2997 0x9c44, 0x9c45, 0x9c46, 0x9c47, 0x9c48, 0x9c49, 0x9c4a, 0x9c4b,
2998 0x9c4c, 0x9c4d, 0x9c4e, 0x9c4f, 0x9c50, 0x9c51, 0x9c52, 0x9c53,
2999 0x9c54, 0x9c55, 0x9c56, 0x9c57, 0x9c58, 0x9c59, 0x9c5a, 0x9c5b,
3000 0x9c5c, 0x9c5d, 0x9c5e, 0x9c5f, 0x9c60, 0x9c61, 0x9c62, 0x9c63,
3001 0x9c64, 0x9c65, 0x9c66, 0x9c67, 0x9c68, 0x9c69, 0x9c6a, 0x9c6b,
3002 0x9c6c, 0x9c6d, 0x9c6e, 0x9c6f, 0x9c70, 0x9c71, 0x9c72, 0x9c73,
3003 0x9c74, 0x9c75, 0x9c76, 0x9c77, 0x9c78, 0x9c79, 0x9c7a, 0x9c7b,
3004 0x9c7d, 0x9c7e, 0x9c80, 0x9c83, 0x9c84, 0x9c89, 0x9c8a, 0x9c8c,
3005 0x9c8f, 0x9c93, 0x9c96, 0x9c97, 0x9c98, 0x9c99, 0x9c9d, 0x9caa,
3006 0x9cac, 0x9caf, 0x9cb9, 0x9cbe, 0x9cbf, 0x9cc0, 0x9cc1, 0x9cc2,
3007 0x9cc8, 0x9cc9, 0x9cd1, 0x9cd2, 0x9cda, 0x9cdb, 0x9ce0, 0x9ce1,
3008 0x9ccc, 0x9ccd, 0x9cce, 0x9ccf, 0x9cd0, 0x9cd3, 0x9cd4, 0x9cd5,
3009 0x9cd7, 0x9cd8, 0x9cd9, 0x9cdc, 0x9cdd, 0x9cdf, 0x9ce2, 0x977c,
3010 0x9785, 0x9791, 0x9792, 0x9794, 0x97af, 0x97ab, 0x97a3, 0x97b2,
3011 0x97b4, 0x9ab1, 0x9ab0, 0x9ab7, 0x9e58, 0x9ab6, 0x9aba, 0x9abc,
3012 0x9ac1, 0x9ac0, 0x9ac5, 0x9ac2, 0x9acb, 0x9acc, 0x9ad1, 0x9b45,
3013 0x9b43, 0x9b47, 0x9b49, 0x9b48, 0x9b4d, 0x9b51, 0x9b51, 0x9b51,
3014 0x992e, 0x9955, 0x9954, 0x9adf, 0x9ae1, 0x9ae6, 0x9aef, 0x9aeb,
3015 0x9afb, 0x9aed, 0x9af9, 0x9b08, 0x9b0f, 0x9b13, 0x9b1f, 0x9b23,
3016 0x9ebd, 0x9ebe, 0x9e3b, 0x9e82, 0x9e87, 0x9e88, 0x9e8b, 0x9e92,
3017 0x93d6, 0x9e9d, 0x9e9f, 0x9edb, 0x9edc, 0x9edd, 0x9ee0, 0x9edf,
3018 0x9ee2, 0x9ee9, 0x9ee7, 0x9ee5, 0x9eea, 0x9eef, 0x9f22, 0x9f2c,
3019 0x9f2f, 0x9f39, 0x9f37, 0x9f3d, 0x9f3e,
3020 /* 0xf8 */
3021 0x9ce3, 0x9ce4, 0x9ce5, 0x9ce6, 0x9ce7, 0x9ce8, 0x9ce9, 0x9cea,
3022 0x9ceb, 0x9cec, 0x9ced, 0x9cee, 0x9cef, 0x9cf0, 0x9cf1, 0x9cf2,
3023 0x9cf3, 0x9cf4, 0x9cf5, 0x9cf6, 0x9cf7, 0x9cf8, 0x9cf9, 0x9cfa,
3024 0x9cfb, 0x9cfc, 0x9cfd, 0x9cfe, 0x9cff, 0x9d00, 0x9d01, 0x9d02,
3025 0x9d03, 0x9d04, 0x9d05, 0x9d06, 0x9d07, 0x9d08, 0x9d09, 0x9d0a,
3026 0x9d0b, 0x9d0c, 0x9d0d, 0x9d0e, 0x9d0f, 0x9d10, 0x9d11, 0x9d12,
3027 0x9d13, 0x9d14, 0x9d15, 0x9d16, 0x9d17, 0x9d18, 0x9d19, 0x9d1a,
3028 0x9d1b, 0x9d1c, 0x9d1d, 0x9d1e, 0x9d1f, 0x9d20, 0x9d21, 0x9d22,
3029 0x9d23, 0x9d24, 0x9d25, 0x9d26, 0x9d27, 0x9d28, 0x9d29, 0x9d2a,
3030 0x9d2b, 0x9d2c, 0x9d2d, 0x9d2e, 0x9d2f, 0x9d30, 0x9d31, 0x9d32,
3031 0x9d33, 0x9d34, 0x9d35, 0x9d36, 0x9d37, 0x9d38, 0x9d39, 0x9d3a,
3032 0x9d3b, 0x9d3c, 0x9d3d, 0x9d3e, 0x9d3f, 0x9d40, 0x9d41, 0x9d42,
3033 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
3034 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
3035 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
3036 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
3037 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
3038 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
3039 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
3040 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
3041 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
3042 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
3043 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
```

```
3044 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
3045 /* 0xf9 */
3046 0x9d43, 0x9d44, 0x9d45, 0x9d46, 0x9d47, 0x9d48, 0x9d49, 0x9d4a,
3047 0x9d4b, 0x9d4c, 0x9d4d, 0x9d4e, 0x9d4f, 0x9d50, 0x9d51, 0x9d52,
3048 0x9d53, 0x9d54, 0x9d55, 0x9d56, 0x9d57, 0x9d58, 0x9d59, 0x9d5a,
3049 0x9d5b, 0x9d5c, 0x9d5d, 0x9d5e, 0x9d5f, 0x9d60, 0x9d61, 0x9d62,
3050 0x9d63, 0x9d64, 0x9d65, 0x9d66, 0x9d67, 0x9d68, 0x9d69, 0x9d6a,
3051 0x9d6b, 0x9d6c, 0x9d6d, 0x9d6e, 0x9d6f, 0x9d70, 0x9d71, 0x9d72,
3052 0x9d73, 0x9d74, 0x9d75, 0x9d76, 0x9d77, 0x9d78, 0x9d79, 0x9d7a,
3053 0x9d7b, 0x9d7c, 0x9d7d, 0x9d7e, 0x9d7f, 0x9d80, 0x9d81, 0x9d82,
3054 0x9d83, 0x9d84, 0x9d85, 0x9d86, 0x9d87, 0x9d88, 0x9d89, 0x9d8a,
3055 0x9d8b, 0x9d8c, 0x9d8d, 0x9d8e, 0x9d8f, 0x9d90, 0x9d91, 0x9d92,
3056 0x9d93, 0x9d94, 0x9d95, 0x9d96, 0x9d97, 0x9d98, 0x9d99, 0x9d9a,
3057 0x9d9b, 0x9d9c, 0x9d9d, 0x9d9e, 0x9d9f, 0x9da0, 0x9da1, 0x9da2,
3058 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
3059 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
3060 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
3061 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
3062 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
3063 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
3064 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
3065 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
3066 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
3067 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
3068 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
3069 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
3070 /* 0xfa */
3071 0x9da3, 0x9da4, 0x9da5, 0x9da6, 0x9da7, 0x9da8, 0x9da9, 0x9daa,
3072 0x9dab, 0x9dac, 0x9dad, 0x9dae, 0x9daf, 0x9db0, 0x9db1, 0x9db2,
3073 0x9db3, 0x9db4, 0x9db5, 0x9db6, 0x9db7, 0x9db8, 0x9db9, 0x9dba,
3074 0x9dbb, 0x9dbc, 0x9dbd, 0x9dbe, 0x9dbf, 0x9dc0, 0x9dc1, 0x9dc2,
3075 0x9dc3, 0x9dc4, 0x9dc5, 0x9dc6, 0x9dc7, 0x9dc8, 0x9dc9, 0x9dca,
3076 0x9dcb, 0x9dcc, 0x9dcd, 0x9dce, 0x9dcf, 0x9dd0, 0x9dd1, 0x9dd2,
3077 0x9dd3, 0x9dd4, 0x9dd5, 0x9dd6, 0x9dd7, 0x9dd8, 0x9dd9, 0x9dda,
3078 0x9ddb, 0x9ddc, 0x9ddd, 0x9dde, 0x9ddf, 0x9de0, 0x9de1, 0x9de2,
3079 0x9de3, 0x9de4, 0x9de5, 0x9de6, 0x9de7, 0x9de8, 0x9de9, 0x9dea,
3080 0x9deb, 0x9dec, 0x9ded, 0x9dee, 0x9def, 0x9df0, 0x9df1, 0x9df2,
3081 0x9df3, 0x9df4, 0x9df5, 0x9df6, 0x9df7, 0x9df8, 0x9df9, 0x9dfa,
3082 0x9dfb, 0x9dfc, 0x9dfd, 0x9dfe, 0x9dff, 0x9e00, 0x9e01, 0x9e02,
3083 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
3084 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
3085 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
3086 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
3087 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
3088 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
3089 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
3090 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
3091 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
3092 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
3093 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
3094 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
3095 /* 0xfb */
3096 0x9e03, 0x9e04, 0x9e05, 0x9e06, 0x9e07, 0x9e08, 0x9e09, 0x9e0a,
3097 0x9e0b, 0x9e0c, 0x9e0d, 0x9e0e, 0x9e0f, 0x9e10, 0x9e11, 0x9e12,
3098 0x9e13, 0x9e14, 0x9e15, 0x9e16, 0x9e17, 0x9e18, 0x9e19, 0x9e1a,
3099 0x9e1b, 0x9e1c, 0x9e1d, 0x9e1e, 0x9e1f, 0x9e20, 0x9e21, 0x9e22, 0x9e23,
3100 0x9e24, 0x9e25, 0x9e26, 0x9e27, 0x9e28, 0x9e29, 0x9e2a, 0x9e2b,
3101 0x9e2c, 0x9e2d, 0x9e2e, 0x9e2f, 0x9e30, 0x9e31, 0x9e32, 0x9e33,
3102 0x9e34, 0x9e35, 0x9e36, 0x9e37, 0x9e38, 0x9e39, 0x9e3a, 0x9e3b,
3103 0x9e3c, 0x9e3d, 0x9e3e, 0x9e3f, 0x9e40, 0x9e41, 0x9e42, 0x9e43,
3104 0x9e44, 0x9e45, 0x9e46, 0x9e47, 0x9e48, 0x9e49, 0x9e4a, 0x9e4b,
3105 0x9e4c, 0x9e4d, 0x9e4e, 0x9e4f, 0x9e50, 0x9e51, 0x9e52, 0x9e53,
3106 0x9e54, 0x9e55, 0x9e56, 0x9e57, 0x9e58, 0x9e59, 0x9e5a, 0x9e5b,
3107 0x9e5c, 0x9e5d, 0x9e5e, 0x9e5f, 0x9e60, 0x9e61, 0x9e62,
3108 0x9e63, 0x9e64, 0x9e65, 0x9e66, 0x9e67, 0x9e68, 0x9e69, 0x9e6a,
3109 0x9e6b, 0x9e6c, 0x9e6d, 0x9e6e, 0x9e6f, 0x9e70, 0x9e71, 0x9e72,
3110 0x9e73, 0x9e74, 0x9e75, 0x9e76, 0x9e77, 0x9e78, 0x9e79, 0x9e7a,
3111 0x9e7b, 0x9e7c, 0x9e7d, 0x9e7e, 0x9e7f, 0x9e80, 0x9e81, 0x9e82,
3112 0x9e83, 0x9e84, 0x9e85, 0x9e86, 0x9e87, 0x9e88, 0x9e89, 0x9e8a,
3113 0x9e8b, 0x9e8c, 0x9e8d, 0x9e8e, 0x9e8f, 0x9e90, 0x9e91, 0x9e92,
3114 0x9e93, 0x9e94, 0x9e95, 0x9e96, 0x9e97, 0x9e98, 0x9e99, 0x9e9a,
3115 0x9e9b, 0x9e9c, 0x9e9d, 0x9e9e, 0x9e9f, 0x9ea0, 0x9ea1, 0x9ea2,
3116 0x9ea3, 0x9ea4, 0x9ea5, 0x9ea6, 0x9ea7, 0x9ea8, 0x9ea9, 0x9eaa,
3117 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
3118 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
3119 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd, 0xffffd,
3120 /* 0xfc */
3121 0x9eab, 0x9eac, 0x9ead, 0x9eae, 0x9eaf, 0x9eb0, 0x9eb1, 0x9eb2,
3122 0x9eb3, 0x9eb4, 0x9eb5, 0x9eb6, 0x9eb7, 0x9eb8, 0x9eba, 0x9ebc, 0x9ebf,
3123 0x9ec0, 0x9ec1, 0x9ec2, 0x9ec3, 0x9ec4, 0x9ec5, 0x9ec6, 0x9ec7, 0x9ec8,
3124 0x9eca, 0x9ecb, 0x9ecc, 0x9ecd, 0x9ece, 0x9ed0, 0x9ed1, 0x9ed2, 0x9ed3,
3125 0x9ed4, 0x9ed5, 0x9ed6, 0x9ed7, 0x9ed8, 0x9eda, 0x9ede, 0x9ee0, 0x9ee1,
3126 0x9ee2, 0x9ee3, 0x9ee4, 0x9ee5, 0x9ee6, 0x9ee7, 0x9ee8, 0x9ee9,
3127 0x9eea, 0x9eeb, 0x9eec, 0x9eed, 0x9eee, 0x9eef, 0x9ef0, 0x9ef1,
3128 0x9ef2, 0x9ef3, 0x9ef4, 0x9ef5, 0x9ef6, 0x9ef7, 0x9ef8, 0x9efa, 0x9efd,
3129 0x9eff, 0x9f00, 0x9f01, 0x9f02, 0x9f03, 0x9f04, 0x9f05, 0x9f06,
3130 0x9f07, 0x9f08, 0x9f09, 0x9f0a, 0x9f0b, 0x9f0c, 0x9f0d, 0x9f0e,
3131 0x9f0f, 0x9f10, 0x9f11, 0x9f12, 0x9f13, 0x9f14, 0x9f15, 0x9f16,
3132 0x9f17, 0x9f18, 0x9f19, 0x9f1a, 0x9f1b, 0x9f1c, 0x9f1d,
```

```

3131 0x9f1e, 0x9f1f, 0x9f21, 0x9f23, 0x9f24, 0x9f25, 0x9f26, 0x9f27,
3132 0x9f28, 0x9f29, 0x9f2a, 0x9f2b, 0x9f2d, 0x9f2e, 0x9f30, 0x9f31,
3133 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
3134 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
3135 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
3136 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
3137 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
3138 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
3139 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
3140 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
3141 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
3142 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
3143 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
3144 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
3145 /* 0xfd */
3146 0x9f32, 0x9f33, 0x9f34, 0x9f35, 0x9f36, 0x9f38, 0x9f3a, 0x9f3c,
3147 0x9f3f, 0x9f40, 0x9f41, 0x9f42, 0x9f43, 0x9f45, 0x9f46, 0x9f47,
3148 0x9f48, 0x9f49, 0x9f4a, 0x9f4b, 0x9f4c, 0x9f4d, 0x9f4e, 0x9f4f,
3149 0x9f52, 0x9f53, 0x9f54, 0x9f55, 0x9f56, 0x9f57, 0x9f58, 0x9f59,
3150 0x9f5a, 0x9f5b, 0x9f5c, 0x9f5d, 0x9f5e, 0x9f5f, 0x9f60, 0x9f61,
3151 0x9f62, 0x9f63, 0x9f64, 0x9f65, 0x9f66, 0x9f67, 0x9f68, 0x9f69,
3152 0x9f6a, 0x9f6b, 0x9f6c, 0x9f6d, 0x9f6e, 0x9f6f, 0x9f70, 0x9f71,
3153 0x9f72, 0x9f73, 0x9f74, 0x9f75, 0x9f76, 0x9f77, 0x9f78, 0x9f79,
3154 0x9f7a, 0x9f7b, 0x9f7c, 0x9f7d, 0x9f7e, 0x9f81, 0x9f82, 0x9f8d,
3155 0x9f8e, 0x9f8f, 0x9f90, 0x9f91, 0x9f92, 0x9f93, 0x9f94, 0x9f95,
3156 0x9f96, 0x9f97, 0x9f98, 0x9f9c, 0x9f9d, 0x9f9e, 0x9fa1, 0x9fa2,
3157 0x9fa3, 0x9fa4, 0x9fa5, 0x9fa8, 0x9fa9, 0x9fab, 0x9fc1, 0x9fc2,
3158 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
3159 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
3160 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
3161 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
3162 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
3163 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
3164 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
3165 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
3166 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
3167 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
3168 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
3169 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
3170 /* 0xfe */
3171 0xfa0c, 0xfa0d, 0xfa0e, 0xfa0f, 0xfa11, 0xfa13, 0xfa14, 0xfa18,
3172 0xfa1f, 0xfa20, 0xfa21, 0xfa23, 0xfa24, 0xfa27, 0xfa28, 0xfa29,
3173 };
3174
3175 static int
3176 cp936ext_mbtowc (conv_t conv, ucs4_t *pwc, const unsigned char *s, int n)
3177 {
3178     unsigned char c1 = s[0];
3179     if ((c1 >= 0x81 && c1 <= 0xfe)) {
3180         if (n >= 2) {
3181             unsigned char c2 = s[1];
3182             if ((c2 >= 0x40 && c2 < 0x7f) || (c2 >= 0x80 && c2 < 0xff)) {
3183                 unsigned int i = 190 * (c1 - 0x81) + (c2 - (c2 >= 0x80 ? 0x41 : 0x40));
3184                 unsigned short wc = 0xffff;
3185                 {
3186                     if (i < 23766)
3187                         wc = cp936ext_2uni_page81[i];
3188                 }
3189                 if (wc != 0xffff) {
3190                     *pwc = (ucs4_t) wc;
3191                     return 2;
3192                 }
3193             }
3194             return RET_ILSEQ;
3195         }
3196         return RET_TOOFEW(0);
3197     }
3198     return RET_ILSEQ;
3199 }
3200 #endif /* NEED_TOWC */
3201
3202 #ifdef NEED_TOMB
3203
3204 static const unsigned short cp936ext_page0014[208] = {
3205     0x0000, 0x0000, 0x0000, 0x0000, 0xale8, 0x0000, 0x0000, 0xalec, /*0xa0-0xa7*/
3206     0xa1a7, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0xa8-0xaf*/
3207     0xale3, 0xalc0, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xala4, /*0xb0-0xb7*/
3208     0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0xb8-0xbf*/
3209     0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0xc0-0xc7*/
3210     0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0xc8-0xcf*/
3211     0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xalc1, /*0xd0-0xd7*/
3212     0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0xd8-0xdf*/
3213     0xa8a4, 0xa8a2, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0xe0-0xe7*/
3214     0xa8a8, 0xa8a6, 0xa8ba, 0x0000, 0xa8ac, 0xa8aa, 0x0000, 0x0000, /*0xe8-0xef*/
3215     0x0000, 0x0000, 0xa8b0, 0xa8ae, 0x0000, 0x0000, 0x0000, 0xalc2, /*0xf0-0xf7*/
3216     0x0000, 0xa8b4, 0xa8b2, 0x0000, 0xa8b9, 0x0000, 0x0000, 0x0000, /*0xf8-0xff*/
3217     /* 0x0100 */

```

```
3218 0x0000, 0xa8a1, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0x00-0x07*/
3219 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0x08-0x0f*/
3220 0x0000, 0x0000, 0x0000, 0xa8a5, 0x0000, 0x0000, 0x0000, 0x0000, /*0x10-0x17*/
3221 0x0000, 0x0000, 0x0000, 0xa8a7, 0x0000, 0x0000, 0x0000, 0x0000, /*0x18-0x1f*/
3222 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0x20-0x27*/
3223 0x0000, 0x0000, 0x0000, 0xa8a9, 0x0000, 0x0000, 0x0000, 0x0000, /*0x28-0x2f*/
3224 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0x30-0x37*/
3225 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0x38-0x3f*/
3226 0x0000, 0x0000, 0x0000, 0x0000, 0xa8bd, 0x0000, 0x0000, 0x0000, /*0x40-0x47*/
3227 0xa8be, 0x0000, 0x0000, 0x0000, 0x0000, 0xa8ad, 0x0000, 0x0000, /*0x48-0x4f*/
3228 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0x50-0x57*/
3229 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0x58-0x5f*/
3230 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0x60-0x67*/
3231 0x0000, 0x0000, 0x0000, 0xa8b1, 0x0000, 0x0000, 0x0000, 0x0000, /*0x68-0x6f*/
3232 };
3233 static const unsigned short cp936ext_page0039[24] = {
3234 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xa8a3, 0x0000, /*0xc8-0xcf*/
3235 0xa8ab, 0x0000, 0xa8af, 0x0000, 0xa8b3, 0x0000, 0xa8b5, 0x0000, /*0xd0-0xd7*/
3236 0xa8b6, 0x0000, 0xa8b7, 0x0000, 0xa8b8, 0x0000, 0x0000, 0x0000, /*0xd8-0xdf*/
3237 };
3238 static const unsigned short cp936ext_page004a[24] = {
3239 0x0000, 0xa8bb, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0x50-0x57*/
3240 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0x58-0x5f*/
3241 0x0000, 0xa8c0, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0x60-0x67*/
3242 };
3243 static const unsigned short cp936ext_page0058[32] = {
3244 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xa1a6, /*0xc0-0xc7*/
3245 0x0000, 0xa1a5, 0xa840, 0xa841, 0x0000, 0x0000, 0x0000, 0x0000, /*0xc8-0xcf*/
3246 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0xd0-0xd7*/
3247 0x0000, 0xa842, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0xd8-0xdf*/
3248 };
3249 static const unsigned short cp936ext_page0072[64] = {
3250 0x0000, 0xa6a1, 0xa6a2, 0xa6a3, 0xa6a4, 0xa6a5, 0xa6a6, 0xa6a7, /*0x90-0x97*/
3251 0xa6a8, 0xa6a9, 0xa6aa, 0xa6ab, 0xa6ac, 0xa6ad, 0xa6ae, 0xa6af, /*0x98-0x9f*/
3252 0xa6b0, 0xa6b1, 0x0000, 0xa6b2, 0xa6b3, 0xa6b4, 0xa6b5, 0xa6b6, /*0xa0-0xa7*/
3253 0xa6b7, 0xa6b8, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0xa8-0xaf*/
3254 0x0000, 0xa6c1, 0xa6c2, 0xa6c3, 0xa6c4, 0xa6c5, 0xa6c6, 0xa6c7, /*0xb0-0xb7*/
3255 0xa6c8, 0xa6c9, 0xa6ca, 0xa6cb, 0xa6cc, 0xa6cd, 0xa6ce, 0xa6cf, /*0xb8-0xbf*/
3256 0xa6d0, 0xa6d1, 0x0000, 0xa6d2, 0xa6d3, 0xa6d4, 0xa6d5, 0xa6d6, /*0xc0-0xc7*/
3257 0xa6d7, 0xa6d8, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0xc8-0xcf*/
3258 };
3259 static const unsigned short cp936ext_page0080[88] = {
3260 0x0000, 0xa7a7, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0x00-0x07*/
3261 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0x08-0x0f*/
3262 0xa7a1, 0xa7a2, 0xa7a3, 0xa7a4, 0xa7a5, 0xa7a6, 0xa7a7, 0xa7a8, 0xa7a9, /*0x10-0x17*/
3263 0xa7aa, 0xa7ab, 0xa7ac, 0xa7ad, 0xa7ae, 0xa7af, 0xa7b0, 0xa7b1, /*0x18-0x1f*/
3264 0xa7b2, 0xa7b3, 0xa7b4, 0xa7b5, 0xa7b6, 0xa7b7, 0xa7b8, 0xa7b9, /*0x20-0x27*/
3265 0xa7ba, 0xa7bb, 0xa7bc, 0xa7bd, 0xa7be, 0xa7bf, 0xa7c0, 0xa7c1, /*0x28-0x2f*/
3266 0xa7d1, 0xa7d2, 0xa7d3, 0xa7d4, 0xa7d5, 0xa7d6, 0xa7d7, 0xa7d8, 0xa7d9, /*0x30-0x37*/
3267 0xa7da, 0xa7db, 0xa7dc, 0xa7dd, 0xa7de, 0xa7df, 0xa7e0, 0xa7e1, /*0x38-0x3f*/
3268 0xa7e2, 0xa7e3, 0xa7e4, 0xa7e5, 0xa7e6, 0xa7e7, 0xa7e8, 0xa7e9, /*0x40-0x47*/
3269 0xa7ea, 0xa7eb, 0xa7ec, 0xa7ed, 0xa7ee, 0xa7ef, 0xa7f0, 0xa7f1, /*0x48-0x4f*/
3270 0x0000, 0xa7d7, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0x50-0x57*/
3271 };
3272 static const unsigned short cp936ext_page0402[48] = {
3273 0xa95c, 0x0000, 0x0000, 0xa843, 0xa1aa, 0xa844, 0xalac, 0x0000, /*0x10-0x17*/
3274 0xa1ae, 0xa1af, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0x18-0x1f*/
3275 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xa845, 0xa1ad, 0x0000, /*0x20-0x27*/
3276 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0x28-0x2f*/
3277 0xa1eb, 0x0000, 0xa1e4, 0xa1e5, 0x0000, 0xa846, 0x0000, 0x0000, /*0x30-0x37*/
3278 0x0000, 0x0000, 0x0000, 0xa1f9, 0x0000, 0x0000, 0x0000, 0x0000, /*0x38-0x3f*/
3279 };
3280 static const unsigned short cp936ext_page0420[160] = {
3281 0x0000, 0x0000, 0x0000, 0xa1e6, 0x0000, 0xa847, 0x0000, 0x0000, /*0x00-0x07*/
3282 0x0000, 0xa848, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0x08-0x0f*/
3283 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xaled, 0x0000, /*0x10-0x17*/
3284 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0x18-0x1f*/
3285 0x0000, 0xa959, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0x20-0x27*/
3286 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0x28-0x2f*/
3287 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0x30-0x37*/
3288 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0x38-0x3f*/
3289 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0x40-0x47*/
3290 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0x48-0x4f*/
3291 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0x50-0x57*/
3292 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0x58-0x5f*/
3293 0xa2f1, 0xa2f2, 0xa2f3, 0xa2f4, 0xa2f5, 0xa2f6, 0xa2f7, 0xa2f8, /*0x60-0x67*/
3294 0xa2f9, 0xa2fa, 0xa2fb, 0xa2fc, 0x0000, 0x0000, 0x0000, 0x0000, /*0x68-0x6f*/
3295 0xa2a1, 0xa2a2, 0xa2a3, 0xa2a4, 0xa2a5, 0xa2a6, 0xa2a7, 0xa2a8, /*0x70-0x77*/
3296 0xa2a9, 0xa2aa, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0x78-0x7f*/
3297 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0x80-0x87*/
3298 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0x88-0x8f*/
3299 0xa1fb, 0xa1fc, 0xa1fa, 0xa1fd, 0x0000, 0x0000, 0xa849, 0xa84a, /*0x90-0x97*/
3300 0xa84b, 0xa84c, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0x98-0x9f*/
3301 };
3302 static const unsigned short cp936ext_page0441[184] = {
3303 0xalca, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xalca, /*0x08-0x0f*/
3304 0x0000, 0xalca, 0x0000, 0x0000, 0x0000, 0xa84d, 0x0000, 0x0000, /*0x10-0x17*/
```

```
3305 0x0000, 0x0000, 0xa1cc, 0x0000, 0x0000, 0xa1d8, 0xa1de, 0xa84e, /*0x18-0x1f*/
3306 0xa1cf, 0x0000, 0x0000, 0xa84f, 0x0000, 0xa1ce, 0x0000, 0xa1c4, /*0x20-0x27*/
3307 0xa1c5, 0xa1c9, 0xa1c8, 0xa1d2, 0x0000, 0x0000, 0xa1d3, 0x0000, /*0x28-0x2f*/
3308 0x0000, 0x0000, 0x0000, 0x0000, 0xa1e0, 0xa1df, 0xa1c3, 0xa1cb, /*0x30-0x37*/
3309 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xa1d7, 0x0000, 0x0000, /*0x38-0x3f*/
3310 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0x40-0x47*/
3311 0xa1d6, 0x0000, 0x0000, 0x0000, 0xa1d5, 0x0000, 0x0000, 0x0000, /*0x48-0x4f*/
3312 0x0000, 0x0000, 0xa850, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0x50-0x57*/
3313 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0x58-0x5f*/
3314 0xa1d9, 0xa1d4, 0x0000, 0x0000, 0xa1dc, 0xa1dd, 0xa851, 0xa852, /*0x60-0x67*/
3315 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xa1da, 0xa1db, /*0x68-0x6f*/
3316 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0x70-0x77*/
3317 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0x78-0x7f*/
3318 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0x80-0x87*/
3319 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0x88-0x8f*/
3320 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xa892, 0x0000, 0x0000, /*0x90-0x97*/
3321 0x0000, 0xa1d1, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0x98-0x9f*/
3322 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xa1cd, 0x0000, 0x0000, /*0xa0-0xa7*/
3323 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0xa8-0xaf*/
3324 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0xb0-0xbf*/
3325 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xa853, /*0xb8-0xbf*/
3326 };
3327 static const unsigned short cp936ext_page048c[64] = {
3328 0xa2d9, 0xa2da, 0xa2db, 0xa2dc, 0xa2dd, 0xa2de, 0xa2df, 0xa2e0, /*0x60-0x67*/
3329 0xa2e1, 0xa2e2, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0x68-0x6f*/
3330 0x0000, 0x0000, 0x0000, 0x0000, 0xa2c5, 0xa2c6, 0xa2c7, 0xa2c8, /*0x70-0x77*/
3331 0xa2c9, 0xa2ca, 0xa2cb, 0xa2cc, 0xa2cd, 0xa2ce, 0xa2cf, 0xa2d0, /*0x78-0x7f*/
3332 0xa2d1, 0xa2d2, 0xa2d3, 0xa2d4, 0xa2d5, 0xa2d6, 0xa2d7, 0xa2d8, /*0x80-0x87*/
3333 0xa2b1, 0xa2b2, 0xa2b3, 0xa2b4, 0xa2b5, 0xa2b6, 0xa2b7, 0xa2b8, /*0x88-0x8f*/
3334 0xa2b9, 0xa2ba, 0xa2bb, 0xa2bc, 0xa2bd, 0xa2be, 0xa2bf, 0xa2c0, /*0x90-0x97*/
3335 0xa2c1, 0xa2c2, 0xa2c3, 0xa2c4, 0x0000, 0x0000, 0x0000, 0x0000, /*0x98-0x9f*/
3336 };
3337 static const unsigned short cp936ext_page04a0[232] = {
3338 0xa9a4, 0xa9a5, 0xa9a6, 0xa9a7, 0xa9a8, 0xa9a9, 0xa9aa, 0xa9ab, /*0x00-0x07*/
3339 0xa9ac, 0xa9ad, 0xa9ae, 0xa9af, 0xa9b0, 0xa9b1, 0xa9b2, 0xa9b3, /*0x08-0x0f*/
3340 0xa9b4, 0xa9b5, 0xa9b6, 0xa9b7, 0xa9b8, 0xa9b9, 0xa9ba, 0xa9bb, /*0x10-0x17*/
3341 0xa9bc, 0xa9bd, 0xa9be, 0xa9bf, 0xa9c0, 0xa9c1, 0xa9c2, 0xa9c3, /*0x18-0x1f*/
3342 0xa9c4, 0xa9c5, 0xa9c6, 0xa9c7, 0xa9c8, 0xa9c9, 0xa9ca, 0xa9cb, /*0x20-0x27*/
3343 0xa9cc, 0xa9cd, 0xa9ce, 0xa9cf, 0xa9d0, 0xa9d1, 0xa9d2, 0xa9d3, /*0x28-0x2f*/
3344 0xa9d4, 0xa9d5, 0xa9d6, 0xa9d7, 0xa9d8, 0xa9d9, 0xa9da, 0xa9db, /*0x30-0x37*/
3345 0xa9dc, 0xa9dd, 0xa9de, 0xa9df, 0xa9e0, 0xa9e1, 0xa9e2, 0xa9e3, /*0x38-0x3f*/
3346 0xa9e4, 0xa9e5, 0xa9e6, 0xa9e7, 0xa9e8, 0xa9e9, 0xa9ea, 0xa9eb, /*0x40-0x47*/
3347 0xa9ec, 0xa9ed, 0xa9ee, 0xa9ef, 0x0000, 0x0000, 0x0000, 0x0000, /*0x48-0x4f*/
3348 0xa854, 0xa855, 0xa856, 0xa857, 0xa858, 0xa859, 0xa85a, 0xa85b, /*0x50-0x57*/
3349 0xa85c, 0xa85d, 0xa85e, 0xa85f, 0xa860, 0xa861, 0xa862, 0xa863, /*0x58-0x5f*/
3350 0xa864, 0xa865, 0xa866, 0xa867, 0xa868, 0xa869, 0xa86a, 0xa86b, /*0x60-0x67*/
3351 0xa86c, 0xa86d, 0xa86e, 0xa86f, 0xa870, 0xa871, 0xa872, 0xa873, /*0x68-0x6f*/
3352 0xa874, 0xa875, 0xa876, 0xa877, 0x0000, 0x0000, 0x0000, 0x0000, /*0x70-0x77*/
3353 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0x78-0x7f*/
3354 0x0000, 0xa878, 0xa879, 0xa87a, 0xa87b, 0xa87c, 0xa87d, 0xa87e, /*0x80-0x87*/
3355 0xa880, 0xa881, 0xa882, 0xa883, 0xa884, 0xa885, 0xa886, 0xa887, /*0x88-0x8f*/
3356 0x0000, 0x0000, 0x0000, 0xa888, 0xa889, 0xa88a, 0x0000, 0x0000, /*0x90-0x97*/
3357 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0x98-0x9f*/
3358 0xa1f6, 0xa1f5, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0xa0-0xa7*/
3359 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0xa8-0xaf*/
3360 0x0000, 0x0000, 0xa1f8, 0xa1f7, 0x0000, 0x0000, 0x0000, 0x0000, /*0xb0-0xbf*/
3361 0x0000, 0x0000, 0x0000, 0x0000, 0xa88b, 0xa88c, 0x0000, 0x0000, /*0xb8-0xbf*/
3362 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xa1f4, 0xa1f3, /*0xc0-0xc7*/
3363 0x0000, 0x0000, 0x0000, 0xa1f0, 0x0000, 0x0000, 0xa1f2, 0xa1f1, /*0xc8-0xcf*/
3364 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0xd0-0xdf*/
3365 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0xe0-0xef*/
3366 0x0000, 0x0000, 0xa88d, 0xa88e, 0xa88f, 0xa890, 0x0000, 0x0000, /*0xe0-0xe7*/
3367 };
3368 static const unsigned short cp936ext_page04c0[72] = {
3369 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xa1ef, 0xa1ee, 0x0000, /*0x00-0x07*/
3370 0x0000, 0xa891, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0x08-0x0f*/
3371 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0x10-0x17*/
3372 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0x18-0x1f*/
3373 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0x20-0x27*/
3374 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0x28-0x2f*/
3375 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0x30-0x37*/
3376 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0x38-0x3f*/
3377 0xa1e2, 0x0000, 0xa1e1, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0x40-0x47*/
3378 };
3379 static const unsigned short cp936ext_page0600[304] = {
3380 0xa1a1, 0xa1a2, 0xa1a3, 0xa1a4, 0x0000, 0xa1a9, 0xa965, 0xa996, /*0x00-0x07*/
3381 0xa1b4, 0xa1b5, 0xa1b6, 0xa1b7, 0xa1b8, 0xa1b9, 0xa1ba, 0xa1bb, /*0x08-0x0f*/
3382 0xa1be, 0xa1bf, 0xa893, 0xa1fe, 0xa1b2, 0xa1b3, 0xa1bc, 0xa1bd, /*0x10-0x17*/
3383 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xa894, 0xa895, 0x0000, /*0x18-0x1f*/
3384 0x0000, 0xa940, 0xa941, 0xa942, 0xa943, 0xa944, 0xa945, 0xa946, /*0x20-0x27*/
3385 0xa947, 0xa948, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0x28-0x2f*/
3386 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0x30-0x37*/
3387 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0x38-0x3f*/
3388 0x0000, 0xa4a1, 0xa4a2, 0xa4a3, 0xa4a4, 0xa4a5, 0xa4a6, 0xa4a7, /*0x40-0x47*/
3389 0xa4a8, 0xa4a9, 0xa4aa, 0xa4ab, 0xa4ac, 0xa4ad, 0xa4ae, 0xa4af, /*0x48-0x4f*/
3390 0xa4b0, 0xa4b1, 0xa4b2, 0xa4b3, 0xa4b4, 0xa4b5, 0xa4b6, 0xa4b7, /*0x50-0x57*/
3391 0xa4b8, 0xa4b9, 0xa4ba, 0xa4bb, 0xa4bc, 0xa4bd, 0xa4be, 0xa4bf, /*0x58-0x5f*/
```



```
3392 0xa4c0, 0xa4c1, 0xa4c2, 0xa4c3, 0xa4c4, 0xa4c5, 0xa4c6, 0xa4c7, /*0x60-0x67*/
3393 0xa4c8, 0xa4c9, 0xa4ca, 0xa4cb, 0xa4cc, 0xa4cd, 0xa4ce, 0xa4cf, /*0x68-0x6f*/
3394 0xa4d0, 0xa4d1, 0xa4d2, 0xa4d3, 0xa4d4, 0xa4d5, 0xa4d6, 0xa4d7, /*0x70-0x77*/
3395 0xa4d8, 0xa4d9, 0xa4da, 0xa4db, 0xa4dc, 0xa4dd, 0xa4de, 0xa4df, /*0x78-0x7f*/
3396 0xa4e0, 0xa4e1, 0xa4e2, 0xa4e3, 0xa4e4, 0xa4e5, 0xa4e6, 0xa4e7, /*0x80-0x87*/
3397 0xa4e8, 0xa4e9, 0xa4ea, 0xa4eb, 0xa4ec, 0xa4ed, 0xa4ee, 0xa4ef, /*0x88-0x8f*/
3398 0xa4f0, 0xa4f1, 0xa4f2, 0xa4f3, 0x0000, 0x0000, 0x0000, 0x0000, /*0x90-0x97*/
3399 0x0000, 0x0000, 0x0000, 0xa961, 0xa962, 0xa966, 0xa967, 0x0000, /*0x98-0x9f*/
3400 0x0000, 0xa5a1, 0xa5a2, 0xa5a3, 0xa5a4, 0xa5a5, 0xa5a6, 0xa5a7, /*0xa0-0xa7*/
3401 0xa5a8, 0xa5a9, 0xa5aa, 0xa5ab, 0xa5ac, 0xa5ad, 0xa5ae, 0xa5af, /*0xa8-0xaf*/
3402 0xa5b0, 0xa5b1, 0xa5b2, 0xa5b3, 0xa5b4, 0xa5b5, 0xa5b6, 0xa5b7, /*0xb0-0xb7*/
3403 0xa5b8, 0xa5b9, 0xa5ba, 0xa5bb, 0xa5bc, 0xa5bd, 0xa5be, 0xa5bf, /*0xb8-0xbf*/
3404 0xa5c0, 0xa5c1, 0xa5c2, 0xa5c3, 0xa5c4, 0xa5c5, 0xa5c6, 0xa5c7, /*0xc0-0xc7*/
3405 0xa5c8, 0xa5c9, 0xa5ca, 0xa5cb, 0xa5cc, 0xa5cd, 0xa5ce, 0xa5cf, /*0xc8-0xcf*/
3406 0xa5d0, 0xa5d1, 0xa5d2, 0xa5d3, 0xa5d4, 0xa5d5, 0xa5d6, 0xa5d7, /*0xd0-0xd7*/
3407 0xa5d8, 0xa5d9, 0xa5da, 0xa5db, 0xa5dc, 0xa5dd, 0xa5de, 0xa5df, /*0xd8-0xdf*/
3408 0xa5e0, 0xa5e1, 0xa5e2, 0xa5e3, 0xa5e4, 0xa5e5, 0xa5e6, 0xa5e7, /*0xe0-0xe7*/
3409 0xa5e8, 0xa5e9, 0xa5ea, 0xa5eb, 0xa5ec, 0xa5ed, 0xa5ee, 0xa5ef, /*0xe8-0xef*/
3410 0xa5f0, 0xa5f1, 0xa5f2, 0xa5f3, 0xa5f4, 0xa5f5, 0xa5f6, 0x0000, /*0xf0-0xf7*/
3411 0x0000, 0x0000, 0x0000, 0x0000, 0xa960, 0xa963, 0xa964, 0x0000, /*0xf8-0xff*/
3412 /* 0x3100 */
3413 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xa8c5, 0xa8c6, 0xa8c7, /*0x00-0x07*/
3414 0xa8c8, 0xa8c9, 0xa8ca, 0xa8cb, 0xa8cc, 0xa8cd, 0xa8ce, 0xa8cf, /*0x08-0x0f*/
3415 0xa8d0, 0xa8d1, 0xa8d2, 0xa8d3, 0xa8d4, 0xa8d5, 0xa8d6, 0xa8d7, /*0x10-0x17*/
3416 0xa8d8, 0xa8d9, 0xa8da, 0xa8db, 0xa8dc, 0xa8dd, 0xa8de, 0xa8df, /*0x18-0x1f*/
3417 0xa8e0, 0xa8e1, 0xa8e2, 0xa8e3, 0xa8e4, 0xa8e5, 0xa8e6, 0xa8e7, /*0x20-0x2f*/
3418 0xa8e8, 0xa8e9, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0x28-0x2f*/
3419 };
3420 static const unsigned short cp936ext_page0644[24] = {
3421 0xa2e5, 0xa2e6, 0xa2e7, 0xa2e8, 0xa2e9, 0xa2ea, 0xa2eb, 0xa2ec, /*0x20-0x27*/
3422 0xa2ed, 0xa2ee, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0x28-0x2f*/
3423 0x0000, 0xa95a, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0x30-0x37*/
3424 };
3425 static const unsigned short cp936ext_page0671[80] = {
3426 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xa94a, 0xa94b, /*0x88-0x8f*/
3427 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0x90-0x97*/
3428 0x0000, 0x0000, 0x0000, 0x0000, 0xa94c, 0xa94d, 0xa94e, 0x0000, /*0x98-0x9f*/
3429 0x0000, 0xa94f, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0xa0-0xaf*/
3430 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0xa8-0xaf*/
3431 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0xb0-0xb7*/
3432 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0xb8-0xbf*/
3433 0x0000, 0x0000, 0x0000, 0x0000, 0xa950, 0xa951, 0x0000, 0x0000, /*0xc0-0xcf*/
3434 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xa951, 0x0000, 0x0000, /*0xc8-0xcf*/
3435 0x0000, 0xa952, 0xa953, 0x0000, 0x0000, 0xa954, 0x0000, 0x0000, /*0xd0-0xdf*/
3436 };
3437 static const unsigned short cp936ext_page09c0[20904] = {
3438 0xd2bb, 0xb6a1, 0x8140, 0xc6df, 0x8141, 0x8142, 0x8143, 0xcdf2, /*0x00-0x07*/
3439 0xd5c9, 0xc8fd, 0xc9cf, 0xcfc2, 0xd8a2, 0xb2bb, 0xd3eb, 0x8144, /*0x08-0x0f*/
3440 0xd8a4, 0xb3f3, 0x8145, 0xd7a8, 0xc7d2, 0xd8a7, 0xcac0, 0x8146, /*0x10-0x17*/
3441 0xc7f0, 0xb1fb, 0xd2b5, 0xb4d4, 0xb6ab, 0xcbbf, 0xd8a9, 0x8147, /*0x18-0x1f*/
3442 0x8148, 0x8149, 0xb6aa, 0x814a, 0xc1bd, 0xd1cf, 0x814b, 0xc9a5, /*0x20-0x27*/
3443 0xd8ad, 0x814c, 0xb8f6, 0xd1be, 0xe3dc, 0xd6d0, 0x814d, 0x814e, /*0x28-0x2f*/
3444 0xb7e1, 0x814f, 0xb4ae, 0x8150, 0xcd1d, 0x8151, 0xd8bc, 0x8152, /*0x30-0x37*/
3445 0xcde8, 0xb5a4, 0xceaa, 0xd6f7, 0x8153, 0xc0f6, 0xbded, 0xd8af, /*0x38-0x3f*/
3446 0x8154, 0x8155, 0x8156, 0xc4cb, 0x8157, 0xbec3, 0x8158, 0xd8b1, /*0x40-0x47*/
3447 0xc3b4, 0xd2e5, 0x8159, 0xd6ae, 0xcda2, 0xd5a7, 0xbaf5, 0xb7a6, /*0x48-0x4f*/
3448 0xc0d6, 0x815a, 0xc6b9, 0xc5d2, 0xc7c7, 0x815b, 0xb9d4, 0x815c, /*0x50-0x57*/
3449 0xb3cb, 0xd2d2, 0x815d, 0x815e, 0xd8bf, 0xbec5, 0xc6f2, 0xd2b2, /*0x58-0x5f*/
3450 0xcfb0, 0xcfe7, 0x815f, 0x8160, 0x8161, 0x8162, 0xcae9, 0x8163, /*0x60-0x67*/
3451 0x8164, 0xd8c0, 0x8165, 0x8166, 0x8167, 0x8168, 0x8169, 0x816a, /*0x68-0x6f*/
3452 0xc2f2, 0xc2d2, 0x816b, 0xc8e9, 0x816c, 0x816d, 0x816e, 0x816f, /*0x70-0x77*/
3453 0x8170, 0x8171, 0x8172, 0x8173, 0x8174, 0x8175, 0xc7ac, 0x8176, /*0x78-0x7f*/
3454 0x8177, 0x8178, 0x8179, 0x817a, 0x817b, 0x817c, 0xc1cb, 0x817d, /*0x80-0x87*/
3455 0xd3e8, 0xd5f9, 0x817e, 0xcac2, 0xb6fe, 0xd8a1, 0xd3da, 0xbff7, /*0x88-0x8f*/
3456 0x8180, 0xd4c6, 0xbba5, 0xd8c1, 0xcce5, 0xbeae, 0x8181, 0x8182, /*0x90-0x97*/
3457 0xd8a8, 0x8183, 0xd1c7, 0xd0a9, 0x8184, 0x8185, 0x8186, 0xd8bd, /*0x98-0x9f*/
3458 0xd9ef, 0xcdf6, 0xbfba, 0x8187, 0xbdab, 0x8188, 0xd2e0, 0xb2fa, /*0xa0-0xaf*/
3459 0xbae0, 0xc4b6, 0x8189, 0xcfed, 0xbea9, 0xcda4, 0xc1c1, 0x818a, /*0xa8-0xaf*/
3460 0x818b, 0x818c, 0xc7d7, 0xd9f1, 0x818d, 0xd9f4, 0x818e, 0x818f, /*0xb0-0xb7*/
3461 0x818f, 0x8190, 0xc8cb, 0xd8e9, 0x8191, 0x8192, 0x8193, 0xd2da, /*0xb8-0xbf*/
3462 0xcab2, 0xc8ca, 0xd8ec, 0xd8ea, 0xd8c6, 0xbdf6, 0xc6cd, 0xb3f0, /*0xc0-0xc7*/
3463 0x8194, 0xd8eb, 0xbdf1, 0xbde9, 0x8195, 0xc8d4, 0xb4d3, 0x8196, /*0xc8-0xcf*/
3464 0x8197, 0xc2d8, 0x8198, 0xb2d6, 0xd7d0, 0xcacb, 0xcbbf, 0xd5cc, /*0xd0-0xd7*/
3465 0xb8b6, 0xcfc9, 0x8199, 0x819a, 0x819b, 0xd9da, 0xd8f0, 0xc7aa, /*0xd8-0xdf*/
3466 0x819c, 0xd8ee, 0x819d, 0xb4fa, 0xclee, 0xd2d4, 0x819e, 0x819f, /*0xe0-0xe7*/
3467 0xd8ed, 0x81a0, 0xd2c7, 0xd8ef, 0xc3c7, 0x81a1, 0x81a2, 0x81a3, /*0xe8-0xef*/
3468 0xd1f6, 0x81a4, 0xd6d9, 0xd8f2, 0x81a5, 0xd8f5, 0xbcf6, 0xbcd6, /*0xf0-0xf7*/
3469 0x81a6, 0x81a7, 0x81a8, 0xc8ce, 0x81a9, 0xb7dd, 0x81aa, 0xb7c2, /*0xf8-0xff*/
3470 /* 0xaf00 */
3471 0x81ab, 0xc6f3, 0x81ac, 0x81ad, 0x81ae, 0x81af, 0x81b0, 0x81b1, /*0x00-0x07*/
3472 0x81b2, 0xd8f8, 0xd2c1, 0x81b3, 0x81b4, 0xcce9, 0xbcbf, 0xb7fc, /*0x08-0x0f*/
3473 0xb7a5, 0xd0dd, 0x81b5, 0x81b6, 0x81b7, 0x81b8, 0x81b9, 0xd6da, /*0x10-0x17*/
3474 0xd3c5, 0xbbef, 0xbbe1, 0xd8f1, 0x81ba, 0x81bb, 0xc9a1, 0xcbeb, /*0x18-0x1f*/
3475 0xb4ab, 0x81bc, 0xd8f3, 0x81bd, 0xc9cb, 0xd8f6, 0xc2d7, 0xd8f7, /*0x20-0x27*/
3476 0x81be, 0x81bf, 0xcbe1, 0xd8f9, 0x81c0, 0x81c1, 0x81c2, 0xb2ae, /*0x28-0x2f*/
3477 0xb9c0, 0x81c3, 0xd9a3, 0x81c4, 0xb0e9, 0x81c5, 0xc1e6, 0x81c6, /*0x30-0x37*/
3478 0xc9ec, 0x81c7, 0xcbc5, 0x81c8, 0xcbc6, 0xd9a4, 0x81c9, 0x81ca, /*0x38-0x3f*/
```



```
4088 0x919c, 0x919d, 0x919e, 0x919f, 0x91a0, 0x91a1, 0xbab6, 0x91a2, /*0xb8-0xbf*/
4089 0x91a3, 0x91a4, 0xb6ae, 0x91a5, 0x91a6, 0x91a7, 0x91a8, 0x91a9, /*0xc0-0xc7*/
4090 0xd0b8, 0x91aa, 0xb0c3, 0xedae, 0x91ab, 0x91ac, 0x91ad, 0x91ae, /*0xc8-0xcf*/
4091 0x91af, 0xedaf, 0xc0c1, 0x91b0, 0xe3c1, 0x91b1, 0x91b2, 0x91b3, /*0xd0-0xd7*/
4092 0x91b4, 0x91b5, 0x91b6, 0x91b7, 0x91b8, 0x91b9, 0x91ba, 0x91bb, /*0xd8-0xdf*/
4093 0x91bc, 0x91bd, 0x91be, 0x91bf, 0x91c0, 0x91c1, 0xc5b3, 0x91c2, /*0xe0-0xef*/
4094 0x91c3, 0x91c4, 0x91c5, 0x91c6, 0x91c7, 0x91c8, 0x91c9, 0x91ca, /*0xe8-0xef*/
4095 0x91cb, 0x91cc, 0x91cd, 0x91ce, 0x91cf, 0xe3c2, 0x91d0, 0x91d1, /*0xf0-0xf7*/
4096 0x91d2, 0x91d3, 0x91d4, 0x91d5, 0x91d6, 0x91d7, 0x91d8, 0xdc2, /*0xf8-0xff*/
4097 /* 0x6200 */
4098 0x91d9, 0x91da, 0x91db, 0x91dc, 0x91dd, 0x91de, 0xedb0, 0x91df, /*0x00-0x07*/
4099 0xb8ea, 0x91e0, 0xccec, 0xea7, 0xd0e7, 0xcacf9, 0xc8d6, 0xcfb7, /*0x08-0x0f*/
4100 0xb3c9, 0xcd2, 0xbde4, 0x91e1, 0x91e2, 0xe3de, 0xbbf2, 0xea8, /*0x10-0x17*/
4101 0xd5bd, 0x91e3, 0xc6dd, 0xea9, 0x91e4, 0x91e5, 0x91e6, 0xea, /*0x18-0x1f*/
4102 0x91e7, 0xeaec, 0xeaab, 0x91e8, 0xea, 0xeaad, 0x91e9, 0x91ea, /*0x20-0x27*/
4103 0x91eb, 0x91ec, 0xbdd8, 0x91ed, 0xeaaf, 0x91ee, 0xc2be, 0x91ef, /*0x28-0x2f*/
4104 0x91f0, 0x91f1, 0x91f2, 0xb4c1, 0xb4f7, 0x91f3, 0x91f4, 0xbba7, /*0x30-0x37*/
4105 0x91f5, 0x91f6, 0x91f7, 0x91f8, 0x91f9, 0xece6, 0xece5, 0xb7bf, /*0x38-0x3f*/
4106 0xcbf9, 0xb1e2, 0x91fa, 0xece7, 0x91fb, 0x91fc, 0x91fd, 0xc9c8, /*0x40-0x47*/
4107 0xece8, 0xece9, 0x91fe, 0xad6, 0xded0, 0xb2c5, 0xd4fa, 0x9240, /*0x48-0x4f*/
4108 0x9241, 0xc6cb, 0xb0c7, 0xb4f2, 0xc8d3, 0x9242, 0x9243, 0x9244, /*0x50-0x57*/
4109 0xcdd0, 0x9245, 0x9246, 0xbfb8, 0x9247, 0x9248, 0x9249, 0x924a, /*0x58-0x5f*/
4110 0x924b, 0x924c, 0x924d, 0xbfdb, 0x924e, 0x924f, 0xc7a4, 0xd6b4, /*0x60-0x67*/
4111 0x9250, 0xc0a9, 0xded1, 0xc9a8, 0xd1ef, 0xc5a4, 0xb0e7, 0xb3b6, /*0x68-0x6f*/
4112 0xc8c5, 0x9251, 0x9252, 0xb0e2, 0x9253, 0x9254, 0xb7f6, 0x9255, /*0x70-0x77*/
4113 0x9256, 0xc5fa, 0x9257, 0x9258, 0xb6f3, 0x9259, 0xd5d2, 0xb3d0, /*0x78-0x7f*/
4114 0xbcbc, 0x925a, 0x925b, 0x925c, 0xb3ad, 0x925d, 0x925e, 0x925f, /*0x80-0x87*/
4115 0x9260, 0xbfef, 0xb0d1, 0x9261, 0x9262, 0x9263, 0x9264, 0x9265, /*0x88-0x8f*/
4116 0x9266, 0xd2d6, 0xcae3, 0xd7a5, 0x9267, 0xcdb6, 0xb6b6, 0xbfb9, /*0x90-0x97*/
4117 0xd5db, 0xd2e8, 0xb8a7, 0xc5d7, 0x9269, 0x926a, 0x926b, 0xded2, /*0x98-0x9f*/
4118 0xbfd9, 0xc2d5, 0xc7c0, 0x926c, 0xbba4, 0xb1a8, 0x926d, 0x926e, /*0xa0-0xaf*/
4119 0xc5ea, 0x926f, 0x9270, 0xc5fb, 0xcac7, 0x9271, 0x9272, 0x9273, /*0xa8-0xaf*/
4120 0x9274, 0xb1a7, 0x9275, 0x9276, 0x9277, 0xb5d6, 0x9278, 0x9279, /*0xb0-0xbf*/
4121 0x927a, 0xc4a8, 0x927b, 0xded3, 0xd1ba, 0xb3e9, 0x927c, 0xc3f2, /*0xb8-0xbf*/
4122 0x927d, 0x927e, 0xb7f7, 0x9280, 0xd6f4, 0xb5a3, 0xb2f0, 0xc4b4, /*0xc0-0xc7*/
4123 0xc4e9, 0xc0ad, 0xded4, 0x9281, 0xb0e8, 0xc5c4, 0xc1e0, 0x9282, /*0xc8-0xcf*/
4124 0xb9d5, 0x9283, 0xbedc, 0xcd8, 0xb0ce, 0x9284, 0xcdcf, 0xded6, /*0xd0-0xd7*/
4125 0xbed0, 0xd7be, 0xded5, 0xd5d0, 0xb0dd, 0x9285, 0x9286, 0xc4e2, /*0xd8-0xdf*/
4126 0x9287, 0x9288, 0xc2a3, 0xbcf0, 0x9289, 0xd3b5, 0xc0b9, 0xc5a1, /*0xe0-0xef*/
4127 0xb2a6, 0xd4f1, 0x928a, 0x928b, 0xc0a8, 0xcac3, 0xded7, 0xd5fc, /*0xe8-0xef*/
4128 0x928c, 0xb9b0, 0x928d, 0xc8ad, 0xcba9, 0x928e, 0xded9, 0xbfb, /*0xf0-0xf7*/
4129 0x928f, 0x9290, 0x9291, 0x9292, 0xc6b4, 0xd7a7, 0xcab0, 0xc4c3, /*0xf8-0xff*/
4130 /* 0x6300 */
4131 0x9293, 0xb3d6, 0xb9d2, 0x9294, 0x9295, 0x9296, 0x9297, 0xd6b8, /*0x00-0x07*/
4132 0xea7, 0xb0b4, 0x9298, 0x9299, 0x929a, 0x929b, 0xbf6, 0x929c, /*0x08-0x0f*/
4133 0x929d, 0xccf4, 0x929e, 0x929f, 0x92a0, 0x92a1, 0xcdda, 0x92a2, /*0x10-0x17*/
4134 0x92a3, 0x92a4, 0xd6bf, 0xc2ce, 0x92a5, 0xcce, 0xcac7, 0xd0ae, /*0x18-0x1f*/
4135 0xc4d3, 0xb5b2, 0xded8, 0xd5f5, 0xbcb7, 0xbbd3, 0x92a6, 0x92a7, /*0x20-0x27*/
4136 0xb0a4, 0x92a8, 0xc5b2, 0xb4ec, 0x92a9, 0x92aa, 0x92ab, 0xd5f1, /*0x28-0x2f*/
4137 0x92ac, 0x92ad, 0xeafd, 0x92ae, 0x92af, 0x92b0, 0x92b1, 0x92b2, /*0x30-0x37*/
4138 0x92b3, 0xdeda, 0xcda6, 0x92b4, 0x92b5, 0xcdec, 0x92b6, 0x92b7, /*0x38-0x3f*/
4139 0x92b8, 0x92b9, 0xcce6, 0xdedc, 0x92ba, 0xcdb1, 0xc0a6, 0x92bb, /*0x40-0x47*/
4140 0x92bc, 0xd7bd, 0x92bd, 0xded, 0xb0c6, 0xbab4, 0xc9d3, 0xc4f3, /*0x48-0x4f*/
4141 0xbec, 0x92be, 0x92bf, 0x92c0, 0x92c1, 0xb2b6, 0x92c2, 0x92c3, /*0x50-0x57*/
4142 0x92c4, 0x92c5, 0x92c6, 0x92c7, 0x92c8, 0x92c9, 0xc0cc, 0xcbf0, /*0x58-0x5f*/
4143 0x92ca, 0xbcf1, 0xbbbb, 0xb5b7, 0x92cd, 0x92ce, 0xc5f5, /*0x60-0x67*/
4144 0x92ce, 0xdee6, 0x92cf, 0x92d0, 0x92d1, 0xded3, 0xbedd, 0x92d2, /*0x68-0x6f*/
4145 0x92d3, 0xdedf, 0x92d4, 0x92d5, 0x92d6, 0x92d7, 0xb4b7, 0xbddd, /*0x70-0x77*/
4146 0x92d8, 0x92d9, 0xded, 0xc4ed, 0x92da, 0x92db, 0x92dc, 0x92dd, /*0x78-0x7f*/
4147 0xcfc6, 0x92de, 0xb5e0, 0x92df, 0x92e0, 0x92e1, 0x92e2, 0xb6de, /*0x80-0x87*/
4148 0xcada, 0xb5f4, 0xddee5, 0x92e3, 0xd5c6, 0x92e4, 0xdde1, 0xcdd, /*0x88-0x8f*/
4149 0xc6fe, 0x92e5, 0xc5c5, 0x92e6, 0x92e7, 0x92e8, 0xd2b4, 0x92e9, /*0x90-0x97*/
4150 0xbf2, 0x92ea, 0x92eb, 0x92ec, 0x92ed, 0x92ee, 0x92ef, 0x92f0, /*0x98-0x9f*/
4151 0xc2d3, 0x92f1, 0xccbd, 0xb3b8, 0x92f2, 0xbdd3, 0x92f3, 0xbfd8, /*0xa0-0xaf*/
4152 0xcdc6, 0xd1da, 0xb4eb, 0x92f4, 0xdde4, 0xdedd, 0xdde7, 0x92f5, /*0xa8-0xaf*/
4153 0xea7, 0x92f6, 0x92f7, 0xc2b0, 0xdde2, 0x92f8, 0x92f9, 0xd6c0, /*0xb0-0xbf*/
4154 0xb5a7, 0x92fa, 0xb2f4, 0x92fb, 0xdde8, 0x92fc, 0xdfe2, 0x92fd, /*0xb8-0xbf*/
4155 0x92fe, 0x9340, 0x9341, 0x9342, 0xddeed, 0x9343, 0xdfe1, 0x9344, /*0xc0-0xc7*/
4156 0x9345, 0xc8e0, 0x9346, 0x9347, 0x9348, 0xd7e1, 0xdfeef, 0xc3e8, /*0xc8-0xcf*/
4157 0xcce1, 0x9349, 0xb2e5, 0x934a, 0x934b, 0x934c, 0xd2be, 0x934d, /*0xd0-0xd7*/
4158 0x934e, 0x934f, 0x9350, 0x9351, 0x9352, 0x9353, 0xddee, 0x9354, /*0xd8-0xdf*/
4159 0xddeb, 0xcce5, 0x9355, 0xb4a7, 0x9356, 0x9357, 0x9358, 0x9359, /*0xe0-0xef*/
4160 0x935a, 0xbfab, 0xbebe, 0x935b, 0x935c, 0xbdd2, 0x935d, 0x935e, /*0xe8-0xef*/
4161 0x935f, 0x9360, 0xdde9, 0x9361, 0xd4ae, 0x9362, 0xdede, 0x9363, /*0xf0-0xf7*/
4162 0xddea, 0x9364, 0x9365, 0x9366, 0x9367, 0xc0bf, 0x9368, 0xddeec, /*0xf8-0xff*/
4163 /* 0x6400 */
4164 0xb2f3, 0xb8e9, 0xc2a7, 0x9369, 0x936a, 0xbdc1, 0x936b, 0x936c, /*0x00-0x07*/
4165 0x936d, 0x936e, 0x936f, 0xdfe5, 0xdfe8, 0x9370, 0x9371, 0xb2ab, /*0x08-0x0f*/
4166 0xb4a4, 0x9372, 0x9373, 0xb4ea, 0xc9a6, 0x9374, 0x9375, 0x9376, /*0x10-0x17*/
4167 0x9377, 0x9378, 0x9379, 0xdfe6, 0xcdb1, 0x937a, 0xb8e3, 0x937b, /*0x18-0x1f*/
4168 0xdfe7, 0xdfe8, 0x937c, 0x937d, 0x937e, 0x9380, 0xdfe9, 0x9381, /*0x20-0x27*/
4169 0x9382, 0x9383, 0xcce2, 0x9384, 0xb0e1, 0xb4ee, 0x9385, 0x9386, /*0x28-0x2f*/
4170 0x9387, 0x9388, 0x9389, 0x938a, 0xe5ba, 0x938b, 0x938c, 0x938d, /*0x30-0x37*/
4171 0x938e, 0x938f, 0xd0af, 0x9390, 0x9391, 0xb2eb, 0x9392, 0xeba1, /*0x38-0x3f*/
4172 0x9393, 0xdfe4, 0x9394, 0x9395, 0xc9e3, 0xdfe3, 0xb0da, 0xd2a1, /*0x40-0x47*/
4173 0xb1f7, 0x9396, 0xccaf, 0x9397, 0x9398, 0x9399, 0x939a, 0x939b, /*0x48-0x4f*/
4174 0x939c, 0x939d, 0xdfe0, 0x939e, 0xcba4, 0x939f, 0x93a0, 0x93a1, /*0x50-0x57*/
```

```

4175 0xd5aa, 0x93a2, 0x93a3, 0x93a4, 0x93a5, 0x93a6, 0xdefb, 0x93a7, /*0x58-0x5f*/
4176 0x93a8, 0x93a9, 0x93aa, 0x93ab, 0x93ac, 0x93ad, 0x93ae, 0xb4dd, /*0x60-0x67*/
4177 0x93af, 0xc4a6, 0x93b0, 0x93b1, 0x93b2, 0xdefd, 0x93b3, 0x93b4, /*0x68-0x6f*/
4178 0x93b5, 0x93b6, 0x93b7, 0x93b8, 0x93b9, 0x93ba, 0x93bb, 0x93bc, /*0x70-0x77*/
4179 0xc3fe, 0xc4a1, 0xdfa1, 0x93bd, 0x93be, 0x93bf, 0x93c0, 0x93c1, /*0x78-0x7f*/
4180 0x93c2, 0x93c3, 0x93c4, 0xc1cc, 0x93c4, 0xdefc, 0x93c5, 0xc6b2, /*0x80-0x87*/
4181 0x93c6, 0x93c7, 0x93c8, 0x93c9, 0x93ca, 0x93cb, 0x93cc, 0x93cd, /*0x88-0x8f*/
4182 0x93ce, 0xb3c5, 0xc8f6, 0x93cf, 0x93d0, 0xcbb8, 0xdefe, 0x93d1, /*0x90-0x97*/
4183 0x93d2, 0xdfa4, 0x93d3, 0x93d4, 0x93d5, 0x93d6, 0xd7b2, 0x93d7, /*0x98-0x9f*/
4184 0x93d8, 0x93d9, 0x93da, 0x93db, 0xb3b7, 0x93dc, 0x93dd, 0x93de, /*0xa0-0xa7*/
4185 0x93df, 0xc1c3, 0x93e0, 0x93e1, 0xc7cb, 0xb2a5, 0xb4e9, 0x93e2, /*0xa8-0xaf*/
4186 0xd7ab, 0x93e3, 0x93e4, 0x93e5, 0x93e6, 0xc4ec, 0x93e7, 0xdfa2, /*0xb0-0xb7*/
4187 0xdfa3, 0x93e8, 0xdfa5, 0x93e9, 0xbab3, 0x93ea, 0x93eb, 0x93ec, /*0xb8-0xbf*/
4188 0xdfa6, 0x93ed, 0xc0de, 0x93ee, 0x93ef, 0xc9c3, 0x93f0, 0x93f1, /*0xc0-0xc7*/
4189 0x93f2, 0x93f3, 0x93f4, 0x93f5, 0x93f6, 0xb2d9, 0xc7e6, 0x93f7, /*0xc8-0xcf*/
4190 0xdfa7, 0x93f8, 0xc7dc, 0x93f9, 0x93fa, 0x93fb, 0x93fc, 0xdfa8, /*0xd0-0xd7*/
4191 0xeba2, 0x93fd, 0x93fe, 0x9400, 0x9441, 0x9442, 0xcdb3, 0x9443, /*0xd8-0xdf*/
4192 0x9444, 0x9445, 0x944a, 0x944b, 0x944c, 0x944d, 0x944e, 0x944f, /*0xe0-0xef*/
4193 0x9449, 0x944a, 0x944b, 0x944c, 0x944d, 0x944e, 0x944f, 0x9450, /*0xe8-0xef*/
4194 0x9451, 0x9452, 0x9453, 0x9454, 0x9455, 0x9456, 0x9457, 0x9458, /*0xf0-0xff*/
4195 0x9459, 0x945a, 0x945b, 0x945c, 0x945d, 0x945e, 0x945f, 0x9460, /*0xf8-0xff*/
4196 /* 0x6500 */
4197 0xc5ca, 0x9461, 0x9462, 0x9463, 0x9464, 0x9465, 0x9466, 0x9467, /*0x00-0x07*/
4198 0x9468, 0xdfab, 0x9469, 0x946a, 0x946b, 0x946c, 0x946d, 0x946e, /*0x08-0x0f*/
4199 0x946f, 0x9470, 0xd4dc, 0x9471, 0x9472, 0x9473, 0x9474, 0x9475, /*0x10-0x17*/
4200 0xc8c1, 0x9476, 0x9477, 0x9478, 0x9479, 0x947a, 0x947b, 0x947c, /*0x18-0x1f*/
4201 0x947d, 0x947e, 0x9480, 0x9481, 0x9482, 0xdfac, 0x9483, 0x9484, /*0x20-0x27*/
4202 0x9485, 0x9486, 0x9487, 0xbf00, 0x9488, 0x9489, 0xdfad, 0xd6a7, /*0x28-0x2f*/
4203 0x948a, 0x948b, 0x948c, 0x948d, 0xeab7, 0xebb6, 0xcad5, 0x948e, /*0x30-0x37*/
4204 0xd8fc, 0xb8c4, 0x948f, 0xb9a5, 0x9490, 0x9491, 0xb7c5, 0xd5fe, /*0x38-0x3f*/
4205 0x9492, 0x9493, 0x9494, 0x9495, 0x9496, 0xb9ca, 0x9497, 0x9498, /*0x40-0x47*/
4206 0xd0a7, 0xf4cd, 0x9499, 0x949a, 0xb5d0, 0x949b, 0x949c, 0xc3f4, /*0x48-0x4f*/
4207 0x949d, 0xbec8, 0x949e, 0x949f, 0x94a0, 0xebb7, 0xb0bd, 0x94a1, /*0x50-0x57*/
4208 0x94a2, 0xbdcc, 0x94a3, 0xc1b2, 0x94a4, 0xb1d6, 0xb3a8, 0x94a5, /*0x58-0x5f*/
4209 0x94a6, 0x94a7, 0xb8d2, 0xc9a2, 0x94a8, 0x94a9, 0xb6d8, 0x94aa, /*0x60-0x67*/
4210 0x94ab, 0x94ac, 0x94ad, 0xebb8, 0xebb4, 0x94ae, 0x94af, 0x94b0, /*0x68-0x6f*/
4211 0xcafd, 0x94b1, 0xc7c3, 0x94b2, 0xd5fb, 0x94b3, 0x94b4, 0xb7f3, /*0x70-0x77*/
4212 0x94b5, 0x94b6, 0x94b7, 0x94b8, 0x94b9, 0x94ba, 0x94bb, 0x94bc, /*0x78-0x7f*/
4213 0x94bd, 0x94be, 0x94bf, 0x94c0, 0x94c1, 0x94c2, 0x94c3, 0xccec4, /*0x80-0x87*/
4214 0x94c4, 0x94c5, 0x94c6, 0xd5ab, 0xb1f3, 0x94c7, 0x94c8, 0x94c9, /*0x88-0x8f*/
4215 0xecb3, 0xb0df, 0x94ca, 0xecb5, 0x94cb, 0x94cc, 0x94cd, 0xb6b7, /*0x90-0x97*/
4216 0x94ce, 0xc1cf, 0x94cf, 0x94d0, 0xf5fa, 0xd0b1, 0x94d1, 0xd5e5, /*0x98-0x9f*/
4217 0x94d2, 0xc3cd, 0x94d3, 0x94d4, 0xbdef, 0xb3e2, 0x94d5, 0xb8ab, /*0xa0-0xa7*/
4218 0x94d6, 0xd5b6, 0x94d7, 0xedbd, 0x94d8, 0xb6cf, 0x94d9, 0xcbb9, /*0xa8-0xaf*/
4219 0xd0c2, 0x94da, 0x94db, 0x94dc, 0x94dd, 0x94de, 0x94df, 0x94e0, /*0xb0-0xbf*/
4220 0x94e1, 0xb7bd, 0x94e2, 0x94e3, 0xecb6, 0xc9a9, 0x94e4, 0x94e5, /*0xb8-0xbf*/
4221 0x94e6, 0xc5d4, 0x94e7, 0xecb9, 0xecb8, 0xc2c3, 0xecb7, 0x94e8, /*0xc0-0xc7*/
4222 0x94e9, 0x94ea, 0x94eb, 0xd0fd, 0xecba, 0x94ec, 0xecbb, 0xd7e5, /*0xc8-0xcf*/
4223 0x94ed, 0x94ee, 0xecbc, 0x94ef, 0x94f0, 0x94f1, 0xecbd, 0xc6ec, /*0xd0-0xd7*/
4224 0x94f2, 0x94f3, 0x94f4, 0x94f5, 0x94f6, 0x94f7, 0x94f8, 0x94f9, /*0xd8-0xdf*/
4225 0xc3de, 0x94fa, 0xbcc8, 0x94fb, 0x94fc, 0xc8d5, 0xb5a9, 0xbec9, /*0xe0-0xef*/
4226 0xd6bc, 0xd4e7, 0x94fd, 0x94fe, 0xd1ae, 0xd0f1, 0xeab8, 0xeab9, /*0xe8-0xef*/
4227 0xeaba, 0xbab5, 0x9540, 0x9541, 0x9542, 0x9543, 0xcab1, 0xbff5, /*0xf0-0xff*/
4228 0x9544, 0x9545, 0xcdfa, 0x9546, 0x9547, 0x9548, 0x9549, 0x954a, /*0xf8-0xff*/
4229 /* 0x6600 */
4230 0xeac0, 0x954b, 0xb0ba, 0xeabe, 0x954c, 0x954d, 0xc0a5, 0x954e, /*0x00-0x07*/
4231 0x954f, 0x9550, 0xeabb, 0x9551, 0xb2fd, 0x9552, 0xc3f7, 0xbbe8, /*0x08-0x0f*/
4232 0x9553, 0x9554, 0x9555, 0xd2d7, 0xcef4, 0xeabf, 0x9556, 0x9557, /*0x10-0x17*/
4233 0x9558, 0xeabc, 0x9559, 0x955a, 0x955b, 0xeac3, 0x955c, 0xd0c7, /*0x18-0x1f*/
4234 0xd3b3, 0x955d, 0x955e, 0x955f, 0x9560, 0xb4ba, 0x9561, 0xc3c1, /*0x20-0x27*/
4235 0xd7f2, 0x9562, 0x9563, 0x9564, 0x9565, 0xd5d1, 0x9566, 0xcac7, /*0x28-0x2f*/
4236 0x9567, 0xeac5, 0x9568, 0x9569, 0xeac4, 0xeac7, 0xeac6, 0x956a, /*0x30-0x37*/
4237 0x956b, 0x956c, 0x956d, 0x956e, 0xd6e7, 0x956f, 0xcfd4, 0x9570, /*0x38-0x3f*/
4238 0x9571, 0xeac8, 0x9572, 0xbfce, 0x9573, 0x9574, 0x9575, 0x9576, /*0x40-0x47*/
4239 0x9577, 0x9578, 0x9579, 0xbdfa, 0xc9ce, 0x957a, 0x957b, 0xeacc, /*0x48-0x4f*/
4240 0x957c, 0x957d, 0xc9b9, 0xcffe, 0xeaca, 0xd4ce, 0xeacd, 0xeacf, /*0x50-0x57*/
4241 0x957e, 0x9580, 0xcded, 0x9581, 0x9582, 0x9583, 0x9584, 0xeac9, /*0x58-0x5f*/
4242 0x9585, 0xeace, 0x9586, 0x9587, 0xccee, 0x9588, 0xbbbe, 0x9589, /*0x60-0x67*/
4243 0xb3bf, 0x958a, 0x958b, 0x958c, 0x958d, 0x958e, 0xc6d5, 0xbbe0, /*0x68-0x6f*/
4244 0xc3fa, 0x958f, 0x9590, 0x9591, 0xc7e7, 0x9592, 0xbea7, 0xead0, /*0x70-0x77*/
4245 0x9593, 0x9594, 0xd6c7, 0x9595, 0x9596, 0x9597, 0xc1c0, 0x9598, /*0x78-0x7f*/
4246 0x9599, 0x959a, 0xd4dd, 0x959b, 0xead1, 0x959c, 0x959d, 0xcfbe, /*0x80-0x87*/
4247 0x959e, 0x959f, 0x95a0, 0x95a1, 0xead2, 0x95a2, 0x95a3, 0x95a4, /*0x88-0x8f*/
4248 0x95a5, 0xc9ae, 0x95a6, 0x95a7, 0x95a8, 0x95a9, 0xc5af, 0xb0b5, /*0x90-0x97*/
4249 0x95aa, 0x95ab, 0x95ac, 0x95ad, 0x95ae, 0xead4, 0x95af, 0x95b0, /*0x98-0x9f*/
4250 0x95b1, 0x95b2, 0x95b3, 0x95b4, 0x95b5, 0x95b6, 0x95b7, 0xead3, /*0xa0-0xa7*/
4251 0xf4df, 0x95b8, 0x95b9, 0x95ba, 0x95bb, 0x95bc, 0xc4ba, 0x95bd, /*0xa8-0xaf*/
4252 0x95be, 0x95bf, 0x95c0, 0x95c1, 0xb1a9, 0x95c2, 0x95c3, 0x95c4, /*0xb0-0xb7*/
4253 0x95c5, 0xe5df, 0x95c6, 0x95c7, 0x95c8, 0x95c9, 0xead5, 0x95ca, /*0xb8-0xbf*/
4254 0x95cb, 0x95cc, 0x95cd, 0x95ce, 0x95cf, 0x95d0, 0x95d1, 0x95d2, /*0xc0-0xc7*/
4255 0x95d3, 0x95d4, 0x95d5, 0x95d6, 0x95d7, 0x95d8, 0x95d9, 0x95da, /*0xc8-0xcf*/
4256 0x95db, 0x95dc, 0x95dd, 0x95de, 0x95df, 0x95e0, 0x95e1, 0x95e2, /*0xd0-0xd7*/
4257 0x95e3, 0xc9af, 0x95e4, 0xeade, 0xeade, 0xc6d8, 0x95e5, 0x95e6, /*0xd8-0xdf*/
4258 0x95e7, 0x95e8, 0x95e9, 0x95ea, 0x95eb, 0x95ec, 0xeade, 0x95ed, /*0xe0-0xef*/
4259 0x95ee, 0xeade, 0x95ef, 0x95f0, 0x95f1, 0x95f2, 0x95f3, 0x95f4, /*0xe8-0xef*/
4260 0xd4bb, 0x95f5, 0xc7fa, 0xd2b7, 0xb8fc, 0x95f6, 0x95f7, 0xeac2, /*0xf0-0xff*/
4261 0x95f8, 0xb2dc, 0x95f9, 0x95fa, 0xc2fc, 0x95fb, 0xd4f8, 0xcce6, /*0xf8-0xff*/

```

```
4262 /* 0x6700 */
4263 0xd7ee, 0x95fd, 0x95fe, 0x9640, 0x9641, 0x9642, 0x9643, /*0x00-0x07*/
4264 0xd4c2, 0xd3d0, 0xebc3, 0xc5f3, 0x9644, 0xb7fe, 0x9645, 0x9646, /*0x08-0x0f*/
4265 0xebd4, 0x9647, 0x9648, 0x9649, 0xcbb7, 0xebde, 0x964a, 0xc0ca, /*0x10-0x17*/
4266 0x964b, 0x964c, 0x964d, 0xcdfb, 0x964e, 0xb3af, 0x964f, 0xc6da, /*0x18-0x1f*/
4267 0x9650, 0x9651, 0x9652, 0x9653, 0x9654, 0x9655, 0xebfc, 0x9656, /*0x20-0x27*/
4268 0xc4be, 0x9657, 0xceb4, 0xc4a9, 0xb1be, 0xd4fd, 0x9658, 0xcaf5, /*0x28-0x2f*/
4269 0x9659, 0xd6ec, 0x965a, 0x965b, 0xc6d3, 0xb6e4, 0x965c, 0x965d, /*0x30-0x37*/
4270 0x965e, 0x965f, 0xbbfa, 0x9660, 0x9661, 0xd0e0, 0x9662, 0x9663, /*0x38-0x3f*/
4271 0xc9b1, 0x9664, 0xd4d3, 0xc8a8, 0x9665, 0x9666, 0xb8cb, 0x9667, /*0x40-0x47*/
4272 0xe8be, 0xc9bc, 0x9668, 0x9669, 0xe8bb, 0x966a, 0xc0ee, 0xd0d3, /*0x48-0x4f*/
4273 0xb2c4, 0xb4e5, 0x966b, 0xe8bc, 0x966c, 0x966d, 0xd5c8, 0x966e, /*0x50-0x57*/
4274 0x966f, 0x9670, 0x9671, 0x9672, 0xb6c5, 0x9673, 0xe8bd, 0xcaf8, /*0x58-0x5f*/
4275 0xb8dc, 0xccf5, 0x9674, 0x9675, 0x9676, 0xc0b4, 0x9677, 0x9678, /*0x60-0x6f*/
4276 0xd1ee, 0xe8bf, 0xe8c2, 0x9679, 0x967a, 0xbabc, 0x967b, 0xb1ad, /*0x68-0x6f*/
4277 0xbddc, 0x967c, 0xeabd, 0xe8c3, 0x967d, 0xe8c6, 0x967e, 0xe8cb, /*0x70-0x77*/
4278 0x9680, 0x9681, 0x9682, 0x9683, 0xe8cc, 0x9684, 0xcbc9, 0xb0e5, /*0x78-0x7f*/
4279 0x9685, 0xbcab, 0x9686, 0x9687, 0xb9b9, 0x9688, 0x9689, 0xe8c1, /*0x80-0x87*/
4280 0x968a, 0xcdf7, 0x968b, 0xe8ca, 0x968c, 0x968d, 0x968e, 0x968f, /*0x88-0x8f*/
4281 0xcef6, 0x9690, 0x9691, 0x9692, 0x9693, 0xd5ed, 0x9694, 0xc1de, /*0x90-0x97*/
4282 0xe8c4, 0x9695, 0xc3b6, 0x9696, 0xb9fb, 0xd6a6, 0xe8c8, 0x9697, /*0x98-0x9f*/
4283 0x9698, 0x9699, 0xcae0, 0xd4e6, 0x969a, 0xe8c0, 0x969b, 0xe8c5, /*0xa0-0xa7*/
4284 0xe8c7, 0x969c, 0xc7b9, 0xb7e3, 0x969d, 0xe8c9, 0x969e, 0xbfd0, /*0xa8-0xaf*/
4285 0xe8d2, 0x969f, 0x96a0, 0xe8d7, 0x96a1, 0xe8d5, 0xbcdc, 0xbccf, /*0xb0-0xbf*/
4286 0xe8db, 0x96a2, 0x96a3, 0x96a4, 0x96a5, 0x96a6, 0x96a7, 0x96a8, /*0xb8-0xbf*/
4287 0x96a9, 0xe8de, 0x96aa, 0xe8da, 0xb1fa, 0x96ab, 0x96ac, 0x96ad, /*0xc0-0xcf*/
4288 0x96ae, 0x96af, 0x96b0, 0x96b1, 0x96b2, 0x96b3, 0x96b4, 0xb0d8, /*0xc8-0xcf*/
4289 0xc4b3, 0xb8cc, 0xc6e2, 0xc8be, 0xc8e1, 0x96b5, 0x96b6, 0x96b7, /*0xd0-0xdf*/
4290 0xe8cf, 0xe8d4, 0xe8d6, 0x96b8, 0xb9f1, 0xe8d8, 0xd7f5, 0x96b9, /*0xd8-0xdf*/
4291 0xc4fb, 0x96ba, 0xe8dc, 0x96bb, 0x96bc, 0xb2e9, 0x96bd, 0x96be, /*0xe0-0xef*/
4292 0x96bf, 0xe8d1, 0x96c0, 0x96c1, 0xbced, 0x96c2, 0x96c3, 0xbfc2, /*0xe8-0xef*/
4293 0xe8cd, 0xd6f9, 0x96c4, 0xc1f8, 0xb2f1, 0x96c5, 0x96c6, 0x96c7, /*0xf0-0xff*/
4294 0x96c8, 0x96c9, 0x96ca, 0x96cb, 0x96cc, 0xe8df, 0x96cd, 0xcac1, /*0xf8-0xff*/
4295 /* 0x6800 */
4296 0xe8d9, 0x96ce, 0x96cf, 0x96d0, 0x96d1, 0xd5a4, 0x96d2, 0xb1ea, /*0x00-0x07*/
4297 0xd5bb, 0xe8ce, 0xe8d0, 0xb6b0, 0xe8d3, 0x96d3, 0xe8dd, 0xc0b8, /*0x08-0x0f*/
4298 0x96d4, 0xcaf7, 0x96d5, 0xcba8, 0x96d6, 0x96d7, 0xc6dc, 0xc0f5, /*0x10-0x17*/
4299 0x96d8, 0x96d9, 0x96da, 0x96db, 0x96dc, 0xe8e9, 0x96dd, 0x96de, /*0x18-0x1f*/
4300 0x96df, 0xd0a3, 0x96e0, 0x96e1, 0x96e2, 0x96e3, 0x96e4, 0x96e5, /*0x20-0x27*/
4301 0x96e6, 0xe8f2, 0xd6ea, 0x96e7, 0x96e8, 0x96e9, 0x96ea, 0x96eb, /*0x28-0x2f*/
4302 0x96ec, 0x96ed, 0xe8e0, 0xe8e1, 0x96ee, 0x96ef, 0x96f0, 0xd1f9, /*0x30-0x37*/
4303 0xbacb, 0xb8f9, 0x96f1, 0x96f2, 0xb8f1, 0xd4d4, 0xe8ef, 0x96f3, /*0x38-0x3f*/
4304 0xe8ee, 0xe8ec, 0xb9f0, 0xcdc2, 0xe8e6, 0xcea6, 0xbff2, 0x96f4, /*0x40-0x47*/
4305 0xb0b8, 0xe8f1, 0xe8f0, 0x96f5, 0xd7c0, 0x96f6, 0xe8e4, 0x96f7, /*0x48-0x4f*/
4306 0xcda9, 0xc9a3, 0x96f8, 0xbdbb, 0xbdbb, 0xe8ea, 0x96f9, 0x96fa, /*0x50-0x57*/
4307 0x96fb, 0x96fc, 0x96fd, 0x96fe, 0x9740, 0x9741, 0x9742, 0x9743, /*0x58-0x5f*/
4308 0xe8e2, 0xe8e3, 0xe8e5, 0xb5b5, 0xe8e7, 0xc7c5, 0xe8eb, 0xe8ed, /*0x60-0x6f*/
4309 0xbdb0, 0xd7ae, 0x9744, 0xe8f8, 0x9745, 0x9746, 0x9747, 0x9748, /*0x68-0x6f*/
4310 0x9749, 0x974a, 0x974b, 0x974c, 0xe8f5, 0x974d, 0xcdb0, 0xe8f6, /*0x70-0x77*/
4311 0x974e, 0x974f, 0x9750, 0x9751, 0x9752, 0x9753, 0x9754, 0x9755, /*0x78-0x7f*/
4312 0x9756, 0xc1ba, 0x9757, 0xe8e8, 0x9758, 0xc3b7, 0xb0f0, 0x9759, /*0x80-0x87*/
4313 0x975a, 0x975b, 0x975c, 0x975d, 0x975e, 0x975f, 0x9760, 0xe8f4, /*0x88-0x8f*/
4314 0x9761, 0x9762, 0x9763, 0xe8f7, 0x9764, 0x9765, 0x9766, 0xb9a3, /*0x90-0x97*/
4315 0x9767, 0x9768, 0x9769, 0x976a, 0x976b, 0x976c, 0x976d, 0x976e, /*0x98-0x9f*/
4316 0x976f, 0x9770, 0xc9d2, 0x9771, 0x9772, 0x9773, 0xc3ce, 0xcce0, /*0xa0-0xa7*/
4317 0xc0e6, 0x9774, 0x9775, 0x9776, 0x9777, 0xcbf3, 0x9778, 0xcddd, /*0xa8-0xaf*/
4318 0xd0b5, 0x9779, 0x977a, 0xcae1, 0x977b, 0xe8f3, 0x977c, 0x977d, /*0xb0-0xbf*/
4319 0x977e, 0x9780, 0x9781, 0x9782, 0x9783, 0x9784, 0x9785, 0x9786, /*0xb8-0xbf*/
4320 0xbcec, 0x9787, 0xe8f9, 0x9788, 0x9789, 0x978a, 0x978b, 0x978c, /*0xc0-0xcf*/
4321 0x978d, 0xc3de, 0x978e, 0xc6e5, 0x978f, 0xb9f7, 0x9790, 0x9791, /*0xc8-0xcf*/
4322 0x9792, 0x9793, 0xb0f4, 0x9794, 0x9795, 0xd7d8, 0x9796, 0x9797, /*0xd0-0xdf*/
4323 0xbcac, 0x9798, 0xc5ef, 0x9799, 0x979a, 0x979b, 0x979c, 0x979d, /*0xd8-0xdf*/
4324 0ccc4, 0x979e, 0x979f, 0xe9a6, 0x97a0, 0x97a1, 0x97a2, 0x97a3, /*0xe0-0xef*/
4325 0x97a4, 0x97a5, 0x97a6, 0x97a7, 0x97a8, 0x97a9, 0xc9ad, 0x97aa, /*0xe8-0xef*/
4326 0xe9a2, 0xc0e2, 0x97ab, 0x97ac, 0x97ad, 0xbfc3, 0x97ae, 0x97af, /*0xf0-0xff*/
4327 0x97b0, 0xe8fe, 0xb9d7, 0x97b1, 0xe8fb, 0x97b2, 0x97b3, 0x97b4, /*0xf8-0xff*/
4328 /* 0x6900 */
4329 0x97b5, 0xe9a4, 0x97b6, 0x97b7, 0x97b8, 0xd2ce, 0x97b9, 0x97ba, /*0x00-0x07*/
4330 0x97bb, 0x97bc, 0x97bd, 0xe9a3, 0x97be, 0xd6b2, 0xd7b5, 0x97bf, /*0x08-0x0f*/
4331 0xe9a7, 0x97c0, 0xbdb7, 0x97c1, 0x97c2, 0x97c3, 0x97c4, 0x97c5, /*0x10-0x17*/
4332 0x97c6, 0x97c7, 0x97c8, 0x97c9, 0x97ca, 0x97cb, 0x97cc, 0xe8fc, /*0x18-0x1f*/
4333 0xe8fd, 0x97cd, 0x97ce, 0x97cf, 0xe9a1, 0x97d0, 0x97d1, 0x97d2, /*0x20-0x27*/
4334 0x97d3, 0x97d4, 0x97d5, 0x97d6, 0x97d7, 0xcdcd, 0x97d8, 0x97d9, /*0x28-0x2f*/
4335 0xd2ac, 0x97da, 0x97db, 0x97dc, 0xe9b2, 0x97dd, 0x97de, 0x97df, /*0x30-0x37*/
4336 0x97e0, 0xe9a9, 0x97e1, 0x97e2, 0x97e3, 0xb4aa, 0x97e4, 0xb4bb, /*0x38-0x3f*/
4337 0x97e5, 0x97e6, 0xe9ab, 0x97e7, 0x97e8, 0x97e9, 0x97ea, 0x97eb, /*0x40-0x47*/
4338 0x97ec, 0x97ed, 0x97ee, 0x97ef, 0x97f0, 0x97f1, 0x97f2, 0x97f3, /*0x48-0x4f*/
4339 0x97f4, 0x97f5, 0x97f6, 0x97f7, 0xd0a8, 0x97f8, 0x97f9, 0xe9a5, /*0x50-0x57*/
4340 0x97fa, 0x97fb, 0xb3fe, 0x97fc, 0x97fd, 0xe9ac, 0xc0e3, 0x97fe, /*0x58-0x5f*/
4341 0xe9aa, 0x9840, 0x9841, 0xe9b9, 0x9842, 0x9843, 0xe9b8, 0x9844, /*0x60-0x6f*/
4342 0x9845, 0x9846, 0x9847, 0xe9ae, 0x9848, 0x9849, 0xe8fa, 0x984a, /*0x68-0x6f*/
4343 0x984b, 0xe9a8, 0x984c, 0x984d, 0x984e, 0x984f, 0x9850, 0xbfac, /*0x70-0x77*/
4344 0xe9b1, 0xe9ba, 0x9851, 0x9852, 0xc2a5, 0x9853, 0x9854, 0x9855, /*0x78-0x7f*/
4345 0xe9af, 0x9856, 0xb8c5, 0x9857, 0xe9ad, 0x9858, 0xd3dc, 0xe9b4, /*0x80-0x87*/
4346 0xe9b5, 0xe9b7, 0x9859, 0x985a, 0x985b, 0xe9c7, 0x985c, 0x985d, /*0x88-0x8f*/
4347 0x985e, 0x985f, 0x9860, 0x9861, 0xc0c6, 0xe9c5, 0x9862, 0x9863, /*0x90-0x97*/
4348 0xe9b0, 0x9864, 0x9865, 0xe9bb, 0xb0f1, 0x9866, 0x9867, 0x9868, /*0x98-0x9f*/
```

```
4349 0x9869, 0x986a, 0x986b, 0x986c, 0x986d, 0x986e, 0x986f, 0x9869, /*0xa0-0xa7*/
4350 0xd5a5, 0x9870, 0x9871, 0x9872, 0x9873, 0x9874, 0x9875, 0x9876, /*0xa8-0xaf*/
4351 0x9877, 0xe9c1, 0x9876, 0x9877, 0xc1f1, 0x9878, 0x9879, 0xc8b6, /*0xb0-0xbf*/
4352 0x987a, 0x987b, 0x987c, 0xe9bd, 0x987d, 0x987e, 0x9880, 0x9881, /*0xb8-0xbf*/
4353 0x9882, 0xe9c2, 0x9883, 0x9884, 0x9885, 0x9886, 0x9887, 0x9888, /*0xc0-0xc7*/
4354 0x9889, 0x988a, 0xe9c3, 0x988b, 0xe9b3, 0x988c, 0xe9b6, 0x988d, /*0xc8-0xcf*/
4355 0xbbb1, 0x988e, 0x988f, 0x9890, 0xe9c0, 0x9891, 0x9892, 0x9893, /*0xd0-0xd7*/
4356 0x9894, 0x9895, 0x9896, 0xbcf7, 0x9897, 0x9898, 0x9899, 0xe9c4, /*0xd8-0xdf*/
4357 0xe9c6, 0x989a, 0x989b, 0x989c, 0x989d, 0x989e, 0x989f, 0x98a0, /*0xe0-0xef*/
4358 0x98a1, 0x98a2, 0x98a3, 0x98a4, 0x98a5, 0xe9ca, 0x98a6, 0x98a7, /*0xe8-0xef*/
4359 0x98a8, 0x98a9, 0xe9ce, 0x98aa, 0x98ab, 0x98ac, 0x98ad, 0x98ae, /*0xf0-0xff*/
4360 0x98af, 0x98b0, 0x98b1, 0x98b2, 0x98b3, 0xb2db, 0x98b4, 0xe9c8, /*0xf8-0xff*/
4361 /* 0x6a00 */
4362 0x98b5, 0x98b6, 0x98b7, 0x98b8, 0x98b9, 0x98ba, 0x98bb, 0x98bc, /*0x00-0x07*/
4363 0x98bd, 0x98be, 0xb7ae, 0x98bf, 0x98c0, 0x98c1, 0x98c2, 0x98c3, /*0x08-0x0f*/
4364 0x98c4, 0x98c5, 0x98c6, 0x98c7, 0x98c8, 0x98c9, 0x98ca, 0xe9cb, /*0x10-0x17*/
4365 0xe9cc, 0x98cb, 0x98cc, 0x98cd, 0x98ce, 0x98cf, 0x98d0, 0xd5c1, /*0x18-0x1f*/
4366 0x98d1, 0xc4a3, 0x98d2, 0x98d3, 0x98d4, 0x98d5, 0x98d6, 0x98d7, /*0x20-0x27*/
4367 0xe9d8, 0x98d8, 0xbae1, 0x98d9, 0x98da, 0x98db, 0x98dc, 0xe9c9, /*0x28-0x2f*/
4368 0x98dd, 0xd3a3, 0x98de, 0x98df, 0x98e0, 0xe9d4, 0x98e1, 0x98e2, /*0x30-0x37*/
4369 0x98e3, 0x98e4, 0x98e5, 0x98e6, 0x98e7, 0xe9d7, 0xe9d0, 0x98e8, /*0x38-0x3f*/
4370 0x98e9, 0x98ea, 0x98eb, 0x98ec, 0xe9cf, 0x98ed, 0x98ee, 0xc7c1, /*0x40-0x47*/
4371 0x98ef, 0x98f0, 0x98f1, 0x98f2, 0x98f3, 0x98f4, 0x98f5, 0x98f6, /*0x48-0x4f*/
4372 0xe9d2, 0x98f7, 0x98f8, 0x98f9, 0x98fa, 0x98fb, 0x98fc, 0x98fd, /*0x50-0x57*/
4373 0xe9d9, 0xb3c8, 0x98fe, 0xe9d3, 0x9940, 0x9941, 0x9942, 0x9943, /*0x58-0x5f*/
4374 0x9944, 0xcff0, 0x9945, 0x9946, 0x9947, 0xe9cd, 0x9948, 0x9949, /*0x60-0x67*/
4375 0x994a, 0x994b, 0x994c, 0x994d, 0x994e, 0x994f, 0x9950, 0x9951, /*0x68-0x6f*/
4376 0x9952, 0xb3f7, 0x9953, 0x9954, 0x9955, 0x9956, 0x9957, 0x9958, /*0x70-0x77*/
4377 0x9959, 0xe9d6, 0x995a, 0x995b, 0xe9da, 0x995c, 0x995d, 0x995e, /*0x78-0x7f*/
4378 0xcdb4, 0x995f, 0x9960, 0x9961, 0xcfad, 0x9962, 0x9963, 0x9964, /*0x80-0x87*/
4379 0x9965, 0x9966, 0x9967, 0x9968, 0x9969, 0x996a, 0xe9d5, 0x996b, /*0x88-0x8f*/
4380 0xe9dc, 0xe9db, 0x996c, 0x996d, 0x996e, 0x996f, 0x9970, 0xe9de, /*0x90-0x97*/
4381 0x9971, 0x9972, 0x9973, 0x9974, 0x9975, 0x9976, 0x9977, 0x9978, /*0x98-0x9f*/
4382 0xe9d1, 0x9979, 0x997a, 0x997b, 0x997c, 0x997d, 0x997e, 0x9980, /*0xa0-0xa7*/
4383 0x9981, 0xe9dd, 0x9982, 0xe9df, 0xc3ca, 0x9983, 0x9984, 0x9985, /*0xa8-0xaf*/
4384 0x9986, 0x9987, 0x9988, 0x9989, 0x998a, 0x998b, 0x998c, 0x998d, /*0xb0-0xbf*/
4385 0x998e, 0x998f, 0x9990, 0x9991, 0x9992, 0x9993, 0x9994, 0x9995, /*0xb8-0xbf*/
4386 0x9996, 0x9997, 0x9998, 0x9999, 0x999a, 0x999b, 0x999c, 0x999d, /*0xc0-0xc7*/
4387 0x999e, 0x999f, 0x99a0, 0x99a1, 0x99a2, 0x99a3, 0x99a4, 0x99a5, /*0xc8-0xcf*/
4388 0x99a6, 0x99a7, 0x99a8, 0x99a9, 0x99aa, 0x99ab, 0x99ac, 0x99ad, /*0xd0-0xdf*/
4389 0x99ae, 0x99af, 0x99b0, 0x99b1, 0x99b2, 0x99b3, 0x99b4, 0x99b5, /*0xe0-0xef*/
4390 0x99b6, 0x99b7, 0x99b8, 0x99b9, 0x99ba, 0x99bb, 0x99bc, 0x99bd, /*0xe8-0xef*/
4391 0x99be, 0x99bf, 0x99c0, 0x99c1, 0x99c2, 0x99c3, 0x99c4, 0x99c5, /*0xf0-0xff*/
4392 0x99c6, 0x99c7, 0x99c8, 0x99c9, 0x99ca, 0x99cb, 0x99cc, 0x99cd, /*0xf0-0xff*/
4393 0x99ce, 0x99cf, 0x99d0, 0x99d1, 0x99d2, 0x99d3, 0x99d4, 0x99d5, /*0xf8-0xff*/
4394 /* 0x6b00 */
4395 0x99d6, 0x99d7, 0x99d8, 0x99d9, 0x99da, 0x99db, 0x99dc, 0x99dd, /*0x00-0x07*/
4396 0x99de, 0x99df, 0x99e0, 0x99e1, 0x99e2, 0x99e3, 0x99e4, 0x99e5, /*0x08-0x0f*/
4397 0x99e6, 0x99e7, 0x99e8, 0x99e9, 0x99ea, 0x99eb, 0x99ec, 0x99ed, /*0x10-0x17*/
4398 0x99ee, 0x99ef, 0x99f0, 0x99f1, 0x99f2, 0x99f3, 0x99f4, 0x99f5, /*0x18-0x1f*/
4399 0xc7b7, 0xb4ce, 0xbbb6, 0xd0c0, 0xecac, 0x99f6, 0x99f7, 0xc5b7, /*0x20-0x27*/
4400 0x99f8, 0x99f9, 0x99fa, 0x99fb, 0x99fc, 0x99fd, 0x99fe, 0x99ff, /*0x28-0x2f*/
4401 0x99a4, 0x99a2, 0xd3fb, 0x99a3, 0x99a4, 0x99a5, 0x99a6, 0xecac, /*0x30-0x37*/
4402 0x99a7, 0xecac, 0xc6db, 0x99a8, 0x99a9, 0x99aa, 0xbfee, 0x99ab, /*0x38-0x3f*/
4403 0x99ac, 0x99ad, 0x99ae, 0xecac, 0x99af, 0x99b0, 0xecac, 0xd0aa, /*0x40-0x47*/
4404 0x99a5, 0xc7b8, 0x99a2, 0x99a3, 0xb8e8, 0x99a4, 0x99a5, 0x99a6, /*0x48-0x4f*/
4405 0x99a7, 0x99a8, 0x99a9, 0x99aa, 0x99ab, 0x99ac, 0x99ad, 0x99ae, /*0x50-0x57*/
4406 0x99af, 0xecac, 0x99a0, 0x99a1, 0x99a2, 0x99a3, 0x99a4, 0x99a5, /*0x58-0x5f*/
4407 0x99a6, 0x99a7, 0xd6b9, 0xd5fd, 0xb4cb, 0xb2bd, 0xcee4, 0xc6e7, /*0x60-0x67*/
4408 0x99a8, 0x99a9, 0xcd1e, 0x99aa, 0x99ab, 0x99ac, 0x99ad, 0x99ae, /*0x68-0x6f*/
4409 0x99af, 0x99b0, 0x99b1, 0x99b2, 0x99b3, 0x99b4, 0x99b5, 0x99b6, /*0x70-0x77*/
4410 0x99b7, 0xb4f5, 0x99b8, 0xcbc0, 0xbcdf, 0x99b9, 0x99ba, 0x99bb, /*0x78-0x7f*/
4411 0x99bc, 0xe9e2, 0xe9e3, 0xd1ea, 0xe9e4, 0x99bd, 0xb4f9, 0xe9e5, /*0x80-0x87*/
4412 0x99be, 0xd1b3, 0xcae2, 0xb2d0, 0x99b8, 0xe9e8, 0x99b1, 0x99b2, /*0x88-0x8f*/
4413 0x99b3, 0x99b4, 0xe9e6, 0xe9e7, 0x99b5, 0x99b6, 0xd6b3, 0x99b7, /*0x90-0x97*/
4414 0x99b8, 0x99b9, 0xe9e9, 0xe9ea, 0x99b8, 0x99b9, 0x99ba, 0x99bb, /*0x98-0x9f*/
4415 0x99be, 0xe9eb, 0x99b8, 0x99b9, 0x99ba, 0x99bb, 0x99bc, 0x99bd, /*0xa0-0xaf*/
4416 0x99c5, 0x99c6, 0xe9ec, 0x99c7, 0x99c8, 0x99c9, 0x99ca, 0x99cb, /*0xa8-0xaf*/
4417 0x99c0, 0x99c1, 0x99c2, 0xcacf, 0xc5b9, 0xb6ce, 0x99c0, 0xd2f3, /*0xb0-0xbf*/
4418 0x99c2, 0x99c3, 0x99c4, 0x99c5, 0x99c6, 0x99c7, 0x99c8, 0xb5ee, /*0xb8-0xbf*/
4419 0x99c9, 0xbdb9, 0xecb1, 0x99c8, 0x99c9, 0xd2e3, 0x99ca, 0x99cb, /*0xc0-0xcf*/
4420 0x99ca, 0x99cb, 0x99cc, 0xcee3, 0x99ca, 0xc4b8, 0x99cb, 0xc3bf, /*0xc8-0xcf*/
4421 0x99cb, 0x99cd, 0xb6be, 0xd8b9, 0xb1c8, 0xb1cf, 0xb1d1, 0xc5fe, /*0xd0-0xdf*/
4422 0x99cb, 0xb1d0, 0x99cb, 0xc3ab, 0x99cb, 0x99cb, 0x99cb, 0x99cb, /*0xd8-0xdf*/
4423 0x99cb, 0xd5b1, 0x99cb, 0x99cb, 0x99cb, 0x99cb, 0x99cb, 0x99cb, /*0xe0-0xef*/
4424 0x99cb, 0x99cb, 0xeba4, 0xbac1, 0x99cb, 0x99cb, 0x99cb, 0xc5ca, /*0xe8-0xef*/
4425 0x99cb, 0x99cb, 0x99cb, 0xeba5, 0x99cb, 0xeba7, 0x99cb, 0x99cb, /*0xf0-0xff*/
4426 0x99cb, 0xeba8, 0x99cb, 0x99cb, 0x99cb, 0xeba6, 0x99cb, 0x99cb, /*0xf8-0xff*/
4427 /* 0x6c00 */
4428 0x99cb, 0x99cb, 0x99cb, 0x99cb, 0x99cb, 0xeba9, 0xebab, 0xebaa, /*0x00-0x07*/
4429 0x99cb, 0x99cb, 0x99cb, 0x99cb, 0x99cb, 0xebac, 0x99cb, 0xcacf, /*0x08-0x0f*/
4430 0xd8b5, 0xc3f1, 0x99cb, 0xc3a5, 0xc6f8, 0xebad, 0xc4ca, 0x99cb, /*0x10-0x17*/
4431 0xebae, 0xebaf, 0xebb0, 0xb7d5, 0x99cb, 0x99cb, 0x99cb, 0xb7fa, /*0x18-0x1f*/
4432 0x99cb, 0xebb1, 0xc7e2, 0x99cb, 0xebb3, 0x99cb, 0xbaa4, 0xd1f5, /*0x20-0x27*/
4433 0xb0b1, 0xebb2, 0xebb3, 0x99cb, 0x99cb, 0x99cb, 0xb5aa, 0xc2c8, /*0x28-0x2f*/
4434 0xc7e8, 0x99cb, 0xebb5, 0x99cb, 0xebb8, 0xcbae, 0xc3df, 0x99cb, /*0x30-0x37*/
4435 0xd3c0, 0x99cb, 0x99cb, 0x99cb, 0x99cb, 0xd9db, 0x99cb, 0x99cb, /*0x38-0x3f*/
```

4436 0xcdal, 0xd6ad, 0xc7f3, 0x9af1, 0x9af2, 0x9af3, 0xd9e0, 0xbbe3, /*0x40-0x47*/
4437 0x9af4, 0xbaba, 0xe3e2, 0x9af5, 0x9af6, 0x9af7, 0x9af8, 0x9af9, /*0x48-0x4f*/
4438 0xcfab, 0x9afa, 0x9afb, 0x9afc, 0xe3e0, 0xc9c7, 0x9afd, 0xbab9, /*0x50-0x57*/
4439 0x9afe, 0x9b40, 0x9b41, 0xd1b4, 0xe3e1, 0xc8ea, 0xb9af, 0xbdad, /*0x58-0x5f*/
4440 0xb3d8, 0xcdcd, 0x9b42, 0x9b43, 0xc9c0, 0x9b44, 0x9b45, 0x9b46, /*0x60-0x67*/
4441 0xe3e8, 0xe3e9, 0xcdf4, 0x9b47, 0x9b48, 0x9b49, 0x9b4a, 0x9b4b, /*0x68-0x6f*/
4442 0xccad, 0x9b4c, 0xbcb3, 0x9b4d, 0xe3ea, 0x9b4e, 0xe3eb, 0x9b4f, /*0x70-0x77*/
4443 0x9b50, 0xd0da, 0x9b51, 0x9b52, 0x9b53, 0xc6fb, 0xb7da, 0x9b54, /*0x78-0x7f*/
4444 0x9b55, 0xc7df, 0xd2ca, 0xcd66, 0x9b56, 0xe3e4, 0xe3ec, 0x9b57, /*0x80-0x87*/
4445 0xc9f2, 0xb3c1, 0x9b58, 0x9b59, 0xe3e7, 0x9b5a, 0x9b5b, 0xc6e3, /*0x88-0x8f*/
4446 0xe3e5, 0x9b5c, 0x9b5d, 0xedb3, 0xe3e6, 0x9b5e, 0x9b5f, 0x9b60, /*0x90-0x97*/
4447 0x9b61, 0xc9b3, 0x9b62, 0xc5e6, 0x9b63, 0x9b64, 0x9b65, 0x9b66, /*0x98-0x9f*/
4448 0x9b66, 0xc3bb, 0x9b67, 0xe3e3, 0xc5bd, 0xc1a4, 0xc2d9, 0xb2d7, /*0xa0-0xa7*/
4449 0x9b68, 0xe3ed, 0xbba6, 0xc4ad, 0x9b69, 0xe3f0, 0xbeda, 0x9b6a, /*0xa8-0xaf*/
4450 0x9b6b, 0xe3fb, 0xe3f5, 0xbad3, 0x9b6c, 0x9b6d, 0x9b6e, 0x9b6f, /*0xb0-0xbf*/
4451 0xb7d0, 0xd3cd, 0x9b70, 0xd6ce, 0xd5d3, 0xb9c1, 0xd5b4, 0xd1d8, /*0xb8-0xbf*/
4452 0x9b71, 0xb7f2, 0x9b73, 0x9b74, 0xd0b9, 0xc7f6, 0x9b75, 0x9b76, /*0xc0-0xc7*/
4453 0x9b77, 0xc8aa, 0xb2b4, 0x9b78, 0xc3da, 0x9b79, 0x9b7a, 0x9b7b, /*0xc8-0xcf*/
4454 0xe3ee, 0x9b7c, 0x9b7d, 0xe3fc, 0xe3ef, 0xb7a8, 0xe3f7, 0xe3f4, /*0xd0-0xd7*/
4455 0x9b7e, 0x9b80, 0x9b81, 0xb7ba, 0x9b82, 0x9b83, 0xc5a2, 0x9b84, /*0xd8-0xdf*/
4456 0xe3f6, 0xc5db, 0xb2a8, 0xc6fc, 0x9b85, 0xc4e0, 0x9b86, 0x9b87, /*0xe0-0xef*/
4457 0xd7a2, 0x9b88, 0xc0e1, 0xe3f9, 0x9b89, 0x9b8a, 0xe3fa, 0xe3fd, /*0xe8-0xef*/
4458 0xc9a9, 0xe3f3, 0x9b8b, 0xd3be, 0x9b8c, 0xb1c3, 0xedb4, 0xe3ff, /*0xf0-0xff*/
4459 0xe3f2, 0x9b8d, 0xe3f8, 0xd0ba, 0xc6c3, 0xd4f3, 0xe3fe, 0x9b8e, /*0xf8-0xff*/
4460 /* 0x6d00 */
4461 0x9b8f, 0xbde0, 0x9b90, 0x9b91, 0xe4a7, 0x9b92, 0x9b93, 0xe4a6, /*0x00-0x07*/
4462 0x9b94, 0x9b95, 0x9b96, 0xd1f3, 0xe4a3, 0x9b97, 0xe4a9, 0x9b98, /*0x08-0x0f*/
4463 0x9b99, 0x9b9a, 0xc8f7, 0x9b9b, 0x9b9c, 0x9b9d, 0x9b9e, 0xcfbc, /*0x10-0x17*/
4464 0x9b9f, 0xe4a8, 0xe4ae, 0xc2e5, 0x9ba0, 0x9ba1, 0xb6b4, 0x9ba2, /*0x18-0x1f*/
4465 0x9ba3, 0x9ba4, 0x9ba5, 0x9ba6, 0x9ba7, 0xbd2f, 0x9ba8, 0xe4a2, /*0x20-0x27*/
4466 0x9ba9, 0x9baa, 0xbae9, 0xe4aa, 0x9bab, 0x9bac, 0xe4ac, 0x9bad, /*0x28-0x2f*/
4467 0x9bae, 0xb6fd, 0xd6de, 0xe4b2, 0x9baf, 0xe4ad, 0x9bb0, 0x9bb1, /*0x30-0x37*/
4468 0x9bb2, 0xe4a1, 0x9bb3, 0xbbee, 0xcdcd, 0xc7a2, 0xc5c9, 0x9bb4, /*0x38-0x3f*/
4469 0x9bb5, 0xc1f7, 0x9bb6, 0xe4a4, 0x9bb7, 0xc7b3, 0xbdac, 0xbdbd, /*0x40-0x47*/
4470 0xe4a5, 0x9bb8, 0xd7c7, 0xb2e2, 0x9bb9, 0xe4ab, 0xbcc3, 0xe4af, /*0x48-0x4f*/
4471 0x9bba, 0xbbeb, 0xe4b0, 0xc5a8, 0xe4b1, 0x9bbb, 0x9bbc, 0x9bbd, /*0x50-0x57*/
4472 0x9bbe, 0xd5e3, 0xbfa3, 0x9bbf, 0xe4ba, 0x9bc0, 0xe4b7, 0x9bc1, /*0x58-0x5f*/
4473 0xe4bb, 0x9bc2, 0x9bc3, 0xe4bd, 0x9bc4, 0x9bc5, 0xc6d6, 0x9bc6, /*0x60-0x67*/
4474 0x9bc7, 0xbac6, 0xc0cb, 0x9bc8, 0x9bc9, 0x9bca, 0xb8a1, 0xe4b4, /*0x68-0x6f*/
4475 0x9bcb, 0x9bcc, 0x9bcd, 0x9bce, 0xd4a1, 0x9bcf, 0x9bd0, 0xbaa3, /*0x70-0x77*/
4476 0xbdfc, 0x9bd1, 0x9bd2, 0x9bd3, 0xe4bc, 0x9bd4, 0x9bd5, 0x9bd6, /*0x78-0x7f*/
4477 0x9bd7, 0x9bd8, 0xcdbf, 0x9bd9, 0x9bda, 0xc4f9, 0x9bdb, 0x9bdc, /*0x80-0x87*/
4478 0xcffb, 0xc9e6, 0x9bdd, 0x9bde, 0xd3bf, 0x9bdf, 0xcfd1, 0x9be0, /*0x88-0x8f*/
4479 0x9be1, 0xe4b3, 0x9be2, 0xe4b8, 0xe4b9, 0xcce9, 0x9be3, 0x9be4, /*0x90-0x97*/
4480 0x9be5, 0x9be6, 0x9be7, 0xc9cc, 0x9be8, 0xc0d4, 0xe4b5, 0xc1b0, /*0x98-0x9f*/
4481 0xe4b6, 0xcd00, 0x9be9, 0xbbc1, 0xb5d3, 0x9bea, 0xc8f3, 0xbda7, /*0xa0-0xaf*/
4482 0xd5c7, 0xc9ac, 0xb8a2, 0xe4ca, 0x9bec, 0x9bec, 0xe4cc, 0xd1c4, /*0xa8-0xaf*/
4483 0x9bed, 0x9bee, 0xd2ba, 0x9bef, 0x9bf0, 0xbaad, 0x9bf1, 0x9bf2, /*0xb0-0xbf*/
4484 0xbad4, 0x9bf3, 0x9bf4, 0x9bf5, 0x9bf6, 0x9bf7, 0x9bf8, 0xe4c3, /*0xb8-0xbf*/
4485 0xb5ed, 0x9bf9, 0x9bfa, 0x9bfb, 0xd7cd, 0xe4c0, 0xcffd, 0xe4bf, /*0xc0-0xc7*/
4486 0x9bfc, 0x9bfd, 0x9bfe, 0xc1dc, 0xc9ca, 0xc9c1, 0xc9c2, 0xc9c3, /*0xc8-0xcf*/
4487 0x9c43, 0xcae7, 0x9c44, 0x9c45, 0x9c46, 0x9c47, 0xc4d7, 0x9c48, /*0xd0-0xd7*/
4488 0xc9cd, 0xe4c8, 0x9c49, 0x9c4a, 0x9c4b, 0xe4c7, 0xe4c1, 0x9c4c, /*0xd8-0xdf*/
4489 0xe4c4, 0xb5ad, 0x9c4d, 0x9c4e, 0xd3d9, 0x9c4f, 0xe4c6, 0x9c50, /*0xe0-0xef*/
4490 0x9c51, 0x9c52, 0x9c53, 0xd2f9, 0xb4e3, 0x9c54, 0xbbb4, 0x9c55, /*0xe8-0xef*/
4491 0x9c56, 0xc9e9, 0x9c57, 0xb4be, 0x9c58, 0x9c59, 0x9c5a, 0xbbec, /*0xf0-0xff*/
4492 0x9c5b, 0xd1cd, 0x9c5c, 0xc9cd, 0xedb5, 0x9c5d, 0x9c5e, 0x9c5f, /*0xf8-0xff*/
4493 /* 0x6e00 */
4494 0x9c60, 0x9c61, 0x9c62, 0x9c63, 0x9c64, 0xc7e5, 0x9c65, 0x9c66, /*0x00-0x07*/
4495 0x9c67, 0x9c68, 0xd4a8, 0x9c69, 0xe4cb, 0xd7d5, 0xe4c2, 0x9c6a, /*0x08-0x0f*/
4496 0xbda5, 0xe4c5, 0x9c6b, 0x9c6c, 0xd3e6, 0x9c6d, 0xe4c9, 0xc9f8, /*0x10-0x17*/
4497 0x9c6e, 0x9c6f, 0xe4be, 0x9c70, 0x9c71, 0xd3e5, 0x9c72, 0x9c73, /*0x18-0x1f*/
4498 0xc7fe, 0xb6c9, 0x9c74, 0xd4fc, 0xb2b3, 0xe4d7, 0x9c75, 0x9c76, /*0x20-0x27*/
4499 0x9c77, 0xc9c2, 0x9c78, 0xe4cd, 0x9c79, 0xc9cb, 0x9c7a, 0xb8db, /*0x28-0x2f*/
4500 0x9c7b, 0x9c7c, 0xe4d6, 0x9c7d, 0xbfca, 0x9c7e, 0x9c80, 0x9c81, /*0x30-0x37*/
4501 0xd3ce, 0x9c82, 0xc3ec, 0x9c83, 0x9c84, 0x9c85, 0x9c86, 0x9c87, /*0x38-0x3f*/
4502 0x9c88, 0x9c89, 0x9c8a, 0xc5c8, 0xe4d8, 0x9c8b, 0x9c8c, 0x9c8d, /*0x40-0x47*/
4503 0x9c8e, 0x9c8f, 0x9c90, 0x9c91, 0x9c92, 0xc9cd, 0xe4cf, 0x9c93, /*0x48-0x4f*/
4504 0x9c94, 0x9c95, 0x9c96, 0xe4d4, 0xe4d5, 0x9c97, 0xbafe, 0x9c98, /*0x50-0x57*/
4505 0xcfe6, 0x9c99, 0x9c9a, 0xd5bf, 0x9c9b, 0x9c9c, 0x9c9d, 0xe4d2, /*0x58-0x5f*/
4506 0x9c9e, 0x9c9f, 0x9ca0, 0x9ca1, 0x9ca2, 0x9ca3, 0x9ca4, 0x9ca5, /*0x60-0x67*/
4507 0x9ca6, 0x9ca7, 0x9ca8, 0xe4d0, 0x9ca9, 0x9caa, 0xe4ce, 0x9cab, /*0x68-0x6f*/
4508 0x9cac, 0x9cad, 0x9cae, 0x9caf, 0x9cb0, 0x9cb1, 0x9cb2, 0x9cb3, /*0x70-0x77*/
4509 0x9cb4, 0x9cb5, 0x9cb6, 0x9cb7, 0x9cb8, 0x9cb9, 0xcde5, 0xc9aa, /*0x78-0x7f*/
4510 0x9cba, 0x9cbb, 0x9cbc, 0xc0a3, 0x9cbd, 0xbda6, 0xe4d3, 0x9cbe, /*0x80-0x87*/
4511 0x9cbf, 0xb8c8, 0x9cc0, 0x9cc1, 0x9cc2, 0x9cc3, 0x9cc4, 0xe4e7, /*0x88-0x8f*/
4512 0xd4b4, 0x9cc5, 0x9cc6, 0x9cc7, 0x9cc8, 0x9cc9, 0x9cca, 0x9ccb, /*0x90-0x97*/
4513 0xe4db, 0x9ccc, 0x9ccd, 0x9cce, 0xc1ef, 0x9ccf, 0x9cd0, 0xe4e9, /*0x98-0x9f*/
4514 0x9cd1, 0x9cd2, 0xd2e7, 0x9cd3, 0x9cd4, 0xe4df, 0x9cd5, 0xe4e0, /*0xa0-0xaf*/
4515 0x9cd6, 0x9cd7, 0xcfaa, 0x9cd8, 0x9cd9, 0x9cda, 0x9cdb, 0xcdbd, /*0xa8-0xaf*/
4516 0x9cdc, 0xe4da, 0xe4d1, 0x9cdd, 0xe4e5, 0x9cde, 0xc8dc, 0xe4e3, /*0xb0-0xbf*/
4517 0x9cdf, 0x9ce0, 0xc4e7, 0xe4e2, 0x9ce1, 0xe4e1, 0x9ce2, 0x9ce3, /*0xb8-0xbf*/
4518 0x9ce4, 0xb3fc, 0xe4e8, 0x9ce5, 0x9ce6, 0x9ce7, 0x9ce8, 0xb5e1, /*0xc0-0xcf*/
4519 0x9ce9, 0x9cea, 0x9ceb, 0xd7cc, 0x9cec, 0x9ced, 0x9cee, 0xe4e6, /*0xc8-0xcf*/
4520 0x9cef, 0xbba6, 0x9cf0, 0xd7d2, 0xc9cc, 0xebf8, 0x9cf1, 0xe4e4, /*0xd0-0xdf*/
4521 0x9cf2, 0x9cf3, 0xb9f6, 0x9cf4, 0x9cf5, 0x9cf6, 0xd6cd, 0xe4d9, /*0xd8-0xdf*/
4522 0xe4dc, 0xc2fa, 0xe4de, 0x9cf7, 0xc2cb, 0xc0c4, 0xc2d0, 0x9cf8, /*0xe0-0xef*/

```

4523 0xb1f5, 0xccb2, 0x9cf9, 0x9cfa, 0x9cfb, 0x9cfc, 0x9cfd, 0x9cfe, /*0xe8-0xef*/
4524 0x9d40, 0x9d41, 0x9d42, 0x9d43, 0xb5ce, 0x9d44, 0x9d45, 0x9d46, /*0xf0-0xf7*/
4525 0x9d47, 0xe4ef, 0x9d48, 0x9d49, 0x9d4a, 0x9d4b, 0x9d4c, 0x9d4d, /*0xf8-0xff*/
4526 /* 0x6f00 */
4527 0x9d4e, 0x9d4f, 0xc6af, 0x9d50, 0x9d51, 0x9d52, 0xc6e1, 0x9d53, /*0x00-0x07*/
4528 0x9d54, 0xe4f5, 0x9d55, 0x9d56, 0x9d57, 0x9d58, 0x9d59, 0xc2a9, /*0x08-0x0f*/
4529 0x9d5a, 0x9d5b, 0x9d5c, 0xc0ec, 0xd1dd, 0xe4ee, 0x9d5d, 0x9d5e, /*0x10-0x17*/
4530 0x9d5f, 0x9d60, 0x9d61, 0x9d62, 0x9d63, 0x9d64, 0x9d65, 0x9d66, /*0x18-0x1f*/
4531 0xc4ae, 0x9d67, 0x9d68, 0x9d69, 0xe4ed, 0x9d6a, 0x9d6b, 0x9d6c, /*0x20-0x27*/
4532 0x9d6d, 0xe4f6, 0xe4f4, 0xc2fe, 0x9d6e, 0xe4dd, 0x9d6f, 0xe4f0, /*0x28-0x2f*/
4533 0x9d70, 0xcafe, 0x9d71, 0xd5c4, 0x9d72, 0x9d73, 0xe4f1, 0x9d74, /*0x30-0x37*/
4534 0x9d75, 0x9d76, 0x9d77, 0x9d78, 0x9d79, 0x9d7a, 0xd1fa, 0x9d7b, /*0x38-0x3f*/
4535 0x9d7c, 0x9d7d, 0x9d7e, 0x9d80, 0x9d81, 0x9d82, 0xe4eb, 0xe4ec, /*0x40-0x47*/
4536 0x9d83, 0x9d84, 0x9d85, 0xe4f2, 0x9d86, 0xceab, 0x9d87, 0x9d88, /*0x48-0x4f*/
4537 0x9d89, 0x9d8a, 0x9d8b, 0x9d8c, 0x9d8d, 0x9d8e, 0x9d8f, 0x9d90, /*0x50-0x57*/
4538 0xc5cb, 0x9d91, 0x9d92, 0x9d93, 0xc7b1, 0x9d94, 0xc2ba, 0x9d95, /*0x58-0x5f*/
4539 0x9d96, 0x9d97, 0xe4ea, 0x9d98, 0x9d99, 0x9da0, 0xc1ca, 0x9d9b, /*0x60-0x67*/
4540 0x9d9c, 0x9d9d, 0x9d9e, 0x9da0, 0xc0cb, 0xb3b1, 0x9da1, /*0x68-0x6f*/
4541 0x9da2, 0x9da3, 0xe4fb, 0x9da4, 0xe4f3, 0x9da5, 0x9da6, 0x9da7, /*0x70-0x77*/
4542 0xe4fa, 0x9da8, 0xe4fd, 0x9da9, 0xe4fc, 0x9daa, 0x9dab, 0x9dac, /*0x78-0x7f*/
4543 0x9dad, 0x9dae, 0x9daf, 0x9db0, 0xb3ce, 0x9db1, 0x9db2, 0x9db3, /*0x80-0x87*/
4544 0xb3ba, 0xe4f7, 0x9db4, 0x9db5, 0xe4f9, 0xe4f8, 0xc5ec, 0x9db6, /*0x88-0x8f*/
4545 0x9db7, 0x9db8, 0x9db9, 0x9dba, 0x9dbb, 0x9dbc, 0x9dbd, 0x9dbe, /*0x90-0x97*/
4546 0x9dbf, 0x9dc0, 0x9dc1, 0x9dc2, 0xc0bd, 0x9dc3, 0x9dc4, 0x9dc5, /*0x98-0x9f*/
4547 0x9dc6, 0xd4e8, 0x9dc7, 0x9dc8, 0x9dc9, 0x9dca, 0x9dcb, 0xe5a2, /*0xa0-0xa7*/
4548 0x9dcc, 0x9dcd, 0x9dce, 0x9dcf, 0x9dd0, 0x9dd1, 0x9dd2, 0x9dd3, /*0xa8-0xaf*/
4549 0x9dda, 0x9ddb, 0x9dde, 0xb0c4, 0x9dd7, 0x9dd8, 0xe5a4, 0x9dd9, /*0xb0-0xb7*/
4550 0x9dda, 0xe5a3, 0x9ddb, 0x9ddc, 0x9ddd, 0x9dde, 0x9ddf, 0x9de0, /*0xb8-0xbf*/
4551 0xbca4, 0x9de1, 0xe5a5, 0x9de2, 0x9de3, 0x9de4, 0x9de5, 0x9de6, /*0xc0-0xc7*/
4552 0x9de7, 0xe5a1, 0x9de8, 0x9de9, 0x9dea, 0x9deb, 0x9dec, 0x9ded, /*0xc8-0xcf*/
4553 0x9dee, 0xe4fe, 0xb1f4, 0x9def, 0x9df0, 0x9df1, 0x9df2, 0x9df3, /*0xd0-0xd7*/
4554 0x9df4, 0x9df5, 0x9df6, 0x9df7, 0x9df8, 0x9df9, 0xe5a8, 0x9dfa, /*0xd8-0xdf*/
4555 0xe5a9, 0xe5a6, 0x9dfb, 0x9dfc, 0x9dfd, 0x9dfe, 0x9e40, 0x9e41, /*0xe0-0xe7*/
4556 0x9e42, 0x9e43, 0x9e44, 0x9e45, 0x9e46, 0x9e47, 0xe5a7, 0xe5aa, /*0xe8-0xef*/
4557 0x9e48, 0x9e49, 0x9e4a, 0x9e4b, 0x9e4c, 0x9e4d, 0x9e4e, 0x9e4f, /*0xf0-0xf7*/
4558 0x9e50, 0x9e51, 0x9e52, 0x9e53, 0x9e54, 0x9e55, 0x9e56, 0x9e57, /*0xf8-0xff*/
4559 /* 0x7000 */
4560 0x9e58, 0x9e59, 0x9e5a, 0x9e5b, 0x9e5c, 0x9e5d, 0x9e5e, 0x9e5f, /*0x00-0x07*/
4561 0x9e60, 0x9e61, 0x9e62, 0x9e63, 0x9e64, 0x9e65, 0x9e66, 0x9e67, /*0x08-0x0f*/
4562 0x9e68, 0xc6d9, 0x9e69, 0x9e6a, 0x9e6b, 0x9e6c, 0x9e6d, 0x9e6e, /*0x10-0x17*/
4563 0x9e6f, 0x9e70, 0xe5ab, 0xe5ad, 0x9e71, 0x9e72, 0x9e73, 0x9e74, /*0x18-0x1f*/
4564 0x9e75, 0x9e76, 0x9e77, 0xe5ac, 0x9e78, 0x9e79, 0x9e7a, 0x9e7b, /*0x20-0x27*/
4565 0x9e7c, 0x9e7d, 0x9e7e, 0x9e80, 0x9e81, 0x9e82, 0x9e83, 0x9e84, /*0x28-0x2f*/
4566 0x9e85, 0x9e86, 0x9e87, 0x9e88, 0x9e89, 0xe5af, 0x9e8a, 0x9e8b, /*0x30-0x37*/
4567 0x9e8c, 0xe5ae, 0x9e8d, 0x9e8e, 0x9e8f, 0x9e90, 0x9e91, 0x9e92, /*0x38-0x3f*/
4568 0x9e93, 0x9e94, 0x9e95, 0x9e96, 0x9e97, 0x9e98, 0x9e99, 0x9e9a, /*0x40-0x47*/
4569 0x9e9b, 0x9e9c, 0x9e9d, 0x9e9e, 0xb9e0, 0x9e9f, 0x9ea0, 0xe5b0, /*0x48-0x4f*/
4570 0x9ea1, 0x9ea2, 0x9ea3, 0x9ea4, 0x9ea5, 0x9ea6, 0x9ea7, 0x9ea8, /*0x50-0x57*/
4571 0x9ea9, 0x9eaa, 0x9eab, 0x9eac, 0x9ead, 0x9eae, 0xe5b1, 0x9eaf, /*0x58-0x5f*/
4572 0x9eb0, 0x9eb1, 0x9eb2, 0x9eb3, 0x9eb4, 0x9eb5, 0x9eb6, 0x9eb7, /*0x60-0x67*/
4573 0x9eb8, 0x9eb9, 0x9eba, 0xbbf0, 0xecc1, 0xc3f0, 0x9ebb, 0xb5c6, /*0x68-0x6f*/
4574 0xbbd2, 0x9ebc, 0x9ebd, 0x9ebe, 0x9ebf, 0xc1e9, 0xd4ee, 0x9ec0, /*0x70-0x77*/
4575 0xbec4, 0x9ec1, 0x9ec2, 0x9ec3, 0xd7c6, 0x9ec4, 0xd4d6, 0xb2d3, /*0x78-0x7f*/
4576 0xecbe, 0x9ec5, 0x9ec6, 0x9ec7, 0x9ec8, 0xeac1, 0x9ec9, 0x9eca, /*0x80-0x87*/
4577 0x9ecb, 0xc2af, 0xb4b6, 0x9ecc, 0x9ecd, 0x9ece, 0xd1d7, 0x9ecf, /*0x88-0x8f*/
4578 0x9ed0, 0x9ed1, 0xb3b4, 0x9ed2, 0xc8b2, 0xbfbf, 0xecc0, 0x9ed3, /*0x90-0x97*/
4579 0x9ed4, 0xd6cb, 0x9ed5, 0x9ed6, 0xecbf, 0xecc1, 0x9ed7, 0x9ed8, /*0x98-0x9f*/
4580 0x9ed9, 0x9eda, 0x9edb, 0x9edc, 0x9edd, 0x9ede, 0x9edf, 0x9ee0, /*0xa0-0xa7*/
4581 0x9ee1, 0x9ee2, 0x9ee3, 0xecc5, 0xbee6, 0xc0bf, 0xc5da, 0xbecb, /*0xa8-0xaf*/
4582 0x9ee4, 0xecc6, 0x9ee5, 0xb1fe, 0x9ee6, 0x9ee7, 0x9ee8, 0xecc4, /*0xb0-0xb7*/
4583 0xd5a8, 0xb5e3, 0x9ee9, 0xecc2, 0xc1b6, 0xb3e3, 0x9eea, 0x9eeb, /*0xb8-0xbf*/
4584 0xecc3, 0xcbb8, 0xc0c3, 0xc0ff, 0x9eed, 0x9eee, 0x9eef, /*0xc0-0xc7*/
4585 0xc1d2, 0x9ef0, 0xecc8, 0x9ef1, 0x9ef2, 0x9ef3, 0x9ef4, 0x9ef5, /*0xc8-0xcf*/
4586 0x9ef6, 0x9ef7, 0x9ef8, 0x9ef9, 0x9efa, 0x9efb, 0x9efc, 0x9efd, /*0xd0-0xd7*/
4587 0xbae6, 0xc0d3, 0x9efe, 0xd6f2, 0x9ff0, 0x9ff1, 0x9ff2, 0xd1cc, /*0xd8-0xdf*/
4588 0x9ff3, 0x9ff4, 0x9ff5, 0x9ff6, 0xbfbf, 0x9ff7, 0xb7b3, 0xc9d5, /*0xe0-0xe7*/
4589 0xecc7, 0xbbe2, 0x9ff8, 0xc0cc, 0xbdff, 0xc8c8, 0x9ff9, 0xcfa9, /*0xe8-0xef*/
4590 0x9ff4a, 0x9ff4b, 0x9ff4c, 0x9ff4d, 0x9ff4e, 0x9ff4f, 0x9ff50, 0xcde9, /*0xf0-0xf7*/
4591 0x9ff51, 0xc5eb, 0x9ff52, 0x9ff53, 0x9ff54, 0xb7e9, 0x9ff55, 0x9ff56, /*0xf8-0xff*/
4592 /* 0x7100 */
4593 0x9ff57, 0x9ff58, 0x9ff59, 0x9ff5a, 0x9ff5b, 0x9ff5c, 0x9ff5d, 0x9ff5e, /*0x00-0x07*/
4594 0x9ff5f, 0xd1c9, 0xbab8, 0x9ff60, 0x9ff61, 0x9ff62, 0x9ff63, 0x9ff64, /*0x08-0x0f*/
4595 0xecc9, 0x9ff65, 0x9ff66, 0xecca, 0x9ff67, 0xbbc0, 0xeccb, 0x9ff68, /*0x10-0x17*/
4596 0xece2, 0xb1ba, 0xb7d9, 0x9ff69, 0x9ff6a, 0x9ff6b, 0x9ff6c, 0x9ff6d, /*0x18-0x1f*/
4597 0x9ff6e, 0x9ff6f, 0x9ff70, 0x9ff71, 0x9ff72, 0x9ff73, 0xbdb9, 0x9ff74, /*0x20-0x27*/
4598 0x9ff75, 0x9ff76, 0x9ff77, 0x9ff78, 0x9ff79, 0x9ff7a, 0x9ff7b, 0xeccc, /*0x28-0x2f*/
4599 0xd1e6, 0xeccd, 0x9ff7c, 0x9ff7d, 0x9ff7e, 0x9ff7f, 0xc8bb, 0x9ff81, /*0x30-0x37*/
4600 0x9ff82, 0x9ff83, 0x9ff84, 0x9ff85, 0x9ff86, 0x9ff87, 0x9ff88, 0x9ff89, /*0x38-0x3f*/
4601 0x9ff8a, 0x9ff8b, 0x9ff8c, 0x9ff8d, 0x9ff8e, 0xecc1, 0x9ff8f, 0x9ff90, /*0x40-0x47*/
4602 0x9ff91, 0x9ff92, 0xecc3, 0x9ff93, 0xbbc0, 0x9ff94, 0xbce5, 0x9ff95, /*0x48-0x4f*/
4603 0x9ff96, 0x9ff97, 0x9ff98, 0x9ff99, 0x9ffa0, 0x9ffa1, 0x9ffa2, 0x9ffa3, /*0x50-0x57*/
4604 0x9ffa4, 0x9ffa5, 0x9ffa6, 0x9ffa7, 0xc3ba, 0x9ffa8, 0xece3, 0xd5d5, /*0x58-0x5f*/
4605 0x9ffa9, 0x9ffa9, 0x9ffa9, 0x9ffab, 0x9ffac, 0x9ffad, 0xd6f3, 0x9ffae, /*0x60-0x6f*/
4606 0x9ffaf, 0x9ffb0, 0xecc2, 0xeccc, 0x9ffb1, 0x9ffb2, 0x9ffb3, 0x9ffb4, /*0x70-0x77*/
4607 0xecc4, 0x9ffb5, 0xecc5, 0x9ffb6, 0x9ffb7, 0xc9bf, 0x9ffb8, 0x9ffb9, /*0x78-0x7f*/
4608 0x9ffba, 0x9ffbb, 0x9ffbc, 0x9ffbd, 0xcfa8, 0x9ffbe, 0x9ffbf, 0x9ffc0, /*0x80-0x87*/

```

```
4610 0x9fc1, 0x9fc2, 0xd0dc, 0x9fc3, 0x9fc4, 0x9fc5, 0x9fc6, 0xd1ac, /*0x88-0x8f*/
4611 0x9fc7, 0x9fc8, 0x9fc9, 0x9fca, 0xc8db, 0x9fcb, 0x9fcc, 0x9fcd, /*0x90-0x97*/
4612 0xecd6, 0xcef5, 0x9fce, 0x9fcf, 0x9fd0, 0x9fd1, 0x9fd2, 0xcaec, /*0x98-0x9f*/
4613 0xecda, 0x9fd3, 0x9fd4, 0x9fd5, 0x9fd6, 0x9fd7, 0x9fd8, 0x9fd9, /*0xa0-0xa7*/
4614 0xecd9, 0x9fda, 0x9fdb, 0x9fdc, 0xb0be, 0x9fdd, 0x9fde, 0x9fde, /*0xa8-0xaf*/
4615 0x9fe0, 0x9fe1, 0x9fe2, 0xecd7, 0x9fe3, 0xecd8, 0x9fe4, 0x9fe5, /*0xb0-0xb7*/
4616 0x9fe6, 0xece4, 0x9fe7, 0x9fe8, 0x9fe9, 0x9fea, 0x9feb, 0x9fec, /*0xb8-0xbf*/
4617 0x9fed, 0x9fee, 0x9fef, 0xc8bc, 0x9ff0, 0x9ff1, 0x9ff2, 0x9ff3, /*0xc0-0xc7*/
4618 0x9ff4, 0x9ff5, 0x9ff6, 0x9ff7, 0x9ff8, 0x9ff9, 0xc1c7, 0x9ffa, /*0xc8-0xcf*/
4619 0x9ffb, 0x9ffc, 0x9ffd, 0x9ffe, 0xecdc, 0xd1e0, 0xa040, 0xa041, /*0xd0-0xd7*/
4620 0xa042, 0xa043, 0xa044, 0xa045, 0xa046, 0xa047, 0xa048, 0xa049, /*0xd8-0xdf*/
4621 0xecdb, 0xa04a, 0xa04b, 0xa04c, 0xa04d, 0xd4ef, 0xa04e, 0xecdd, /*0xe0-0xe7*/
4622 0xa04f, 0xa050, 0xa051, 0xa052, 0xa053, 0xa054, 0xdbcb, 0xa055, /*0xe8-0xef*/
4623 0xa056, 0xa057, 0xa058, 0xa059, 0xa05a, 0xa05b, 0xa05c, 0xa05d, /*0xf0-0xf7*/
4624 0xa05e, 0xecde, 0xa05f, 0xa060, 0xa061, 0xa062, 0xa063, 0xa064, /*0xf8-0xff*/
4625 /* 0x7200 */
4626 0xa065, 0xa066, 0xa067, 0xa068, 0xa069, 0xa06a, 0xb1ac, 0xa06b, /*0x00-0x07*/
4627 0xa06c, 0xa06d, 0xa06e, 0xa06f, 0xa070, 0xa071, 0xa072, 0xa073, /*0x08-0x0f*/
4628 0xa074, 0xa075, 0xa076, 0xa077, 0xa078, 0xa079, 0xa07a, 0xa07b, /*0x10-0x17*/
4629 0xa07c, 0xa07d, 0xa07e, 0xa080, 0xa081, 0xecdf, 0xa082, 0xa083, /*0x18-0x1f*/
4630 0xa084, 0xa085, 0xa086, 0xa087, 0xa088, 0xa089, 0xa08a, 0xa08b, /*0x20-0x27*/
4631 0xece0, 0xa08c, 0xd7a6, 0xa08d, 0xc5c0, 0xa08e, 0xa08f, 0xa090, /*0x28-0x2f*/
4632 0xebbc, 0xb0ae, 0xa091, 0xa092, 0xa093, 0xbef4, 0xb8b8, 0xd2af, /*0x30-0x37*/
4633 0xb0d6, 0xb5f9, 0xa094, 0xd8b3, 0xa095, 0xcba0, 0xa096, 0xe3dd, /*0x38-0x3f*/
4634 0xa097, 0xa098, 0xa099, 0xa09a, 0xa09b, 0xa09c, 0xa09d, 0xc6ac, /*0x40-0x47*/
4635 0xb0e6, 0xa09e, 0xa09f, 0xa0a0, 0xc5c6, 0xebb9, 0xa0a1, 0xa0a2, /*0x48-0x4f*/
4636 0xa0a3, 0xa0a4, 0xebba, 0xa0a5, 0xa0a6, 0xa0a7, 0xebbb, 0xa0a8, /*0x50-0x57*/
4637 0xa0a9, 0xd1c0, 0xa0aa, 0xc5a3, 0xa0ab, 0xeaf2, 0xa0ac, 0xc4b2, /*0x58-0x5f*/
4638 0xa0ad, 0xc4b5, 0xc0ce, 0xa0ae, 0xa0af, 0xa0b0, 0xeaf3, 0xc4c1, /*0x60-0x67*/
4639 0xa0b1, 0xccef, 0xa0b2, 0xa0b3, 0xa0b4, 0xa0b5, 0xeaf0, 0xeaf4, /*0x68-0x6f*/
4640 0xa0b6, 0xa0b7, 0xc9fc, 0xa0b8, 0xa0b9, 0xc7a3, 0xa0ba, 0xa0bb, /*0x70-0x77*/
4641 0xa0bc, 0xccd8, 0xcfe, 0xa0bd, 0xa0be, 0xa0bf, 0xeaf5, 0xeaf6, /*0x78-0x7f*/
4642 0xcfac, 0xc0f2, 0xa0c0, 0xa0c1, 0xeaf7, 0xa0c2, 0xa0c3, 0xa0c4, /*0x80-0x87*/
4643 0xa0c5, 0xa0c6, 0xb6bf, 0xeaf8, 0xa0c7, 0xeaf9, 0xa0c8, 0xeafa, /*0x88-0x8f*/
4644 0xa0c9, 0xa0ca, 0xeafb, 0xa0cb, 0xa0cc, 0xa0cd, 0xa0ce, 0xa0cf, /*0x90-0x97*/
4645 0xa0d0, 0xa0d1, 0xa0d2, 0xa0d3, 0xa0d4, 0xa0d5, 0xa0d6, 0xeaf1, /*0x98-0x9f*/
4646 0xa0d7, 0xa0d8, 0xa0d9, 0xa0da, 0xa0db, 0xa0dc, 0xa0dd, 0xa0de, /*0xa0-0xa7*/
4647 0xa0df, 0xa0e0, 0xa0e1, 0xa0e2, 0xc8ae, 0xe1eb, 0xa0e3, 0xb7b8, /*0xa8-0xaf*/
4648 0xe1ec, 0xa0e4, 0xa0e5, 0xa0e6, 0xe1ed, 0xa0e7, 0xd7b4, 0xe1ee, /*0xb0-0xb7*/
4649 0xe1ef, 0xd3cc, 0xa0e8, 0xa0e9, 0xa0ea, 0xa0eb, 0xa0ec, 0xa0ed, /*0xb8-0xbf*/
4650 0xa0ee, 0xe1f1, 0xbff1, 0xe1f2, 0xb5d2, 0xa0ef, 0xa0f0, 0xa0f1, /*0xc0-0xc7*/
4651 0xeb1b, 0xa0f2, 0xa0f3, 0xa0f4, 0xa0f5, 0xe1f3, 0xe1f2, 0xa0f6, /*0xc8-0xcf*/
4652 0xbafc, 0xa0f7, 0xe1f4, 0xa0f8, 0xa0f9, 0xa0fa, 0xa0fb, 0xb9b7, /*0xd0-0xd7*/
4653 0xa0fc, 0xbed1, 0xa0fd, 0xa0fe, 0xaa40, 0xaa41, 0xc4fc, 0xaa42, /*0xd8-0xdf*/
4654 0xbadd, 0xbdc6, 0xaa43, 0xaa44, 0xaa45, 0xaa46, 0xaa47, 0xaa48, /*0xe0-0xe7*/
4655 0xe1f5, 0xe1f7, 0xaa49, 0xaa4a, 0xb6c0, 0xcfc1, 0xcaa8, 0xe1f6, /*0xe8-0xef*/
4656 0xd5f8, 0xd3fc, 0xe1f8, 0xe1f9, 0xe1fa, 0xaa4b, 0xaa4c, 0xe1fa, /*0xf0-0xf7*/
4657 0xc0ea, 0xaa4d, 0xe1fe, 0xe2a1, 0xc0c7, 0xaa4e, 0xaa4f, 0xaa50, /*0xf8-0xff*/
4658 /* 0x7300 */
4659 0xaa51, 0xe1fb, 0xaa52, 0xe1fd, 0xaa53, 0xaa54, 0xaa55, 0xaa56, /*0x00-0x07*/
4660 0xaa57, 0xaa58, 0xe2a5, 0xaa59, 0xaa5a, 0xaa5b, 0xc1d4, 0xaa5c, /*0x08-0x0f*/
4661 0xaa5d, 0xaa5e, 0xaa5f, 0xe2a3, 0xaa60, 0xe2a8, 0xb2fe, 0xe2a2, /*0x10-0x17*/
4662 0xaa61, 0xaa62, 0xaa63, 0xc3cd, 0xb2c2, 0xe2a7, 0xe2a6, 0xaa64, /*0x18-0x1f*/
4663 0xaa65, 0xe2a4, 0xe2a9, 0xaa66, 0xaa67, 0xe2ab, 0xaa68, 0xaa69, /*0x20-0x27*/
4664 0xaa6a, 0xd0c9, 0xd6ed, 0xc3a8, 0xe2ac, 0xaa6b, 0xcfd7, 0xaa6c, /*0x28-0x2f*/
4665 0xaa6d, 0xe2ae, 0xaa6e, 0xaa6f, 0xbaef, 0xaa70, 0xaa71, 0xe9e0, /*0x30-0x37*/
4666 0xe2ad, 0xe2aa, 0xaa72, 0xaa73, 0xaa74, 0xaa75, 0xbbab, 0xd4b3, /*0x38-0x3f*/
4667 0xaa76, 0xaa77, 0xaa78, 0xaa79, 0xaa7a, 0xaa7b, 0xaa7c, 0xaa7d, /*0x40-0x47*/
4668 0xaa7e, 0xaa80, 0xaa81, 0xaa82, 0xaa83, 0xe2b0, 0xaa84, 0xaa85, /*0x48-0x4f*/
4669 0xe2af, 0xaa86, 0xe9e1, 0xaa87, 0xaa88, 0xaa89, 0xaa8a, 0xe2b1, /*0x50-0x57*/
4670 0xaa8b, 0xaa8c, 0xaa8d, 0xaa8e, 0xaa8f, 0xaa90, 0xaa91, 0xaa92, /*0x58-0x5f*/
4671 0xe2b2, 0xaa93, 0xaa94, 0xaa95, 0xaa96, 0xaa97, 0xaa98, 0xaa99, /*0x60-0x67*/
4672 0xaa9a, 0xaa9b, 0xaa9c, 0xaa9d, 0xe2b3, 0xccal, 0xaa9e, 0xe2b4, /*0x68-0x6f*/
4673 0xaa9f, 0xaaa0, 0xab40, 0xab41, 0xab42, 0xab43, 0xab44, 0xab45, /*0x70-0x77*/
4674 0xab46, 0xab47, 0xab48, 0xab49, 0xab4a, 0xab4b, 0xe2b5, 0xab4c, /*0x78-0x7f*/
4675 0xab4d, 0xab4e, 0xab4f, 0xab50, 0xd0fe, 0xab51, 0xab52, 0xc2ca, /*0x80-0x87*/
4676 0xab53, 0xd3f1, 0xab54, 0xcdf5, 0xab55, 0xab56, 0xe7e0, 0xab57, /*0x88-0x8f*/
4677 0xab58, 0xe7e1, 0xab59, 0xab5a, 0xab5b, 0xab5c, 0xbec1, 0xab5d, /*0x90-0x97*/
4678 0xab5e, 0xab5f, 0xab60, 0xc2ea, 0xab61, 0xab62, 0xab63, 0xe7e4, /*0x98-0x9f*/
4679 0xab64, 0xab65, 0xe7e3, 0xab66, 0xab67, 0xab68, 0xab69, 0xab6a, /*0xa0-0xa7*/
4680 0xab6b, 0xcde6, 0xab6c, 0xc3b5, 0xab6d, 0xab6e, 0xe7e2, 0xbbb7, /*0xa8-0xaf*/
4681 0xcfd6, 0xab6f, 0xc1e1, 0xe7e9, 0xab70, 0xab71, 0xab72, 0xe7e8, /*0xb0-0xb7*/
4682 0xab73, 0xab74, 0xe7f4, 0xb2a3, 0xab75, 0xab76, 0xab77, 0xab78, /*0xb8-0xbf*/
4683 0xe7ea, 0xab79, 0xe7e6, 0xab7a, 0xab7b, 0xab7c, 0xab7d, 0xab7e, /*0xc0-0xc7*/
4684 0xe7ec, 0xe7eb, 0xc9ba, 0xab80, 0xab81, 0xd5e4, 0xab82, 0xe7e5, /*0xc8-0xcf*/
4685 0xb7a9, 0xe7ee, 0xab83, 0xab84, 0xab85, 0xab86, 0xab87, 0xab88, /*0xd0-0xd7*/
4686 0xab89, 0xe7ee, 0xab8a, 0xab8b, 0xab8c, 0xab8d, 0xe7f3, 0xab8e, /*0xd8-0xdf*/
4687 0xd6e9, 0xab8f, 0xab90, 0xab91, 0xab92, 0xe7ed, 0xab93, 0xe7f2, /*0xe0-0xe7*/
4688 0xab94, 0xe7f1, 0xab95, 0xab96, 0xab97, 0xb0e0, 0xab98, 0xab99, /*0xe8-0xef*/
4689 0xab9a, 0xab9b, 0xe7f5, 0xab9c, 0xab9d, 0xab9e, 0xab9f, 0xaba0, /*0xf0-0xf7*/
4690 0xac40, 0xac41, 0xac42, 0xac43, 0xac44, 0xac45, 0xac46, 0xac47, /*0xf8-0xff*/
4691 /* 0x7400 */
4692 0xac48, 0xac49, 0xac4a, 0xc7f2, 0xac4b, 0xc0c5, 0xc0ed, 0xac4c, /*0x00-0x07*/
4693 0xac4d, 0xc1f0, 0xe7f0, 0xac4e, 0xac4f, 0xac50, 0xac51, 0xe7f6, /*0x08-0x0f*/
4694 0xcbf6, 0xac52, 0xac53, 0xac54, 0xac55, 0xac56, 0xac57, 0xac58, /*0x10-0x17*/
4695 0xac59, 0xac5a, 0xe8a2, 0xe8a1, 0xac5b, 0xac5c, 0xac5d, 0xac5e, /*0x18-0x1f*/
4696 0xac5f, 0xac60, 0xd7c1, 0xac61, 0xac62, 0xe7fa, 0xe7f9, 0xac63, /*0x20-0x27*/
```

```

4697 0xe7fb, 0xac64, 0xe7f7, 0xac65, 0xe7fe, 0xac66, 0xe7fd, 0xac67, /*0x28-0x2f*/
4698 0xe7fc, 0xac68, 0xac69, 0xcld5, 0xc7d9, 0xc5fd, 0xc5c3, 0xac6a, /*0x30-0x37*/
4699 0xac6b, 0xac6c, 0xac6d, 0xac6e, 0xc7ed, 0xac6f, 0xac70, 0xac71, /*0x38-0x3f*/
4700 0xac72, 0xe8a3, 0xac73, 0xac74, 0xac75, 0xac76, 0xac77, 0xac78, /*0x40-0x47*/
4701 0xac79, 0xac7a, 0xac7b, 0xac7c, 0xac7d, 0xac7e, 0xac80, 0xac81, /*0x48-0x4f*/
4702 0xac82, 0xac83, 0xac84, 0xac85, 0xac86, 0xe8a6, 0xac87, 0xe8a5, /*0x50-0x57*/
4703 0xac88, 0xe8a7, 0xbaf7, 0xe7f8, 0xe8a4, 0xac89, 0xc8f0, 0xc9aa, /*0x58-0x5f*/
4704 0xac8a, 0xac8b, 0xac8c, 0xac8d, 0xac8e, 0xac8f, 0xac90, 0xac91, /*0x60-0x67*/
4705 0xac92, 0xac93, 0xac94, 0xac95, 0xac96, 0xe8a9, 0xac97, 0xac98, /*0x68-0x6f*/
4706 0xb9e5, 0xac99, 0xac9a, 0xac9b, 0xac9c, 0xac9d, 0xd1fe, 0xe8a8, /*0x70-0x77*/
4707 0xac9e, 0xac9f, 0xaca0, 0xad40, 0xad41, 0xad42, 0xe8aa, 0xad43, /*0x78-0x7f*/
4708 0xe8ad, 0xe8ae, 0xad44, 0xc1a7, 0xad45, 0xad46, 0xad47, 0xe8af, /*0x80-0x87*/
4709 0xad48, 0xad49, 0xad4a, 0xe8b0, 0xad4b, 0xad4c, 0xe8ac, 0xad4d, /*0x88-0x8f*/
4710 0xe8b4, 0xad4e, 0xad4f, 0xad50, 0xad51, 0xad52, 0xad53, 0xad54, /*0x90-0x97*/
4711 0xad55, 0xad56, 0xad57, 0xad58, 0xe8ab, 0xad59, 0xe8b1, 0xad5a, /*0x98-0x9f*/
4712 0xad5b, 0xad5c, 0xad5d, 0xad5e, 0xad5f, 0xad60, 0xad61, 0xe8b5, /*0xa0-0xa7*/
4713 0xe8b2, 0xe8b3, 0xad62, 0xad63, 0xad64, 0xad65, 0xad66, 0xad67, /*0xa8-0xaf*/
4714 0xad68, 0xad69, 0xad6a, 0xad6b, 0xad6c, 0xad6d, 0xad6e, 0xad6f, /*0xb0-0xbf*/
4715 0xad70, 0xad71, 0xe8b7, 0xad72, 0xad73, 0xad74, 0xad75, 0xad76, /*0xb8-0xbf*/
4716 0xad77, 0xad78, 0xad79, 0xad7a, 0xad7b, 0xad7c, 0xad7d, 0xad7e, /*0xc0-0xcf*/
4717 0xad80, 0xad81, 0xad82, 0xad83, 0xad84, 0xad85, 0xad86, 0xad87, /*0xc8-0xcf*/
4718 0xad88, 0xad89, 0xe8b6, 0xad8a, 0xad8b, 0xad8c, 0xad8d, 0xad8e, /*0xd0-0xdf*/
4719 0xad8f, 0xad90, 0xad91, 0xad92, 0xb9cf, 0xad93, 0xf0ac, 0xad94, /*0xd8-0xdf*/
4720 0xf0ad, 0xad95, 0xc6b0, 0xb0ea, 0xc8bf, 0xad96, 0xcddf, 0xad97, /*0xe0-0xef*/
4721 0xad98, 0xad99, 0xad9a, 0xad9b, 0xad9c, 0xad9d, 0xcecd, 0xeab1, /*0xe8-0xef*/
4722 0xad9e, 0xad9f, 0xada0, 0xae40, 0xeab2, 0xae41, 0xc6bf, 0xb4c9, /*0xf0-0xff*/
4723 0xae42, 0xae43, 0xae44, 0xae45, 0xae46, 0xae47, 0xae48, 0xeab3, /*0xf8-0xff*/
4724 /* 0x7500 */
4725 0xae49, 0xae4a, 0xae4b, 0xae4c, 0xd5e7, 0xae4d, 0xae4e, 0xae4f, /*0x00-0x07*/
4726 0xae50, 0xae51, 0xae52, 0xae53, 0xae54, 0xae55, 0xeab4, /*0x08-0x0f*/
4727 0xae56, 0xeab5, 0xae57, 0xeab6, 0xae58, 0xae59, 0xae5a, 0xae5b, /*0x10-0x17*/
4728 0xb8ca, 0xdfb0, 0xc9f5, 0xae5c, 0xccf0, 0xae5d, 0xae5e, 0xc9fa, /*0x18-0x1f*/
4729 0xae5f, 0xae60, 0xae61, 0xae62, 0xae63, 0xc9fb, 0xae64, 0xae65, /*0x20-0x27*/
4730 0xd3c3, 0xcba6, 0xae66, 0xb8a6, 0xf0ae, 0xb1c2, 0xae67, 0xe5b8, /*0x28-0x2f*/
4731 0xccef, 0xd3c9, 0xbcd7, 0xc9ea, 0xae68, 0xb5e7, 0xae69, 0xc4d0, /*0x30-0x37*/
4732 0xe5e9, 0xae6a, 0xeeae, 0xbbad, 0xae6b, 0xae6c, 0xe7de, 0xae6d, /*0x38-0x3f*/
4733 0xeeaf, 0xae6e, 0xae6f, 0xae70, 0xae71, 0xb3a9, 0xae72, 0xae73, /*0x40-0x47*/
4734 0xeeb2, 0xae74, 0xae75, 0xeeb1, 0xbde7, 0xae76, 0xeeb0, 0xceb7, /*0x48-0x4f*/
4735 0xae77, 0xae78, 0xae79, 0xae7a, 0xc5cf, 0xae7b, 0xae7c, 0xae7d, /*0x50-0x57*/
4736 0xae7e, 0xc1f4, 0xdabc, 0xeeb3, 0xd0f3, 0xae80, 0xae81, 0xae82, /*0x58-0x5f*/
4737 0xae83, 0xae84, 0xae85, 0xae86, 0xae87, 0xc2d4, 0xc6e8, 0xae88, /*0x60-0x67*/
4738 0xae89, 0xae8a, 0xb7ac, 0xae8b, 0xae8c, 0xae8d, 0xae8e, 0xae8f, /*0x68-0x6f*/
4739 0xae90, 0xae91, 0xeeb4, 0xae92, 0xb3eb, 0xae93, 0xae94, 0xae95, /*0x70-0x77*/
4740 0xbbf6, 0xeeb5, 0xae96, 0xae97, 0xae98, 0xae99, 0xae9a, 0xe7dc, /*0x78-0x7f*/
4741 0xae9b, 0xae9c, 0xae9d, 0xeeb6, 0xae9e, 0xae9f, 0xbdae, 0xaea0, /*0x80-0x87*/
4742 0xaf40, 0xaf41, 0xaf42, 0xf1e2, 0xaf43, 0xaf44, 0xaf45, 0xcae8, /*0x88-0x8f*/
4743 0xaf46, 0xd2c9, 0xf0da, 0xaf47, 0xf0db, 0xaf48, 0xf0dc, 0xc1c6, /*0x90-0x97*/
4744 0xaf49, 0xb8ed, 0xbece, 0xaf4a, 0xaf4b, 0xf0de, 0xaf4c, 0xc5b1, /*0x98-0x9f*/
4745 0xf0dd, 0xd1f1, 0xaf4d, 0xf0e0, 0xb0cc, 0xbdea, 0xaf4e, 0xaf4f, /*0xa0-0xaf*/
4746 0xaf50, 0xaf51, 0xaf52, 0xd2df, 0xf0df, 0xaf53, 0xb4af, 0xb7e8, /*0xa8-0xaf*/
4747 0xf0e6, 0xf0e5, 0xc6a3, 0xf0e1, 0xf0e2, 0xb4c3, 0xaf54, 0xaf55, /*0xb0-0xbf*/
4748 0xf0e3, 0xd5ee, 0xaf56, 0xaf57, 0xccdb, 0xbcd2, 0xbcb2, 0xaf58, /*0xb8-0xbf*/
4749 0xaf59, 0xaf5a, 0xf0e8, 0xf0e7, 0xf0e4, 0xb2a1, 0xaf5b, 0xd6a2, /*0xc0-0xcf*/
4750 0xd3b8, 0xbcb7, 0xc8ac, 0xaf5c, 0xaf5d, 0xf0ea, 0xaf5e, 0xaf5f, /*0xc8-0xcf*/
4751 0xaf60, 0xaf61, 0xd1f7, 0xaf62, 0xd6cc, 0xbadb, 0xf0e9, 0xaf63, /*0xd0-0xdf*/
4752 0xb6bb, 0xaf64, 0xaf65, 0xcdb4, 0xaf66, 0xaf67, 0xc6a6, 0xaf68, /*0xd8-0xdf*/
4753 0xaf69, 0xaf6a, 0xc1a1, 0xf0eb, 0xf0ee, 0xaf6b, 0xf0ed, 0xf0f0, /*0xe0-0xef*/
4754 0xf0ec, 0xaf6c, 0xbbbe, 0xf0ef, 0xaf6d, 0xaf6e, 0xaf6f, 0xaf70, /*0xe8-0xef*/
4755 0xccb5, 0xf0f2, 0xaf71, 0xaf72, 0xb3d5, 0xaf73, 0xaf74, 0xaf75, /*0xf0-0xff*/
4756 0xaf76, 0xb1d4, 0xaf77, 0xaf78, 0xf0f3, 0xaf79, 0xaf7a, 0xf0f4, /*0xf8-0xff*/
4757 /* 0x7600 */
4758 0xf0f6, 0xb4e1, 0xaf7b, 0xf0f1, 0xaf7c, 0xf0f7, 0xaf7d, 0xaf7e, /*0x00-0x07*/
4759 0xaf80, 0xaf81, 0xf0fa, 0xaf82, 0xf0f8, 0xaf83, 0xaf84, 0xaf85, /*0x08-0x0f*/
4760 0xf0f5, 0xaf86, 0xaf87, 0xaf88, 0xaf89, 0xf0fd, 0xaf8a, 0xf0f9, /*0x10-0x17*/
4761 0xf0fc, 0xf0fe, 0xaf8b, 0xf1a1, 0xaf8c, 0xaf8d, 0xaf8e, 0xcce1, /*0x18-0x1f*/
4762 0xf1a4, 0xaf8f, 0xf1a3, 0xaf90, 0xc1f6, 0xf0fb, 0xcadd, 0xaf91, /*0x20-0x27*/
4763 0xaf92, 0xb4f1, 0xb1f1, 0xccb1, 0xaf93, 0xf1a6, 0xaf94, 0xaf95, /*0x28-0x2f*/
4764 0xf1a7, 0xaf96, 0xaf97, 0xf1ac, 0xd5ce, 0xf1a9, 0xaf98, 0xaf99, /*0x30-0x37*/
4765 0xc8b3, 0xaf9a, 0xaf9b, 0xaf9c, 0xf1a2, 0xaf9d, 0xf1ab, 0xf1a8, /*0x38-0x3f*/
4766 0xf1a5, 0xaf9e, 0xaf9f, 0xf1aa, 0xf1a0, 0xb040, 0xb041, 0xb042, /*0x40-0x47*/
4767 0xb043, 0xb044, 0xb045, 0xb046, 0xb0a9, 0xf1ad, 0xb047, 0xb048, /*0x48-0x4f*/
4768 0xb049, 0xb04a, 0xb04b, 0xb04c, 0xf1af, 0xb04d, 0xf1b1, 0xb04e, /*0x50-0x57*/
4769 0xb04f, 0xb050, 0xb051, 0xb052, 0xf1b0, 0xb053, 0xf1ae, 0xb054, /*0x58-0x5f*/
4770 0xb055, 0xb056, 0xb057, 0xd1a2, 0xb058, 0xb059, 0xb05a, 0xb05b, /*0x60-0x67*/
4771 0xb05c, 0xb05d, 0xb05e, 0xf1b2, 0xb05f, 0xb060, 0xb061, 0xf1b3, /*0x68-0x6f*/
4772 0xb062, 0xb063, 0xb064, 0xb065, 0xb066, 0xb067, 0xb068, 0xb069, /*0x70-0x77*/
4773 0xb06e, 0xb06a, 0xb06b, 0xb06c, 0xb06d, 0xb06e, 0xb06f, 0xb06d, /*0x78-0x7f*/
4774 0xb06e, 0xb06f, 0xd4ed, 0xb070, 0xb05c4, 0xb071, 0xbdd4, 0xbbaa, /*0x80-0x87*/
4775 0xf0a7, 0xb072, 0xb073, 0xb074, 0xb075, 0xf0a8, 0xb076, 0xb077, /*0x88-0x8f*/
4776 0xb077, 0xb0a8, 0xb078, 0xf0a9, 0xb079, 0xb07a, 0xcdee, 0xb07b, /*0x90-0x97*/
4777 0xb07c, 0xf0aa, 0xb07d, 0xb07e, 0xb080, 0xb081, 0xb082, 0xb083, /*0x98-0x9f*/
4778 0xb084, 0xb085, 0xb086, 0xb087, 0xf0ab, 0xb088, 0xb089, 0xb08a, /*0xa0-0xaf*/
4779 0xb08b, 0xb08c, 0xb08d, 0xb08e, 0xb08f, 0xb090, 0xc6a4, 0xb091, /*0xa8-0xaf*/
4780 0xb092, 0xd6e5, 0xf1e4, 0xb093, 0xf1e5, 0xb094, 0xb095, 0xb096, /*0xb0-0xbf*/
4781 0xb097, 0xb098, 0xb099, 0xb09a, 0xb09b, 0xb09c, 0xb09d, 0xc3f3, /*0xb8-0xbf*/
4782 0xb09e, 0xb09f, 0xd3db, 0xb0a0, 0xb140, 0xd6d1, 0xc5e8, 0xb141, /*0xc0-0xcf*/
4783 0xd3af, 0xb142, 0xd2e6, 0xb143, 0xb144, 0xeec1, 0xb0bb, 0xd5b5, /*0xc8-0xcf*/

```



```
4784 0xd1ce, 0xbce0, 0xbad0, 0xb145, 0xbff8, 0xb146, 0xb8c7, 0xb5c1, /*0xd0-0xd7*/
4785 0xc5cc, 0xb147, 0xb148, 0xcaa2, 0xb149, 0xb14a, 0xb14b, 0xc3cb, /*0xd8-0xdf*/
4786 0xb14c, 0xb14d, 0xb14e, 0xb14f, 0xb150, 0xeec2, 0xb151, 0xb152, /*0xe0-0xef*/
4787 0xb153, 0xb154, 0xb155, 0xb156, 0xb157, 0xb158, 0xc4bf, 0xb6a2, /*0xe8-0xef*/
4788 0xb159, 0xedec, 0xc3a4, 0xb15a, 0xd6b1, 0xb15b, 0xb15c, 0xb15d, /*0xf0-0xff*/
4789 0xcfe0, 0xedef, 0xb15e, 0xb15f, 0xc5ce, 0xb160, 0xb6dc, 0xb161, /*0xf8-0xff*/
4790 /* 0x7700 */
4791 0xb162, 0xcaa1, 0xb163, 0xb164, 0xeded, 0xb165, 0xb166, 0xedf0, /*0x00-0x07*/
4792 0xedf1, 0xc3bc, 0xb167, 0xbfb4, 0xb168, 0xedee, 0xb169, 0xb16a, /*0x08-0x0f*/
4793 0xb16b, 0xb16c, 0xb16d, 0xb16e, 0xb16f, 0xb170, 0xb171, 0xb172, /*0x10-0x17*/
4794 0xb173, 0xedf4, 0xedf2, 0xb174, 0xb175, 0xb176, 0xb177, 0xd5e6, /*0x18-0x1f*/
4795 0xc3df, 0xb178, 0xedf3, 0xb179, 0xb17a, 0xb17b, 0xedf6, 0xb17c, /*0x20-0x27*/
4796 0xd5a3, 0xd1a3, 0xb17d, 0xb17e, 0xb180, 0xedf5, 0xb181, 0xc3d0, /*0x28-0x2f*/
4797 0xb182, 0xb183, 0xb184, 0xb185, 0xb186, 0xedf7, 0xbff4, 0xbee, /*0x30-0x37*/
4798 0xedf8, 0xb187, 0xccf7, 0xb188, 0xd1db, 0xb189, 0xb18a, 0xb18b, /*0x38-0x3f*/
4799 0xd7c5, 0xd5f6, 0xb18c, 0xedfc, 0xb18d, 0xb18e, 0xb18f, 0xedfb, /*0x40-0x47*/
4800 0xb190, 0xb191, 0xb192, 0xb193, 0xb194, 0xb195, 0xb196, 0xb197, /*0x48-0x4f*/
4801 0xedf9, 0xedfa, 0xb198, 0xb199, 0xb19a, 0xb19b, 0xb19c, 0xb19d, /*0x50-0x57*/
4802 0xb19e, 0xb19f, 0xedfd, 0xbea6, 0xb1a0, 0xb240, 0xb241, 0xb242, /*0x58-0x5f*/
4803 0xb243, 0xcba, 0xeea1, 0xb6bd, 0xb244, 0xeea2, 0xc4c0, 0xb245, /*0x60-0x67*/
4804 0xedfe, 0xb246, 0xb247, 0xb248, 0xb249, 0xb24a, /*0x68-0x6f*/
4805 0xb24b, 0xb24c, 0xb24d, 0xb24e, 0xb24f, 0xb250, 0xb251, 0xb252, /*0x70-0x77*/
4806 0xb253, 0xb6c3, 0xb254, 0xb255, 0xb256, 0xeea5, 0xd8ba, 0xeea3, /*0x78-0x7f*/
4807 0xeea6, 0xb257, 0xb258, 0xc3e9, 0xb3f2, 0xb25a, 0xb25b, /*0x80-0x87*/
4808 0xb25c, 0xb25d, 0xb25e, 0xb25f, 0xeea7, 0xeea4, 0xcf9, 0xb260, /*0x88-0x8f*/
4809 0xb261, 0xeea8, 0xc2f7, 0xb262, 0xb263, 0xb264, 0xb265, 0xb266, /*0x90-0x97*/
4810 0xb267, 0xb268, 0xb269, 0xb26a, 0xb26b, 0xb26c, 0xb26d, 0xeea9, /*0x98-0x9f*/
4811 0xeeaa, 0xb26e, 0xdeab, 0xb26f, 0xb270, 0xc6b3, 0xb271, 0xc7c6, /*0xa0-0xaf*/
4812 0xb272, 0xd6f5, 0xb5c9, 0xb273, 0xcbb2, 0xb274, 0xb275, 0xb276, /*0xa8-0xaf*/
4813 0xeeab, 0xb277, 0xb278, 0xcadb, 0xb279, 0xeeac, 0xb27a, 0xb27b, /*0xb0-0xbf*/
4814 0xb27c, 0xb27d, 0xb27e, 0xd5b0, 0xb280, 0xeead, 0xb281, 0xf6c4, /*0xb8-0xbf*/
4815 0xb282, 0xb283, 0xb284, 0xb285, 0xb286, 0xb287, 0xb288, 0xb289, /*0xc0-0xcf*/
4816 0xb28a, 0xb28b, 0xb28c, 0xb28d, 0xb28e, 0xd8c7, 0xb28f, 0xb290, /*0xc8-0xcf*/
4817 0xb291, 0xb292, 0xb293, 0xb294, 0xb295, 0xb296, 0xb297, 0xb4a3, /*0xd0-0xd7*/
4818 0xb298, 0xb299, 0xb29a, 0xc3ac, 0xf1e6, 0xb29b, 0xb29c, 0xb29d, /*0xd8-0xdf*/
4819 0xb29e, 0xb29f, 0xcab8, 0xd2d3, 0xb2a0, 0xd6aa, 0xb340, 0xefff2, /*0xe0-0xef*/
4820 0xb341, 0xbed8, 0xb342, 0xbdc3, 0xefff3, 0xb6cc, 0xb0ab, 0xb343, /*0xe8-0xef*/
4821 0xb344, 0xb345, 0xb346, 0xcaaf, 0xb347, 0xb348, 0xedb6, 0xb349, /*0xf0-0xff*/
4822 0xedb7, 0xb34a, 0xb34b, 0xb34c, 0xb34d, 0xcef9, 0xb7af, 0xbfff3, /*0xf8-0xff*/
4823 /* 0x7800 */
4824 0xedb8, 0xc2eb, 0xc9b0, 0xb34e, 0xb34f, 0xb350, 0xb351, 0xb352, /*0x00-0x07*/
4825 0xb353, 0xedb9, 0xb354, 0xb355, 0xc6f6, 0xbfb3, 0xb356, 0xb357, /*0x08-0x0f*/
4826 0xb358, 0xedbc, 0xc5f8, 0xb359, 0xd1d0, 0xb35a, 0xd7a9, 0xedba, /*0x10-0x17*/
4827 0xedbb, 0xb35b, 0xd1e2, 0xb35c, 0xedbf, 0xedc0, 0xb35d, 0xedc4, /*0x18-0x1f*/
4828 0xb35e, 0xb35f, 0xb360, 0xedc8, 0xb361, 0xedc6, 0xedce, 0xd5e8, /*0x20-0x27*/
4829 0xb362, 0xedc9, 0xb363, 0xb364, 0xedc7, 0xedbe, 0xb365, 0xb366, /*0x28-0x2f*/
4830 0xc5e9, 0xb367, 0xb368, 0xb369, 0xc6c6, 0xb36a, 0xb36b, 0xc9e9, /*0x30-0x37*/
4831 0xd4d2, 0xedc1, 0xedc2, 0xedc3, 0xedc5, 0xb36c, 0xc0f9, 0xb36d, /*0x38-0x3f*/
4832 0xb4a1, 0xb36e, 0xb36f, 0xb370, 0xb371, 0xb9e8, 0xb372, 0xedd0, /*0x40-0x47*/
4833 0xb373, 0xb374, 0xb375, 0xb376, 0xedd1, 0xb377, 0xedca, 0xb378, /*0x48-0x4f*/
4834 0xedcf, 0xb379, 0xcdf8, 0xb37a, 0xb37b, 0xcbb6, 0xedcc, 0xedcd, /*0x50-0x57*/
4835 0xb37c, 0xb37d, 0xb37e, 0xb380, 0xb381, 0xcfff5, 0xb382, 0xb383, /*0x58-0x5f*/
4836 0xb384, 0xb385, 0xb386, 0xb387, 0xb388, 0xb389, 0xb38a, 0xb38b, /*0x60-0x67*/
4837 0xb38c, 0xb38d, 0xedd2, 0xc1f2, 0xd3b2, 0xedcb, 0xc8b7, 0xb38e, /*0x68-0x6f*/
4838 0xb38f, 0xb390, 0xb391, 0xb392, 0xb393, 0xb394, 0xb395, 0xbcef, /*0x70-0x77*/
4839 0xb396, 0xb397, 0xb398, 0xb399, 0xc5f0, 0xb39a, 0xb39b, 0xb39c, /*0x78-0x7f*/
4840 0xb39d, 0xb39e, 0xb39f, 0xb3a0, 0xb440, 0xb441, 0xb442, 0xedd6, /*0x80-0x87*/
4841 0xb443, 0xb5ef, 0xb444, 0xb445, 0xc2b5, 0xb0ad, 0xcbe9, 0xb446, /*0x88-0x8f*/
4842 0xb447, 0xb1ae, 0xb448, 0xedd4, 0xb449, 0xb44a, 0xb44b, 0xcdeb, /*0x90-0x97*/
4843 0xb5e2, 0xb44c, 0xedd5, 0xedd3, 0xedd7, 0xb44d, 0xb44e, 0xb5fa, /*0x98-0x9f*/
4844 0xb44f, 0xedd8, 0xb450, 0xedd9, 0xb451, 0xeddc, 0xb452, 0xb1cc, /*0xa0-0xaf*/
4845 0xb453, 0xb454, 0xb455, 0xb456, 0xb457, 0xb458, 0xb459, 0xb45a, /*0xa8-0xaf*/
4846 0xc5f6, 0xbcee, 0xedda, 0xcchc, 0xb2ea, 0xb45b, 0xb45c, 0xb45d, /*0xb0-0xbf*/
4847 0xb45e, 0xeddb, 0xb45f, 0xb460, 0xb461, 0xb462, 0xc4eb, 0xb463, /*0xb8-0xbf*/
4848 0xb464, 0xb4c5, 0xb465, 0xb466, 0xb467, 0xb0f5, 0xb468, 0xb469, /*0xc0-0xcf*/
4849 0xb46a, 0xeddf, 0xc0da, 0xb46b, 0xb46c, 0xb46d, 0xb46e, /*0xc8-0xcf*/
4850 0xc5cd, 0xb46f, 0xb470, 0xb471, 0xeddd, 0xbfc4, 0xb472, 0xb473, /*0xd0-0xd7*/
4851 0xb474, 0xedde, 0xb475, 0xb476, 0xb477, 0xb478, 0xb479, 0xb47a, /*0xd8-0xdf*/
4852 0xb47b, 0xb47c, 0xb47d, 0xb47e, 0xb480, 0xb481, 0xb482, 0xb483, /*0xe0-0xef*/
4853 0xc4a5, 0xb484, 0xb485, 0xb486, 0xede0, 0xb487, 0xb488, 0xb489, /*0xe8-0xef*/
4854 0xb48a, 0xb48b, 0xb48d, 0xb48e, 0xede3, 0xb48d, 0xb48e, 0xbcd7, /*0xf0-0xff*/
4855 0xb48f, 0xb490, 0xbbc7, 0xb491, 0xb492, 0xb493, 0xb494, 0xb495, /*0xf8-0xff*/
4856 /* 0x7900 */
4857 0xb496, 0xbdb8, 0xb497, 0xb498, 0xb499, 0xede2, 0xb49a, 0xb49b, /*0x00-0x07*/
4858 0xb49c, 0xb49d, 0xb49e, 0xb49f, 0xb4a0, 0xb540, 0xb541, 0xb542, /*0x08-0x0f*/
4859 0xb543, 0xb544, 0xb545, 0xede4, 0xb546, 0xb547, 0xb548, 0xb549, /*0x10-0x17*/
4860 0xb54a, 0xb54b, 0xb54c, 0xb54d, 0xb54e, 0xb54f, 0xede6, 0xb550, /*0x18-0x1f*/
4861 0xb551, 0xb552, 0xb553, 0xb554, 0xede5, 0xb555, 0xb556, 0xb557, /*0x20-0x27*/
4862 0xb558, 0xb559, 0xb55a, 0xb55b, 0xb55c, 0xb55d, 0xb55e, 0xb55f, /*0x28-0x2f*/
4863 0xb560, 0xb561, 0xb562, 0xb563, 0xede7, 0xb564, 0xb565, 0xb566, /*0x30-0x37*/
4864 0xb567, 0xb568, 0xcabe, 0xecea, 0xc0f1, 0xb569, 0xc9e7, 0xb56a, /*0x38-0x3f*/
4865 0xeceb, 0xc6ee, 0xb56b, 0xb56c, 0xb56d, 0xb56e, 0xecec, 0xb56f, /*0x40-0x47*/
4866 0xc6ed, 0xeced, 0xb570, 0xb571, 0xb572, 0xb573, 0xb574, 0xb575, /*0x48-0x4f*/
4867 0xb576, 0xb577, 0xb578, 0xecf0, 0xb579, 0xb57a, 0xd7e6, 0xecf3, /*0x50-0x57*/
4868 0xb57b, 0xb57c, 0xecf1, 0xecee, 0xecef, 0xd7a3, 0xc9f1, 0xcbee, /*0x58-0x5f*/
4869 0xecf4, 0xb57d, 0xecf2, 0xb57e, 0xb580, 0xcfe9, 0xb581, 0xecf6, /*0x60-0x67*/
4870 0xc6b1, 0xb582, 0xb583, 0xb584, 0xb585, 0xbcc0, 0xb586, 0xecf5, /*0x68-0x6f*/
```

```
4871 0xb587, 0xb588, 0xb589, 0xb58a, 0xb58b, 0xb58c, 0xb58d, 0xb5bb, /*0x70-0x77*/
4872 0xbbf6, 0xb58e, 0xecf7, 0xb58f, 0xb590, 0xb591, 0xb592, 0xb593, /*0x78-0x7f*/
4873 0xd9f7, 0xbdaf, 0xb594, 0xb595, 0xc2bb, 0xecf8, 0xb596, 0xb597, /*0x80-0x87*/
4874 0xb598, 0xb599, 0xecf9, 0xb59a, 0xb59b, 0xb59c, 0xb59d, 0xb8a3, /*0x88-0x8f*/
4875 0xb59e, 0xb59f, 0xb5a0, 0xb640, 0xb641, 0xb642, 0xb643, 0xb644, /*0x90-0x97*/
4876 0xb645, 0xb646, 0xecfa, 0xb647, 0xb648, 0xb649, 0xb64a, 0xb64b, /*0x98-0x9f*/
4877 0xb64c, 0xb64d, 0xb64e, 0xb64f, 0xb650, 0xb651, 0xb652, 0xecfb, /*0xa0-0xa7*/
4878 0xb653, 0xb654, 0xb655, 0xb656, 0xb657, 0xb658, 0xb659, 0xb65a, /*0xa8-0xaf*/
4879 0xb65b, 0xb65c, 0xb65d, 0xecfc, 0xb65e, 0xb65f, 0xb660, 0xb661, /*0xb0-0xb7*/
4880 0xb662, 0xd3ed, 0xd8ae, 0xc0eb, 0xb663, 0xc7dd, 0xbacc, 0xb664, /*0xb8-0xbf*/
4881 0xd0e3, 0xcbbd, 0xb665, 0xcdba, 0xb666, 0xb667, 0xb8d1, 0xb668, /*0xc0-0xc7*/
4882 0xb669, 0xb1fc, 0xb66a, 0xc7ef, 0xb66b, 0xd6d6, 0xb66c, 0xb66d, /*0xc8-0xcf*/
4883 0xb66e, 0xbfc6, 0xc3eb, 0xb66f, 0xb670, 0xeff5, 0xb671, 0xb672, /*0xd0-0xd7*/
4884 0xc3d8, 0xb673, 0xb674, 0xb675, 0xb676, 0xb677, 0xb678, 0xd7e2, /*0xd8-0xdf*/
4885 0xb679, 0xb67a, 0xb67b, 0xeff7, 0xb3d3, 0xb67c, 0xc7d8, 0xd1ed, /*0xe0-0xe7*/
4886 0xb67d, 0xd6c8, 0xb67e, 0xeff8, 0xb680, 0xeff6, 0xb681, 0xbbfd, /*0xe8-0xef*/
4887 0xb3c6, 0xb682, 0xb683, 0xb684, 0xb685, 0xb686, 0xb687, 0xb688, /*0xf0-0xf7*/
4888 0xbdd5, 0xb689, 0xb68a, 0xd2c6, 0xb68b, 0xbbe0, 0xb68c, 0xb68d, /*0xf8-0xff*/
4889 /* 0x7a00 */
4890 0xcfa1, 0xb68e, 0xeffc, 0xeffb, 0xb68f, 0xb690, 0xeff9, 0xb691, /*0x00-0x07*/
4891 0xb692, 0xb693, 0xb694, 0xb3cc, 0xb695, 0xc9d4, 0xcbb0, 0xb696, /*0x08-0x0f*/
4892 0xb697, 0xb698, 0xb699, 0xb69a, 0xeffe, 0xb69b, 0xb69c, 0xb0de, /*0x10-0x17*/
4893 0xb69d, 0xb69e, 0xd6c9, 0xb69f, 0xb6a0, 0xb740, 0xeffd, 0xb741, /*0x18-0x1f*/
4894 0xb3ed, 0xb742, 0xb743, 0xf6d5, 0xb744, 0xb745, 0xb746, 0xb747, /*0x20-0x27*/
4895 0xb748, 0xb749, 0xb74a, 0xb74b, 0xb74c, 0xb74d, 0xb74e, 0xb74f, /*0x28-0x2f*/
4896 0xb750, 0xb751, 0xb752, 0xccec, 0xb753, 0xb754, 0xb755, 0xf0a2, /*0x30-0x37*/
4897 0xb756, 0xf0a1, 0xb757, 0xb5be, 0xbdcda, 0xb758, 0xb759, 0xb8e5, /*0x38-0x3f*/
4898 0xb759, 0xb75a, 0xb75b, 0xb75c, 0xb75d, 0xb75e, 0xc4c2, 0xb75f, /*0x40-0x47*/
4899 0xb760, 0xb761, 0xb762, 0xb763, 0xb764, 0xb765, 0xb766, 0xb767, /*0x48-0x4f*/
4900 0xb768, 0xf0a3, 0xb769, 0xb76a, 0xb76b, 0xb76c, 0xb76d, 0xcbeb, /*0x50-0x57*/
4901 0xb76e, 0xb76f, 0xb770, 0xb771, 0xb772, 0xb773, 0xb774, 0xb775, /*0x58-0x5f*/
4902 0xb776, 0xb777, 0xb778, 0xb779, 0xb77a, 0xb77b, 0xb77c, 0xb77d, /*0x60-0x67*/
4903 0xb77e, 0xb780, 0xb781, 0xb782, 0xb783, 0xb784, 0xb785, 0xb786, /*0x68-0x6f*/
4904 0xf0a6, 0xb787, 0xb788, 0xb789, 0xd1a8, 0xb78a, 0xb78b, 0xc7ee, /*0x70-0x77*/
4905 0xf1b6, 0xf1b7, 0xbfd5, 0xb78b, 0xb78c, 0xb78d, 0xb78e, 0xb4a9, /*0x78-0x7f*/
4906 0xf1b8, 0xcdbb, 0xb78f, 0xc7d4, 0xd5ad, 0xb790, 0xf1b9, 0xb791, /*0x80-0x87*/
4907 0xf1ba, 0xb792, 0xb793, 0xb794, 0xb795, 0xc7cf, 0xb796, 0xb797, /*0x88-0x8f*/
4908 0xb798, 0xd2a4, 0xd6cf, 0xb799, 0xb79a, 0xf1bb, 0xbdd1, 0xb4b0, /*0x90-0x97*/
4909 0xb79c, 0xb79d, 0xb79e, 0xb79f, 0xb79c, 0xc7d4, 0xb79e, 0xb79f, /*0x98-0x9f*/
4910 0xf1bd, 0xb79f, 0xb7a0, 0xb840, 0xb841, 0xbffa, 0xf1bc, 0xb842, /*0xa0-0xa7*/
4911 0xf1bf, 0xb843, 0xb844, 0xb845, 0xf1be, 0xf1c0, 0xb846, 0xb847, /*0xa8-0xaf*/
4912 0xb848, 0xb849, 0xb84a, 0xf1c1, 0xb84b, 0xb84c, 0xb84d, 0xb84e, /*0xb0-0xbf*/
4913 0xb84f, 0xb850, 0xb851, 0xb852, 0xb853, 0xb854, 0xb855, 0xc1fe, /*0xb8-0xbf*/
4914 0xb856, 0xb857, 0xb858, 0xb859, 0xb85a, 0xb85b, 0xb85c, 0xb85d, /*0xc0-0xc7*/
4915 0xb85e, 0xb85f, 0xb860, 0xc1a2, 0xb861, 0xb862, 0xb863, 0xb864, /*0xc8-0xcf*/
4916 0xb865, 0xb866, 0xb867, 0xb868, 0xb869, 0xb86a, 0xcafa, 0xb86b, /*0xd0-0xdf*/
4917 0xb86c, 0xd5be, 0xb86d, 0xb86e, 0xb86f, 0xb870, 0xb871, 0xb872, /*0xe0-0xef*/
4918 0xd5c2, 0xb871, 0xb872, 0xbfa2, 0xb873, 0xcdaf, 0xf1b5, 0xb874, /*0xe8-0xef*/
4919 0xb875, 0xb876, 0xb877, 0xb878, 0xb879, 0xbddf, 0xb87a, 0xb6cb, /*0xf0-0xf7*/
4920 0xb87b, 0xb87c, 0xb87d, 0xb87e, 0xb880, 0xb881, 0xb882, 0xb883, /*0xf8-0xff*/
4921 0xb884, 0xd6f1, 0xf3c3, 0xb885, 0xb886, 0xf3c4, 0xb887, 0xb88d, /*0xf8-0xff*/
4922 /* 0x7b00 */
4923 0xb888, 0xb889, 0xb88a, 0xf3c6, 0xf3c7, 0xb88b, 0xb0ca, 0xb88c, /*0x00-0x07*/
4924 0xf3c5, 0xb88d, 0xf3c9, 0xcbf1, 0xb88e, 0xb88f, 0xb890, 0xf3cb, /*0x08-0x0f*/
4925 0xb891, 0xd0a6, 0xb892, 0xb893, 0xb1ca, 0xf3c8, 0xb894, 0xb895, /*0x10-0x17*/
4926 0xb896, 0xf3cf, 0xb897, 0xb5d1, 0xb898, 0xb899, 0xf3d7, 0xb89a, /*0x18-0x1f*/
4927 0xf3d2, 0xb89b, 0xb89c, 0xb89d, 0xf3d4, 0xf3d3, 0xb7fb, 0xb89e, /*0x20-0x27*/
4928 0xb1bf, 0xb89f, 0xf3ce, 0xf3ca, 0xb5da, 0xb8a0, 0xf3d0, 0xb940, /*0x28-0x2f*/
4929 0xb941, 0xf3d1, 0xb942, 0xf3d5, 0xb943, 0xb944, 0xb945, 0xb946, /*0x30-0x37*/
4930 0xf3cd, 0xb947, 0xb948, 0xc1fd, 0xb949, 0xf3d6, 0xb94a, 0xb94b, /*0x38-0x3f*/
4931 0xb94c, 0xb94d, 0xb94e, 0xb94f, 0xf3da, 0xb950, 0xf3cc, 0xb951, /*0x40-0x47*/
4932 0xb952, 0xb953, 0xb954, 0xb955, 0xb956, 0xb957, 0xb958, 0xb959, /*0x48-0x4f*/
4933 0xb95a, 0xb95b, 0xb95c, 0xb95d, 0xb95e, 0xb95f, 0xb960, 0xb961, /*0x50-0x57*/
4934 0xf3d8, 0xb958, 0xf3d9, 0xc9b8, 0xb959, 0xf3dd, 0xb95a, 0xb95b, /*0x58-0x5f*/
4935 0xf3de, 0xb95c, 0xf3e1, 0xb95d, 0xb95e, 0xb95f, 0xb960, 0xb961, /*0x60-0x67*/
4936 0xb962, 0xb963, 0xb964, 0xb965, 0xb966, 0xb967, 0xf3df, 0xb968, /*0x68-0x6f*/
4937 0xb969, 0xf3e3, 0xf3e2, 0xb96a, 0xb96b, 0xf3db, 0xb96c, 0xbfea, /*0x70-0x77*/
4938 0xb96d, 0xb96e, 0xb96f, 0xf3e0, 0xb96f, 0xb970, 0xc7a9, 0xb971, /*0x78-0x7f*/
4939 0xb972, 0xb973, 0xb974, 0xb975, 0xf3eb, 0xb976, 0xb977, 0xb978, /*0x80-0x87*/
4940 0xb979, 0xb97a, 0xb97b, 0xb97c, 0xb97d, 0xb97e, 0xb97f, 0xb980, /*0x88-0x8f*/
4941 0xf3e4, 0xb980, 0xb981, 0xb982, 0xb983, 0xb984, 0xcbe3, 0xcbe4, /*0x90-0x97*/
4942 0xb985, 0xb986, 0xb987, 0xb988, 0xf3ed, 0xf3e9, 0xb989, 0xb98a, /*0x98-0x9f*/
4943 0xb98b, 0xb98c, 0xf3ee, 0xb98d, 0xb98e, 0xb98f, 0xf3e5, 0xf3e6, /*0xa0-0xaf*/
4944 0xf3ea, 0xc2e1, 0xf3ec, 0xf3ef, 0xf3e8, 0xb98f, 0xb990, 0xb991, /*0xa8-0xaf*/
4945 0xb992, 0xb993, 0xf3f0, 0xb994, 0xb995, 0xb996, 0xb997, 0xb998, /*0xb0-0xbf*/
4946 0xf3e7, 0xb996, 0xb997, 0xb998, 0xb999, 0xb99a, 0xb99b, 0xb99c, /*0xc0-0xc7*/
4947 0xb99d, 0xf3f2, 0xb99e, 0xb99f, 0xb9a0, 0xb9a1, 0xd7ad, 0xc6aa, /*0xc8-0xcf*/
4948 0xba41, 0xba42, 0xba43, 0xba44, 0xf3f3, 0xba45, 0xba46, 0xba47, /*0xd0-0xdf*/
4949 0xba48, 0xf3f1, 0xba49, 0xc2a8, 0xba4a, 0xba4b, 0xba4c, 0xba4d, /*0xd8-0xdf*/
4950 0xba4e, 0xb8dd, 0xf3f5, 0xba4f, 0xba50, 0xf3f4, 0xba51, 0xba52, /*0xe0-0xef*/
4951 0xba53, 0xba54, 0xba55, 0xba56, 0xf3f6, 0xf3f7, 0xba57, 0xb9e7, /*0xe8-0xef*/
4952 0xba58, 0xba59, 0xf3f8, 0xba5a, 0xba5b, 0xba5c, 0xc0ba, 0xba5d, /*0xf0-0xf7*/
4953 0xba5e, 0xc0e9, 0xba5f, 0xba60, 0xba61, 0xba62, 0xba63, 0xc5f1, /*0xf8-0xff*/
4954 0xba64, 0xba65, 0xba66, 0xba67, 0xf3fb, 0xba68, 0xf3fa, 0xba69, /*0xf8-0xff*/
4955 /* 0x7c00 */
4956 0xba6a, 0xba6b, 0xba6c, 0xba6d, 0xba6e, 0xba6f, 0xba70, 0xb4d8, /*0x00-0x07*/
4957 0xba71, 0xba72, 0xba73, 0xf3fe, 0xf3f9, 0xba74, 0xba75, 0xf3fc, /*0x08-0x0f*/
```

```

4958 0xba76, 0xba77, 0xba78, 0xba79, 0xba7a, 0xba7b, 0xf3fd, 0xba7c, /*0x10-0x17*/
4959 0xba7d, 0xba7e, 0xba80, 0xba81, 0xba82, 0xba83, 0xba84, 0xf4a1, /*0x18-0x1f*/
4960 0xba85, 0xba86, 0xba87, 0xba88, 0xba89, 0xba8a, 0xf4a3, 0xbbc9, /*0x20-0x27*/
4961 0xba8b, 0xba8c, 0xf4a2, 0xba8d, 0xba8e, 0xba8f, 0xba90, 0xba91, /*0x28-0x2f*/
4962 0xba92, 0xba93, 0xba94, 0xba95, 0xba96, 0xba97, 0xba98, 0xba99, /*0x30-0x37*/
4963 0xf4a4, 0xba9a, 0xba9b, 0xba9c, 0xba9d, 0xba9e, 0xba9f, 0xb2be, /*0x38-0x3f*/
4964 0xf4a6, 0xf4a5, 0xbaa0, 0xbb40, 0xbb41, 0xbb42, 0xbb43, 0xbb44, /*0x40-0x47*/
4965 0xbb45, 0xbb46, 0xbb47, 0xbb48, 0xbb49, 0xbcae, 0xbb4a, 0xbb4b, /*0x48-0x4f*/
4966 0xbb4c, 0xbb4d, 0xbb4e, 0xbb4f, 0xbb50, 0xbb51, 0xbb52, 0xbb53, /*0x50-0x57*/
4967 0xbb54, 0xbb55, 0xbb56, 0xbb57, 0xbb58, 0xbb59, 0xbb5a, 0xbb5b, /*0x58-0x5f*/
4968 0xbb5c, 0xbb5d, 0xbb5e, 0xbb5f, 0xbb60, 0xbb61, 0xbb62, 0xbb63, /*0x60-0x6f*/
4969 0xbb64, 0xbb65, 0xbb66, 0xbb67, 0xbb68, 0xbb69, 0xbb6a, 0xbb6b, /*0x68-0x6f*/
4970 0xbb6c, 0xbb6d, 0xbb6e, 0xc3d7, 0xd9e1, 0xbb6f, 0xbb70, 0xbb71, /*0x70-0x77*/
4971 0xbb72, 0xbb73, 0xbb74, 0xc0e0, 0xf4cc, 0xd7d1, 0xbb75, 0xbb76, /*0x78-0x7f*/
4972 0xbb77, 0xbb78, 0xbb79, 0xbb7a, 0xbb7b, 0xbb7c, 0xbb7d, 0xbb7e, /*0x80-0x87*/
4973 0xbb80, 0xb7db, 0xbb81, 0xbb82, 0xbb83, 0xbb84, 0xbb85, 0xbb86, /*0x88-0x8f*/
4974 0xbb87, 0xf4ce, 0xc1a3, 0xbb88, 0xbb89, 0xc6c9, 0xbb8a, 0xb4d6, /*0x90-0x97*/
4975 0xc5b3, 0xbb8b, 0xbb8c, 0xbb8d, 0xf4d0, 0xf4cf, 0xf4d1, 0xc3da, /*0x98-0x9f*/
4976 0xbb8e, 0xbb8f, 0xf4d2, 0xbb90, 0xd4c1, 0xd6e0, 0xbb91, 0xbb92, /*0xa0-0xa7*/
4977 0xbb93, 0xbb94, 0xb7e0, 0xbb95, 0xbb96, 0xbb97, 0xc1b8, 0xbb98, /*0xa8-0xaf*/
4978 0xbb99, 0xc1bb, 0xf4d3, 0xbeac, 0xbb9a, 0xbb9b, 0xbb9c, 0xbb9d, /*0xb0-0xbf*/
4979 0xbb9e, 0xb4e2, 0xbb9f, 0xbba0, 0xf4d4, 0xf4d5, 0xbeab, 0xbc40, /*0xb8-0xbf*/
4980 0xbc41, 0xf4d6, 0xbc42, 0xbc43, 0xbc44, 0xf4db, 0xbc45, 0xf4d7, /*0xc0-0xc7*/
4981 0xf4da, 0xbc46, 0xbafd, 0xbc47, 0xf4d8, 0xf4d9, 0xbc48, 0xbc49, /*0xc8-0xcf*/
4982 0xbc4a, 0xbc4b, 0xbc4c, 0xbc4d, 0xbc4e, 0xb8e2, 0xcc7, 0xf4dc, /*0xd0-0xd7*/
4983 0xbc4f, 0xb2da, 0xbc50, 0xbc51, 0xc3d3, 0xbc52, 0xbc53, 0xd4e3, /*0xd8-0xdf*/
4984 0xbfb7, 0xbc54, 0xbc55, 0xbc56, 0xbc57, 0xbc58, 0xbc59, 0xbc5a, /*0xe0-0xef*/
4985 0xf4dd, 0xbc5b, 0xbc5c, 0xbc5d, 0xbc5e, 0xbc5f, 0xbc60, 0xc5b4, /*0xe8-0xef*/
4986 0xbc61, 0xbc62, 0xbc63, 0xbc64, 0xbc65, 0xbc66, 0xbc67, 0xbc68, /*0xf0-0xf7*/
4987 0xf4e9, 0xbc69, 0xbc6a, 0xcfb5, 0xbc6b, 0xbc6c, 0xbc6d, 0xbc6e, /*0xf8-0xff*/
4988 /* 0x7d00 */
4989 0xbc6f, 0xbc70, 0xbc71, 0xbc72, 0xbc73, 0xbc74, 0xbc75, 0xbc76, /*0x00-0x07*/
4990 0xbc77, 0xbc78, 0xc9c9, 0xbc79, 0xbc7a, 0xbc7b, 0xbc7c, 0xbc7d, /*0x08-0x0f*/
4991 0xbc7e, 0xbc80, 0xbc81, 0xbc82, 0xbc83, 0xbc84, 0xbc85, 0xbc86, /*0x10-0x17*/
4992 0xbc87, 0xbc88, 0xbc89, 0xbc8a, 0xbc8b, 0xbc8c, 0xbc8d, 0xbc8e, /*0x18-0x1f*/
4993 0xcdb8, 0xbc8f, 0xcbf7, 0xbc90, 0xbc91, 0xbc92, 0xbc93, 0xbd4f, /*0x20-0x27*/
4994 0xbc94, 0xbc95, 0xbc96, 0xd7cf, 0xbc97, 0xbc98, 0xbc99, 0xc0db, /*0x28-0x2f*/
4995 0xbc9a, 0xbc9b, 0xbc9c, 0xbc9d, 0xbc9e, 0xbc9f, 0xbca0, 0xbd40, /*0x30-0x37*/
4996 0xbd41, 0xbd42, 0xbd43, 0xbd44, 0xbd45, 0xbd46, 0xbd47, 0xbd48, /*0x38-0x3f*/
4997 0xbd49, 0xbd4a, 0xbd4b, 0xbd4c, 0xbd4d, 0xbd4e, 0xbd4f, 0xbd50, /*0x40-0x47*/
4998 0xbd51, 0xbd52, 0xbd53, 0xbd54, 0xbd55, 0xbd56, 0xbd57, 0xbd58, /*0x48-0x4f*/
4999 0xbd59, 0xbd5a, 0xbd5b, 0xbd5c, 0xbd5d, 0xbd5e, 0xbd5f, 0xbd60, /*0x50-0x57*/
5000 0xbd61, 0xbd62, 0xbd63, 0xbd64, 0xbd65, 0xbd66, 0xbd67, 0xbd68, /*0x58-0x5f*/
5001 0xbd69, 0xbd6a, 0xbd6b, 0xbd6c, 0xbd6d, 0xbd6e, 0xbd6f, 0xbd70, /*0x60-0x67*/
5002 0xbd71, 0xbd72, 0xbd73, 0xbd74, 0xbd75, 0xbd76, 0xd0f5, 0xbd77, /*0x68-0x6f*/
5003 0xbd78, 0xbd79, 0xbd7a, 0xbd7b, 0xbd7c, 0xbd7d, 0xbd7e, 0xf4ea, /*0x70-0x77*/
5004 0xbd80, 0xbd81, 0xbd82, 0xbd83, 0xbd84, 0xbd85, 0xbd86, 0xbd87, /*0x78-0x7f*/
5005 0xbd88, 0xbd89, 0xbd8a, 0xbd8b, 0xbd8c, 0xbd8d, 0xbd8e, 0xbd8f, /*0x80-0x87*/
5006 0xbd90, 0xbd91, 0xbd92, 0xbd93, 0xbd94, 0xbd95, 0xbd96, 0xbd97, /*0x88-0x8f*/
5007 0xbd98, 0xbd99, 0xbd9a, 0xbd9b, 0xbd9c, 0xbd9d, 0xbd9e, 0xbd9f, /*0x90-0x97*/
5008 0xbda0, 0xbe40, 0xbe41, 0xbe42, 0xbe43, 0xbe44, 0xbe45, 0xbe46, /*0x98-0x9f*/
5009 0xbe47, 0xbe48, 0xbe49, 0xbe4a, 0xbe4b, 0xbe4c, 0xf4eb, 0xbe4d, /*0xa0-0xaf*/
5010 0xbe4e, 0xbe4f, 0xbe50, 0xbe51, 0xbe52, 0xbe53, 0xf4ec, 0xbe54, /*0xa8-0xaf*/
5011 0xbe55, 0xbe56, 0xbe57, 0xbe58, 0xbe59, 0xbe5a, 0xbe5b, 0xbe5c, /*0xb0-0xbf*/
5012 0xbe5d, 0xbe5e, 0xbe5f, 0xbe60, 0xbe61, 0xbe62, 0xbe63, 0xbe64, /*0xb8-0xbf*/
5013 0xbe65, 0xbe66, 0xbe67, 0xbe68, 0xbe69, 0xbe6a, 0xbe6b, 0xbe6c, /*0xc0-0xc7*/
5014 0xbe6d, 0xbe6e, 0xbe6f, 0xbe70, 0xbe71, 0xbe72, 0xbe73, 0xbe74, /*0xc8-0xcf*/
5015 0xbe75, 0xbe76, 0xbe77, 0xbe78, 0xbe79, 0xbe7a, 0xbe7b, 0xbe7c, /*0xd0-0xd7*/
5016 0xbe7d, 0xbe7e, 0xbe80, 0xbe81, 0xbe82, 0xbe83, 0xbe84, 0xbe85, /*0xd8-0xdf*/
5017 0xbe86, 0xbe87, 0xbe88, 0xbe89, 0xbe8a, 0xbe8b, 0xbe8c, 0xbe8d, /*0xe0-0xef*/
5018 0xbe8e, 0xbe8f, 0xbe90, 0xbe91, 0xbe92, 0xbe93, 0xbe94, 0xbe95, /*0xe8-0xef*/
5019 0xbe96, 0xbe97, 0xbe98, 0xbe99, 0xbe9a, 0xbe9b, 0xbe9c, 0xbe9d, /*0xf0-0xf7*/
5020 0xbe9e, 0xbe9f, 0xbea0, 0xbf40, 0xbf41, 0xbf42, 0xbf43, 0xbf44, /*0xf8-0xff*/
5021 /* 0x7e00 */
5022 0xbf45, 0xbf46, 0xbf47, 0xbf48, 0xbf49, 0xbf4a, 0xbf4b, 0xbf4c, /*0x00-0x07*/
5023 0xbf4d, 0xbf4e, 0xbf4f, 0xbf50, 0xbf51, 0xbf52, 0xbf53, 0xbf54, /*0x08-0x0f*/
5024 0xbf55, 0xbf56, 0xbf57, 0xbf58, 0xbf59, 0xbf5a, 0xbf5b, 0xbf5c, /*0x10-0x17*/
5025 0xbf5d, 0xbf5e, 0xbf5f, 0xbf60, 0xbf61, 0xbf62, 0xbf63, 0xbf64, /*0x18-0x1f*/
5026 0xbf65, 0xbf66, 0xbf67, 0xbf68, 0xbf69, 0xbf6a, 0xbf6b, 0xbf6c, /*0x20-0x27*/
5027 0xbf6d, 0xbf6e, 0xbf6f, 0xbf70, 0xbf71, 0xbf72, 0xbf73, 0xbf74, /*0x28-0x2f*/
5028 0xbf75, 0xbf76, 0xbf77, 0xbf78, 0xbf79, 0xbf7a, 0xbf7b, 0xbf7c, /*0x30-0x37*/
5029 0xbf7d, 0xbf7e, 0xbf80, 0xf7e3, 0xbf81, 0xbf82, 0xbf83, 0xbf84, /*0x38-0x3f*/
5030 0xbf85, 0xb7b1, 0xbf86, 0xbf87, 0xbf88, 0xbf89, 0xbf8a, 0xf4ed, /*0x40-0x47*/
5031 0xbf8b, 0xbf8c, 0xbf8d, 0xbf8e, 0xbf8f, 0xbf90, 0xbf91, 0xbf92, /*0x48-0x4f*/
5032 0xbf93, 0xbf94, 0xbf95, 0xbf96, 0xbf97, 0xbf98, 0xbf99, 0xbf9a, /*0x50-0x57*/
5033 0xbf9b, 0xbf9c, 0xbf9d, 0xbf9e, 0xbf9f, 0xbfa0, 0xc040, 0xc041, /*0x58-0x5f*/
5034 0xc042, 0xc043, 0xc044, 0xc045, 0xc046, 0xc047, 0xc048, 0xc049, /*0x60-0x67*/
5035 0xc04a, 0xc04b, 0xc04c, 0xc04d, 0xc04e, 0xc04f, 0xc050, 0xc051, /*0x68-0x6f*/
5036 0xc052, 0xc053, 0xc054, 0xc055, 0xc056, 0xc057, 0xc058, 0xc059, /*0x70-0x77*/
5037 0xc05a, 0xc05b, 0xc05c, 0xc05d, 0xc05e, 0xc05f, 0xc060, 0xc061, /*0x78-0x7f*/
5038 0xc062, 0xc063, 0xd7eb, 0xc064, 0xc065, 0xc066, 0xc067, 0xc068, /*0x80-0x87*/
5039 0xc069, 0xc06a, 0xc06b, 0xc06c, 0xc06d, 0xc06e, 0xc06f, 0xc070, /*0x88-0x8f*/
5040 0xc071, 0xc072, 0xc073, 0xc074, 0xc075, 0xc076, 0xc077, 0xc078, /*0x90-0x97*/
5041 0xc079, 0xc07a, 0xc07b, 0xf4ee, 0xc07c, 0xc07d, 0xc07e, 0xe6f9, /*0x98-0x9f*/
5042 0xc0ec, 0xe6fa, 0xbaec, 0xe6fb, 0xcfc3, 0xe6fc, 0xd4bc, 0xbcb6, /*0xa0-0xaf*/
5043 0xe6fd, 0xe6fe, 0xbccd, 0xc8d2, 0xc8cb, 0xe7a1, 0xc080, 0xb4bf, /*0xa8-0xaf*/
5044 0xe7a2, 0xc9b4, 0xb8d9, 0xc4c9, 0xc081, 0xd7dd, 0xc2da, 0xb7d7, /*0xb0-0xbf*/

```

```
5045 0xd6bd, 0xcec6, 0xb7c4, 0xc082, 0xc083, 0xc5a6, 0xe7a3, 0xcfd9, /*0xb8-0xbf*/
5046 0xe7a4, 0xe7a5, 0xe7a6, 0xc1b7, 0xd7e9, 0xc9f0, 0xcfb8, 0xd6af, /*0xc0-0xc7*/
5047 0xd6d5, 0xe7a7, 0xb0ed, 0xe7a8, 0xe7a9, 0xc9dc, 0xd2ef, 0xbead, /*0xc8-0xcf*/
5048 0xe7aa, 0xb0f3, 0xc8de, 0xbde1, 0xe7ab, 0xc8c6, 0xc084, 0xe7ac, /*0xd0-0xd7*/
5049 0xbbe6, 0xb8f8, 0xd1a4, 0xe7ad, 0xc2e7, 0xbef8, 0xbdc4, 0xcdb3, /*0xd8-0xdf*/
5050 0xe7ae, 0xe7af, 0xbee, 0xd0e5, 0xc085, 0xcbe7, 0xcdd0, 0xbccc, /*0xe0-0xef*/
5051 0xe7b0, 0xbca8, 0xd0f7, 0xe7b1, 0xc086, 0xd0f8, 0xe7b2, 0xe7b3, /*0xe8-0xef*/
5052 0xb4c2, 0xe7b4, 0xe7b5, 0xc9fe, 0xceac, 0xc3e0, 0xe7b7, 0xb1c1, /*0xf0-0xff*/
5053 0xb3f1, 0xc087, 0xe7b8, 0xe7b9, 0xd7db, 0xd5c0, 0xe7ba, 0xc2cc, /*0xf8-0xff*/
5054 /* 0xf00 */
5055 0xd7ba, 0xe7bb, 0xe7bc, 0xe7bd, 0xbcea, 0xc3e5, 0xc0c2, 0xe7be, /*0x00-0x07*/
5056 0xe7bf, 0xbca9, 0xc088, 0xe7c0, 0xe7c1, 0xe7b6, 0xb6d0, 0xe7c2, /*0x08-0x0f*/
5057 0xc089, 0xe7c3, 0xe7c4, 0xbba, 0xb5de, 0xc2c6, 0xb1e0, 0xe7c5, /*0x10-0x17*/
5058 0xd4b5, 0xe7c6, 0xb8f, 0xe7c8, 0xe7c7, 0xb7ec, 0xc08a, 0xe7c9, /*0x18-0x1f*/
5059 0xb2f8, 0xe7ca, 0xe7cb, 0xe7cc, 0xe7cd, 0xe7ce, 0xe7cf, 0xe7d0, /*0x20-0x27*/
5060 0xd3a7, 0xcbf5, 0xe7d1, 0xe7d2, 0xe7d3, 0xe7d4, 0xc9c9, 0xe7d5, /*0x28-0x2f*/
5061 0xe7d6, 0xe7d7, 0xe7d8, 0xe7d9, 0xbdc9, 0xe7da, 0xf3be, 0xc08b, /*0x30-0x37*/
5062 0xb8d7, 0xc08c, 0xc08d, 0xc8b1, 0xc08e, 0xc08f, 0xc090, 0xc091, /*0x38-0x3f*/
5063 0xc092, 0xc093, 0xf3bf, 0xc094, 0xf3c0, 0xf3c1, 0xc095, 0xc096, /*0x40-0x47*/
5064 0xc097, 0xc098, 0xc099, 0xc09a, 0xc09b, 0xc09c, 0xc09d, 0xc09e, /*0x48-0x4f*/
5065 0xb9de, 0xcdf8, 0xc09f, 0xc0a0, 0xd8e8, 0xbab1, 0xc140, 0xc2de, /*0x50-0x57*/
5066 0xeeb7, 0xc141, 0xb7a3, 0xc142, 0xc143, 0xc144, 0xc145, 0xeeb9, /*0x58-0x5f*/
5067 0xc146, 0xeeb8, 0xb0d5, 0xc147, 0xc148, 0xc149, 0xc14a, 0xc14b, /*0x60-0x67*/
5068 0xeebb, 0xd5d6, 0xd7ef, 0xc14c, 0xc14d, 0xc14e, 0xd6c3, 0xc14f, /*0x68-0x6f*/
5069 0xc150, 0xeebd, 0xcaf0, 0xc151, 0xeebc, 0xc152, 0xc153, 0xc154, /*0x70-0x77*/
5070 0xc155, 0xeebe, 0xc156, 0xc157, 0xc158, 0xc159, 0xeecc, 0xc15a, /*0x78-0x7f*/
5071 0xc15b, 0xeebf, 0xc15c, 0xc15d, 0xc15e, 0xc15f, 0xc160, 0xc161, /*0x80-0x87*/
5072 0xc162, 0xc163, 0xd1f2, 0xc164, 0xc7bc, 0xc165, 0xc3c0, 0xc166, /*0x88-0x8f*/
5073 0xc167, 0xc168, 0xc169, 0xc16a, 0xb8e1, 0xc16b, 0xc16c, 0xc16d, /*0x90-0x97*/
5074 0xc16e, 0xc16f, 0xc1e7, 0xc170, 0xc171, 0xf4c6, 0xd0df, 0xf4c7, /*0x98-0x9f*/
5075 0xc172, 0xcfdb, 0xc173, 0xc174, 0xc8ba, 0xc175, 0xc176, 0xf4c8, /*0xa0-0xaf*/
5076 0xc177, 0xc178, 0xc179, 0xc17a, 0xc17b, 0xc17c, 0xc17d, 0xf4c9, /*0xa8-0xaf*/
5077 0xf4ca, 0xc17e, 0xf4cb, 0xc180, 0xc181, 0xc182, 0xc183, 0xc184, /*0xb0-0xbf*/
5078 0xd9fa, 0xb8fe, 0xc185, 0xc186, 0xe5f1, 0xd3f0, 0xc187, 0xf4e0, /*0xb8-0xbf*/
5079 0xc188, 0xcecc, 0xc189, 0xc18a, 0xc18b, 0xb3e1, 0xc18c, 0xc18d, /*0xc0-0xc7*/
5080 0xc18e, 0xc18f, 0xf1b4, 0xc190, 0xd2ee, 0xc191, 0xf4e1, 0xc192, /*0xc8-0xcf*/
5081 0xc193, 0xc194, 0xc195, 0xc196, 0xcfe8, 0xf4e2, 0xc197, 0xc198, /*0xd0-0xd7*/
5082 0xc7cc, 0xc199, 0xc19a, 0xc19b, 0xc19c, 0xc19d, 0xc19e, 0xb5d4, /*0xd8-0xdf*/
5083 0xb4e4, 0xf4e4, 0xc19f, 0xc1a0, 0xc240, 0xf4e3, 0xf4e5, 0xc241, /*0xe0-0xef*/
5084 0xc242, 0xf4e6, 0xc243, 0xc244, 0xc245, 0xc246, 0xf4e7, 0xc247, /*0xe8-0xef*/
5085 0xbab2, 0xb0bf, 0xc248, 0xf4e8, 0xc249, 0xc24a, 0xc24b, 0xc24c, /*0xf0-0xff*/
5086 0xc24d, 0xc24e, 0xc24f, 0xb7ad, 0xd2ed, 0xc250, 0xc251, 0xc252, /*0xf8-0xff*/
5087 /* 0x8000 */
5088 0xd2ab, 0xc0cf, 0xc253, 0xbfbc, 0xeba3, 0xd5df, 0xeac8, 0xc254, /*0x00-0x07*/
5089 0xc255, 0xc256, 0xc257, 0xf1f3, 0xb6f8, 0xcba3, 0xc258, 0xc259, /*0x08-0x0f*/
5090 0xc4cd, 0xc25a, 0xf1e7, 0xc25b, 0xf1e8, 0xb8fb, 0xf1e9, 0xbac4, /*0x10-0x17*/
5091 0xd4c5, 0xb0d2, 0xc25c, 0xc25d, 0xf1ea, 0xc25e, 0xc25f, 0xc260, /*0x18-0x1f*/
5092 0xf1eb, 0xc261, 0xf1ec, 0xc262, 0xc263, 0xf1ed, 0xf1ee, 0xf1ef, /*0x20-0x27*/
5093 0xf1f1, 0xf1f0, 0xc5d5, 0xc264, 0xc265, 0xc266, 0xc267, 0xc268, /*0x28-0x2f*/
5094 0xc269, 0xf1f2, 0xc26a, 0xb6fa, 0xc26b, 0xf1f4, 0xd2ae, 0xd2c7, /*0x30-0x37*/
5095 0xc26c, 0xc26e, 0xc26d, 0xb3dc, 0xc26f, 0xb5a2, 0xc26f, 0xb9a2, /*0x38-0x3f*/
5096 0xc270, 0xc271, 0xc4f4, 0xf1f5, 0xc272, 0xc273, 0xf1f6, 0xc274, /*0x40-0x47*/
5097 0xc275, 0xc276, 0xc1c4, 0xc1fb, 0xd6b0, 0xf1f7, 0xc277, 0xc278, /*0x48-0x4f*/
5098 0xc279, 0xc27a, 0xf1f8, 0xc27b, 0xc1aa, 0xc27c, 0xc27d, 0xc27e, /*0x50-0x57*/
5099 0xc6b8, 0xc280, 0xbedb, 0xc281, 0xc282, 0xc283, 0xc284, 0xc285, /*0x58-0x5f*/
5100 0xc286, 0xc287, 0xc288, 0xc289, 0xc28a, 0xc28b, 0xc28c, 0xc28d, /*0x60-0x67*/
5101 0xc28e, 0xf1f9, 0xb4cf, 0xc28f, 0xc290, 0xc291, 0xc292, 0xc293, /*0x68-0x6f*/
5102 0xc294, 0xf1fa, 0xc295, 0xc296, 0xc297, 0xc298, 0xc299, 0xc29a, /*0x70-0x77*/
5103 0xc29b, 0xc29c, 0xc29d, 0xc29e, 0xc29f, 0xc2a0, 0xc340, 0xedb2, /*0x78-0x7f*/
5104 0xedb1, 0xc341, 0xc342, 0xcbe0, 0xd2de, 0xc343, 0xc3c1, 0xd5d8, /*0x80-0x87*/
5105 0xc344, 0xc8e2, 0xc345, 0xc0df, 0xbca1, 0xc346, 0xc347, 0xc348, /*0x88-0x8f*/
5106 0xc349, 0xc34a, 0xc34b, 0xebc1, 0xc34c, 0xc34d, 0xd0a4, 0xc34e, /*0x90-0x97*/
5107 0xd6e2, 0xc34f, 0xb6c7, 0xb8d8, 0xebc0, 0xb8ce, 0xc350, 0xebbE, /*0x98-0x9f*/
5108 0xb3a6, 0xb9c9, 0xd6ab, 0xc351, 0xb7f4, 0xb7ca, 0xc352, 0xc353, /*0xa0-0xaf*/
5109 0xc354, 0xbce7, 0xb7be, 0xebc6, 0xc355, 0xebc7, 0xb0b9, 0xbfcf, /*0xa8-0xaf*/
5110 0xc356, 0xebc5, 0xd3fd, 0xc357, 0xebc8, 0xc358, 0xc359, 0xebc9, /*0xb0-0xbf*/
5111 0xc35a, 0xc35b, 0xb7ce, 0xc35c, 0xebc2, 0xebc4, 0xc9f6, 0xd6d7, /*0xb8-0xbf*/
5112 0xd5cd, 0xd0b2, 0xebcf, 0xebd8, 0xebd0, 0xc35d, 0xb5a8, 0xc35e, /*0xc0-0xc7*/
5113 0xc35f, 0xc360, 0xc361, 0xc362, 0xb1b3, 0xebd2, 0xc363, 0xc364, /*0xc8-0xcf*/
5114 0xc36a, 0xc365, 0xc366, 0xc367, 0xc368, 0xc369, 0xc5d6, 0xebd3, /*0xd0-0xd7*/
5115 0xc36a, 0xebd1, 0xc5df, 0xebce, 0xc36a, 0xebd5, 0xb0fb, 0xc36b, /*0xd8-0xdf*/
5116 0xc36c, 0xbafa, 0xc36d, 0xc36e, 0xd8b7, 0xf1e3, 0xc36f, 0xebca, /*0xe0-0xef*/
5117 0xebcb, 0xebcc, 0xebcd, 0xebd6, 0xe6c0, 0xebd9, 0xc370, 0xbfe8, /*0xe8-0xef*/
5118 0xd2c8, 0xebd7, 0xebdc, 0xb8ec, 0xebd8, 0xc371, 0xbdba, 0xc372, /*0xf0-0xff*/
5119 0xd0d8, 0xc373, 0xb0b7, 0xc374, 0xebdd, 0xc4dc, 0xc375, 0xc376, /*0xf8-0xff*/
5120 /* 0x8100 */
5121 0xc377, 0xc378, 0xd6ac, 0xc379, 0xc37a, 0xc37b, 0xb4e0, 0xc37c, /*0x00-0x07*/
5122 0xc37d, 0xc2f6, 0xbcb9, 0xc37e, 0xc380, 0xebda, 0xebdb, 0xd4e0, /*0x08-0x0f*/
5123 0xc6ea, 0xc4d4, 0xebdf, 0xc5a7, 0xd9f5, 0xc381, 0xb2b1, 0xc382, /*0x10-0x17*/
5124 0xebe4, 0xc383, 0xbdc5, 0xc384, 0xc385, 0xc386, 0xebc2, 0xc387, /*0x18-0x1f*/
5125 0xc388, 0xc389, 0xc38a, 0xc38b, 0xc38c, 0xc38d, 0xc38e, 0xc38f, /*0x20-0x27*/
5126 0xc390, 0xc391, 0xc392, 0xc393, 0xebc3, 0xc394, 0xc395, 0xb8ac, /*0x28-0x2f*/
5127 0xc396, 0xcdd1, 0xebc5, 0xc397, 0xc398, 0xc399, 0xebc1, 0xc39a, /*0x30-0x37*/
5128 0xc1b3, 0xc39b, 0xc39c, 0xc39d, 0xc39e, 0xc39f, 0xc6a2, 0xc3a0, /*0x38-0x3f*/
5129 0xc440, 0xc441, 0xc442, 0xc443, 0xc444, 0xc445, 0xc446, 0xc447, /*0x40-0x47*/
5130 0xebc6, 0xc447, 0xc0b0, 0xd2b8, 0xebc7, 0xc448, 0xc449, 0xc44a, /*0x48-0x4f*/
5131 0xb8af, 0xb8ad, 0xc44b, 0xebc8, 0xc7bb, 0xcdf3, 0xc44c, 0xc44d, /*0x50-0x57*/
```

```
5132 0xc44e, 0xebea, 0xebeb, 0xc44f, 0xc450, 0xc451, 0xc452, 0xc453, /*0x58-0x5f*/
5133 0xebed, 0xc454, 0xc455, 0xc456, 0xc457, 0xd0c8, 0xc458, 0xebf2, /*0x60-0x67*/
5134 0xc459, 0xebee, 0xc45a, 0xc45b, 0xc45c, 0xebf1, 0xc45d, 0xc45e, /*0x68-0x6f*/
5135 0xd1fc, 0xebec, 0xc45e, 0xc45f, 0xebe9, 0xc460, 0xc461, 0xc462, /*0x70-0x77*/
5136 0xc463, 0xb8b9, 0xc4f9, 0xc4e5, 0xebef, 0xebf0, 0xccda, 0xcdc8, /*0x78-0x7f*/
5137 0xb0f2, 0xc464, 0xebf6, 0xc465, 0xc466, 0xc467, 0xc468, 0xc469, /*0x80-0x87*/
5138 0xebf5, 0xc46a, 0xb2b2, 0xc46b, 0xc46c, 0xc46d, 0xc46e, 0xb8e0, /*0x88-0x8f*/
5139 0xc46f, 0xebf7, 0xc470, 0xc471, 0xc472, 0xc473, 0xc474, 0xc475, /*0x90-0x97*/
5140 0xb1ec, 0xc476, 0xc477, 0xccc5, 0xc4a4, 0xcfa5, 0xc478, 0xc479, /*0x98-0x9f*/
5141 0xc47a, 0xc47b, 0xc47c, 0xebf9, 0xc47d, 0xc47e, 0xeca2, 0xc480, /*0xa0-0xa7*/
5142 0xc5f2, 0xc481, 0xebfa, 0xc482, 0xc483, 0xc484, 0xc485, 0xc486, /*0xa8-0xaf*/
5143 0xc487, 0xc488, 0xc489, 0xc9c5, 0xc48a, 0xc48b, 0xc48c, 0xc48d, /*0xb0-0xb7*/
5144 0xc48e, 0xc48f, 0xe2df, 0xebfe, 0xc490, 0xc491, 0xc492, 0xc493, /*0xb8-0xbf*/
5145 0xcdce, 0xeca1, 0xb1bd, 0xd3b7, 0xc494, 0xc495, 0xd2dc, 0xc496, /*0xc0-0xc7*/
5146 0xc497, 0xc498, 0xebfd, 0xc499, 0xebfb, 0xc49a, 0xc49b, 0xc49c, /*0xc8-0xcf*/
5147 0xc49d, 0xc49e, 0xc49f, 0xc4a0, 0xc540, 0xc541, 0xc542, 0xc543, /*0xd0-0xd7*/
5148 0xc544, 0xc545, 0xc546, 0xc547, 0xc548, 0xc549, 0xc54a, 0xc54b, /*0xd8-0xdf*/
5149 0xc54c, 0xc54d, 0xc54e, 0xb3bc, 0xc54f, 0xc550, 0xc551, 0xeb0, /*0xe0-0xe7*/
5150 0xc552, 0xc553, 0xd7d4, 0xc554, 0xf4ab, 0xb3f4, 0xc555, 0xc556, /*0xe8-0xef*/
5151 0xc557, 0xc558, 0xc559, 0xd6c1, 0xd6c2, 0xc55a, 0xc55b, 0xc55c, /*0xf0-0xf7*/
5152 0xc55d, 0xc55e, 0xc55f, 0xd5e9, 0xbeca, 0xc560, 0xf4a7, 0xc561, /*0xf8-0xff*/
5153 /* 0x8200 */
5154 0xd2a8, 0xf4a8, 0xf4a9, 0xc562, 0xf4aa, 0xbecb, 0xd3df, 0xc563, /*0x00-0x07*/
5155 0xc564, 0xc565, 0xc566, 0xc567, 0xc9e0, 0xc9e1, 0xc568, 0xc569, /*0x08-0x0f*/
5156 0xf3c2, 0xc56a, 0xcae6, 0xc56b, 0xccf2, 0xc56c, 0xc56d, 0xc56e, /*0x10-0x17*/
5157 0xc56f, 0xc570, 0xc571, 0xe2b6, 0xcbb4, 0xc572, 0xcce8, 0xd6db, /*0x18-0x1f*/
5158 0xc573, 0xf4ad, 0xf4ae, 0xf4af, 0xc574, 0xc575, 0xc576, 0xc577, /*0x20-0x27*/
5159 0xf4b2, 0xc578, 0xbabd, 0xf4b3, 0xb0e3, 0xf4b0, 0xc579, 0xf4b1, /*0x28-0x2f*/
5160 0xbda2, 0xb2d5, 0xc57a, 0xf4b6, 0xf4b7, 0xb6e6, 0xb2b0, 0xcfcf, /*0x30-0x37*/
5161 0xf4b4, 0xb4ac, 0xc57b, 0xf4b5, 0xc57c, 0xc57d, 0xf4b8, 0xc57e, /*0x38-0x3f*/
5162 0xc580, 0xc581, 0xc582, 0xc583, 0xf4b9, 0xc584, 0xc585, 0xcda7, /*0x40-0x47*/
5163 0xc586, 0xf4ba, 0xc587, 0xf4bb, 0xc588, 0xc589, 0xc58a, 0xf4bc, /*0x48-0x4f*/
5164 0xc58b, 0xc58c, 0xc58d, 0xc58e, 0xc58f, 0xc590, 0xc591, 0xc592, /*0x50-0x57*/
5165 0xcbd2, 0xc593, 0xf4bd, 0xc594, 0xc595, 0xc596, 0xc597, 0xf4be, /*0x58-0x5f*/
5166 0xc598, 0xc599, 0xc59a, 0xc59b, 0xc59c, 0xc59d, 0xc59e, 0xc59f, /*0x60-0x67*/
5167 0xf4bf, 0xc5a0, 0xc640, 0xc641, 0xc642, 0xc643, 0xf4de, 0xc1bc, /*0x68-0x6f*/
5168 0xbce8, 0xc644, 0xc9ab, 0xd1de, 0xe5f5, 0xc645, 0xc646, 0xc647, /*0x70-0x77*/
5169 0xc648, 0xdc3, 0xd2d5, 0xc649, 0xc64a, 0xdc34, 0xb0ac, 0xdc35, /*0x78-0x7f*/
5170 0xc64b, 0xc64c, 0xbdda, 0xc64d, 0xdc39, 0xc64e, 0xc64f, 0xc650, /*0x80-0x8f*/
5171 0xd8c2, 0xc651, 0xdc37, 0xd3f3, 0xc652, 0xc9d6, 0xdc3a, 0xdc3b, /*0x88-0x8f*/
5172 0xc653, 0xdc3c, 0xc3a2, 0xc654, 0xc655, 0xc656, 0xc657, 0xdc3c, /*0x90-0x97*/
5173 0xdcc5, 0xdc3d, 0xdc3e, 0xc658, 0xc659, 0xc65a, 0xc65b, 0xdcc5, /*0x98-0x9f*/
5174 0xc65b, 0xdccd, 0xc65c, 0xc65d, 0xdcd2, 0xbde6, 0xc2ab, 0xc65e, /*0xa0-0xaf*/
5175 0xdc38, 0xdc39, 0xdcc6, 0xdc3e, 0xb7d2, 0xb0c5, 0xdcc7, 0xd0be, /*0xa8-0xaf*/
5176 0xdcc1, 0xbba8, 0xc65f, 0xb7bc, 0xdccc, 0xc660, 0xc661, 0xdccc, /*0xb0-0xbf*/
5177 0xdc3f, 0xc7db, 0xc662, 0xc663, 0xc664, 0xd1bf, 0xdccc, 0xc665, /*0xb8-0xbf*/
5178 0xc666, 0xdcca, 0xc667, 0xc668, 0xdcd0, 0xc669, 0xc66a, 0xead, /*0xc0-0xc7*/
5179 0xdcc2, 0xc66b, 0xdcc3, 0xdcc4, 0xdcc5, 0xb2d4, 0xdcd1, 0xc66d, /*0xc8-0xcf*/
5180 0xc66c, 0xd4b7, 0xdc3d, 0xdcd1, 0xc66e, 0xdce6, 0xc66d, 0xc3e7, /*0xd0-0xd7*/
5181 0xdcdc, 0xc66e, 0xc66f, 0xbf31, 0xdcd9, 0xc670, 0xb0fa, 0xb9b6, /*0xd8-0xdf*/
5182 0xdce5, 0xdcd3, 0xc671, 0xdcc4, 0xdcd6, 0xc8f4, 0xbf0, 0xc672, /*0xe0-0xe7*/
5183 0xc673, 0xc674, 0xc675, 0xc9bb, 0xc676, 0xc677, 0xc678, 0xb1bd, /*0xe8-0xef*/
5184 0xc679, 0xd3a2, 0xc67a, 0xc67b, 0xdcdc, 0xc67c, 0xc67d, 0xdcd5, /*0xf0-0xf7*/
5185 0xc67e, 0xc6bb, 0xc680, 0xdcdc, 0xc681, 0xc682, 0xc683, 0xc684, /*0xf8-0xff*/
5186 /* 0x8300 */
5187 0xc685, 0xd7c2, 0xc3af, 0xb7b6, 0xc7d1, 0xc3a9, 0xdce2, 0xdcd8, /*0x00-0x07*/
5188 0xdceb, 0xdcd4, 0xc686, 0xc687, 0xdcd1, 0xc688, 0xbea5, 0xdcd7, /*0x08-0x0f*/
5189 0xc689, 0xdce0, 0xc68a, 0xc68b, 0xdce3, 0xdce4, 0xc68c, 0xdcf8, /*0x10-0x17*/
5190 0xc68d, 0xc68e, 0xdce1, 0xdda2, 0xdce7, 0xc68f, 0xc690, 0xc691, /*0x18-0x1f*/
5191 0xc692, 0xc693, 0xc694, 0xc695, 0xc696, 0xc697, 0xc698, 0xbceb, /*0x20-0x27*/
5192 0xb4c4, 0xc699, 0xc69a, 0xc3a3, 0xb2e7, 0xdcf8, 0xc69b, 0xdcf2, /*0x28-0x2f*/
5193 0xc69c, 0xdcef, 0xc69d, 0xdcf9, 0xdcee, 0xd2f0, 0xb2e8, 0xc69e, /*0x30-0x37*/
5194 0xc8d7, 0xc8e3, 0xdcfb, 0xc69f, 0xdced, 0xc6a0, 0xc740, 0xc741, /*0x38-0x3f*/
5195 0xdcf7, 0xc742, 0xc743, 0xdcf5, 0xc744, 0xc745, 0xbea3, 0xdcf4, /*0x40-0x47*/
5196 0xc746, 0xb2dd, 0xc747, 0xc748, 0xc749, 0xc74a, 0xc74b, 0xdcf3, /*0x48-0x4f*/
5197 0xbcf6, 0xdce8, 0xbbc4, 0xc74c, 0xc0f3, 0xc74d, 0xc74e, 0xc74f, /*0x50-0x57*/
5198 0xc750, 0xc751, 0xbcd4, 0xdce9, 0xdcea, 0xc752, 0xdcf1, 0xdcf6, /*0x58-0x5f*/
5199 0xdcf9, 0xc5b4, 0xc753, 0xc8d9, 0xbbe7, 0xdcf9, 0xdcf3, 0xd3ab, /*0x60-0x67*/
5200 0xdda1, 0xdda3, 0xdda5, 0xd2f1, 0xdda4, 0xdda6, 0xdda7, 0xd2a9, /*0x68-0x6f*/
5201 0xc754, 0xc755, 0xc756, 0xc757, 0xc758, 0xc759, 0xc75a, 0xbac9, /*0x70-0x77*/
5202 0xdda9, 0xc75b, 0xc75c, 0xddb6, 0xddb1, 0xddb4, 0xc75d, 0xc75e, /*0x78-0x7f*/
5203 0xc75f, 0xc760, 0xc761, 0xc762, 0xc763, 0xddb0, 0xc6ce, 0xc764, /*0x80-0x87*/
5204 0xc765, 0xc0f2, 0xc766, 0xc767, 0xc768, 0xc769, 0xc9af, 0xc76a, /*0x88-0x8f*/
5205 0xc76b, 0xc76c, 0xdcec, 0xddae, 0xc76d, 0xc76e, 0xc76f, 0xc770, /*0x90-0x97*/
5206 0xddb7, 0xc771, 0xc772, 0xdcf0, 0xddaf, 0xc773, 0xddb8, 0xc774, /*0x98-0x9f*/
5207 0xddac, 0xc775, 0xc776, 0xc777, 0xc778, 0xc779, 0xc77a, 0xc77b, /*0xa0-0xaf*/
5208 0xddb9, 0xddb3, 0xddad, 0xddaa, 0xc77c, 0xc77d, 0xc77e, 0xc780, /*0xa8-0xaf*/
5209 0xdda8, 0xc0b3, 0xc1ab, 0xddab, 0xddab, 0xc781, 0xddb2, 0xbbf1, /*0xb0-0xbf*/
5210 0xddb5, 0xd3a8, 0xddba, 0xc782, 0xddbb, 0xc3a7, 0xc783, 0xc784, /*0xb8-0xbf*/
5211 0xddd2, 0xddbc, 0xc785, 0xc786, 0xc787, 0xddd1, 0xc788, 0xb9bd, /*0xc0-0xc7*/
5212 0xc789, 0xc78a, 0xbed5, 0xc78b, 0xbefa, 0xc78c, 0xc78d, 0xbaca, /*0xc8-0xcf*/
5213 0xc78e, 0xc78f, 0xc790, 0xc791, 0xddc, 0xc792, 0xddc5, 0xc793, /*0xd0-0xdf*/
5214 0xddbf, 0xc794, 0xc795, 0xc796, 0xb2cb, 0xddc3, 0xc797, 0xddcb, /*0xd8-0xdf*/
5215 0xb2a4, 0xddd5, 0xc798, 0xc799, 0xc79a, 0xddbe, 0xc79b, 0xc79c, /*0xe0-0xe7*/
5216 0xc79d, 0xc6d0, 0xddd0, 0xc79e, 0xc79f, 0xc7a0, 0xc840, 0xc841, /*0xe8-0xef*/
5217 0xddd4, 0xc1e2, 0xb7c6, 0xc842, 0xc843, 0xc844, 0xc845, 0xc846, /*0xf0-0xf7*/
5218 0xddce, 0xddcf, 0xc847, 0xc848, 0xc849, 0xddc4, 0xc84a, 0xc84b, /*0xf8-0xff*/
```

```

5219 /* 0x8400 */
5220 0xc84c, 0xddbd, 0xc84d, 0xddcd, 0xccd1, 0xc84e, 0xddc9, 0xc84f, /*0x00-0x07*/
5221 0xc850, 0xc851, 0xc852, 0xddc2, 0xc3c8, 0xc6bc, 0xceae, 0xddcc, /*0x08-0x0f*/
5222 0xc853, 0xddc8, 0xc854, 0xc855, 0xc856, 0xc857, 0xc858, 0xc859, /*0x10-0x17*/
5223 0xddc1, 0xc85a, 0xc85b, 0xc85c, 0xddc6, 0xc2dc, 0xc85d, 0xc85e, /*0x18-0x1f*/
5224 0xc85f, 0xc860, 0xc861, 0xc862, 0xd3a9, 0xd3aa, 0xddd3, 0xcfff, /*0x20-0x27*/
5225 0xc8f8, 0xc863, 0xc864, 0xc865, 0xc866, 0xc867, 0xc868, 0xc869, /*0x28-0x2f*/
5226 0xc86a, 0xddde, 0xc86b, 0xc86c, 0xc86d, 0xc86e, 0xc86f, 0xc870, /*0x30-0x37*/
5227 0xddc7, 0xc871, 0xc872, 0xc873, 0xddde, 0xc2e4, 0xc874, 0xc875, /*0x38-0x3f*/
5228 0xc876, 0xc877, 0xc878, 0xc879, 0xc87a, 0xc87b, 0xddde, 0xc87c, /*0x40-0x47*/
5229 0xc87d, 0xc87e, 0xc880, 0xc881, 0xc882, 0xc883, 0xc884, 0xc885, /*0x48-0x4f*/
5230 0xc886, 0xdddd, 0xc887, 0xc888, 0xc889, 0xc88a, 0xc88b, 0xd6f8, /*0x50-0x57*/
5231 0xc88c, 0xdddd, 0xdddd, 0xb8f0, 0xdddd, 0xc88d, 0xc88e, 0xc88f, /*0x58-0x5f*/
5232 0xc890, 0xc6cf, 0xc891, 0xb6ad, 0xc892, 0xc893, 0xc894, 0xc895, /*0x60-0x67*/
5233 0xc896, 0xddde, 0xc897, 0xc898, 0xbaf9, 0xd4e1, 0xddde, 0xc899, /*0x68-0x6f*/
5234 0xc89a, 0xb4d0, 0xc89b, 0xddda, 0xc89c, 0xbffb, 0xddde, 0xc89d, /*0x70-0x77*/
5235 0xdddf, 0xc89e, 0xdddd, 0xc89f, 0xc8a0, 0xc940, 0xc941, 0xc942, /*0x78-0x7f*/
5236 0xc943, 0xc944, 0xc945, 0xc946, 0xc947, 0xc948, 0xdddb, /*0x80-0x87*/
5237 0xdddc, 0xddde, 0xc949, 0xbdaf, 0xddde, 0xc94a, 0xddde, 0xc94b, /*0x88-0x8f*/
5238 0xc94c, 0xc94d, 0xc94e, 0xc94f, 0xc950, 0xc951, 0xc952, 0xdddf, /*0x90-0x97*/
5239 0xc953, 0xc3c9, 0xc954, 0xc955, 0xc956, 0xc957, 0xc958, /*0x98-0x9f*/
5240 0xc959, 0xddf2, 0xc95a, 0xc95b, 0xc95c, 0xc95d, 0xc95e, 0xc95f, /*0xa0-0xa7*/
5241 0xc960, 0xc961, 0xc962, 0xc963, 0xc964, 0xc965, 0xc966, 0xd8e1, /*0xa8-0xaf*/
5242 0xc967, 0xc968, 0xc969, 0xc96a, 0xc96b, 0xc96c, 0xc96d, /*0xb0-0xbf*/
5243 0xd5f4, 0xddf3, 0xddf0, 0xc96d, 0xc96e, 0xddec, 0xc96f, 0xddef, /*0xb8-0xbf*/
5244 0xc970, 0xddde, 0xc971, 0xc972, 0xd0ee, 0xc973, 0xc974, 0xc975, /*0xc0-0xc7*/
5245 0xc976, 0xc8d8, 0xc8d9, 0xc977, 0xc978, 0xddde, 0xc979, 0xc97a, /*0xc8-0xcf*/
5246 0xddea, 0xcbf2, 0xc97b, 0xddde, 0xc97c, 0xc97d, 0xb1cd, 0xc97e, /*0xd0-0xd7*/
5247 0xc980, 0xc981, 0xc982, 0xc983, 0xc984, 0xc0b6, 0xc985, 0xbcb6, /*0xd8-0xdf*/
5248 0xddf1, 0xc986, 0xc987, 0xdddf, 0xc988, 0xdddf, 0xc989, /*0xe0-0xef*/
5249 0xc98a, 0xc98b, 0xc98c, 0xc98d, 0xc5ee, 0xc98e, 0xc98f, 0xc990, /*0xe8-0xef*/
5250 0xddfb, 0xc991, 0xc992, 0xc993, 0xc994, 0xc995, 0xc996, 0xc997, /*0xf0-0xf7*/
5251 0xc998, 0xc999, 0xc99a, 0xc99b, 0xdea4, 0xc99c, 0xc99d, 0xdea3, /*0xf8-0xff*/
5252 /* 0x8500 */
5253 0xc99e, 0xc99f, 0xc9a0, 0xc9a1, 0xc9a2, 0xc9a3, 0xc9a4, 0xc9a5, /*0x00-0x07*/
5254 0xc9a6, 0xc9a7, 0xc9a8, 0xc9a9, 0xc9aa, 0xc9ab, 0xc9ac, 0xc9ad, /*0x08-0x0f*/
5255 0xc9ae, 0xc9af, 0xc9b0, 0xc9b1, 0xc9b2, 0xc9b3, 0xc9b4, 0xc9b5, /*0x10-0x17*/
5256 0xc9b6, 0xc9b7, 0xc9b8, 0xc9b9, 0xc9ba, 0xc9bb, 0xc9bc, 0xc9bd, /*0x18-0x1f*/
5257 0xc9be, 0xc9bf, 0xc9c0, 0xc9c1, 0xc9c2, 0xc9c3, 0xc9c4, 0xc9c5, /*0x20-0x27*/
5258 0xc9c6, 0xc9c7, 0xc9c8, 0xc9c9, 0xc9ca, 0xc9cb, 0xc9cc, 0xc9cd, /*0x28-0x2f*/
5259 0xc9ce, 0xc9cf, 0xc9d0, 0xc9d1, 0xc9d2, 0xc9d3, 0xc9d4, 0xc9d5, /*0x30-0x37*/
5260 0xc9d6, 0xc9d7, 0xc9d8, 0xc9d9, 0xc9da, 0xc9db, 0xc9dc, 0xc9dd, /*0x38-0x3f*/
5261 0xc9de, 0xc9df, 0xc9e0, 0xc9e1, 0xc9e2, 0xc9e3, 0xc9e4, 0xc9e5, /*0x40-0x47*/
5262 0xc9e6, 0xc9e7, 0xc9e8, 0xc9e9, 0xc9ea, 0xc9eb, 0xc9ec, 0xc9ed, /*0x48-0x4f*/
5263 0xc9ee, 0xc9ef, 0xc9f0, 0xc9f1, 0xc9f2, 0xc9f3, 0xc9f4, 0xc9f5, /*0x50-0x57*/
5264 0xc9f6, 0xc9f7, 0xc9f8, 0xc9f9, 0xc9fa, 0xc9fb, 0xc9fc, 0xc9fd, /*0x58-0x5f*/
5265 0xc9fe, 0xc9ff, 0xc9a0, 0xc9a1, 0xc9a2, 0xc9a3, 0xc9a4, 0xc9a5, /*0x60-0x67*/
5266 0xc9a6, 0xc9a7, 0xc9a8, 0xc9a9, 0xc9aa, 0xc9ab, 0xc9ac, 0xc9ad, /*0x68-0x6f*/
5267 0xc9ae, 0xc9af, 0xc9b0, 0xc9b1, 0xc9b2, 0xc9b3, 0xc9b4, 0xc9b5, /*0x70-0x77*/
5268 0xc9b6, 0xc9b7, 0xc9b8, 0xc9b9, 0xc9ca, 0xc9cb, 0xc9cc, 0xc9cd, /*0x78-0x7f*/
5269 0xc9ce, 0xc9cf, 0xc9d0, 0xc9d1, 0xc9d2, 0xc9d3, 0xc9d4, 0xc9d5, /*0x80-0x87*/
5270 0xc9d6, 0xc9d7, 0xc9d8, 0xc9d9, 0xc9da, 0xc9db, 0xc9dc, 0xc9dd, /*0x88-0x8f*/
5271 0xc9de, 0xc9df, 0xc9e0, 0xc9e1, 0xc9e2, 0xc9e3, 0xc9e4, 0xc9e5, /*0x90-0x97*/
5272 0xc9e6, 0xc9e7, 0xc9e8, 0xc9e9, 0xc9ea, 0xc9eb, 0xc9ec, 0xc9ed, /*0x98-0x9f*/
5273 0xc9ee, 0xc9ef, 0xc9f0, 0xc9f1, 0xc9f2, 0xc9f3, 0xc9f4, 0xc9f5, /*0xa0-0xa7*/
5274 0xc9f6, 0xc9f7, 0xc9f8, 0xc9f9, 0xc9fa, 0xc9fb, 0xc9fc, 0xc9fd, /*0xa8-0xaf*/
5275 0xc9fe, 0xc9ff, 0xc9a0, 0xc9a1, 0xc9a2, 0xc9a3, 0xc9a4, 0xc9a5, /*0xb0-0xbf*/
5276 0xc9ae, 0xc9af, 0xc9b0, 0xc9b1, 0xc9b2, 0xc9b3, 0xc9b4, 0xc9b5, /*0xb8-0xbf*/
5277 0xc9b6, 0xc9b7, 0xc9b8, 0xc9b9, 0xc9ca, 0xc9cb, 0xc9cc, 0xc9cd, /*0xc0-0xc7*/
5278 0xc9ce, 0xc9cf, 0xc9d0, 0xc9d1, 0xc9d2, 0xc9d3, 0xc9d4, 0xc9d5, /*0xc8-0xcf*/
5279 0xc9de, 0xc9df, 0xc9e0, 0xc9e1, 0xc9e2, 0xc9e3, 0xc9e4, 0xc9e5, /*0xd0-0xd7*/
5280 0xc9e6, 0xc9e7, 0xc9e8, 0xc9e9, 0xc9ea, 0xc9eb, 0xc9ec, 0xc9ed, /*0xd8-0xdf*/
5281 0xc9ee, 0xc9ef, 0xc9f0, 0xc9f1, 0xc9f2, 0xc9f3, 0xc9f4, 0xc9f5, /*0xe0-0xef*/
5282 0xc9f6, 0xc9f7, 0xc9f8, 0xc9f9, 0xc9fa, 0xc9fb, 0xc9fc, 0xc9fd, /*0xe8-0xef*/
5283 0xc9fe, 0xc9ff, 0xc9a0, 0xc9a1, 0xc9a2, 0xc9a3, 0xc9a4, 0xc9a5, /*0xf0-0xf7*/
5284 0xc9ae, 0xc9af, 0xc9b0, 0xc9b1, 0xc9b2, 0xc9b3, 0xc9b4, 0xc9b5, /*0xf8-0xff*/
5285 /* 0x8600 */
5286 0xc9ae, 0xc9af, 0xc9b0, 0xc9b1, 0xc9b2, 0xc9b3, 0xc9b4, 0xc9b5, /*0x00-0x07*/
5287 0xc9b6, 0xc9b7, 0xc9b8, 0xc9b9, 0xc9ca, 0xc9cb, 0xc9cc, 0xc9cd, /*0x08-0x0f*/
5288 0xc9ce, 0xc9cf, 0xc9d0, 0xc9d1, 0xc9d2, 0xc9d3, 0xc9d4, 0xc9d5, /*0x10-0x17*/
5289 0xc9de, 0xc9df, 0xc9e0, 0xc9e1, 0xc9e2, 0xc9e3, 0xc9e4, 0xc9e5, /*0x18-0x1f*/
5290 0xc9ee, 0xc9ef, 0xc9f0, 0xc9f1, 0xc9f2, 0xc9f3, 0xc9f4, 0xc9f5, /*0x20-0x27*/
5291 0xc9fe, 0xc9ff, 0xc9a0, 0xc9a1, 0xc9a2, 0xc9a3, 0xc9a4, 0xc9a5, /*0x28-0x2f*/
5292 0xc9ae, 0xc9af, 0xc9b0, 0xc9b1, 0xc9b2, 0xc9b3, 0xc9b4, 0xc9b5, /*0x30-0x37*/
5293 0xc9b6, 0xc9b7, 0xc9b8, 0xc9b9, 0xc9ca, 0xc9cb, 0xc9cc, 0xc9cd, /*0x38-0x3f*/
5294 0xc9ce, 0xc9cf, 0xc9d0, 0xc9d1, 0xc9d2, 0xc9d3, 0xc9d4, 0xc9d5, /*0x40-0x47*/
5295 0xc9de, 0xc9df, 0xc9e0, 0xc9e1, 0xc9e2, 0xc9e3, 0xc9e4, 0xc9e5, /*0x48-0x4f*/
5296 0xc9ee, 0xc9ef, 0xc9f0, 0xc9f1, 0xc9f2, 0xc9f3, 0xc9f4, 0xc9f5, /*0x50-0x57*/
5297 0xc9fe, 0xc9ff, 0xc9a0, 0xc9a1, 0xc9a2, 0xc9a3, 0xc9a4, 0xc9a5, /*0x58-0x5f*/
5298 0xc9ae, 0xc9af, 0xc9b0, 0xc9b1, 0xc9b2, 0xc9b3, 0xc9b4, 0xc9b5, /*0x60-0x67*/
5299 0xc9b6, 0xc9b7, 0xc9b8, 0xc9b9, 0xc9ca, 0xc9cb, 0xc9cc, 0xc9cd, /*0x68-0x6f*/
5300 0xc9ce, 0xc9cf, 0xc9d0, 0xc9d1, 0xc9d2, 0xc9d3, 0xc9d4, 0xc9d5, /*0x70-0x77*/
5301 0xc9de, 0xc9df, 0xc9e0, 0xc9e1, 0xc9e2, 0xc9e3, 0xc9e4, 0xc9e5, /*0x78-0x7f*/
5302 0xc9ee, 0xc9ef, 0xc9f0, 0xc9f1, 0xc9f2, 0xc9f3, 0xc9f4, 0xc9f5, /*0x80-0x87*/
5303 0xc9fe, 0xc9ff, 0xc9a0, 0xc9a1, 0xc9a2, 0xc9a3, 0xc9a4, 0xc9a5, /*0x88-0x8f*/
5304 0xc9ae, 0xc9af, 0xc9b0, 0xc9b1, 0xc9b2, 0xc9b3, 0xc9b4, 0xc9b5, /*0x90-0x97*/
5305 0xc9b6, 0xc9b7, 0xc9b8, 0xc9b9, 0xc9ca, 0xc9cb, 0xc9cc, 0xc9cd, /*0x98-0x9f*/

```

```
5306 0xcd5f, 0xcd60, 0xcd61, 0xf2bc, 0xd4e9, 0xcd62, 0xcd63, 0xf2bb, /*0xa0-0xa7*/
5307 0xf2b6, 0xf2bf, 0xf2bd, 0xcd64, 0xf2b9, 0xcd65, 0xcd66, 0xf2c7, /*0xa8-0xaf*/
5308 0xf2c4, 0xf2c6, 0xcd67, 0xcd68, 0xf2ca, 0xf2c2, 0xf2c0, 0xcd69, /*0xb0-0xb7*/
5309 0xcd6a, 0xcd6b, 0xf2c5, 0xcd6c, 0xcd6d, 0xcd6e, 0xcd6f, 0xcd70, /*0xb8-0xbf*/
5310 0xd6fb, 0xcd71, 0xcd72, 0xcd73, 0xf2c1, 0xcd74, 0xc7f9, 0xc9df, /*0xc0-0xc7*/
5311 0xcd75, 0xf2c8, 0xb9c6, 0xb5b0, 0xcd76, 0xcd77, 0xf2c3, 0xf2c9, /*0xc8-0xcf*/
5312 0xf2d0, 0xf2d6, 0xcd78, 0xcd79, 0xbbd7, 0xcd7a, 0xcd7b, 0xcd7c, /*0xd0-0xd7*/
5313 0xf2d5, 0xcdde, 0xcd7d, 0xd6eb, 0xcd7e, 0xcd80, 0xf2d2, 0xf2d4, /*0xd8-0xdf*/
5314 0xcd81, 0xcd82, 0xcd83, 0xcd84, 0xb8f2, 0xcd85, 0xcd86, 0xcd87, /*0xe0-0xe7*/
5315 0xcd88, 0xf2cb, 0xcd89, 0xcd8a, 0xcd8b, 0xf2ce, 0xc2f9, 0xcd8c, /*0xe8-0xef*/
5316 0xd5dd, 0xf2cc, 0xf2cd, 0xf2cf, 0xf2d3, 0xcd8d, 0xcd8e, 0xcd8f, /*0xf0-0xf7*/
5317 0xf2d9, 0xd3bc, 0xcd90, 0xcd91, 0xcd92, 0xcd93, 0xb6ea, 0xcd94, /*0xf8-0xff*/
5318 /* 0x8700 */
5319 0xcaf1, 0xcd95, 0xb7e4, 0xf2d7, 0xcd96, 0xcd97, 0xcd98, 0xf2d8, /*0x00-0x07*/
5320 0xf2da, 0xf2dd, 0xf2db, 0xcd99, 0xcd9a, 0xf2dc, 0xcd9b, 0xcd9c, /*0x08-0x0f*/
5321 0xcd9d, 0xcd9e, 0xd1d1, 0xf2d1, 0xcd9f, 0xcdc9, 0xcda0, 0xccecf, /*0x10-0x17*/
5322 0xd6a9, 0xce40, 0xf2e3, 0xce41, 0xc3db, 0xce42, 0xf2e0, 0xce43, /*0x18-0x1f*/
5323 0xce44, 0xce4f, 0xf2ec, 0xf2ed, 0xf2ee, 0xf2e1, 0xce46, 0xce47, /*0x20-0x27*/
5324 0xce48, 0xf2e8, 0xce49, 0xce4a, 0xce4b, 0xce4c, 0xf2e2, 0xce4d, /*0x28-0x2f*/
5325 0xce4e, 0xf2e7, 0xce4f, 0xce50, 0xf2e6, 0xce51, 0xce52, 0xf2e9, /*0x30-0x37*/
5326 0xce53, 0xce54, 0xce55, 0xf2df, 0xce56, 0xce57, 0xf2ea, 0xf2eb, /*0x38-0x3f*/
5327 0xce58, 0xce59, 0xce5a, 0xce5b, 0xce5c, 0xce5d, 0xce5e, 0xd3ac, /*0x40-0x47*/
5328 0xf2e5, 0xb2f5, 0xce5f, 0xce60, 0xf2f2, 0xce61, 0xd0ab, 0xce62, /*0x48-0x4f*/
5329 0xce63, 0xce64, 0xce65, 0xf2f5, 0xce66, 0xce67, 0xce68, 0xbbc8, /*0x50-0x57*/
5330 0xce69, 0xf2f9, 0xce6a, 0xce6b, 0xce6c, 0xce6d, 0xce6e, 0xce6f, /*0x58-0x5f*/
5331 0xf2f0, 0xce70, 0xce71, 0xf2f6, 0xf2f8, 0xf2fa, 0xce72, 0xce73, /*0x60-0x67*/
5332 0xce74, 0xce75, 0xce76, 0xce77, 0xce78, 0xce79, 0xf2f3, 0xce7a, /*0x68-0x6f*/
5333 0xf2f1, 0xce7b, 0xce7c, 0xce7d, 0xbafb, 0xce7e, 0xb5fb, 0xce80, /*0x70-0x77*/
5334 0xce81, 0xce82, 0xce83, 0xf2ef, 0xf2f7, 0xf2ed, 0xf2ee, 0xce84, /*0x78-0x7f*/
5335 0xce85, 0xce86, 0xf2eb, 0xf3a6, 0xce87, 0xf3a3, 0xce88, 0xce89, /*0x80-0x87*/
5336 0xf3a2, 0xce8a, 0xce8b, 0xf2f4, 0xce8c, 0xc8da, 0xce8d, 0xce8e, /*0x88-0x8f*/
5337 0xce8f, 0xce90, 0xce91, 0xf2fb, 0xce92, 0xce93, 0xce94, 0xf3a5, /*0x90-0x97*/
5338 0xce95, 0xce96, 0xce97, 0xce98, 0xce99, 0xce9a, 0xce9b, 0xc3f8, /*0x98-0x9f*/
5339 0xce9c, 0xce9d, 0xce9e, 0xce9f, 0xcea0, 0xcf40, 0xcf41, 0xcf42, /*0xa0-0xa7*/
5340 0xf2fd, 0xcf43, 0xcf44, 0xf3a7, 0xf3a9, 0xf3a4, 0xcf45, 0xf2fc, /*0xa8-0xaf*/
5341 0xcf46, 0xcf47, 0xcf48, 0xf3ab, 0xcf49, 0xf3aa, 0xcf4a, 0xcf4b, /*0xb0-0xb7*/
5342 0xcf4c, 0xcf4d, 0xc2dd, 0xcf4e, 0xcf4f, 0xf3ae, 0xcf50, 0xcf51, /*0xb8-0xbf*/
5343 0xf3b0, 0xcf52, 0xcf53, 0xcf54, 0xcf55, 0xcf56, 0xf3a1, 0xcf57, /*0xc0-0xc7*/
5344 0xcf58, 0xcf59, 0xcf5b, 0xf3ac, 0xcf5a, 0xcf5b, 0xcf5c, 0xcf5d, /*0xc8-0xcf*/
5345 0xcf5e, 0xf3af, 0xf2fe, 0xf3ad, 0xcf5f, 0xcf60, 0xcf61, 0xcf62, /*0xd0-0xd7*/
5346 0xcf63, 0xcf64, 0xcf65, 0xf3b2, 0xcf66, 0xcf67, 0xcf68, 0xcf69, /*0xd8-0xdf*/
5347 0xf3b4, 0xcf6a, 0xcf6b, 0xcf6c, 0xcf6d, 0xf3a8, 0xcf6e, 0xcf6f, /*0xe0-0xe7*/
5348 0xcf70, 0xcf71, 0xf3b3, 0xcf72, 0xcf73, 0xcf74, 0xf3b5, 0xcf75, /*0xe8-0xef*/
5349 0xcf76, 0xcf77, 0xcf78, 0xcf79, 0xcf7a, 0xcf7b, 0xcf7c, 0xcf7d, /*0xf0-0xf7*/
5350 0xcf7e, 0xd0b7, 0xcf80, 0xcf81, 0xcf82, 0xcf83, 0xf3b8, 0xcf84, /*0xf8-0xff*/
5351 /* 0x8800 */
5352 0xcf85, 0xcf86, 0xcf87, 0xd9f9, 0xcf88, 0xcf89, 0xcf8a, 0xcf8b, /*0x00-0x07*/
5353 0xcf8c, 0xcf8d, 0xcf8f, 0xcf8e, 0xcf8f, 0xcf90, 0xcf91, 0xcf92, /*0x08-0x0f*/
5354 0xcf93, 0xcf94, 0xcf95, 0xf3b7, 0xcf96, 0xc8e4, 0xf3b6, 0xcf97, /*0x10-0x17*/
5355 0xcf98, 0xcf99, 0xcf9a, 0xf3ba, 0xcf9b, 0xcf9c, 0xcf9d, 0xcf9e, /*0x18-0x1f*/
5356 0xcf9f, 0xf3bb, 0xb4c0, 0xcfa0, 0xd040, 0xd041, 0xd042, 0xd043, /*0x20-0x27*/
5357 0xd044, 0xd045, 0xd046, 0xd047, 0xd048, 0xd049, 0xd04a, 0xd04b, /*0x28-0x2f*/
5358 0xd04c, 0xd04d, 0xeec3, 0xd04e, 0xd04f, 0xd050, 0xd051, 0xd052, /*0x30-0x37*/
5359 0xd053, 0xf3bc, 0xd054, 0xd055, 0xf3bd, 0xd056, 0xd057, 0xd058, /*0x38-0x3f*/
5360 0xd1aa, 0xd059, 0xd05a, 0xd05b, 0xf4ac, 0xd0c6, 0xd05c, 0xd05d, /*0x40-0x47*/
5361 0xd05e, 0xd05f, 0xd060, 0xd061, 0xd062, 0xd1dc, 0xd062, 0xd063, /*0x48-0x4f*/
5362 0xd064, 0xd065, 0xd066, 0xd067, 0xcfce, 0xd068, 0xd069, 0xbdd6, /*0x50-0x57*/
5363 0xd06a, 0xd1c3, 0xd06b, 0xd06c, 0xd06d, 0xd06e, 0xd06f, 0xd070, /*0x58-0x5f*/
5364 0xd071, 0xbae2, 0xe1e9, 0xd2c2, 0xf1c2, 0xb2b9, 0xd072, 0xd073, /*0x60-0x67*/
5365 0xb1ed, 0xf1c3, 0xd074, 0xc9c0, 0xb3c4, 0xd075, 0xd9f2, 0xd076, /*0x68-0x6f*/
5366 0xcba5, 0xd077, 0xf1c4, 0xd078, 0xd079, 0xd07a, 0xd07b, 0xd6d4, /*0x70-0x77*/
5367 0xd07c, 0xd07d, 0xd07e, 0xd080, 0xd081, 0xf1c5, 0xf4c0, 0xf1c6, /*0x78-0x7f*/
5368 0xd082, 0xd4ac, 0xf1c7, 0xd083, 0xb0c0, 0xf4c1, 0xd084, 0xd085, /*0x80-0x87*/
5369 0xf4c2, 0xd086, 0xd087, 0xb4fc, 0xd088, 0xc5db, 0xd089, 0xd08a, /*0x88-0x8f*/
5370 0xd08b, 0xd08c, 0xcbb, 0xd08d, 0xd08e, 0xd08f, 0xd0e4, 0xd090, /*0x90-0x97*/
5371 0xd091, 0xd092, 0xd093, 0xd094, 0xcde0, 0xd095, 0xd096, 0xd097, /*0x98-0x9f*/
5372 0xd098, 0xd099, 0xf1c8, 0xd09a, 0xd09f3, 0xd09b, 0xd09c, 0xd09d, /*0xa0-0xaf*/
5373 0xd09e, 0xd09f, 0xd0a0, 0xb1bb, 0xd140, 0xcfae, 0xd141, 0xd142, /*0xa8-0xaf*/
5374 0xd143, 0xb8a4, 0xd144, 0xd145, 0xd146, 0xd147, 0xd148, 0xf1ca, /*0xb0-0xb7*/
5375 0xd149, 0xd14a, 0xd14b, 0xd14c, 0xf1cb, 0xd14d, 0xd14e, 0xd14f, /*0xb8-0xbf*/
5376 0xd150, 0xb2c3, 0xc1d1, 0xd151, 0xd152, 0xd7b0, 0xf1c9, 0xd153, /*0xc0-0xc7*/
5377 0xd154, 0xf1cc, 0xd155, 0xd156, 0xd157, 0xd158, 0xf1ce, 0xd159, /*0xc8-0xcf*/
5378 0xd15a, 0xd15b, 0xd9f6, 0xd15c, 0xd2e1, 0xd4a3, 0xd15d, 0xd15e, /*0xd0-0xd7*/
5379 0xf4c3, 0xc8b9, 0xd15f, 0xd160, 0xd161, 0xd162, 0xd163, 0xf4c4, /*0xd8-0xdf*/
5380 0xd164, 0xd165, 0xf1cd, 0xf1cf, 0xbfe3, 0xf1d0, 0xd166, 0xd167, /*0xe0-0xe7*/
5381 0xf1d4, 0xd168, 0xd169, 0xd16a, 0xd16b, 0xd16c, 0xd16d, 0xd16e, /*0xe8-0xef*/
5382 0xf1d6, 0xf1d1, 0xd16f, 0xc9d1, 0xc5e1, 0xd170, 0xd171, 0xd172, /*0xf0-0xf7*/
5383 0xc2e3, 0xb9fc, 0xd173, 0xd174, 0xf1d3, 0xd175, 0xf1d5, 0xd176, /*0xf8-0xff*/
5384 /* 0x8900 */
5385 0xd177, 0xd178, 0xb9d3, 0xd179, 0xd17a, 0xd17b, 0xd17c, 0xd17d, /*0x00-0x07*/
5386 0xd17e, 0xd180, 0xf1db, 0xd181, 0xd182, 0xd183, 0xd184, 0xd185, /*0x08-0x0f*/
5387 0xbad6, 0xd186, 0xb0fd, 0xf1da, 0xd187, 0xd188, 0xd189, 0xd18a, /*0x10-0x17*/
5388 0xd18b, 0xf1d8, 0xf1d2, 0xf1da, 0xd18c, 0xd18d, 0xd18e, 0xd18f, /*0x18-0x1f*/
5389 0xd190, 0xf1d7, 0xd191, 0xd192, 0xd193, 0xc8ec, 0xd194, 0xd195, /*0x20-0x27*/
5390 0xd196, 0xd197, 0xcdca, 0xf1dd, 0xd198, 0xd199, 0xd19a, 0xd19b, /*0x28-0x2f*/
5391 0xe5bd, 0xd19c, 0xd19d, 0xd19e, 0xf1dc, 0xd19f, 0xf1de, 0xd1a0, /*0x30-0x37*/
5392 0xd240, 0xd241, 0xd242, 0xd243, 0xd244, 0xd245, 0xd246, 0xd247, /*0x38-0x3f*/
```

```
5393 0xd248, 0xf1df, 0xd249, 0xd24a, 0xcfe5, 0xd24b, 0xd24c, 0xd24d, /*0x40-0x47*/
5394 0xd24e, 0xd24f, 0xd250, 0xd251, 0xd252, 0xd253, 0xd254, 0xd255, /*0x48-0x4f*/
5395 0xd256, 0xd257, 0xd258, 0xd259, 0xd25a, 0xd25b, 0xd25c, 0xd25d, /*0x50-0x57*/
5396 0xd25e, 0xd25f, 0xd260, 0xd261, 0xd262, 0xd263, 0xf4c5, 0xbdf3, /*0x58-0x5f*/
5397 0xd264, 0xd265, 0xd266, 0xd267, 0xd268, 0xd269, 0xf1e0, 0xd26a, /*0x60-0x67*/
5398 0xd26b, 0xd26c, 0xd26d, 0xd26e, 0xd26f, 0xd270, 0xd271, 0xd272, /*0x68-0x6f*/
5399 0xd273, 0xd274, 0xd275, 0xd276, 0xd277, 0xd278, 0xd279, 0xd27a, /*0x70-0x77*/
5400 0xd27b, 0xd27c, 0xd27d, 0xf1e1, 0xd27e, 0xd280, 0xd281, 0xcef7, /*0x78-0x7f*/
5401 0xd282, 0xd2aa, 0xd283, 0xf1fb, 0xd284, 0xd285, 0xb8b2, 0xd286, /*0x80-0x87*/
5402 0xd287, 0xd288, 0xd289, 0xd28a, 0xd28b, 0xd28c, 0xd28d, 0xd28e, /*0x88-0x8f*/
5403 0xd28f, 0xd290, 0xd291, 0xd292, 0xd293, 0xd294, 0xd295, 0xd296, /*0x90-0x97*/
5404 0xd297, 0xd298, 0xd299, 0xd29a, 0xd29b, 0xd29c, 0xd29d, 0xd29e, /*0x98-0x9f*/
5405 0xd29f, 0xd2a0, 0xd340, 0xd341, 0xd342, 0xd343, 0xd344, 0xd345, /*0xa0-0xa7*/
5406 0xd346, 0xd347, 0xd348, 0xd349, 0xd34a, 0xd34b, 0xd34c, 0xd34d, /*0xa8-0xaf*/
5407 0xd34e, 0xd34f, 0xd350, 0xd351, 0xd352, 0xd353, 0xd354, 0xd355, /*0xb0-0xb7*/
5408 0xd356, 0xd357, 0xd358, 0xd359, 0xd35a, 0xd35b, 0xd35c, 0xd35d, /*0xb8-0xbf*/
5409 0xd35e, 0xbcbf, 0xb9db, 0xd35f, 0xb9e6, 0xc3d9, 0xcad3, 0xaeae, /*0xc0-0xc7*/
5410 0xc0c0, 0xbef5, 0xaeae, 0xaeae, 0xaeae, 0xd360, 0xaeac, 0xaeae, /*0xc8-0xcf*/
5411 0xaeae, 0xaeaf, 0xbdc7, 0xd361, 0xd362, 0xd363, 0xf5fb, 0xd364, /*0xd0-0xd7*/
5412 0xd365, 0xd366, 0xf5fd, 0xd367, 0xf5fe, 0xd368, 0xf5fc, 0xd369, /*0xd8-0xdf*/
5413 0xd36a, 0xd36b, 0xd36c, 0xbde2, 0xd36d, 0xf6a1, 0xb4a5, 0xd36e, /*0xe0-0xef*/
5414 0xd36f, 0xd370, 0xd371, 0xf6a2, 0xd372, 0xd373, 0xd374, 0xf6a3, /*0xe8-0xef*/
5415 0xd375, 0xd376, 0xd377, 0xecb2, 0xd378, 0xd379, 0xd37a, 0xd37b, /*0xf0-0xff*/
5416 0xd37c, 0xd37d, 0xd37e, 0xd380, 0xd381, 0xd382, 0xd383, 0xd384, /*0xf8-0xff*/
5417 /* 0x8a00 */
5418 0xd1d4, 0xd385, 0xd386, 0xd387, 0xd388, 0xd389, 0xd38a, 0xd9ea, /*0x00-0x07*/
5419 0xd38b, 0xd38c, 0xd38d, 0xd38e, 0xd38f, 0xd390, 0xd391, 0xd392, /*0x08-0x0f*/
5420 0xd393, 0xd394, 0xd395, 0xd396, 0xd397, 0xd398, 0xd399, 0xd39a, /*0x10-0x17*/
5421 0xd39b, 0xd39c, 0xd39d, 0xd39e, 0xd39f, 0xd3a0, 0xd440, 0xd441, /*0x18-0x1f*/
5422 0xd442, 0xd443, 0xd444, 0xd445, 0xd446, 0xd447, 0xd448, 0xd449, /*0x20-0x27*/
5423 0xd44a, 0xd44b, 0xd44c, 0xd44d, 0xd44e, 0xd44f, 0xd450, 0xd451, /*0x28-0x2f*/
5424 0xd452, 0xd453, 0xd454, 0xd455, 0xd456, 0xd457, 0xd458, 0xd459, /*0x30-0x37*/
5425 0xd45a, 0xd45b, 0xd45c, 0xd45d, 0xd45e, 0xd45f, 0xf6a4, 0xd460, /*0x38-0x3f*/
5426 0xd461, 0xd462, 0xd463, 0xd464, 0xd465, 0xd466, 0xd467, 0xd468, /*0x40-0x47*/
5427 0xeeba, 0xd469, 0xd46a, 0xd46b, 0xd46c, 0xd46d, 0xd46e, 0xd46f, /*0x48-0x4f*/
5428 0xd470, 0xd471, 0xd472, 0xd473, 0xd474, 0xd475, 0xd476, 0xd477, /*0x50-0x57*/
5429 0xd478, 0xd479, 0xd47a, 0xd47b, 0xd47c, 0xd47d, 0xd47e, 0xd480, /*0x58-0x5f*/
5430 0xd481, 0xd482, 0xd483, 0xd484, 0xd485, 0xd486, 0xd487, 0xd488, /*0x60-0x67*/
5431 0xd489, 0xd48a, 0xd48b, 0xd48c, 0xd48d, 0xd48e, 0xd48f, 0xd490, /*0x68-0x6f*/
5432 0xd491, 0xd492, 0xd493, 0xd494, 0xd495, 0xd496, 0xd497, 0xd498, /*0x70-0x77*/
5433 0xd499, 0xd5b2, 0xd49a, 0xd49b, 0xd49c, 0xd49d, 0xd49e, 0xd49f, /*0x78-0x7f*/
5434 0xd4a0, 0xd540, 0xd541, 0xd542, 0xd543, 0xd544, 0xd545, 0xd546, /*0x80-0x87*/
5435 0xd547, 0xd3fe, 0xccdc, 0xd548, 0xd549, 0xd54a, 0xd54b, 0xd54c, /*0x88-0x8f*/
5436 0xd54d, 0xd54e, 0xd54f, 0xcac4, 0xd550, 0xd551, 0xd552, 0xd553, /*0x90-0x97*/
5437 0xd554, 0xd555, 0xd556, 0xd557, 0xd558, 0xd559, 0xd55a, 0xd55b, /*0x98-0x9f*/
5438 0xd55c, 0xd55d, 0xd55e, 0xd55f, 0xd560, 0xd561, 0xd562, 0xd563, /*0xa0-0xaf*/
5439 0xd564, 0xd565, 0xd566, 0xd567, 0xd568, 0xd569, 0xd56a, 0xd56b, /*0xaa-0xaf*/
5440 0xd56c, 0xd56d, 0xd56e, 0xd56f, 0xd570, 0xd571, 0xd572, 0xd573, /*0xb0-0xbf*/
5441 0xd574, 0xd575, 0xd576, 0xd577, 0xd578, 0xd579, 0xd57a, 0xd57b, /*0xb8-0xbf*/
5442 0xd57c, 0xd57d, 0xd57e, 0xd580, 0xd581, 0xd582, 0xd583, 0xd584, /*0xc0-0xc7*/
5443 0xd585, 0xd586, 0xd587, 0xd588, 0xd589, 0xd58a, 0xd58b, 0xd58c, /*0xc8-0xcf*/
5444 0xd58d, 0xd58e, 0xd58f, 0xd590, 0xd591, 0xd592, 0xd593, 0xd594, /*0xd0-0xd7*/
5445 0xd595, 0xd596, 0xd597, 0xd598, 0xd599, 0xd59a, 0xd59b, 0xd59c, /*0xd8-0xdf*/
5446 0xd59d, 0xd59e, 0xd59f, 0xd5a0, 0xd640, 0xd641, 0xd642, 0xd643, /*0xe0-0xef*/
5447 0xd644, 0xd645, 0xd646, 0xd647, 0xd648, 0xd649, 0xd64a, 0xd64b, /*0xe8-0xef*/
5448 0xd64c, 0xd64d, 0xd64e, 0xd64f, 0xd650, 0xd651, 0xd652, 0xd653, /*0xf0-0xff*/
5449 0xd654, 0xd655, 0xd656, 0xd657, 0xd658, 0xd659, 0xd65a, 0xd65b, /*0xf8-0xff*/
5450 /* 0x8b00 */
5451 0xd65c, 0xd65d, 0xd65e, 0xd65f, 0xd660, 0xd661, 0xd662, 0xe5c0, /*0x00-0x07*/
5452 0xd663, 0xd664, 0xd665, 0xd666, 0xd667, 0xd668, 0xd669, 0xd66a, /*0x08-0x0f*/
5453 0xd66b, 0xd66c, 0xd66d, 0xd66e, 0xd66f, 0xd670, 0xd671, 0xd672, /*0x10-0x17*/
5454 0xd673, 0xd674, 0xd675, 0xd676, 0xd677, 0xd678, 0xd679, 0xd67a, /*0x18-0x1f*/
5455 0xd67b, 0xd67c, 0xd67d, 0xd67e, 0xd680, 0xd681, 0xf6a5, 0xd682, /*0x20-0x27*/
5456 0xd683, 0xd684, 0xd685, 0xd686, 0xd687, 0xd688, 0xd689, 0xd68a, /*0x28-0x2f*/
5457 0xd68b, 0xd68c, 0xd68d, 0xd68e, 0xd68f, 0xd690, 0xd691, 0xd692, /*0x30-0x37*/
5458 0xd693, 0xd694, 0xd695, 0xd696, 0xd697, 0xd698, 0xd699, 0xd69a, /*0x38-0x3f*/
5459 0xd69b, 0xd69c, 0xd69d, 0xd69e, 0xd69f, 0xd6a0, 0xd740, 0xd741, /*0x40-0x4f*/
5460 0xd742, 0xd743, 0xd744, 0xd745, 0xd746, 0xd747, 0xd748, 0xd749, /*0x48-0x4f*/
5461 0xd74a, 0xd74b, 0xd74c, 0xd74d, 0xd74e, 0xd74f, 0xd750, 0xd751, /*0x50-0x57*/
5462 0xd752, 0xd753, 0xd754, 0xd755, 0xd756, 0xd757, 0xd758, 0xd759, /*0x58-0x5f*/
5463 0xd75a, 0xd75b, 0xd75c, 0xd75d, 0xd75e, 0xd75f, 0xbeaf, 0xd760, /*0x60-0x6f*/
5464 0xd761, 0xd762, 0xd763, 0xd764, 0xc6a9, 0xd765, 0xd766, 0xd767, /*0x68-0x6f*/
5465 0xd768, 0xd769, 0xd76a, 0xd76b, 0xd76c, 0xd76d, 0xd76e, 0xd76f, /*0x70-0x77*/
5466 0xd770, 0xd771, 0xd772, 0xd773, 0xd774, 0xd775, 0xd776, 0xd777, /*0x78-0x7f*/
5467 0xd778, 0xd779, 0xd77a, 0xd77b, 0xd77c, 0xd77d, 0xd77e, 0xd780, /*0x80-0x87*/
5468 0xd781, 0xd782, 0xd783, 0xd784, 0xd785, 0xd786, 0xd787, 0xd788, /*0x88-0x8f*/
5469 0xd789, 0xd78a, 0xd78b, 0xd78c, 0xd78d, 0xd78e, 0xd78f, 0xd790, /*0x90-0x97*/
5470 0xd791, 0xd792, 0xd793, 0xd794, 0xd795, 0xd796, 0xd797, 0xd798, /*0x98-0x9f*/
5471 0xdaa5, 0xbccc, 0xb6a9, 0xb8bc, 0xc8cf, 0xbca5, 0xdaa6, 0xdaa7, /*0xa0-0xaf*/
5472 0xccd6, 0xc8c3, 0xdaa8, 0xc6fd, 0xd799, 0xd1b5, 0xd2e9, 0xd1b6, /*0xaa8-0xaf*/
5473 0xbcc7, 0xd79a, 0xbdb2, 0xbbe4, 0xdaa9, 0xdaaa, 0xd1c8, 0xdaab, /*0xb0-0xbf*/
5474 0xd0ed, 0xb6ef, 0xc2db, 0xd79b, 0xcbcf, 0xb7ed, 0xc9e8, 0xb7c3, /*0xb8-0xbf*/
5475 0xbef7, 0xd6a4, 0xdaac, 0xdaac, 0xc6c0, 0xd7e7, 0xcab6, 0xd79c, /*0xc0-0xcf*/
5476 0xd5a9, 0xcdbf, 0xd5ef, 0xdaae, 0xd6df, 0xb4ca, 0xdab0, 0xdaaf, /*0xc8-0xcf*/
5477 0xd79d, 0xd2eb, 0xdab1, 0xdab2, 0xdab3, 0xcad4, 0xdab4, 0xcaab, /*0xd0-0xdf*/
5478 0xdab5, 0xdab6, 0xb3cf, 0xd6ef, 0xdab7, 0xbbb0, 0xb5ae, 0xdab8, /*0xd8-0xdf*/
5479 0xdab9, 0xb9ee, 0xd1af, 0xd2e8, 0xdaba, 0xb8c3, 0xcfea, 0xb2ef, /*0xe0-0xef*/
```



```

5480 0xdabb, 0xdabc, 0xd79e, 0xbdeb, 0xcdec, 0xd3ef, 0xdabd, 0xcef3, /*0xe8-0xef*/
5481 0xdabe, 0xd3d5, 0xbbe5, 0xdabf, 0xcbb5, 0xcbd0, 0xdac0, 0xc7eb, /*0xf0-0xf7*/
5482 0xd6ee, 0xdacl, 0xc5b5, 0xb6c1, 0xdac2, 0xb7cc, 0xbfce, 0xdac3, /*0xf8-0xff*/
5483 /* 0x8c00 */
5484 0xdac4, 0xcbad, 0xdac5, 0xb5f7, 0xdac6, 0xc1c2, 0xd7bb, 0xdac7, /*0x00-0x07*/
5485 0xccb8, 0xd79f, 0xd2ea, 0xc4b1, 0xdac8, 0xb5fd, 0xbbd1, 0xdac9, /*0x08-0x0f*/
5486 0xd0b3, 0xdaca, 0xdacb, 0xcabd, 0xdacc, 0xdacd, 0xdace, 0xb2f7, /*0x10-0x17*/
5487 0xdad1, 0xdacf, 0xd1e8, 0xdad0, 0xc3d5, 0xdad2, 0xd7a0, 0xdad3, /*0x18-0x1f*/
5488 0xdad4, 0xdad5, 0xd0bb, 0xd2a5, 0xb0f9, 0xdad6, 0xc7ab, 0xdad7, /*0x20-0x27*/
5489 0xbdf7, 0xc3a1, 0xdad8, 0xdad9, 0xc3fd, 0xccb7, 0xdada, 0xdadb, /*0x28-0x2f*/
5490 0xc0be, 0xc6d7, 0xdadc, 0xdadd, 0xc7b4, 0xdade, 0xdadf, 0xb9c8, /*0x30-0x37*/
5491 0xd840, 0xd841, 0xd842, 0xd843, 0xd844, 0xd845, 0xd846, 0xd847, /*0x38-0x3f*/
5492 0xd848, 0xbbed, 0xd849, 0xd84a, 0xd84b, 0xd84c, 0xb6b9, 0xf4f8, /*0x40-0x47*/
5493 0xd84d, 0xf4f9, 0xd84e, 0xd84f, 0xcde3, 0xd850, 0xd851, 0xd852, /*0x48-0x4f*/
5494 0xd853, 0xd854, 0xd855, 0xd856, 0xd857, 0xf5b9, 0xd858, 0xd859, /*0x50-0x57*/
5495 0xd85a, 0xd85b, 0xebe0, 0xd85c, 0xd85d, 0xd85e, 0xd85f, 0xd860, /*0x58-0x5f*/
5496 0xd861, 0xcff3, 0xbbbf, 0xd862, 0xd863, 0xd864, 0xd865, 0xd866, /*0x60-0x67*/
5497 0xd867, 0xd868, 0xbac0, 0xd4a5, 0xd869, 0xd86a, 0xd86b, 0xd86c, /*0x68-0x6f*/
5498 0xd86d, 0xd86e, 0xd86f, 0xeld9, 0xd870, 0xd871, 0xd872, 0xd873, /*0x70-0x77*/
5499 0xf5f4, 0xb1aa, 0xb2f2, 0xd874, 0xd875, 0xd876, 0xd877, 0xd878, /*0x78-0x7f*/
5500 0xd879, 0xd87a, 0xf5f5, 0xd87b, 0xd87c, 0xf5f7, 0xd87d, 0xd87e, /*0x80-0x87*/
5501 0xd880, 0xbad1, 0xf5f6, 0xd881, 0xc3b2, 0xd882, 0xd883, 0xd884, /*0x88-0x8f*/
5502 0xd885, 0xd886, 0xd887, 0xd888, 0xf5f9, 0xd889, 0xd88a, 0xd88b, /*0x90-0x97*/
5503 0xf5f8, 0xd88c, 0xd88d, 0xd88e, 0xd88f, 0xd890, 0xd891, 0xd892, /*0x98-0x9f*/
5504 0xd893, 0xd894, 0xd895, 0xd896, 0xd897, 0xd898, 0xd899, 0xd89a, /*0xa0-0xa7*/
5505 0xd89b, 0xd89c, 0xd89d, 0xd89e, 0xd89f, 0xd8a0, 0xd940, 0xd941, /*0xa8-0xaf*/
5506 0xd942, 0xd943, 0xd944, 0xd945, 0xd946, 0xd947, 0xd948, 0xd949, /*0xb0-0xb7*/
5507 0xd94a, 0xd94b, 0xd94c, 0xd94d, 0xd94e, 0xd94f, 0xd950, 0xd951, /*0xb8-0xbf*/
5508 0xd952, 0xd953, 0xd954, 0xd955, 0xd956, 0xd957, 0xd958, 0xd959, /*0xc0-0xc7*/
5509 0xd95a, 0xd95b, 0xd95c, 0xd95d, 0xd95e, 0xd95f, 0xd960, 0xd961, /*0xc8-0xcf*/
5510 0xd962, 0xd963, 0xd964, 0xd965, 0xd966, 0xd967, 0xd968, 0xd969, /*0xd0-0xd7*/
5511 0xd96a, 0xd96b, 0xd96c, 0xd96d, 0xd96e, 0xd96f, 0xd970, 0xd971, /*0xd8-0xdf*/
5512 0xd972, 0xd973, 0xd974, 0xd975, 0xd976, 0xd977, 0xd978, 0xd979, /*0xe0-0xef*/
5513 0xd97a, 0xd97b, 0xd97c, 0xd97d, 0xd97e, 0xd980, 0xd981, 0xd982, /*0xe8-0xef*/
5514 0xd983, 0xd984, 0xd985, 0xd986, 0xd987, 0xd988, 0xd989, 0xd98a, /*0xf0-0xf7*/
5515 0xd98b, 0xd98c, 0xd98d, 0xd98e, 0xd98f, 0xd990, 0xd991, 0xd992, /*0xf8-0xff*/
5516 /* 0x8d00 */
5517 0xd993, 0xd994, 0xd995, 0xd996, 0xd997, 0xd998, 0xd999, 0xd99a, /*0x00-0x07*/
5518 0xd99b, 0xd99c, 0xd99d, 0xd99e, 0xd99f, 0xd9a0, 0xda40, 0xda41, /*0x08-0x0f*/
5519 0xda42, 0xda43, 0xda44, 0xda45, 0xda46, 0xda47, 0xda48, 0xda49, /*0x10-0x17*/
5520 0xda4a, 0xda4b, 0xda4c, 0xda4d, 0xda4e, 0xb1b4, 0xd5ea, 0xb8ba, /*0x18-0x1f*/
5521 0xda4f, 0xb9b1, 0xb2c6, 0xd4f0, 0xcfd4, 0xb0dc, 0xd5cb, 0xbbbf, /*0x20-0x27*/
5522 0xd6ca, 0xb7b7, 0xccb0, 0xc6b6, 0xb1e1, 0xb9ba, 0xd6fc, 0xb9e1, /*0x28-0x2f*/
5523 0xb7a1, 0xbcf4, 0xeada, 0xeadd, 0xccf9, 0xb9f3, 0xeadc, 0xb4fb, /*0x30-0x37*/
5524 0xc3b3, 0xb7d1, 0xbad8, 0xeadd, 0xd4f4, 0xeade, 0xbcd6, 0xbbdf, /*0x38-0x3f*/
5525 0xeadf, 0xc1de, 0xc2b8, 0xd4df, 0xd7ca, 0xae0, 0xae1, 0xae4, /*0x40-0x47*/
5526 0xae2, 0xae3, 0xc9de, 0xb8b3, 0xb6c4, 0xae5, 0xcaea, 0xc9cd, /*0x48-0x4f*/
5527 0xb4cd, 0xda50, 0xda51, 0xe2d9, 0xc5e2, 0xae6, 0xc0b5, 0xda52, /*0x50-0x57*/
5528 0xd7b8, 0xae7, 0xd7ac, 0xc8fc, 0xd8d3, 0xd8cd, 0xd4de, 0xda53, /*0x58-0x5f*/
5529 0xd4f9, 0xc9c4, 0xd3ae, 0xb8d3, 0xb3e0, 0xda54, 0xc9e2, 0xf4f6, /*0x60-0x67*/
5530 0xda55, 0xda56, 0xda57, 0xbad5, 0xda58, 0xf4f7, 0xda59, 0xda5a, /*0x68-0x6f*/
5531 0xd7df, 0xda5b, 0xda5c, 0xf4f1, 0xb8b0, 0xd5d4, 0xb8cf, 0xc6f0, /*0x70-0x77*/
5532 0xda5d, 0xda5e, 0xda5f, 0xda60, 0xda61, 0xda62, 0xda63, 0xda64, /*0x78-0x7f*/
5533 0xda65, 0xb3c3, 0xda66, 0xda67, 0xf4f2, 0xb3ac, 0xda68, 0xda69, /*0x80-0x87*/
5534 0xda6a, 0xda6b, 0xd4bd, 0xc7f7, 0xda6c, 0xda6d, 0xda6e, 0xda6f, /*0x88-0x8f*/
5535 0xda70, 0xf4f4, 0xda71, 0xda72, 0xf4f3, 0xda73, 0xda74, 0xda75, /*0x90-0x97*/
5536 0xda76, 0xda77, 0xda78, 0xda79, 0xda7a, 0xda7b, 0xda7c, 0xcccb, /*0x98-0x9f*/
5537 0xda7d, 0xda7e, 0xda80, 0xc8a4, 0xda81, 0xda82, 0xda83, 0xda84, /*0xa0-0xa7*/
5538 0xda85, 0xda86, 0xda87, 0xda88, 0xda89, 0xda8a, 0xda8b, 0xda8c, /*0xa8-0xaf*/
5539 0xda8d, 0xf4f5, 0xda8e, 0xda8f, 0xc5bf, 0xf5c0, 0xda8f, 0xda90, /*0xb0-0xb7*/
5540 0xf5bb, 0xda91, 0xf5c3, 0xda92, 0xf5c2, 0xda93, 0xd6ba, 0xf5c1, /*0xb8-0xbf*/
5541 0xda94, 0xda95, 0xda96, 0xd4be, 0xf5c4, 0xda97, 0xf5cc, 0xda98, /*0xc0-0xc7*/
5542 0xda99, 0xda9a, 0xda9b, 0xb0cf, 0xb5f8, 0xda9c, 0xf5c9, 0xf5ca, /*0xc8-0xcf*/
5543 0xda9d, 0xc5dc, 0xda9e, 0xda9f, 0xdaa0, 0xdb40, 0xf5c5, 0xf5c6, /*0xd0-0xd7*/
5544 0xdb41, 0xdb42, 0xf5c7, 0xf5cb, 0xdb43, 0xbee0, 0xf5c8, 0xb8fa, /*0xd8-0xdf*/
5545 0xdb44, 0xdb45, 0xdb46, 0xf5d0, 0xf5d3, 0xdb47, 0xdb48, 0xdb49, /*0xe0-0xef*/
5546 0xbfe7, 0xdb4a, 0xb9f2, 0xf5bc, 0xf5cd, 0xdb4b, 0xdb4c, 0xc2b7, /*0xe8-0xef*/
5547 0xdb4d, 0xdb4e, 0xdb4f, 0xcdf8, 0xdb50, 0xbcf9, 0xdb51, 0xf5ce, /*0xf0-0xf7*/
5548 0xf5cf, 0xf5d1, 0xb6e5, 0xf5d2, 0xdb52, 0xf5d5, 0xdb53, 0xdb54, /*0xf8-0xff*/
5549 /* 0xe000 */
5550 0xdb55, 0xdb56, 0xdb57, 0xdb58, 0xdb59, 0xf5bd, 0xdb5a, 0xdb5b, /*0x00-0x07*/
5551 0xdb5c, 0xf5d4, 0xd3bb, 0xdb5d, 0xb3ec, 0xdb5e, 0xdb5f, 0xcca4, /*0x08-0x0f*/
5552 0xdb60, 0xdb61, 0xdb62, 0xdb63, 0xf5d6, 0xdb64, 0xdb65, 0xdb66, /*0x10-0x17*/
5553 0xdb67, 0xdb68, 0xdb69, 0xdb6a, 0xdb6b, 0xf5d7, 0xbee1, 0xf5d8, /*0x18-0x1f*/
5554 0xdb6c, 0xdb6d, 0xcdf, 0xf5db, 0xdb6e, 0xdb6f, 0xdb70, 0xdb71, /*0x20-0x27*/
5555 0xdb72, 0xb2c8, 0xd7d9, 0xdb73, 0xf5d9, 0xdb74, 0xf5da, 0xf5dc, /*0x28-0x2f*/
5556 0xdb75, 0xf5e2, 0xdb76, 0xdb77, 0xdb78, 0xf5e0, 0xdb79, 0xdb7a, /*0x30-0x37*/
5557 0xdb7b, 0xf5df, 0xf5dd, 0xdb7c, 0xdb7d, 0xf5e1, 0xdb7e, 0xdb80, /*0x38-0x3f*/
5558 0xf5de, 0xf5e4, 0xf5e5, 0xdb81, 0xcce3, 0xdb82, 0xdb83, 0xe5bf, /*0x40-0x47*/
5559 0xb5b8, 0xf5e3, 0xf5e8, 0xcca3, 0xdb84, 0xdb85, 0xdb86, 0xdb87, /*0x48-0x4f*/
5560 0xdb88, 0xf5e6, 0xf5e7, 0xdb89, 0xdb8a, 0xdb8b, 0xdb8c, 0xdb8d, /*0x50-0x57*/
5561 0xdb8e, 0xf5be, 0xdb8f, 0xdb90, 0xdb91, 0xdb92, 0xdb93, 0xdb94, /*0x58-0x5f*/
5562 0xdb95, 0xdb96, 0xdb97, 0xdb98, 0xdb99, 0xdb9a, 0xb1c4, 0xdb9b, /*0x60-0x67*/
5563 0xdb9c, 0xf5bf, 0xdb9d, 0xdb9e, 0xb5c5, 0xb2e4, 0xdb9f, 0xf5ec, /*0x68-0x6f*/
5564 0xf5e9, 0xdba0, 0xb6d7, 0xdc40, 0xf5ed, 0xdc41, 0xf5ea, 0xdc42, /*0x70-0x77*/
5565 0xdc43, 0xdc44, 0xdc45, 0xdc46, 0xf5eb, 0xdc47, 0xdc48, 0xb4da, /*0x78-0x7f*/
5566 0xdc49, 0xdc4ea, 0xdc4a, 0xdc4b, 0xdc4c, 0xf5ee, 0xdc4d, 0xb3f9, /*0x80-0x87*/

```

```

5567 0xdc4e, 0xdc4f, 0xdc50, 0xdc51, 0xdc52, 0xdc53, 0xdc54, 0xf5ef, /*0x88-0x8f*/
5568 0xf5f1, 0xdc55, 0xdc56, 0xdc57, 0xf5f0, 0xdc58, 0xdc59, 0xdc5a, /*0x90-0x97*/
5569 0xdc5b, 0xdc5c, 0xdc5d, 0xdc5e, 0xf5f2, 0xdc5f, 0xf5f3, 0xdc60, /*0x98-0x9f*/
5570 0xdc61, 0xdc62, 0xdc63, 0xdc64, 0xdc65, 0xdc66, 0xdc67, 0xdc68, /*0xa0-0xa7*/
5571 0xdc69, 0xdc6a, 0xdc6b, 0xc9ed, 0xb9aa, 0xdc6c, 0xdc6d, 0xc7fb, /*0xa8-0xaf*/
5572 0xdc6e, 0xdc6f, 0xb6e3, 0xdc70, 0xdc71, 0xdc72, 0xdc73, 0xdc74, /*0xb0-0xb7*/
5573 0xdc75, 0xdc76, 0xcc9, 0xdc77, 0xdc78, 0xdc79, 0xdc7a, 0xdc7b, /*0xb8-0xbf*/
5574 0xdc7c, 0xdc7d, 0xdc7e, 0xdc80, 0xdc81, 0xdc82, 0xdc83, 0xdc84, /*0xc0-0xc7*/
5575 0xdc85, 0xdc86, 0xdc87, 0xdc88, 0xdc89, 0xdc8a, 0xea6, 0xdc8b, /*0xc8-0xcf*/
5576 0xdc8c, 0xdc8d, 0xdc8e, 0xdc8f, 0xdc90, 0xdc91, 0xdc92, 0xdc93, /*0xd0-0xd7*/
5577 0xdc94, 0xdc95, 0xdc96, 0xdc97, 0xdc98, 0xdc99, 0xdc9a, 0xdc9b, /*0xd8-0xdf*/
5578 0xdc9c, 0xdc9d, 0xc9e, 0xdc9f, 0xdca0, 0xdd40, 0xdd41, 0xdd42, /*0xe0-0xef*/
5579 0xdd43, 0xdd44, 0xdd45, 0xdd46, 0xdd47, 0xdd48, 0xdd49, 0xdd4a, /*0xe8-0xef*/
5580 0xdd4b, 0xdd4c, 0xdd4d, 0xdd4e, 0xdd4f, 0xdd50, 0xdd51, 0xdd52, /*0xf0-0xf7*/
5581 0xdd53, 0xdd54, 0xdd55, 0xdd56, 0xdd57, 0xdd58, 0xdd59, 0xdd5a, /*0xf8-0xff*/
5582 /* 0x8f00 */
5583 0xdd5b, 0xdd5c, 0xdd5d, 0xdd5e, 0xdd5f, 0xdd60, 0xdd61, 0xdd62, /*0x00-0x07*/
5584 0xdd63, 0xdd64, 0xdd65, 0xdd66, 0xdd67, 0xdd68, 0xdd69, 0xdd6a, /*0x08-0x0f*/
5585 0xdd6b, 0xdd6c, 0xdd6d, 0xdd6e, 0xdd6f, 0xdd70, 0xdd71, 0xdd72, /*0x10-0x17*/
5586 0xdd73, 0xdd74, 0xdd75, 0xdd76, 0xdd77, 0xdd78, 0xdd79, 0xdd7a, /*0x18-0x1f*/
5587 0xdd7b, 0xdd7c, 0xdd7d, 0xdd7e, 0xdd80, 0xdd81, 0xdd82, 0xdd83, /*0x20-0x27*/
5588 0xdd84, 0xdd85, 0xdd86, 0xdd87, 0xdd88, 0xdd89, 0xdd8a, 0xdd8b, /*0x28-0x2f*/
5589 0xdd8c, 0xdd8d, 0xdd8e, 0xdd8f, 0xdd90, 0xdd91, 0xdd92, 0xdd93, /*0x30-0x37*/
5590 0xdd94, 0xdd95, 0xdd96, 0xdd97, 0xdd98, 0xdd99, 0xdd9a, 0xdd9b, /*0x38-0x3f*/
5591 0xdd9c, 0xdd9d, 0xdd9e, 0xdd9f, 0xdda0, 0xde40, 0xde41, 0xde42, /*0x40-0x47*/
5592 0xde43, 0xde44, 0xde45, 0xde46, 0xde47, 0xde48, 0xde49, 0xde4a, /*0x48-0x4f*/
5593 0xde4b, 0xde4c, 0xde4d, 0xde4e, 0xde4f, 0xde50, 0xde51, 0xde52, /*0x50-0x57*/
5594 0xde53, 0xde54, 0xde55, 0xde56, 0xde57, 0xde58, 0xde59, 0xde5a, /*0x58-0x5f*/
5595 0xde5b, 0xde5c, 0xde5d, 0xde5e, 0xde5f, 0xde60, 0xb3b5, 0xd4fe, /*0x60-0x67*/
5596 0xb9ec, 0xd0f9, 0xde61, 0xe9ed, 0xd7aa, 0xe9ee, 0xc2d6, 0xc8ed, /*0x68-0x6f*/
5597 0xbae4, 0xe9ef, 0xe9f0, 0xe9f1, 0xd6e1, 0xe9f2, 0xe9f3, 0xe9f5, /*0x70-0x77*/
5598 0xe9f4, 0xe9f6, 0xe9f7, 0xc7e1, 0xe9f8, 0xd4d8, 0xe9f9, 0xbdce, /*0x78-0x7f*/
5599 0xde62, 0xe9fa, 0xe9fb, 0xbdcf, 0xe9fc, 0xb8a8, 0xc1be, 0xe9fd, /*0x80-0x87*/
5600 0xb1b2, 0xbbd4, 0xb9f5, 0xe9fe, 0xde63, 0xea1, 0xea2, 0xea3, /*0x88-0x8f*/
5601 0xb7f8, 0xbcad, 0xde64, 0xcae4, 0xe0ce, 0xd4af, 0xcfb, 0xd5b7, /*0x90-0x97*/
5602 0xea4, 0xd5de, 0xea5, 0xd0c1, 0xb9bc, 0xde65, 0xb4c7, 0xb1d9, /*0x98-0x9f*/
5603 0xde66, 0xde67, 0xde68, 0xc0b1, 0xde69, 0xde6a, 0xde6b, 0xde6c, /*0xa0-0xa7*/
5604 0xb1e6, 0xb1e7, 0xde6d, 0xb1e8, 0xde6e, 0xde6f, 0xde70, 0xde71, /*0xa8-0xaf*/
5605 0xc3bd, 0xc8e8, 0xde72, 0xde73, 0xde74, 0xde75, 0xe5c1, 0xde76, /*0xb0-0xb7*/
5606 0xde77, 0xb1df, 0xde78, 0xde79, 0xde7a, 0xc1c9, 0xb4ef, 0xde7b, /*0xb8-0xbf*/
5607 0xde7c, 0xc7a8, 0xd3d8, 0xde7d, 0xc6f9, 0xd1b8, 0xde7e, 0xb9fd, /*0xc0-0xc7*/
5608 0xc2f5, 0xcde8, 0xde81, 0xde82, 0xde83, 0xde84, 0xd3ad, 0xde85, /*0xc8-0xcf*/
5609 0xd4cb, 0xbdfc, 0xde86, 0xe5c2, 0xb7b5, 0xe5c3, 0xde87, 0xde88, /*0xd0-0xd7*/
5610 0xbbb9, 0xd5e2, 0xde89, 0xbdf8, 0xd4b6, 0xcea5, 0xc1ac, 0xb3d9, /*0xd8-0xdf*/
5611 0xde8a, 0xde8b, 0xccf6, 0xde8c, 0xe5c6, 0xe5c4, 0xe5c8, 0xde8d, /*0xe0-0xef*/
5612 0xe5ca, 0xe5c7, 0xb5cf, 0xc6c8, 0xde8e, 0xb5fc, 0xe5c5, 0xde8f, /*0xf0-0xf7*/
5613 0xcaf6, 0xde90, 0xde91, 0xe5c9, 0xde92, 0xde93, 0xde94, 0xc3d4, /*0xf8-0xff*/
5614 0xb1c5, 0xbca3, 0xde95, 0xde96, 0xde97, 0xd7b7, 0xde98, 0xde99,
5615 /* 0x9000 */
5616 0xcdcb, 0xcbcd, 0xcaca, 0xccd3, 0xe5cc, 0xe5cb, 0xc4e6, 0xde9a, /*0x00-0x07*/
5617 0xd9b, 0xd1a1, 0xd1b7, 0xe5cd, 0xde9c, 0xe5d0, 0xde9d, 0xcdb8, /*0x08-0x0f*/
5618 0xd6f0, 0xe5cf, 0xb5dd, 0xde9e, 0xcdb, 0xde9f, 0xe5d1, 0xb6ba, /*0x10-0x17*/
5619 0xdea0, 0xdf40, 0xcda8, 0xb9e4, 0xdf41, 0xcac5, 0xb3d1, 0xcbd9, /*0x18-0x1f*/
5620 0xd4ec, 0xe5d2, 0xb7ea, 0xdf42, 0xdf43, 0xdf44, 0xe5ce, 0xdf45, /*0x20-0x27*/
5621 0xdf46, 0xdf47, 0xdf48, 0xdf49, 0xdf4a, 0xe5d5, 0xb4fe, 0xe5d6, /*0x28-0x2f*/
5622 0xdf4b, 0xdf4c, 0xdf4d, 0xdf4e, 0xdf4f, 0xe5d3, 0xe5d4, 0xdf50, /*0x30-0x37*/
5623 0xd2dd, 0xdf51, 0xdf52, 0xc2df, 0xb1c6, 0xdf53, 0xc3e2, 0xdf54, /*0x38-0x3f*/
5624 0xdf55, 0xb6dd, 0xcbec, 0xdf56, 0xe5d7, 0xdf57, 0xdf58, 0xd3f6, /*0x40-0x47*/
5625 0xdf59, 0xdf5a, 0xdf5b, 0xdf5c, 0xdf5d, 0xb1e9, 0xdf5e, 0xb6f4, /*0x48-0x4f*/
5626 0xe5da, 0xe5d8, 0xe5d9, 0xb5c0, 0xdf5f, 0xdf60, 0xdf61, 0xd2c5, /*0x50-0x57*/
5627 0xe5dc, 0xdf62, 0xdf63, 0xe5de, 0xdf64, 0xdf65, 0xdf66, 0xdf67, /*0x58-0x5f*/
5628 0xdf68, 0xdf69, 0xe5dd, 0xc7b2, 0xdf6a, 0xd2a3, 0xdf6b, 0xdf6c, /*0x60-0x67*/
5629 0xe5db, 0xdf6d, 0xdf6e, 0xdf6f, 0xdf70, 0xd4e2, 0xd5da, 0xdf71, /*0x68-0x6f*/
5630 0xdf72, 0xdf73, 0xdf74, 0xdf75, 0xe5e0, 0xdf71, 0xdf76, 0xdf77, /*0x70-0x77*/
5631 0xdf78, 0xdf79, 0xdf7a, 0xdf7b, 0xdf7c, 0xe5e1, 0xdf7d, 0xb1dc, /*0x78-0x7f*/
5632 0xd1fb, 0xdf7e, 0xe5e2, 0xe5e4, 0xdf80, 0xdf81, 0xdf82, 0xdf83, /*0x80-0x87*/
5633 0xe5e3, 0xdf84, 0xdf85, 0xe5e5, 0xdf86, 0xdf87, 0xdf88, 0xdf89, /*0x88-0x8f*/
5634 0xdf8a, 0xd2d8, 0xdf8b, 0xb5cb, 0xdf8c, 0xe7df, 0xdf8d, 0xdaf5, /*0x90-0x97*/
5635 0xdf8e, 0xdaf8, 0xdf8f, 0xdaf6, 0xdf90, 0xdaf7, 0xdf91, 0xdf92, /*0x98-0x9f*/
5636 0xdf93, 0xdafa, 0xd0cf, 0xc4c7, 0xdf94, 0xdf95, 0xb0ee, 0xdf96, /*0xa0-0xa7*/
5637 0xdf97, 0xdf98, 0xd0b0, 0xdf99, 0xdaf9, 0xdf9a, 0xd3ca, 0xbaaa, /*0xa8-0xaf*/
5638 0xdba2, 0xc7f1, 0xdf9b, 0xdafc, 0xdafb, 0xc9db, 0xdafd, 0xdf9c, /*0xb0-0xb7*/
5639 0xdba1, 0xd7de, 0xdafe, 0xc1da, 0xdf9d, 0xdf9e, 0xdba5, 0xdf9f, /*0xb8-0xbf*/
5640 0xdfa0, 0xd3f4, 0xe040, 0xdba7, 0xdba4, 0xe042, 0xdba8, /*0xc0-0xc7*/
5641 0xe043, 0xe044, 0xbdb, 0xe045, 0xe046, 0xe047, 0xc0c9, 0xdba3, /*0xc8-0xcf*/
5642 0xdba6, 0xd6a3, 0xe048, 0xdba9, 0xe049, 0xe04a, 0xe04b, 0xdba, /*0xd0-0xd7*/
5643 0xe04c, 0xe04d, 0xe04e, 0xdbae, 0xdbac, 0xbac2, 0xe04f, 0xe050, /*0xd8-0xdf*/
5644 0xe051, 0xbfa4, 0xdbab, 0xe052, 0xe053, 0xe054, 0xdbaa, 0xd4c7, /*0xe0-0xef*/
5645 0xb2bf, 0xe055, 0xe056, 0xdbaf, 0xe057, 0xb9f9, 0xe058, 0xdbb0, /*0xf0-0xf7*/
5646 0xe059, 0xe05a, 0xe05b, 0xe05c, 0xe05d, 0xe05e, 0xe05f, /*0xf8-0xff*/
5647 0xb5a6, 0xe060, 0xe061, 0xe062, 0xe063, 0xb6bc, 0xdbb1, 0xe064,
5648 /* 0x9100 */
5649 0xe065, 0xe066, 0xb6f5, 0xe067, 0xdbb2, 0xe068, 0xe069, 0xe06a, /*0x00-0x07*/
5650 0xe06b, 0xe06c, 0xe06d, 0xe06e, 0xe06f, 0xe070, 0xe071, 0xe072, /*0x08-0x0f*/
5651 0xe073, 0xe074, 0xe075, 0xe076, 0xe077, 0xe078, 0xe079, 0xe07a, /*0x10-0x17*/
5652 0xe07b, 0xb1c9, 0xe07c, 0xe07d, 0xe07e, 0xe080, 0xdbb4, 0xe081, /*0x18-0x1f*/
5653 0xe082, 0xe083, 0xdbb3, 0xdbb5, 0xe084, 0xe085, 0xe086, 0xe087, /*0x20-0x27*/

```

```
5654 0xe088, 0xe089, 0xe08a, 0xe08b, 0xe08c, 0xe08d, 0xe08e, 0xdbb7, /*0x28-0x2f*/
5655 0xe08f, 0xdbb6, 0xe090, 0xe091, 0xe092, 0xe093, 0xe094, 0xe095, /*0x30-0x37*/
5656 0xe096, 0xdbb8, 0xe097, 0xe098, 0xe099, 0xe09a, 0xe09b, 0xe09c, /*0x38-0x3f*/
5657 0xe09d, 0xe09e, 0xe09f, 0xdbb9, 0xe0a0, 0xe140, 0xdbba, 0xe141, /*0x40-0x47*/
5658 0xe142, 0xd3cf, 0xf4fa, 0xc7f5, 0xd7c3, 0xc5e4, 0xf4fc, 0xf4fd, /*0x48-0x4f*/
5659 0xf4fb, 0xe143, 0xbec6, 0xe144, 0xe145, 0xe146, 0xe147, 0xd0ef, /*0x50-0x57*/
5660 0xe148, 0xe149, 0xb7d3, 0xe14a, 0xe14b, 0xd4cd, 0xccaa, 0xe14c, /*0x58-0x5f*/
5661 0xe14d, 0xf5a2, 0xf5a1, 0xbaa8, 0xf4fe, 0xcbd6, 0xe14e, 0xe14f, /*0x60-0x67*/
5662 0xe150, 0xf5a4, 0xc0d2, 0xe151, 0xb3ea, 0xe152, 0xcdaa, 0xf5a5, /*0x68-0x6f*/
5663 0xf5a3, 0xbdb4, 0xf5a8, 0xe153, 0xf5a9, 0xbdcd, 0xc3b8, 0xbfe1, /*0x70-0x77*/
5664 0xcbel, 0xf5aa, 0xe154, 0xe155, 0xe156, 0xf5a6, 0xf5a7, 0xc4f0, /*0x78-0x7f*/
5665 0xe157, 0xe158, 0xe159, 0xe15a, 0xe15b, 0xf5ac, 0xe15c, 0xb4bc, /*0x80-0x87*/
5666 0xe15d, 0xd7ed, 0xe15e, 0xb4d7, 0xf5ab, 0xf5ae, 0xe15f, 0xe160, /*0x88-0x8f*/
5667 0xf5ad, 0xf5af, 0xd0d1, 0xe161, 0xe162, 0xe163, 0xe164, 0xe165, /*0x90-0x97*/
5668 0xe166, 0xe167, 0xc3d1, 0xc8a9, 0xe168, 0xe169, 0xe16a, 0xe16b, /*0x98-0x9f*/
5669 0xe16c, 0xe16d, 0xf5b0, 0xf5b1, 0xe16e, 0xe16f, 0xe170, 0xe171, /*0xa0-0xa7*/
5670 0xe172, 0xe173, 0xf5b2, 0xe174, 0xe175, 0xf5b3, 0xf5b4, 0xf5b5, /*0xa8-0xaf*/
5671 0xe176, 0xe177, 0xe178, 0xe179, 0xf5b7, 0xf5b6, 0xe17a, 0xe17b, /*0xb0-0xb7*/
5672 0xe17c, 0xe17d, 0xf5b8, 0xe17e, 0xe180, 0xe181, 0xe182, 0xe183, /*0xb8-0xbf*/
5673 0xe184, 0xe185, 0xe186, 0xe187, 0xe188, 0xe189, 0xe18a, 0xb2c9, /*0xc0-0xc7*/
5674 0xe18b, 0xd3d4, 0xcacd, 0xe18c, 0xc0ef, 0xd6d8, 0xd2b0, 0xc1bf, /*0xc8-0xcf*/
5675 0xe18d, 0xbdff, 0xe18e, 0xe18f, 0xe190, 0xe191, 0xe192, 0xe193, /*0xd0-0xd7*/
5676 0xe194, 0xe195, 0xe196, 0xe197, 0xb8aa, 0xe198, 0xe199, 0xe19a, /*0xd8-0xdf*/
5677 0xe19b, 0xe19c, 0xe19d, 0xe19e, 0xe19f, 0xe1a0, 0xe240, 0xe241, /*0xe0-0xef*/
5678 0xe242, 0xe243, 0xe244, 0xe245, 0xe246, 0xe247, 0xe248, 0xe249, /*0xf0-0xff*/
5679 0xe24a, 0xe24b, 0xe24c, 0xe24d, 0xe24e, 0xe24f, 0xe250, 0xe251,
5680 0xe252, 0xe253, 0xe254, 0xe255, 0xe256, 0xe257, 0xe258, 0xe259,
5681 /* 0x9200 */
5682 0xe25a, 0xe25b, 0xe25c, 0xe25d, 0xe25e, 0xe25f, 0xe260, 0xe261, /*0x00-0x07*/
5683 0xe262, 0xe263, 0xe264, 0xe265, 0xe266, 0xe267, 0xe268, 0xe269, /*0x08-0x0f*/
5684 0xe26a, 0xe26b, 0xe26c, 0xe26d, 0xe26e, 0xe26f, 0xe270, 0xe271, /*0x10-0x17*/
5685 0xe272, 0xe273, 0xe274, 0xe275, 0xe276, 0xe277, 0xe278, 0xe279, /*0x18-0x1f*/
5686 0xe27a, 0xe27b, 0xe27c, 0xe27d, 0xe27e, 0xe27f, 0xe280, 0xe281, /*0x20-0x27*/
5687 0xe282, 0xe283, 0xe284, 0xe285, 0xe286, 0xe287, 0xe288, 0xe289, /*0x28-0x2f*/
5688 0xe28b, 0xe28c, 0xe28d, 0xe28e, 0xe28f, 0xe290, 0xe291, 0xe292, /*0x30-0x37*/
5689 0xe293, 0xe294, 0xe295, 0xe296, 0xe297, 0xe298, 0xe299, 0xe29a, /*0x38-0x3f*/
5690 0xe29b, 0xe29c, 0xe29d, 0xe29e, 0xe29f, 0xe2a0, 0xe340, 0xe341, /*0x40-0x47*/
5691 0xe342, 0xe343, 0xe344, 0xe345, 0xe346, 0xe347, 0xe348, 0xe349, /*0x48-0x4f*/
5692 0xe34a, 0xe34b, 0xe34c, 0xe34d, 0xe34e, 0xe34f, 0xe350, 0xe351, /*0x50-0x57*/
5693 0xe352, 0xe353, 0xe354, 0xe355, 0xe356, 0xe357, 0xe358, 0xe359, /*0x58-0x5f*/
5694 0xe35a, 0xe35b, 0xe35c, 0xe35d, 0xe35e, 0xe35f, 0xe360, 0xe361, /*0x60-0x67*/
5695 0xe362, 0xe363, 0xe364, 0xe365, 0xe366, 0xe367, 0xe368, 0xe369, /*0x68-0x6f*/
5696 0xe36a, 0xe36b, 0xe36c, 0xe36d, 0xbcfc, 0xe36e, 0xe36f, 0xe370, /*0x70-0x77*/
5697 0xe371, 0xe372, 0xe373, 0xe374, 0xe375, 0xe376, 0xe377, 0xe378, /*0x78-0x7f*/
5698 0xe379, 0xe37a, 0xe37b, 0xe37c, 0xe37d, 0xe37e, 0xe37f, 0xe380, /*0x80-0x87*/
5699 0xe382, 0xe383, 0xe384, 0xe385, 0xe386, 0xe387, 0xfc6c, 0xe388, /*0x88-0x8f*/
5700 0xe389, 0xe38a, 0xe38b, 0xe38c, 0xe38d, 0xe38e, 0xe38f, 0xe390, /*0x90-0x97*/
5701 0xe391, 0xe392, 0xe393, 0xe394, 0xe395, 0xe396, 0xe397, 0xe398, /*0x98-0x9f*/
5702 0xe399, 0xe39a, 0xe39b, 0xe39c, 0xe39d, 0xe39e, 0xe39f, 0xe3a0, /*0xa0-0xaf*/
5703 0xe440, 0xe441, 0xe442, 0xe443, 0xe444, 0xe445, 0xfc67, 0xe446, /*0xa8-0xaf*/
5704 0xe447, 0xe448, 0xe449, 0xe44a, 0xe44b, 0xe44c, 0xe44d, 0xe44e, /*0xb0-0xbf*/
5705 0xe44f, 0xe450, 0xe451, 0xe452, 0xe453, 0xe454, 0xe455, 0xe456, /*0xb8-0xbf*/
5706 0xe457, 0xe458, 0xe459, 0xe45a, 0xe45b, 0xe45c, 0xe45d, 0xe45e, /*0xc0-0xc7*/
5707 0xfc68, 0xe45f, 0xe460, 0xe461, 0xe462, 0xe463, 0xe464, 0xe465, /*0xc8-0xcf*/
5708 0xe466, 0xe467, 0xe468, 0xe469, 0xe46a, 0xe46b, 0xe46c, 0xe46d, /*0xd0-0xd7*/
5709 0xe46e, 0xe46f, 0xe470, 0xe471, 0xe472, 0xe473, 0xe474, 0xe475, /*0xd8-0xdf*/
5710 0xe476, 0xe477, 0xe478, 0xe479, 0xe47a, 0xe47b, 0xe47c, 0xe47d, /*0xe0-0xef*/
5711 0xe47e, 0xe480, 0xe481, 0xe482, 0xe483, 0xe484, 0xe485, 0xe486, /*0xf0-0xff*/
5712 0xe487, 0xe488, 0xe489, 0xe48a, 0xe48b, 0xe48c, 0xe48d, 0xe48e,
5713 0xe48f, 0xe490, 0xe491, 0xe492, 0xe493, 0xe494, 0xe495, 0xe496,
5714 /* 0x9300 */
5715 0xe497, 0xe498, 0xe499, 0xe49a, 0xe49b, 0xe49c, 0xe49d, 0xe49e, /*0x00-0x07*/
5716 0xe49f, 0xe4a0, 0xe4a1, 0xe4a2, 0xe4a3, 0xe4a4, 0xe4a5, 0xe4a6, /*0x08-0x0f*/
5717 0xe4a7, 0xe4a8, 0xe4a9, 0xe4aa, 0xe4ab, 0xe4ac, 0xe4ad, 0xe4ae, /*0x10-0x17*/
5718 0xe4af, 0xe4b0, 0xe4b1, 0xe4b2, 0xe4b3, 0xe4b4, 0xe4b5, 0xe4b6, /*0x18-0x1f*/
5719 0xe4b7, 0xe4b8, 0xe4b9, 0xe4ba, 0xe4bb, 0xe4bc, 0xe4bd, 0xe4be, /*0x20-0x27*/
5720 0xe4bf, 0xe4c0, 0xe4c1, 0xe4c2, 0xe4c3, 0xe4c4, 0xe4c5, 0xe4c6, /*0x28-0x2f*/
5721 0xe4c7, 0xe4c8, 0xe4c9, 0xe4ca, 0xe4cb, 0xe4cc, 0xe4cd, 0xe4ce, /*0x30-0x37*/
5722 0xe4cf, 0xe4d0, 0xe4d1, 0xe4d2, 0xe4d3, 0xe4d4, 0xe4d5, 0xe4d6, /*0x38-0x3f*/
5723 0xe4d7, 0xe4d8, 0xe4d9, 0xe4da, 0xe4db, 0xe4dc, 0xe4dd, 0xe4de, /*0x40-0x47*/
5724 0xe4df, 0xe4e0, 0xe4e1, 0xe4e2, 0xe4e3, 0xe4e4, 0xe4e5, 0xe4e6, /*0x48-0x4f*/
5725 0xe4e7, 0xe4e8, 0xe4e9, 0xe4ea, 0xe4eb, 0xe4ec, 0xe4ed, 0xe4ee, /*0x50-0x57*/
5726 0xe4ef, 0xe4f0, 0xe4f1, 0xe4f2, 0xe4f3, 0xe4f4, 0xe4f5, 0xe4f6, /*0x58-0x5f*/
5727 0xe4f7, 0xe4f8, 0xe4f9, 0xe4fa, 0xe4fb, 0xe4fc, 0xe4fd, 0xe4fe, /*0x60-0x6f*/
5728 0xe4ff, 0xe500, 0xe501, 0xe502, 0xe503, 0xe504, 0xe505, 0xe506, /*0x68-0x6f*/
5729 0xe507, 0xe508, 0xe509, 0xe50a, 0xe50b, 0xe50c, 0xe50d, 0xe50e, /*0x70-0x77*/
5730 0xe50f, 0xe510, 0xe511, 0xe512, 0xe513, 0xe514, 0xe515, 0xe516, /*0x78-0x7f*/
5731 0xe517, 0xe518, 0xe519, 0xe51a, 0xe51b, 0xe51c, 0xe51d, 0xe51e, /*0x80-0x87*/
5732 0xe51f, 0xe520, 0xe521, 0xe522, 0xe523, 0xe524, 0xe525, 0xe526, /*0x88-0x8f*/
5733 0xe527, 0xe528, 0xe529, 0xe52a, 0xe52b, 0xe52c, 0xe52d, 0xe52e, /*0x90-0x97*/
5734 0xe52f, 0xe530, 0xe531, 0xe532, 0xe533, 0xe534, 0xe535, 0xe536, /*0x98-0x9f*/
5735 0xe537, 0xe538, 0xe539, 0xe53a, 0xe53b, 0xe53c, 0xe53d, 0xe53e, /*0xa0-0xaf*/
5736 0xe53f, 0xe540, 0xe541, 0xe542, 0xe543, 0xe544, 0xe545, 0xe546, /*0xa8-0xaf*/
5737 0xe547, 0xe548, 0xe549, 0xe54a, 0xe54b, 0xe54c, 0xe54d, 0xe54e, /*0xb0-0xbf*/
5738 0xe54f, 0xe550, 0xe551, 0xe552, 0xe553, 0xe554, 0xe555, 0xe556, /*0xb8-0xbf*/
5739 0xe557, 0xe558, 0xe559, 0xe55a, 0xe55b, 0xe55c, 0xe55d, 0xe55e, /*0xc0-0xc7*/
5740 0xe55f, 0xe560, 0xe561, 0xe562, 0xe563, 0xe564, 0xe565, 0xe566, /*0xc8-0xcf*/
```



```

5828 0xea8e, 0xea8f, 0xdaef, 0xea90, 0xdaf0, 0xc1ea, 0xccd5, 0xcffd, /*0x70-0x77*/
5829 0xea91, 0xea92, 0xea93, 0xea94, 0xea95, 0xea96, 0xea97, 0xea98, /*0x78-0x7f*/
5830 0xea99, 0xea9a, 0xea9b, 0xea9c, 0xea9d, 0xd3e7, 0xc2a1, 0xea9e, /*0x80-0x87*/
5831 0xdaf1, 0xea9f, 0xeaa0, 0xcbe5, 0xeb40, 0xdaf2, 0xeb41, 0xcbe6, /*0x88-0x8f*/
5832 0xd2fe, 0xeb42, 0xeb43, 0xeb44, 0xb8f4, 0xeb45, 0xeb46, 0xdaf3, /*0x90-0x97*/
5833 0xb0af, 0xcfb6, 0xcfb7, 0xeb47, 0xeb48, 0xd5cf, 0xeb49, 0xeb4a, 0xeb4b, /*0x98-0x9f*/
5834 0xeb4c, 0xeb4d, 0xeb4e, 0xeb4f, 0xeb50, 0xeb51, 0xeb52, 0xcbed, /*0xa0-0xa7*/
5835 0xeb53, 0xeb54, 0xeb55, 0xeb56, 0xeb57, 0xeb58, 0xeb59, 0xeb5a, /*0xa8-0xaf*/
5836 0xdaf4, 0xeb5b, 0xeb5c, 0xeb5d, 0xeb5e, 0xc1a5, 0xeb5f, /*0xb0-0xbf*/
5837 0xeb60, 0xf6bf, 0xeb61, 0xeb62, 0xf6c0, 0xf6c1, 0xc4d1, 0xeb63, /*0xb8-0xbf*/
5838 0xc8b8, 0xd1e3, 0xeb64, 0xeb65, 0xd0db, 0xd1c5, 0xbcaf, 0xb9cd, /*0xc0-0xc7*/
5839 0xeb66, 0xef64, 0xeb67, 0xeb68, 0xb4c6, 0xd3ba, 0xf6c2, 0xb3fb, /*0xc8-0xcf*/
5840 0xeb69, 0xeb6a, 0xf6c3, 0xeb6b, 0xeb6c, 0xb5f1, 0xeb6d, 0xeb6e, /*0xd0-0xd7*/
5841 0xeb6f, 0xeb70, 0xeb71, 0xeb72, 0xeb73, 0xeb74, 0xeb75, 0xeb76, /*0xd8-0xdf*/
5842 0xf6c5, 0xeb77, 0xeb78, 0xeb79, 0xeb7a, 0xeb7b, 0xeb7c, 0xeb7d, /*0xe0-0xef*/
5843 0xd3ea, 0xf6a7, 0xd1a9, 0xeb7e, 0xeb80, 0xeb81, 0xeb82, 0xf6a9, /*0xe8-0xef*/
5844 0xeb83, 0xeb84, 0xeb85, 0xf6a8, 0xeb86, 0xeb87, 0xc1e3, 0xc0d7, /*0xf0-0xf7*/
5845 0xeb88, 0xb1a2, 0xeb89, 0xeb8a, 0xeb8b, 0xeb8c, 0xceed, 0xeb8d, /*0xf8-0xff*/
5846 /* 0x9700 */
5847 0xd0e8, 0xf6ab, 0xeb8e, 0xeb8f, 0xcfff, 0xeb90, 0xf6aa, 0xd5f0, /*0x00-0x07*/
5848 0xf6ac, 0xc3b9, 0xeb91, 0xeb92, 0xeb93, 0xbbf4, 0xf6ae, 0xf6ad, /*0x08-0x0f*/
5849 0xeb94, 0xeb95, 0xeb96, 0xc4de, 0xeb97, 0xeb98, 0xc1d8, 0xeb99, /*0x10-0x17*/
5850 0xeb9a, 0xeb9b, 0xeb9c, 0xeb9d, 0xcbaa, 0xeb9e, 0xcfb3, 0xeb9f, /*0x18-0x1f*/
5851 0xeba0, 0xec40, 0xec41, 0xec42, 0xec43, 0xec44, 0xec45, 0xec46, /*0x20-0x27*/
5852 0xec47, 0xec48, 0xf6af, 0xec49, 0xec4a, 0xf6b0, 0xec4b, 0xec4c, /*0x28-0x2f*/
5853 0xf6b1, 0xec4d, 0xc2b6, 0xec4e, 0xec4f, 0xec50, 0xec51, 0xec52, /*0x30-0x37*/
5854 0xb0d4, 0xc5f9, 0xec53, 0xec54, 0xec55, 0xec56, 0xf6b2, 0xec57, /*0x38-0x3f*/
5855 0xec58, 0xec59, 0xec5a, 0xec5b, 0xec5c, 0xec5d, 0xec5e, 0xec5f, /*0x40-0x47*/
5856 0xec60, 0xec61, 0xec62, 0xec63, 0xec64, 0xec65, 0xec66, 0xec67, /*0x48-0x4f*/
5857 0xec68, 0xec69, 0xc7e0, 0xf6a6, 0xec6a, 0xec6b, 0xeb88, 0xec6c, /*0x50-0x57*/
5858 0xec6d, 0xbbe2, 0xec6e, 0xb5e5, 0xec6f, 0xec70, 0xb7c7, 0xec71, /*0x58-0x5f*/
5859 0xbfbf, 0xc3d2, 0xc3e6, 0xec72, 0xec73, 0xd8cc, 0xec74, 0xec75, /*0x60-0x67*/
5860 0xec76, 0xb8ef, 0xec77, 0xec78, 0xec79, 0xec7a, 0xec7b, 0xec7c, /*0x68-0x6f*/
5861 0xec7d, 0xec7e, 0xec80, 0xbdf9, 0xd1a5, 0xec81, 0xb0d0, 0xec82, /*0x70-0x77*/
5862 0xec83, 0xec84, 0xec85, 0xec86, 0xf7b0, 0xec87, 0xec88, 0xec89, /*0x78-0x7f*/
5863 0xec8a, 0xec8b, 0xec8c, 0xec8d, 0xec8e, 0xf7b1, 0xec8f, 0xec90, /*0x80-0x87*/
5864 0xec91, 0xec92, 0xec93, 0xd0ac, 0xec94, 0xb0b0, 0xec95, 0xec96, /*0x88-0x8f*/
5865 0xec97, 0xf7b2, 0xf7b3, 0xec98, 0xf7b4, 0xec99, 0xec9a, 0xec9b, /*0x90-0x97*/
5866 0xc7ca, 0xec9c, 0xec9d, 0xec9e, 0xec9f, 0xeaca, 0xed40, 0xed41, /*0x98-0x9f*/
5867 0xbecf, 0xed42, 0xed43, 0xf7b7, 0xed44, 0xed45, 0xed46, 0xed47, /*0xa0-0xaf*/
5868 0xed48, 0xed49, 0xed4a, 0xf7b6, 0xed4b, 0xb1de, 0xed4c, 0xf7b5, /*0xa8-0xaf*/
5869 0xed4d, 0xed4e, 0xf7b8, 0xed4f, 0xf7b9, 0xed50, 0xed51, 0xed52, /*0xb0-0xbf*/
5870 0xed53, 0xed54, 0xed55, 0xed56, 0xed57, 0xed58, 0xed59, 0xed5a, /*0xb8-0xbf*/
5871 0xed5b, 0xed5c, 0xed5d, 0xed5e, 0xed5f, 0xed60, 0xed61, 0xed62, /*0xc0-0xc7*/
5872 0xed63, 0xed64, 0xed65, 0xed66, 0xed67, 0xed68, 0xed69, 0xed6a, /*0xc8-0xcf*/
5873 0xed6b, 0xed6c, 0xed6d, 0xed6e, 0xed6f, 0xed70, 0xed71, 0xed72, /*0xd0-0xdf*/
5874 0xed73, 0xed74, 0xed75, 0xed76, 0xed77, 0xed78, 0xed79, 0xed7a, /*0xd8-0xdf*/
5875 0xed7b, 0xed7c, 0xed7d, 0xed7e, 0xed80, 0xed81, 0xcea4, 0xc8cd, /*0xe0-0xef*/
5876 0xed82, 0xbaab, 0xeb88, 0xeb89, 0xeb8a, 0xbec2, 0xed83, 0xed84, /*0xe8-0xef*/
5877 0xed85, 0xed86, 0xed87, 0xd2f4, 0xed88, 0xd4cf, 0xc9d8, 0xed89, /*0xf0-0xf7*/
5878 0xed8a, 0xed8b, 0xed8c, 0xed8d, 0xed8e, 0xed8f, 0xed90, 0xed91, /*0xf8-0xff*/
5879 /* 0x9800 */
5880 0xed92, 0xed93, 0xed94, 0xed95, 0xed96, 0xed97, 0xed98, 0xed99, /*0x00-0x07*/
5881 0xed9a, 0xed9b, 0xed9c, 0xed9d, 0xed9e, 0xed9f, 0xeda0, 0xee40, /*0x08-0x0f*/
5882 0xee41, 0xee42, 0xee43, 0xee44, 0xee45, 0xee46, 0xee47, 0xee48, /*0x10-0x17*/
5883 0xee49, 0xee4a, 0xee4b, 0xee4c, 0xee4d, 0xee4e, 0xee4f, 0xee50, /*0x18-0x1f*/
5884 0xee51, 0xee52, 0xee53, 0xee54, 0xee55, 0xee56, 0xee57, 0xee58, /*0x20-0x27*/
5885 0xee59, 0xee5a, 0xee5b, 0xee5c, 0xee5d, 0xee5e, 0xee5f, 0xee60, /*0x28-0x2f*/
5886 0xee61, 0xee62, 0xee63, 0xee64, 0xee65, 0xee66, 0xee67, 0xee68, /*0x30-0x37*/
5887 0xee69, 0xee6a, 0xee6b, 0xee6c, 0xee6d, 0xee6e, 0xee6f, 0xee70, /*0x38-0x3f*/
5888 0xee71, 0xee72, 0xee73, 0xee74, 0xee75, 0xee76, 0xee77, 0xee78, /*0x40-0x47*/
5889 0xee79, 0xee7a, 0xee7b, 0xee7c, 0xee7d, 0xee7e, 0xee80, 0xee81, /*0x48-0x4f*/
5890 0xee82, 0xee83, 0xee84, 0xee85, 0xee86, 0xee87, 0xee88, 0xee89, /*0x50-0x57*/
5891 0xee8a, 0xee8b, 0xee8c, 0xee8d, 0xee8e, 0xee8f, 0xee90, 0xee91, /*0x58-0x5f*/
5892 0xee92, 0xee93, 0xee94, 0xee95, 0xee96, 0xee97, 0xee98, 0xee99, /*0x60-0x67*/
5893 0xee9a, 0xee9b, 0xee9c, 0xee9d, 0xee9e, 0xee9f, 0xeea0, 0xef40, /*0x68-0x6f*/
5894 0xef41, 0xef42, 0xef43, 0xef44, 0xef45, 0xd2b3, 0xb6a5, 0xc7ea, /*0x70-0x77*/
5895 0xf1fc, 0xcfee, 0xcbb3, 0xd0eb, 0xe7ef, 0xcde7, 0xb9cb, 0xb6d9, /*0x78-0x7f*/
5896 0xf1fd, 0xb0e4, 0xcbbc, 0xf1fe, 0xd4a4, 0xc2ad, 0xc1ec, 0xc6c4, /*0x80-0x87*/
5897 0xbeb1, 0xf2a1, 0xbcd5, 0xef46, 0xf2a2, 0xf2a3, 0xef47, 0xf2a4, /*0x88-0x8f*/
5898 0xd2c3, 0xc6b5, 0xef48, 0xcdc7, 0xf2a5, 0xef49, 0xd3b1, 0xbfc5, /*0x90-0x97*/
5899 0xcce2, 0xef4a, 0xf2a6, 0xf2a7, 0xd1d5, 0xb6ee, 0xf2a8, 0xf2a9, /*0x98-0x9f*/
5900 0xb5df, 0xf2aa, 0xf2ab, 0xf2ac, 0xb2fc, 0xf2ad, 0xf2ae, 0xc8a7, /*0xa0-0xaf*/
5901 0xef4c, 0xef4d, 0xef4e, 0xef4f, 0xef50, 0xef51, 0xef52, 0xef53, /*0xa8-0xaf*/
5902 0xef54, 0xef55, 0xef56, 0xef57, 0xef58, 0xef59, 0xef5a, 0xef5b, /*0xb0-0xbf*/
5903 0xef5c, 0xef5d, 0xef5e, 0xef5f, 0xef60, 0xef61, 0xef62, 0xef63, /*0xb8-0xbf*/
5904 0xef64, 0xef65, 0xef66, 0xef67, 0xef68, 0xef69, 0xef6a, 0xef6b, /*0xc0-0xc7*/
5905 0xef6c, 0xef6d, 0xef6e, 0xef6f, 0xef70, 0xef71, 0xb7e7, 0xef72, /*0xc8-0xcf*/
5906 0xef73, 0xecad, 0xecae, 0xecab, 0xef74, 0xecac, 0xef75, 0xef76, /*0xd0-0xdf*/
5907 0xc6ae, 0xecad, 0xecae, 0xecae, 0xef77, 0xef78, 0xef79, 0xb7c9, 0xcab3, /*0xd8-0xdf*/
5908 0xef7a, 0xef7b, 0xef7c, 0xef7d, 0xef7e, 0xef7f, 0xef80, 0xef81, 0xe2b8, /*0xe0-0xef*/
5909 0xf7cf, 0xef82, 0xef83, 0xef84, 0xef85, 0xef86, 0xef87, 0xef88, /*0xe8-0xef*/
5910 0xef89, 0xef8a, 0xef8b, 0xef8c, 0xef8d, 0xef8e, 0xef8f, 0xef90, /*0xf0-0xf7*/
5911 0xef91, 0xef92, 0xef93, 0xef94, 0xef95, 0xef96, 0xef97, 0xef98, /*0xf8-0xff*/
5912 /* 0x9900 */
5913 0xef99, 0xef9a, 0xef9b, 0xef9c, 0xef9d, 0xef9e, 0xef9f, 0xefa0, /*0x00-0x07*/
5914 0xf040, 0xf041, 0xf042, 0xf043, 0xf044, 0xf7d0, 0xf045, 0xf046, /*0x08-0x0f*/

```

```

5915 0xb2cd, 0xf047, 0xf048, 0xf049, 0xf04a, 0xf04b, 0xf04c, 0xf04d, /*0x10-0x17*/
5916 0xf04e, 0xf04f, 0xf050, 0xf051, 0xf052, 0xf053, 0xf054, 0xf055, /*0x18-0x1f*/
5917 0xf056, 0xf057, 0xf058, 0xf059, 0xf05a, 0xf05b, 0xf05c, 0xf05d, /*0x20-0x27*/
5918 0xf05e, 0xf05f, 0xf060, 0xf061, 0xf062, 0xf063, 0xf064, 0xf065, /*0x28-0x2f*/
5919 0xf066, 0xf067, 0xf068, 0xf069, 0xf06a, 0xf06b, 0xf06c, 0xf06d, /*0x30-0x37*/
5920 0xf06e, 0xf06f, 0xf070, 0xf071, 0xf072, 0xf073, 0xf074, /*0x38-0x3f*/
5921 0xf075, 0xf076, 0xf077, 0xf078, 0xf079, 0xf07a, 0xf07b, 0xf07c, /*0x40-0x47*/
5922 0xf07d, 0xf07e, 0xf080, 0xf081, 0xf082, 0xf083, 0xf084, 0xf085, /*0x48-0x4f*/
5923 0xf086, 0xf087, 0xf088, 0xf089, 0xf08a, 0xf08b, 0xf08c, 0xf08d, /*0x50-0x57*/
5924 0xf08e, 0xf08f, 0xf090, 0xf091, 0xf092, 0xf093, 0xf094, 0xf095, /*0x58-0x5f*/
5925 0xf096, 0xf097, 0xf098, 0xf099, 0xf09a, 0xf09b, 0xf09c, 0xf09d, /*0x60-0x67*/
5926 0xf09e, 0xf09f, 0xf0a0, 0xf0a1, 0xf0a2, 0xf0a3, 0xf0a4, 0xf0a5, /*0x68-0x6f*/
5927 0xf0a6, 0xf0a7, 0xf0a8, 0xf0a9, 0xf0aa, 0xf0ab, 0xf0ac, 0xf0ad, /*0x70-0x77*/
5928 0xf0ae, 0xf0af, 0xf0b0, 0xf0b1, 0xf0b2, 0xf0b3, 0xf0b4, 0xf0b5, /*0x78-0x7f*/
5929 0xf0b6, 0xf0b7, 0xf0b8, 0xf0b9, 0xf0ba, 0xf0bb, 0xf0bc, 0xf0bd, /*0x80-0x87*/
5930 0xf0be, 0xf0bf, 0xf0c0, 0xf0c1, 0xf0c2, 0xf0c3, 0xf0c4, 0xf0c5, /*0x88-0x8f*/
5931 0xf0c6, 0xf0c7, 0xf0c8, 0xf0c9, 0xf0ca, 0xf0cb, 0xf0cc, 0xf0cd, /*0x90-0x97*/
5932 0xf0ce, 0xf0cf, 0xf0d0, 0xf0d1, 0xf0d2, 0xf0d3, 0xf0d4, 0xf0d5, /*0x98-0x9f*/
5933 0xf0de, 0xf0df, 0xf0e0, 0xf0e1, 0xf0e2, 0xf0e3, 0xf0e4, 0xf0e5, /*0xa0-0xa7*/
5934 0xf0e6, 0xf0e7, 0xf0e8, 0xf0e9, 0xf0ea, 0xf0eb, 0xf0ec, 0xf0ed, /*0xa8-0xaf*/
5935 0xf0ee, 0xf0ef, 0xf0f0, 0xf0f1, 0xf0f2, 0xf0f3, 0xf0f4, 0xf0f5, /*0xb0-0xb7*/
5936 0xf0f6, 0xf0f7, 0xf0f8, 0xf0f9, 0xf0fa, 0xf0fb, 0xf0fc, 0xf0fd, /*0xb8-0xbf*/
5937 0xf0fe, 0xf0ff, 0xf100, 0xf101, 0xf102, 0xf103, 0xf104, 0xf105, /*0xc0-0xc7*/
5938 0xf106, 0xf107, 0xf108, 0xf109, 0xf10a, 0xf10b, 0xf10c, 0xf10d, /*0xc8-0xcd*/
5939 0xf10e, 0xf10f, 0xf110, 0xf111, 0xf112, 0xf113, 0xf114, 0xf115, /*0xd0-0xd7*/
5940 0xf116, 0xf117, 0xf118, 0xf119, 0xf11a, 0xf11b, 0xf11c, 0xf11d, /*0xd8-0xdd*/
5941 0xf11e, 0xf11f, 0xf120, 0xf121, 0xf122, 0xf123, 0xf124, 0xf125, /*0xde-0xdf*/
5942 0xf126, 0xf127, 0xf128, 0xf129, 0xf12a, 0xf12b, 0xf12c, 0xf12d, /*0xe0-0xe7*/
5943 0xf12e, 0xf12f, 0xf130, 0xf131, 0xf132, 0xf133, 0xf134, 0xf135, /*0xe8-0xef*/
5944 0xf136, 0xf137, 0xf138, 0xf139, 0xf13a, 0xf13b, 0xf13c, 0xf13d, /*0xf0-0xf7*/
5945 0xf13e, 0xf13f, 0xf140, 0xf141, 0xf142, 0xf143, 0xf144, 0xf145, /*0xf8-0xff*/
5945 /* 0x9a00 */
5946 0xf246, 0xf247, 0xf248, 0xf249, 0xf24a, 0xf24b, 0xf24c, 0xf24d, /*0x00-0x07*/
5947 0xf24e, 0xf24f, 0xf250, 0xf251, 0xf252, 0xf253, 0xf254, 0xf255, /*0x08-0x0f*/
5948 0xf256, 0xf257, 0xf258, 0xf259, 0xf25a, 0xf25b, 0xf25c, 0xf25d, /*0x10-0x17*/
5949 0xf25e, 0xf25f, 0xf260, 0xf261, 0xf262, 0xf263, 0xf264, 0xf265, /*0x18-0x1f*/
5950 0xf266, 0xf267, 0xf268, 0xf269, 0xf26a, 0xf26b, 0xf26c, 0xf26d, /*0x20-0x27*/
5951 0xf26e, 0xf26f, 0xf270, 0xf271, 0xf272, 0xf273, 0xf274, 0xf275, /*0x28-0x2f*/
5952 0xf276, 0xf277, 0xf278, 0xf279, 0xf27a, 0xf27b, 0xf27c, 0xf27d, /*0x30-0x37*/
5953 0xf27e, 0xf27f, 0xf280, 0xf281, 0xf282, 0xf283, 0xf284, 0xf285, /*0x38-0x3f*/
5954 0xf286, 0xf287, 0xf288, 0xf289, 0xf28a, 0xf28b, 0xf28c, 0xf28d, /*0x40-0x47*/
5955 0xf28e, 0xf28f, 0xf290, 0xf291, 0xf292, 0xf293, 0xf294, 0xf295, /*0x48-0x4f*/
5956 0xf296, 0xf297, 0xf298, 0xf299, 0xf29a, 0xf29b, 0xf29c, 0xf29d, /*0x50-0x57*/
5957 0xf29e, 0xf29f, 0xf2a0, 0xf2a1, 0xf2a2, 0xf2a3, 0xf2a4, 0xf2a5, /*0x58-0x5f*/
5958 0xf2a6, 0xf2a7, 0xf2a8, 0xf2a9, 0xf2aa, 0xf2ab, 0xf2ac, 0xf2ad, /*0x60-0x67*/
5959 0xf2ae, 0xf2af, 0xf2b0, 0xf2b1, 0xf2b2, 0xf2b3, 0xf2b4, 0xf2b5, /*0x68-0x6f*/
5960 0xf2b6, 0xf2b7, 0xf2b8, 0xf2b9, 0xf2ba, 0xf2bb, 0xf2bc, 0xf2bd, /*0x70-0x77*/
5961 0xf2be, 0xf2bf, 0xf2c0, 0xf2c1, 0xf2c2, 0xf2c3, 0xf2c4, 0xf2c5, /*0x78-0x7f*/
5962 0xf2c6, 0xf2c7, 0xf2c8, 0xf2c9, 0xf2ca, 0xf2cb, 0xf2cc, 0xf2cd, /*0x80-0x87*/
5963 0xf2ce, 0xf2cf, 0xf2d0, 0xf2d1, 0xf2d2, 0xf2d3, 0xf2d4, 0xf2d5, /*0x88-0x8f*/
5964 0xf2de, 0xf2df, 0xf2e0, 0xf2e1, 0xf2e2, 0xf2e3, 0xf2e4, 0xf2e5, /*0x90-0x97*/
5965 0xf2e6, 0xf2e7, 0xf2e8, 0xf2e9, 0xf2ea, 0xf2eb, 0xf2ec, 0xf2ed, /*0x98-0x9f*/
5966 0xf2ee, 0xf2ef, 0xf2f0, 0xf2f1, 0xf2f2, 0xf2f3, 0xf2f4, 0xf2f5, /*0xa0-0xa7*/
5967 0xf2f6, 0xf2f7, 0xf2f8, 0xf2f9, 0xf2fa, 0xf2fb, 0xf2fc, 0xf2fd, /*0xa8-0xaf*/
5968 0xf2fe, 0xf2ff, 0xf300, 0xf301, 0xf302, 0xf303, 0xf304, 0xf305, /*0xb0-0xb7*/
5969 0xf306, 0xf307, 0xf308, 0xf309, 0xf30a, 0xf30b, 0xf30c, 0xf30d, /*0xb8-0xbf*/
5970 0xf30e, 0xf30f, 0xf310, 0xf311, 0xf312, 0xf313, 0xf314, 0xf315, /*0xc0-0xc7*/
5971 0xf316, 0xf317, 0xf318, 0xf319, 0xf31a, 0xf31b, 0xf31c, 0xf31d, /*0xc8-0xcd*/
5972 0xf31e, 0xf31f, 0xf320, 0xf321, 0xf322, 0xf323, 0xf324, 0xf325, /*0xd0-0xd7*/
5973 0xf326, 0xf327, 0xf328, 0xf329, 0xf32a, 0xf32b, 0xf32c, 0xf32d, /*0xd8-0xdd*/
5974 0xf32e, 0xf32f, 0xf330, 0xf331, 0xf332, 0xf333, 0xf334, 0xf335, /*0xde-0xdf*/
5975 0xf336, 0xf337, 0xf338, 0xf339, 0xf33a, 0xf33b, 0xf33c, 0xf33d, /*0xe0-0xe7*/
5976 0xf33e, 0xf33f, 0xf340, 0xf341, 0xf342, 0xf343, 0xf344, 0xf345, /*0xe8-0xef*/
5977 0xf346, 0xf347, 0xf348, 0xf349, 0xf34a, 0xf34b, 0xf34c, 0xf34d, /*0xf0-0xf7*/
5978 0xf34e, 0xf34f, 0xf350, 0xf351, 0xf352, 0xf353, 0xf354, 0xf355, /*0xf8-0xff*/
5978 /* 0x9b00 */
5979 0xf39b, 0xf39c, 0xf39d, 0xf39e, 0xf39f, 0xf3a0, 0xf3a1, 0xf3a2, /*0x00-0x07*/
5980 0xf3a3, 0xf3a4, 0xf3a5, 0xf3a6, 0xf3a7, 0xf3a8, 0xf3a9, 0xf3aa, /*0x08-0x0f*/
5981 0xf3ab, 0xf3ac, 0xf3ad, 0xf3ae, 0xf3af, 0xf3b0, 0xf3b1, 0xf3b2, /*0x10-0x17*/
5982 0xf3b3, 0xf3b4, 0xf3b5, 0xf3b6, 0xf3b7, 0xf3b8, 0xf3b9, 0xf3ba, /*0x18-0x1f*/
5983 0xf3bb, 0xf3bc, 0xf3bd, 0xf3be, 0xf3bf, 0xf3c0, 0xf3c1, 0xf3c2, /*0x20-0x27*/
5984 0xf3c3, 0xf3c4, 0xf3c5, 0xf3c6, 0xf3c7, 0xf3c8, 0xf3c9, 0xf3ca, /*0x28-0x2f*/
5985 0xf3cb, 0xf3cc, 0xf3cd, 0xf3ce, 0xf3cf, 0xf3d0, 0xf3d1, 0xf3d2, /*0x30-0x37*/
5986 0xf3d3, 0xf3d4, 0xf3d5, 0xf3d6, 0xf3d7, 0xf3d8, 0xf3d9, 0xf3da, /*0x38-0x3f*/
5987 0xf3db, 0xf3dc, 0xf3dd, 0xf3de, 0xf3df, 0xf3e0, 0xf3e1, 0xf3e2, /*0x40-0x47*/
5988 0xf3e3, 0xf3e4, 0xf3e5, 0xf3e6, 0xf3e7, 0xf3e8, 0xf3e9, 0xf3ea, /*0x48-0x4f*/
5989 0xf3eb, 0xf3ec, 0xf3ed, 0xf3ee, 0xf3ef, 0xf3f0, 0xf3f1, 0xf3f2, /*0x50-0x57*/
5990 0xf3f3, 0xf3f4, 0xf3f5, 0xf3f6, 0xf3f7, 0xf3f8, 0xf3f9, 0xf3fa, /*0x58-0x5f*/
5991 0xf3fb, 0xf3fc, 0xf3fd, 0xf3fe, 0xf3ff, 0xf400, 0xf401, 0xf402, /*0x60-0x67*/
5992 0xf403, 0xf404, 0xf405, 0xf406, 0xf407, 0xf408, 0xf409, 0xf40a, /*0x68-0x6f*/
5993 0xf40b, 0xf40c, 0xf40d, 0xf40e, 0xf40f, 0xf410, 0xf411, 0xf412, /*0x70-0x77*/
5994 0xf413, 0xf414, 0xf415, 0xf416, 0xf417, 0xf418, 0xf419, 0xf41a, /*0x78-0x7f*/
5995 0xf41b, 0xf41c, 0xf41d, 0xf41e, 0xf41f, 0xf420, 0xf421, 0xf422, /*0x80-0x87*/
5996 0xf423, 0xf424, 0xf425, 0xf426, 0xf427, 0xf428, 0xf429, 0xf42a, /*0x88-0x8f*/
5997 0xf42b, 0xf42c, 0xf42d, 0xf42e, 0xf42f, 0xf430, 0xf431, 0xf432, /*0x90-0x97*/
5998 0xf433, 0xf434, 0xf435, 0xf436, 0xf437, 0xf438, 0xf439, 0xf43a, /*0x98-0x9f*/
5999 0xf43b, 0xf43c, 0xf43d, 0xf43e, 0xf43f, 0xf440, 0xf441, 0xf442, /*0xa0-0xaf*/
6000 0xf443, 0xf444, 0xf445, 0xf446, 0xf447, 0xf448, 0xf449, 0xf44a, /*0xa8-0xaf*/
6001 0xf44b, 0xf44c, 0xf44d, 0xf44e, 0xf44f, 0xf450, 0xf451, 0xf452, /*0xb0-0xb7*/

```

```

6002 0xf57c, 0xf57d, 0xf57e, 0xf580, 0xf581, 0xf582, 0xf583, 0xf584, /*0xb8-0xbf*/
6003 0xf585, 0xf586, 0xf587, 0xf588, 0xf589, 0xf58a, 0xf58b, 0xf58c, /*0xc0-0xc7*/
6004 0xf58d, 0xf58e, 0xf58f, 0xf590, 0xf591, 0xf592, 0xf593, 0xf594, /*0xc8-0xcf*/
6005 0xf595, 0xf596, 0xf597, 0xf598, 0xf599, 0xf59a, 0xf59b, 0xf59c, /*0xd0-0xd7*/
6006 0xf59d, 0xf59e, 0xf59f, 0xf5a0, 0xf5a1, 0xf5a2, 0xf5a3, 0xf5a4, /*0xd8-0xdf*/
6007 0xf5a5, 0xf5a6, 0xf5a7, 0xf5a8, 0xf5a9, 0xf5aa, 0xf5ab, 0xf5ac, /*0xe0-0xef*/
6008 0xf5ad, 0xf5ae, 0xf5af, 0xf5b0, 0xf5b1, 0xf5b2, 0xf5b3, 0xf5b4, /*0xf0-0xff*/
6009 0xf5b5, 0xf5b6, 0xf5b7, 0xf5b8, 0xf5b9, 0xf5ba, 0xf5bb, 0xf5bc, /*0xc0-0xc7*/
6010 0xf5bd, 0xf5be, 0xf5bf, 0xf5c0, 0xf5c1, 0xf5c2, 0xf5c3, 0xf5c4, /*0xc8-0xcf*/
6011 /* 0x9c00 */
6012 0xf664, 0xf665, 0xf666, 0xf667, 0xf668, 0xf669, 0xf66a, 0xf66b, /*0x00-0x07*/
6013 0xf66c, 0xf66d, 0xf66e, 0xf66f, 0xf670, 0xf671, 0xf672, 0xf673, /*0x08-0x0f*/
6014 0xf674, 0xf675, 0xf676, 0xf677, 0xf678, 0xf679, 0xf67a, 0xf67b, /*0x10-0x17*/
6015 0xf67c, 0xf67d, 0xf67e, 0xf67f, 0xf680, 0xf681, 0xf682, 0xf683, 0xf684, /*0x18-0x1f*/
6016 0xf685, 0xf686, 0xf687, 0xf688, 0xf689, 0xf68a, 0xf68b, 0xf68c, /*0x20-0x27*/
6017 0xf68d, 0xf68e, 0xf68f, 0xf690, 0xf691, 0xf692, 0xf693, 0xf694, /*0x28-0x2f*/
6018 0xf695, 0xf696, 0xf697, 0xf698, 0xf699, 0xf69a, 0xf69b, 0xf69c, /*0x30-0x37*/
6019 0xf69d, 0xf69e, 0xf69f, 0xf6a0, 0xf6a1, 0xf6a2, 0xf6a3, 0xf6a4, /*0x38-0x3f*/
6020 0xf6a5, 0xf6a6, 0xf6a7, 0xf6a8, 0xf6a9, 0xf6aa, 0xf6ab, 0xf6ac, /*0x40-0x47*/
6021 0xf6ad, 0xf6ae, 0xf6af, 0xf6b0, 0xf6b1, 0xf6b2, 0xf6b3, 0xf6b4, /*0x48-0x4f*/
6022 0xf6b5, 0xf6b6, 0xf6b7, 0xf6b8, 0xf6b9, 0xf6ba, 0xf6bb, 0xf6bc, /*0x50-0x57*/
6023 0xf6bd, 0xf6be, 0xf6bf, 0xf6c0, 0xf6c1, 0xf6c2, 0xf6c3, 0xf6c4, /*0x58-0x5f*/
6024 0xf6c5, 0xf6c6, 0xf6c7, 0xf6c8, 0xf6c9, 0xf6ca, 0xf6cb, 0xf6cc, /*0x60-0x67*/
6025 0xf6cd, 0xf6ce, 0xf6cf, 0xf6d0, 0xf6d1, 0xf6d2, 0xf6d3, 0xf6d4, /*0x68-0x6f*/
6026 0xf6d5, 0xf6d6, 0xf6d7, 0xf6d8, 0xf6d9, 0xf6da, 0xf6db, 0xf6dc, /*0x70-0x77*/
6027 0xf6dd, 0xf6de, 0xf6df, 0xf6e0, 0xf6e1, 0xf6e2, 0xf6e3, 0xf6e4, /*0x78-0x7f*/
6028 0xf6e5, 0xf6e6, 0xf6e7, 0xf6e8, 0xf6e9, 0xf6ea, 0xf6eb, 0xf6ec, /*0x80-0x87*/
6029 0xf6ed, 0xf6ee, 0xf6ef, 0xf6f0, 0xf6f1, 0xf6f2, 0xf6f3, 0xf6f4, /*0x88-0x8f*/
6030 0xf6f5, 0xf6f6, 0xf6f7, 0xf6f8, 0xf6f9, 0xf6fa, 0xf6fb, 0xf6fc, /*0x90-0x97*/
6031 0xf6fd, 0xf6fe, 0xf6ff, 0xf700, 0xf701, 0xf702, 0xf703, 0xf704, /*0x98-0x9f*/
6032 0xf705, 0xf706, 0xf707, 0xf708, 0xf709, 0xf70a, 0xf70b, 0xf70c, /*0xa0-0xaf*/
6033 0xf70d, 0xf70e, 0xf70f, 0xf710, 0xf711, 0xf712, 0xf713, 0xf714, /*0xab-0xaf*/
6034 0xf715, 0xf716, 0xf717, 0xf718, 0xf719, 0xf71a, 0xf71b, 0xf71c, /*0xb0-0xbf*/
6035 0xf71d, 0xf71e, 0xf71f, 0xf720, 0xf721, 0xf722, 0xf723, 0xf724, /*0xc0-0xc7*/
6036 0xf725, 0xf726, 0xf727, 0xf728, 0xf729, 0xf72a, 0xf72b, 0xf72c, /*0xc8-0xcf*/
6037 0xf72d, 0xf72e, 0xf72f, 0xf730, 0xf731, 0xf732, 0xf733, 0xf734, /*0xd0-0xd7*/
6038 0xf735, 0xf736, 0xf737, 0xf738, 0xf739, 0xf73a, 0xf73b, 0xf73c, /*0xd8-0xdf*/
6039 0xf73d, 0xf73e, 0xf73f, 0xf740, 0xf741, 0xf742, 0xf743, 0xf744, /*0xe0-0xef*/
6040 0xf745, 0xf746, 0xf747, 0xf748, 0xf749, 0xf74a, 0xf74b, 0xf74c, /*0xf0-0xff*/
6041 0xf74d, 0xf74e, 0xf74f, 0xf750, 0xf751, 0xf752, 0xf753, 0xf754, /*0xc0-0xc7*/
6042 0xf755, 0xf756, 0xf757, 0xf758, 0xf759, 0xf75a, 0xf75b, 0xf75c, /*0xc8-0xcf*/
6043 0xf75d, 0xf75e, 0xf75f, 0xf760, 0xf761, 0xf762, 0xf763, 0xf764, /*0xd0-0xd7*/
6044 /* 0x9d00 */
6045 0xf85d, 0xf85e, 0xf85f, 0xf860, 0xf861, 0xf862, 0xf863, 0xf864, /*0x00-0x07*/
6046 0xf865, 0xf866, 0xf867, 0xf868, 0xf869, 0xf86a, 0xf86b, 0xf86c, /*0x08-0x0f*/
6047 0xf86d, 0xf86e, 0xf86f, 0xf870, 0xf871, 0xf872, 0xf873, 0xf874, /*0x10-0x17*/
6048 0xf875, 0xf876, 0xf877, 0xf878, 0xf879, 0xf87a, 0xf87b, 0xf87c, /*0x18-0x1f*/
6049 0xf87d, 0xf87e, 0xf87f, 0xf880, 0xf881, 0xf882, 0xf883, 0xf884, 0xf885, /*0x20-0x27*/
6050 0xf886, 0xf887, 0xf888, 0xf889, 0xf88a, 0xf88b, 0xf88c, 0xf88d, /*0x28-0x2f*/
6051 0xf88e, 0xf88f, 0xf890, 0xf891, 0xf892, 0xf893, 0xf894, 0xf895, /*0x30-0x37*/
6052 0xf896, 0xf897, 0xf898, 0xf899, 0xf89a, 0xf89b, 0xf89c, 0xf89d, /*0x38-0x3f*/
6053 0xf89e, 0xf89f, 0xf8a0, 0xf8a1, 0xf8a2, 0xf8a3, 0xf8a4, 0xf8a5, /*0x40-0x47*/
6054 0xf8a6, 0xf8a7, 0xf8a8, 0xf8a9, 0xf8aa, 0xf8ab, 0xf8ac, 0xf8ad, /*0x48-0x4f*/
6055 0xf8ae, 0xf8af, 0xf8b0, 0xf8b1, 0xf8b2, 0xf8b3, 0xf8b4, 0xf8b5, /*0x50-0x57*/
6056 0xf8b6, 0xf8b7, 0xf8b8, 0xf8b9, 0xf8ba, 0xf8bb, 0xf8bc, 0xf8bd, /*0x58-0x5f*/
6057 0xf8be, 0xf8bf, 0xf8c0, 0xf8c1, 0xf8c2, 0xf8c3, 0xf8c4, 0xf8c5, /*0x60-0x67*/
6058 0xf8c6, 0xf8c7, 0xf8c8, 0xf8c9, 0xf8ca, 0xf8cb, 0xf8cc, 0xf8cd, /*0x68-0x6f*/
6059 0xf8ce, 0xf8cf, 0xf8d0, 0xf8d1, 0xf8d2, 0xf8d3, 0xf8d4, 0xf8d5, /*0x70-0x77*/
6060 0xf8d6, 0xf8d7, 0xf8d8, 0xf8d9, 0xf8da, 0xf8db, 0xf8dc, 0xf8dd, /*0x78-0x7f*/
6061 0xf8de, 0xf8df, 0xf8e0, 0xf8e1, 0xf8e2, 0xf8e3, 0xf8e4, 0xf8e5, /*0x80-0x87*/
6062 0xf8e6, 0xf8e7, 0xf8e8, 0xf8e9, 0xf8ea, 0xf8eb, 0xf8ec, 0xf8ed, /*0x88-0x8f*/
6063 0xf8ee, 0xf8ef, 0xf8f0, 0xf8f1, 0xf8f2, 0xf8f3, 0xf8f4, 0xf8f5, /*0x90-0x97*/
6064 0xf8f6, 0xf8f7, 0xf8f8, 0xf8f9, 0xf8fa, 0xf8fb, 0xf8fc, 0xf8fd, /*0x98-0x9f*/
6065 0xf8fe, 0xf8ff, 0xf900, 0xf901, 0xf902, 0xf903, 0xf904, 0xf905, /*0xa0-0xaf*/
6066 0xf906, 0xf907, 0xf908, 0xf909, 0xf90a, 0xf90b, 0xf90c, 0xf90d, /*0xab-0xaf*/
6067 0xf90e, 0xf90f, 0xf910, 0xf911, 0xf912, 0xf913, 0xf914, 0xf915, /*0xb0-0xbf*/
6068 0xf916, 0xf917, 0xf918, 0xf919, 0xf91a, 0xf91b, 0xf91c, 0xf91d, /*0xc0-0xc7*/
6069 0xf91e, 0xf91f, 0xf920, 0xf921, 0xf922, 0xf923, 0xf924, 0xf925, /*0xc8-0xcf*/
6070 0xf926, 0xf927, 0xf928, 0xf929, 0xf92a, 0xf92b, 0xf92c, 0xf92d, /*0xd0-0xd7*/
6071 0xf92e, 0xf92f, 0xf930, 0xf931, 0xf932, 0xf933, 0xf934, 0xf935, /*0xd8-0xdf*/
6072 0xf936, 0xf937, 0xf938, 0xf939, 0xf93a, 0xf93b, 0xf93c, 0xf93d, /*0xe0-0xef*/
6073 0xf93e, 0xf93f, 0xf940, 0xf941, 0xf942, 0xf943, 0xf944, 0xf945, /*0xf0-0xff*/
6074 0xf946, 0xf947, 0xf948, 0xf949, 0xf94a, 0xf94b, 0xf94c, 0xf94d, /*0xc0-0xc7*/
6075 0xf94e, 0xf94f, 0xf950, 0xf951, 0xf952, 0xf953, 0xf954, 0xf955, /*0xc8-0xcf*/
6076 0xf956, 0xf957, 0xf958, 0xf959, 0xf95a, 0xf95b, 0xf95c, 0xf95d, /*0xd0-0xd7*/
6077 /* 0x9e00 */
6078 0xf99e, 0xf99f, 0xf9a0, 0xf9a1, 0xf9a2, 0xf9a3, 0xf9a4, 0xf9a5, /*0x00-0x07*/
6079 0xf9a6, 0xf9a7, 0xf9a8, 0xf9a9, 0xf9aa, 0xf9ab, 0xf9ac, 0xf9ad, /*0x08-0x0f*/
6080 0xf9ae, 0xf9af, 0xf9b0, 0xf9b1, 0xf9b2, 0xf9b3, 0xf9b4, 0xf9b5, /*0x10-0x17*/
6081 0xf9b6, 0xf9b7, 0xf9b8, 0xf9b9, 0xf9ba, 0xf9bb, 0xf9bc, 0xf9bd, /*0x18-0x1f*/
6082 0xf9be, 0xf9bf, 0xf9c0, 0xf9c1, 0xf9c2, 0xf9c3, 0xf9c4, 0xf9c5, /*0x20-0x27*/
6083 0xf9c6, 0xf9c7, 0xf9c8, 0xf9c9, 0xf9ca, 0xf9cb, 0xf9cc, 0xf9cd, /*0x28-0x2f*/
6084 0xf9ce, 0xf9cf, 0xf9d0, 0xf9d1, 0xf9d2, 0xf9d3, 0xf9d4, 0xf9d5, /*0x30-0x37*/
6085 0xf9d6, 0xf9d7, 0xf9d8, 0xf9d9, 0xf9da, 0xf9db, 0xf9dc, 0xf9dd, /*0x38-0x3f*/
6086 0xf9de, 0xf9df, 0xf9e0, 0xf9e1, 0xf9e2, 0xf9e3, 0xf9e4, 0xf9e5, /*0x40-0x4f*/
6087 0xf9e6, 0xf9e7, 0xf9e8, 0xf9e9, 0xf9ea, 0xf9eb, 0xf9ec, 0xf9ed, /*0x48-0x4f*/
6088 0xf9ee, 0xf9ef, 0xf9f0, 0xf9f1, 0xf9f2, 0xf9f3, 0xf9f4, 0xf9f5, /*0x50-0x57*/

```

```

6089 0xf7bd, 0xfb6a, 0xf0cb, 0xf0cc, 0xf0cd, 0xfb6b, 0xf0ce, 0xfb6c, /*0x58-0x5f*/
6090 0xfb6d, 0xfb6e, 0xfb6f, 0xf0cf, 0xbad7, 0xfb70, 0xf0d0, 0xf0d1, /*0x60-0x67*/
6091 0xf0d2, 0xf0d3, 0xf0d4, 0xf0d5, 0xf0d6, 0xf0d8, 0xfb71, 0xfb72, /*0x68-0x6f*/
6092 0xd3a5, 0xf0d7, 0xfb73, 0xf0d9, 0xfb74, 0xfb75, 0xfb76, 0xfb77, /*0x70-0x77*/
6093 0xfb78, 0xfb79, 0xfb7a, 0xfb7b, 0xfb7c, 0xfb7d, 0xf5ba, 0xc2b9, /*0x78-0x7f*/
6094 0xfb7e, 0xfb80, 0xf7e4, 0xfb81, 0xfb82, 0xfb83, 0xfb84, 0xf7e5, /*0x80-0x87*/
6095 0xf7e6, 0xfb85, 0xfb86, 0xf7e7, 0xfb87, 0xfb88, 0xfb89, 0xfb8a, /*0x88-0x8f*/
6096 0xfb8b, 0xfb8c, 0xf7e8, 0xc2b4, 0xfb8d, 0xfb8e, 0xfb8f, 0xfb90, /*0x90-0x97*/
6097 0xfb91, 0xfb92, 0xfb93, 0xfb94, 0xfb95, 0xf7ea, 0xfb96, 0xf7eb, /*0x98-0x9f*/
6098 0xfb97, 0xfb98, 0xfb99, 0xfb9a, 0xfb9b, 0xfb9c, 0xc2f3, 0xfb9d, /*0xa0-0xa7*/
6099 0xfb9e, 0xfb9f, 0xfba0, 0xfc40, 0xfc41, 0xfc42, 0xfc43, 0xfc44, /*0xa8-0xaf*/
6100 0xfc45, 0xfc46, 0xfc47, 0xfc48, 0xfc49, 0xfc4a, 0xfc4b, /*0xb0-0xbf*/
6101 0xf4ef, 0xfc4c, 0xfc4d, 0xc2e9, 0xfc4e, 0xf7e1, 0xf7e2, 0xfc4f, /*0xb8-0xbf*/
6102 0xfc50, 0xfc51, 0xfc52, 0xfc53, 0xbbc6, 0xfc54, 0xfc55, 0xfc56, /*0xc0-0xc7*/
6103 0xfc57, 0xd9e4, 0xfc58, 0xfc59, 0xfc5a, 0xcaf2, 0xc0e8, 0xf0a4, /*0xc8-0xcf*/
6104 0xfc5b, 0xbada, 0xfc5c, 0xfc5d, 0xc7ad, 0xfc5e, 0xfc5f, 0xfc60, /*0xd0-0xd7*/
6105 0xc4ac, 0xfc61, 0xfc62, 0xf7ec, 0xf7ed, 0xf7ee, 0xfc63, 0xf7f0, /*0xd8-0xdf*/
6106 0xf7ef, 0xfc64, 0xf7f1, 0xfc65, 0xfc66, 0xf7f4, 0xfc67, 0xf7f3, /*0xe0-0xef*/
6107 0xfc68, 0xf7f2, 0xf7f5, 0xfc69, 0xfc6a, 0xfc6b, 0xfc6c, 0xf7f6, /*0xe8-0xef*/
6108 0xfc6d, 0xfc6e, 0xfc6f, 0xfc70, 0xfc71, 0xfc72, 0xfc73, 0xfc74, /*0xf0-0xff*/
6109 0xfc75, 0xede9, 0xfc76, 0xede8, 0xede9, 0xede8, 0xfc77, 0xfb6c, 0xfc78, /*0xf8-0xff*/
6110 /* 0x9f00 */
6111 0xfc79, 0xfc7a, 0xfc7b, 0xfc7c, 0xfc7d, 0xfc7e, 0xfc80, 0xfc81, /*0x00-0x07*/
6112 0xfc82, 0xfc83, 0xfc84, 0xf6bd, 0xfc85, 0xf6be, 0xb6a6, 0xfc86, /*0x08-0x0f*/
6113 0xd8be, 0xfc87, 0xfc88, 0xb9c4, 0xfc89, 0xfc8a, 0xfc8b, 0xd8bb, /*0x10-0x17*/
6114 0xfc8c, 0xdcbl, 0xfc8d, 0xfc8e, 0xfc8f, 0xfc90, 0xfc91, 0xfc92, /*0x18-0x1f*/
6115 0xcaf3, 0xfc93, 0xf7f7, 0xfc94, 0xfc95, 0xfc96, 0xfc97, 0xfc98, /*0x20-0x27*/
6116 0xfc99, 0xfc9a, 0xfc9b, 0xfc9c, 0xf7f8, 0xfc9d, 0xfc9e, 0xf7f9, /*0x28-0x2f*/
6117 0xfc9f, 0xfca0, 0xfd40, 0xfd41, 0xfd42, 0xfd43, 0xfd44, 0xf7fb, /*0x30-0x37*/
6118 0xfd45, 0xf7fa, 0xfd46, 0xb1c7, 0xfd47, 0xf7fc, 0xf7fd, 0xfd48, /*0x38-0x3f*/
6119 0xfd49, 0xfd4a, 0xfd4b, 0xfd4c, 0xf7fe, 0xfd4d, 0xfd4e, 0xfd4f, /*0x40-0x47*/
6120 0xfd50, 0xfd51, 0xfd52, 0xfd53, 0xfd54, 0xfd55, 0xfd56, 0xfd57, /*0x48-0x4f*/
6121 0xc6eb, 0xecb4, 0xfd58, 0xfd59, 0xfd5a, 0xfd5b, 0xfd5c, 0xfd5d, /*0x50-0x57*/
6122 0xfd5e, 0xfd5f, 0xfd60, 0xfd61, 0xfd62, 0xfd63, 0xfd64, 0xfd65, /*0x58-0x5f*/
6123 0xfd66, 0xfd67, 0xfd68, 0xfd69, 0xfd6a, 0xfd6b, 0xfd6c, 0xfd6d, /*0x60-0x67*/
6124 0xfd6e, 0xfd6f, 0xfd70, 0xfd71, 0xfd72, 0xfd73, 0xfd74, 0xfd75, /*0x68-0x6f*/
6125 0xfd76, 0xfd77, 0xfd78, 0xfd79, 0xfd7a, 0xfd7b, 0xfd7c, 0xfd7d, /*0x70-0x77*/
6126 0xfd7e, 0xfd80, 0xfd81, 0xfd82, 0xfd83, 0xfd84, 0xfd85, 0xb3dd, /*0x78-0x7f*/
6127 0xf6b3, 0xf6b4, 0xf6b7, 0xf6b8, 0xc1e4, 0xf6b5, 0xf6b6, 0xf6b7, /*0x80-0x87*/
6128 0xf6b8, 0xf6b9, 0xf6ba, 0xc8a3, 0xf6bb, 0xfd88, 0xfd89, 0xfd8a, /*0x88-0x8f*/
6129 0xfd8b, 0xfd8c, 0xfd8d, 0xfd8e, 0xfd8f, 0xfd90, 0xfd91, 0xfd92, /*0x90-0x97*/
6130 0xfd93, 0xc1fa, 0xb9a8, 0xede8, 0xfd94, 0xfd95, 0xfd96, 0xb9ea, /*0x98-0x9f*/
6131 0xd9df, 0xfd97, 0xfd98, 0xfd99, 0xfd9a, 0xfd9b, 0x0000, 0x0000, /*0xa0-0xa7*/
6132 };
6133 static const unsigned short cp936ext_pagelf2f[32] = {
6134 0x0000, 0xfd9d, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0x78-0x7f*/
6135 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0x80-0x87*/
6136 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0x88-0x8f*/
6137 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xfd9e, 0x0000, 0x0000, /*0x90-0x97*/
6138 };
6139 static const unsigned short cp936ext_pagelf3c[24] = {
6140 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xfd9f, /*0xe0-0xe7*/
6141 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0xe8-0xef*/
6142 0x0000, 0xfda0, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0xf0-0xff*/
6143 };
6144 static const unsigned short cp936ext_pagelf41[40] = {
6145 0x0000, 0x0000, 0x0000, 0x0000, 0xfe40, 0xfe41, 0xfe42, 0xfe43, /*0x08-0x0f*/
6146 0x0000, 0xfe44, 0x0000, 0xfe45, 0xfe46, 0x0000, 0x0000, 0x0000, /*0x10-0x17*/
6147 0xfe47, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xfe48, /*0x18-0x1f*/
6148 0xfe49, 0xfe4a, 0x0000, 0xfe4b, 0xfe4c, 0x0000, 0x0000, 0xfe4d, /*0x20-0x27*/
6149 0xfe4e, 0xfe4f, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, /*0x28-0x2f*/
6150 };
6151 static const unsigned short cp936ext_pagelfc6[64] = {
6152 0xa955, 0xa6f2, 0x0000, 0xa6f4, 0xa6f5, 0xa6e0, 0xa6e1, 0xa6f0, /*0x30-0x37*/
6153 0xa6f1, 0xa6e2, 0xa6e3, 0xa6ee, 0xa6ef, 0xa6e6, 0xa6e7, 0xa6e4, /*0x38-0x3f*/
6154 0xa6e5, 0xa6e8, 0xa6e9, 0xa6ea, 0xa6eb, 0x0000, 0x0000, /*0x40-0x47*/
6155 0x0000, 0xa968, 0xa969, 0xa96a, 0xa96b, 0xa96c, 0xa96d, 0xa96e, /*0x48-0x4f*/
6156 0xa96f, 0xa970, 0xa971, 0x0000, 0xa972, 0xa973, 0xa974, 0xa975, /*0x50-0x57*/
6157 0x0000, 0xa976, 0xa977, 0xa978, 0xa979, 0xa97a, 0xa97b, 0xa97c, /*0x58-0x5f*/
6158 0xa97d, 0xa97e, 0xa980, 0xa981, 0xa982, 0xa983, 0xa984, 0x0000, /*0x60-0x67*/
6159 0xa985, 0xa986, 0xa987, 0xa988, 0x0000, 0x0000, 0x0000, 0x0000, /*0x68-0x6f*/
6160 };
6161 static const unsigned short cp936ext_pagelfe0[96] = {
6162 0x0000, 0xa3a1, 0xa3a2, 0xa3a3, 0xa1e7, 0xa3a5, 0xa3a6, 0xa3a7, /*0x00-0x07*/
6163 0xa3a8, 0xa3a9, 0xa3aa, 0xa3ab, 0xa3ac, 0xa3ad, 0xa3ae, 0xa3af, /*0x08-0x0f*/
6164 0xa3b0, 0xa3b1, 0xa3b2, 0xa3b3, 0xa3b4, 0xa3b5, 0xa3b6, 0xa3b7, /*0x10-0x17*/
6165 0xa3b8, 0xa3b9, 0xa3ba, 0xa3bb, 0xa3bc, 0xa3bd, 0xa3be, 0xa3bf, /*0x18-0x1f*/
6166 0xa3c0, 0xa3c1, 0xa3c2, 0xa3c3, 0xa3c4, 0xa3c5, 0xa3c6, 0xa3c7, /*0x20-0x27*/
6167 0xa3c8, 0xa3c9, 0xa3ca, 0xa3cb, 0xa3cc, 0xa3cd, 0xa3ce, 0xa3cf, /*0x28-0x2f*/
6168 0xa3d0, 0xa3d1, 0xa3d2, 0xa3d3, 0xa3d4, 0xa3d5, 0xa3d6, 0xa3d7, /*0x30-0x37*/
6169 0xa3d8, 0xa3d9, 0xa3da, 0xa3db, 0xa3dc, 0xa3dd, 0xa3de, 0xa3df, /*0x38-0x3f*/
6170 0xa3e0, 0xa3e1, 0xa3e2, 0xa3e3, 0xa3e4, 0xa3e5, 0xa3e6, 0xa3e7, /*0x40-0x47*/
6171 0xa3e8, 0xa3e9, 0xa3ea, 0xa3eb, 0xa3ec, 0xa3ed, 0xa3ee, 0xa3ef, /*0x48-0x4f*/
6172 0xa3f0, 0xa3f1, 0xa3f2, 0xa3f3, 0xa3f4, 0xa3f5, 0xa3f6, 0xa3f7, /*0x50-0x57*/
6173 0xa3f8, 0xa3f9, 0xa3fa, 0xa3fb, 0xa3fc, 0xa3fd, 0xalab, 0x0000, /*0x58-0x5f*/
6174 };
6175 static const unsigned short cp936ext_pagelffc[8] = {

```



```

6176 0xale9, 0xalea, 0xa956, 0xa3fe, 0xa957, 0xa3a4, 0x0000, 0x0000, /*0xe0-0xe7*/
6177 };
6178
6179 static int
6180 cp936ext_wctomb (conv_t conv, unsigned char *, ucs4_t wc, int n)
6181 {
6182     if (n >= 2) {
6183         unsigned short c = 0;
6184         if (wc >= 0x00a0 && wc < 0x0170)
6185             c = cp936ext_page0014[wc-0x00a0];
6186         else if (wc >= 0x01c8 && wc < 0x01e0)
6187             c = cp936ext_page0039[wc-0x01c8];
6188         else if (wc >= 0x0250 && wc < 0x0268)
6189             c = cp936ext_page004a[wc-0x0250];
6190         else if (wc >= 0x02c0 && wc < 0x02e0)
6191             c = cp936ext_page0058[wc-0x02c0];
6192         else if (wc >= 0x0390 && wc < 0x03d0)
6193             c = cp936ext_page0072[wc-0x0390];
6194         else if (wc >= 0x0400 && wc < 0x0458)
6195             c = cp936ext_page0080[wc-0x0400];
6196         else if (wc >= 0x2010 && wc < 0x2040)
6197             c = cp936ext_page0402[wc-0x2010];
6198         else if (wc >= 0x2100 && wc < 0x21a0)
6199             c = cp936ext_page0420[wc-0x2100];
6200         else if (wc >= 0x2208 && wc < 0x22c0)
6201             c = cp936ext_page0441[wc-0x2208];
6202         else if (wc == 0x2312)
6203             c = 0xald0;
6204         else if (wc >= 0x2460 && wc < 0x24a0)
6205             c = cp936ext_page048c[wc-0x2460];
6206         else if (wc >= 0x2500 && wc < 0x25e8)
6207             c = cp936ext_page04a0[wc-0x2500];
6208         else if (wc >= 0x2600 && wc < 0x2648)
6209             c = cp936ext_page04c0[wc-0x2600];
6210         else if (wc >= 0x3000 && wc < 0x3130)
6211             c = cp936ext_page0600[wc-0x3000];
6212         else if (wc >= 0x3220 && wc < 0x3238)
6213             c = cp936ext_page0644[wc-0x3220];
6214         else if (wc == 0x32a3)
6215             c = 0xa949;
6216         else if (wc >= 0x3388 && wc < 0x33d8)
6217             c = cp936ext_page0671[wc-0x3388];
6218         else if (wc >= 0x4e00 && wc < 0x9fa8)
6219             c = cp936ext_page09c0[wc-0x4e00];
6220         else if (wc == 0xf92c)
6221             c = 0xfd9c;
6222         else if (wc >= 0xf978 && wc < 0xf998)
6223             c = cp936ext_page1f2f[wc-0xf978];
6224         else if (wc >= 0xf9e0 && wc < 0xf9f8)
6225             c = cp936ext_page1f3c[wc-0xf9e0];
6226         else if (wc >= 0xfa08 && wc < 0xfa30)
6227             c = cp936ext_page1f41[wc-0xfa08];
6228         else if (wc >= 0xfe30 && wc < 0xfe70)
6229             c = cp936ext_page1fc6[wc-0xfe30];
6230         else if (wc >= 0xff00 && wc < 0xff60)
6231             c = cp936ext_page1fe0[wc-0xff00];
6232         else if (wc >= 0xffe0 && wc < 0xffe8)
6233             c = cp936ext_page1ffc[wc-0xffe0];
6234         if (c != 0) {
6235             r[0] = (c >> 8); r[1] = (c & 0xff);
6236             return 2;
6237         }
6238         return RET_ILSEQ;
6239     }
6240     return RET_TOOSMALL;
6241 }
6242 #endif /* NEED_TOMB */
6243
6244 #endif /* CP936 */
6245
6246 #endif /* __APPLE__ WIN32 */
6247
6248 /*
6249 * End of "$Id$".
6250 */

```

32.208 gb2312.h

```

1 /* $XFree86: xc/lib/X11/lcUniConv/gb2312.h,v 1.5 2003/05/27 22:26:29 tsi Exp $ */
2
3 /*
4 * GB2312.1980-0
5 */
6 #ifdef NEED_TOWC
7 static const unsigned short gb2312_2uni_page21[831] = {

```

```
8 /* 0x21 */
9 0x3000, 0x3001, 0x3002, 0x30fb, 0x02c9, 0x02c7, 0x00a8, 0x3003,
10 0x3005, 0x2015, 0xff5e, 0x2016, 0x2026, 0x2018, 0x2019, 0x201c,
11 0x201d, 0x3014, 0x3015, 0x3008, 0x3009, 0x300a, 0x300b, 0x300c,
12 0x300d, 0x300e, 0x300f, 0x3016, 0x3017, 0x3010, 0x3011, 0x00b1,
13 0x00d7, 0x00f7, 0x2236, 0x2227, 0x2228, 0x2211, 0x220f, 0x222a,
14 0x2229, 0x2208, 0x2237, 0x221a, 0x22a5, 0x2225, 0x2220, 0x2312,
15 0x2299, 0x222b, 0x222e, 0x2261, 0x224c, 0x2248, 0x223d, 0x221d,
16 0x2260, 0x226e, 0x226f, 0x2264, 0x2265, 0x221e, 0x2235, 0x2234,
17 0x2642, 0x2640, 0x00b0, 0x2032, 0x2033, 0x2103, 0xff04, 0x00a4,
18 0xfe0, 0xfe1, 0x2030, 0x00a7, 0x2116, 0x2606, 0x2605, 0x25cb,
19 0x25cf, 0x25ce, 0x25c7, 0x25c6, 0x25a1, 0x25a0, 0x25b3, 0x25b2,
20 0x203b, 0x2192, 0x2190, 0x2191, 0x2193, 0x3013,
21 /* 0x22 */
22 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
23 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
24 0x2488, 0x2489, 0x248a, 0x248b, 0x248c, 0x248d, 0x248e, 0x248f,
25 0x2490, 0x2491, 0x2492, 0x2493, 0x2494, 0x2495, 0x2496, 0x2497,
26 0x2498, 0x2499, 0x249a, 0x249b, 0x2474, 0x2475, 0x2476, 0x2477,
27 0x2478, 0x2479, 0x247a, 0x247b, 0x247c, 0x247d, 0x247e, 0x247f,
28 0x2480, 0x2481, 0x2482, 0x2483, 0x2484, 0x2485, 0x2486, 0x2487,
29 0x2460, 0x2461, 0x2462, 0x2463, 0x2464, 0x2465, 0x2466, 0x2467,
30 0x2468, 0x2469, 0xffff, 0xffff, 0x3220, 0x3221, 0x3222, 0x3223,
31 0x3224, 0x3225, 0x3226, 0x3227, 0x3228, 0x3229, 0xffff, 0xffff,
32 0x2160, 0x2161, 0x2162, 0x2163, 0x2164, 0x2165, 0x2166, 0x2167,
33 0x2168, 0x2169, 0x216a, 0x216b, 0xffff, 0xffff,
34 /* 0x23 */
35 0xff01, 0xff02, 0xff03, 0xffe5, 0xff05, 0xff06, 0xff07, 0xff08,
36 0xff09, 0xff0a, 0xff0b, 0xff0c, 0xff0d, 0xff0e, 0xff0f, 0xff10,
37 0xff11, 0xff12, 0xff13, 0xff14, 0xff15, 0xff16, 0xff17, 0xff18,
38 0xff19, 0xff1a, 0xff1b, 0xff1c, 0xff1d, 0xff1e, 0xff1f, 0xff20,
39 0xff21, 0xff22, 0xff23, 0xff24, 0xff25, 0xff26, 0xff27, 0xff28,
40 0xff29, 0xff2a, 0xff2b, 0xff2c, 0xff2d, 0xff2e, 0xff2f, 0xff30,
41 0xff31, 0xff32, 0xff33, 0xff34, 0xff35, 0xff36, 0xff37, 0xff38,
42 0xff39, 0xff3a, 0xff3b, 0xff3c, 0xff3d, 0xff3e, 0xff3f, 0xff40,
43 0xff41, 0xff42, 0xff43, 0xff44, 0xff45, 0xff46, 0xff47, 0xff48,
44 0xff49, 0xff4a, 0xff4b, 0xff4c, 0xff4d, 0xff4e, 0xff4f, 0xff50,
45 0xff51, 0xff52, 0xff53, 0xff54, 0xff55, 0xff56, 0xff57, 0xff58,
46 0xff59, 0xff5a, 0xff5b, 0xff5c, 0xff5d, 0xffe3,
47 /* 0x24 */
48 0x3041, 0x3042, 0x3043, 0x3044, 0x3045, 0x3046, 0x3047, 0x3048,
49 0x3049, 0x304a, 0x304b, 0x304c, 0x304d, 0x304e, 0x304f, 0x3050,
50 0x3051, 0x3052, 0x3053, 0x3054, 0x3055, 0x3056, 0x3057, 0x3058,
51 0x3059, 0x305a, 0x305b, 0x305c, 0x305d, 0x305e, 0x305f, 0x3060,
52 0x3061, 0x3062, 0x3063, 0x3064, 0x3065, 0x3066, 0x3067, 0x3068,
53 0x3069, 0x306a, 0x306b, 0x306c, 0x306d, 0x306e, 0x306f, 0x3070,
54 0x3071, 0x3072, 0x3073, 0x3074, 0x3075, 0x3076, 0x3077, 0x3078,
55 0x3079, 0x307a, 0x307b, 0x307c, 0x307d, 0x307e, 0x307f, 0x3080,
56 0x3081, 0x3082, 0x3083, 0x3084, 0x3085, 0x3086, 0x3087, 0x3088,
57 0x3089, 0x308a, 0x308b, 0x308c, 0x308d, 0x308e, 0x308f, 0x3090,
58 0x3091, 0x3092, 0x3093, 0xffff, 0xffff, 0xffff, 0xffff,
59 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
60 /* 0x25 */
61 0x30a1, 0x30a2, 0x30a3, 0x30a4, 0x30a5, 0x30a6, 0x30a7, 0x30a8,
62 0x30a9, 0x30aa, 0x30ab, 0x30ac, 0x30ad, 0x30ae, 0x30af, 0x30b0,
63 0x30b1, 0x30b2, 0x30b3, 0x30b4, 0x30b5, 0x30b6, 0x30b7, 0x30b8,
64 0x30b9, 0x30ba, 0x30bb, 0x30bc, 0x30bd, 0x30be, 0x30bf, 0x30c0,
65 0x30c1, 0x30c2, 0x30c3, 0x30c4, 0x30c5, 0x30c6, 0x30c7, 0x30c8,
66 0x30c9, 0x30ca, 0x30cb, 0x30cc, 0x30cd, 0x30ce, 0x30cf, 0x30d0,
67 0x30d1, 0x30d2, 0x30d3, 0x30d4, 0x30d5, 0x30d6, 0x30d7, 0x30d8,
68 0x30d9, 0x30da, 0x30db, 0x30dc, 0x30dd, 0x30de, 0x30df, 0x30e0,
69 0x30e1, 0x30e2, 0x30e3, 0x30e4, 0x30e5, 0x30e6, 0x30e7, 0x30e8,
70 0x30e9, 0x30ea, 0x30eb, 0x30ec, 0x30ed, 0x30ee, 0x30ef, 0x30f0,
71 0x30f1, 0x30f2, 0x30f3, 0x30f4, 0x30f5, 0x30f6, 0xffff, 0xffff,
72 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
73 /* 0x26 */
74 0x0391, 0x0392, 0x0393, 0x0394, 0x0395, 0x0396, 0x0397, 0x0398,
75 0x0399, 0x039a, 0x039b, 0x039c, 0x039d, 0x039e, 0x039f, 0x03a0,
76 0x03a1, 0x03a3, 0x03a4, 0x03a5, 0x03a6, 0x03a7, 0x03a8, 0x03a9,
77 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
78 0x03b1, 0x03b2, 0x03b3, 0x03b4, 0x03b5, 0x03b6, 0x03b7, 0x03b8,
79 0x03b9, 0x03ba, 0x03bb, 0x03bc, 0x03bd, 0x03be, 0x03bf, 0x03c0,
80 0x03c1, 0x03c3, 0x03c4, 0x03c5, 0x03c6, 0x03c7, 0x03c8, 0x03c9,
81 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
82 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
83 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
84 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
85 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
86 /* 0x27 */
87 0x0410, 0x0411, 0x0412, 0x0413, 0x0414, 0x0415, 0x0401, 0x0416,
88 0x0417, 0x0418, 0x0419, 0x041a, 0x041b, 0x041c, 0x041d, 0x041e,
89 0x041f, 0x0420, 0x0421, 0x0422, 0x0423, 0x0424, 0x0425, 0x0426,
90 0x0427, 0x0428, 0x0429, 0x042a, 0x042b, 0x042c, 0x042d, 0x042e,
91 0x042f, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
92 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
93 0x0430, 0x0431, 0x0432, 0x0433, 0x0434, 0x0435, 0x0451, 0x0436,
94 0x0437, 0x0438, 0x0439, 0x043a, 0x043b, 0x043c, 0x043d, 0x043e,
```

```
95 0x043f, 0x0440, 0x0441, 0x0442, 0x0443, 0x0444, 0x0445, 0x0446,
96 0x0447, 0x0448, 0x0449, 0x044a, 0x044b, 0x044c, 0x044d, 0x044e,
97 0x044f, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
98 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
99 /* 0x28 */
100 0x0101, 0x00e1, 0x01ce, 0x00e0, 0x0113, 0x00e9, 0x011b, 0x00e8,
101 0x012b, 0x00ed, 0x01d0, 0x00ec, 0x014d, 0x00f3, 0x01d2, 0x00f2,
102 0x016b, 0x00fa, 0x01d4, 0x00f9, 0x01d6, 0x01d8, 0x01da, 0x01dc,
103 0x00fc, 0x00ea, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
104 0xffff, 0xffff, 0xffff, 0xffff, 0x3105, 0x3106, 0x3107, 0x3108,
105 0x3109, 0x310a, 0x310b, 0x310c, 0x310d, 0x310e, 0x310f, 0x3110,
106 0x3111, 0x3112, 0x3113, 0x3114, 0x3115, 0x3116, 0x3117, 0x3118,
107 0x3119, 0x311a, 0x311b, 0x311c, 0x311d, 0x311e, 0x311f, 0x3120,
108 0x3121, 0x3122, 0x3123, 0x3124, 0x3125, 0x3126, 0x3127, 0x3128,
109 0x3129, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
110 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
111 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
112 /* 0x29 */
113 0xffff, 0xffff, 0xffff, 0x2500, 0x2501, 0x2502, 0x2503, 0x2504,
114 0x2505, 0x2506, 0x2507, 0x2508, 0x2509, 0x250a, 0x250b, 0x250c,
115 0x250d, 0x250e, 0x250f, 0x2510, 0x2511, 0x2512, 0x2513, 0x2514,
116 0x2515, 0x2516, 0x2517, 0x2518, 0x2519, 0x251a, 0x251b, 0x251c,
117 0x251d, 0x251e, 0x251f, 0x2520, 0x2521, 0x2522, 0x2523, 0x2524,
118 0x2525, 0x2526, 0x2527, 0x2528, 0x2529, 0x252a, 0x252b, 0x252c,
119 0x252d, 0x252e, 0x252f, 0x2530, 0x2531, 0x2532, 0x2533, 0x2534,
120 0x2535, 0x2536, 0x2537, 0x2538, 0x2539, 0x253a, 0x253b, 0x253c,
121 0x253d, 0x253e, 0x253f, 0x2540, 0x2541, 0x2542, 0x2543, 0x2544,
122 0x2545, 0x2546, 0x2547, 0x2548, 0x2549, 0x254a, 0x254b,
123 };
124 static const unsigned short gb2312_2uni_page30[6768] = {
125 /* 0x30 */
126 0x554a, 0x963f, 0x57c3, 0x6328, 0x54ce, 0x5509, 0x54c0, 0x7691,
127 0x764c, 0x853c, 0x77ee, 0x827e, 0x788d, 0x7231, 0x9698, 0x978d,
128 0x6c28, 0x5b89, 0x4ffa, 0x6309, 0x6697, 0x5cb8, 0x80fa, 0x6848,
129 0x80ae, 0x6602, 0x76ce, 0x51f9, 0x6556, 0x71ac, 0x7ff1, 0x8884,
130 0x50b2, 0x5965, 0x61ca, 0x6fb3, 0x82ad, 0x634c, 0x6252, 0x53ed,
131 0x5427, 0x7b06, 0x516b, 0x75a4, 0x5df4, 0x62d4, 0x8dcb, 0x9776,
132 0x628a, 0x8019, 0x575d, 0x9738, 0x7f62, 0x7238, 0x767d, 0x67cf,
133 0x767e, 0x6446, 0x4f70, 0x8d25, 0x62dc, 0x7a17, 0x6591, 0x73de,
134 0x642c, 0x6273, 0x822c, 0x9881, 0x677f, 0x7248, 0x626e, 0x62cc,
135 0x4f34, 0x74e3, 0x534a, 0x529e, 0x7eca, 0x90a6, 0x5e2e, 0x6886,
136 0x699c, 0x8180, 0x7ed1, 0x68d2, 0x78c5, 0x868c, 0x9551, 0x508d,
137 0x8c24, 0x82de, 0x80de, 0x5305, 0x8912, 0x5265,
138 /* 0x31 */
139 0x8584, 0x96f9, 0x4fdd, 0x5821, 0x9971, 0x5b9d, 0x62b1, 0x62a5,
140 0x66b4, 0x8c79, 0x9c8d, 0x7206, 0x676f, 0x7891, 0x60b2, 0x5351,
141 0x5317, 0x8f88, 0x80cc, 0x8d1d, 0x94a1, 0x500d, 0x72c8, 0x5907,
142 0x60eb, 0x7119, 0x88ab, 0x5954, 0x82ef, 0x672c, 0x7b28, 0x5d29,
143 0x7ef7, 0x752d, 0x6cf5, 0x8e66, 0x8ff8, 0x903c, 0x9f3b, 0x6bd4,
144 0x9119, 0x7b14, 0x5f7c, 0x78a7, 0x84d6, 0x853d, 0x6bd5, 0x6bd9,
145 0x6bd6, 0x5e01, 0x5e87, 0x75f9, 0x95ed, 0x655d, 0x5f0a, 0x5fc5,
146 0x8f9f, 0x58c1, 0x81c2, 0x907f, 0x965b, 0x97ad, 0x8fb9, 0x7f16,
147 0x8d2c, 0x6241, 0x4fbf, 0x53d8, 0x535e, 0x8fa8, 0x8fa9, 0x8fab,
148 0x904d, 0x6807, 0x5f6a, 0x8198, 0x8868, 0x9cd6, 0x618b, 0x522b,
149 0x762a, 0x5f6c, 0x658c, 0x6fd2, 0x6ee8, 0x5bbe, 0x6448, 0x5175,
150 0x51b0, 0x67c4, 0x4e19, 0x79c9, 0x997c, 0x70b3,
151 /* 0x32 */
152 0x75c5, 0x5e76, 0x73bb, 0x83e0, 0x64ad, 0x62e8, 0x94b5, 0x6ce2,
153 0x535a, 0x52c3, 0x640f, 0x94c2, 0x7b94, 0x4f2f, 0x5e1b, 0x8236,
154 0x8116, 0x818a, 0x6e24, 0x6cca, 0x9a73, 0x6355, 0x535c, 0x54fa,
155 0x8865, 0x57e0, 0x4e0d, 0x5e03, 0x6b65, 0x7c3f, 0x90e8, 0x6016,
156 0x64e6, 0x731c, 0x88c1, 0x6750, 0x624d, 0x8d22, 0x776c, 0x8e29,
157 0x91c7, 0x5f69, 0x83dc, 0x8521, 0x9910, 0x53c2, 0x8695, 0x6b8b,
158 0x60ed, 0x60e8, 0x707f, 0x82cd, 0x8231, 0x4ed3, 0x6ca7, 0x85cf,
159 0x64cd, 0x7cd9, 0x69fd, 0x66f9, 0x8349, 0x5395, 0x7b56, 0x4fa7,
160 0x518c, 0x6d4b, 0x5c42, 0x8e6d, 0x63d2, 0x53c9, 0x832c, 0x8336,
161 0x67e5, 0x78b4, 0x643d, 0x5bdf, 0x5c94, 0x5dee, 0x8be7, 0x62c6,
162 0x67f4, 0x8c7a, 0x6400, 0x63ba, 0x8749, 0x998b, 0x8c17, 0x7f20,
163 0x94f2, 0x4ea7, 0x9610, 0x98a4, 0x660c, 0x7316,
164 /* 0x33 */
165 0x573a, 0x5c1d, 0x5e38, 0x957f, 0x507f, 0x80a0, 0x5382, 0x655e,
166 0x7545, 0x5531, 0x5021, 0x8d85, 0x6284, 0x949e, 0x671d, 0x5632,
167 0x6f6e, 0x5de2, 0x5435, 0x7092, 0x8f66, 0x626f, 0x64a4, 0x63a3,
168 0x5f7b, 0x6ff8, 0x90f4, 0x81e3, 0x8fb0, 0x5c18, 0x6668, 0x5ff1,
169 0x6c89, 0x9648, 0x8d81, 0x886c, 0x6491, 0x79f0, 0x57ce, 0x6a59,
170 0x6210, 0x5448, 0x4e58, 0x7a0b, 0x60e9, 0x6f84, 0x8bda, 0x627f,
171 0x901e, 0x9a8b, 0x79e4, 0x5403, 0x75f4, 0x6301, 0x5319, 0x6c60,
172 0x8fd1, 0x5f1b, 0x9a70, 0x803b, 0x9f7f, 0x4f88, 0x5c3a, 0x8d64,
173 0x7fc5, 0x65a5, 0x70bd, 0x5145, 0x51b2, 0x866b, 0x5d07, 0x5ba0,
174 0x62bd, 0x916c, 0x7574, 0x8e0c, 0x7a20, 0x6101, 0x7b79, 0x4ec7,
175 0x7ef8, 0x7785, 0x4e11, 0x81ed, 0x521d, 0x51fa, 0x6a71, 0x53a8,
176 0x8e87, 0x9504, 0x96cf, 0x6ec1, 0x9664, 0x695a,
177 /* 0x34 */
178 0x7840, 0x50a8, 0x77d7, 0x6410, 0x89e6, 0x5904, 0x63e3, 0x5ddd,
179 0x7a7f, 0x693d, 0x4f20, 0x8239, 0x5598, 0x4e32, 0x75ae, 0x7a97,
180 0x5e62, 0x5e8a, 0x95ef, 0x521b, 0x5439, 0x708a, 0x6376, 0x9524,
181 0x5782, 0x6625, 0x693f, 0x9187, 0x5507, 0x6df3, 0x7eaf, 0x8822,
```

```
182 0x6233, 0x7ef0, 0x75b5, 0x8328, 0x78c1, 0x96cc, 0x8f9e, 0x6148,
183 0x74f7, 0x8bcd, 0x6b64, 0x523a, 0x8d50, 0x6b21, 0x806a, 0x8471,
184 0x56f1, 0x5306, 0x4ece, 0x4e1b, 0x51d1, 0x7c97, 0x918b, 0x7c07,
185 0x4fc3, 0x8e7f, 0x7be1, 0x7a9c, 0x6467, 0x5d14, 0x50ac, 0x8106,
186 0x7601, 0x7cb9, 0x6dec, 0x7fe0, 0x6751, 0x5b58, 0x5bf8, 0x78cb,
187 0x64ae, 0x6413, 0x63aa, 0x632b, 0x9519, 0x642d, 0x8fbe, 0x7b54,
188 0x7629, 0x6253, 0x5927, 0x5446, 0x6b79, 0x50a3, 0x6234, 0x5e26,
189 0x6b86, 0x4ee3, 0x8d37, 0x888b, 0x5f85, 0x902e,
190 /* 0x35 */
191 0x6020, 0x803d, 0x62c5, 0x4e39, 0x5355, 0x90f8, 0x63b8, 0x80c6,
192 0x65e6, 0x6c2e, 0x4f46, 0x60ee, 0x6de1, 0x8bde, 0x5f39, 0x86cb,
193 0x5f53, 0x6321, 0x515a, 0x8361, 0x6863, 0x5200, 0x6363, 0x8e48,
194 0x5012, 0x5c9b, 0x7977, 0x5bfc, 0x5230, 0x7a3b, 0x60bc, 0x9053,
195 0x76d7, 0x5fb7, 0x5f97, 0x7684, 0x8e6c, 0x706f, 0x767b, 0x7b49,
196 0x77aa, 0x51f3, 0x9093, 0x5824, 0x4f4e, 0x6ef4, 0x8fea, 0x654c,
197 0x7b1b, 0x72c4, 0x6da4, 0x7fdf, 0x5ae1, 0x62b5, 0x5e95, 0x5730,
198 0x8482, 0x7b2c, 0x5e1d, 0x5f1f, 0x9012, 0x7f14, 0x98a0, 0x6382,
199 0x6ec7, 0x7898, 0x70b9, 0x5178, 0x975b, 0x57ab, 0x7535, 0x4f43,
200 0x7538, 0x5e97, 0x60e6, 0x5960, 0x6dc0, 0x6bbf, 0x7889, 0x53fc,
201 0x96d5, 0x51cb, 0x5201, 0x6389, 0x540a, 0x9493, 0x8c03, 0x8dcc,
202 0x7239, 0x789f, 0x8776, 0x8fed, 0x8c0d, 0x53e0,
203 /* 0x36 */
204 0x4e01, 0x76ef, 0x53ee, 0x9489, 0x9876, 0x9f0e, 0x952d, 0x5b9a,
205 0x8ba2, 0x4e22, 0x4e1c, 0x51ac, 0x8463, 0x61c2, 0x52a8, 0x680b,
206 0x4f97, 0x606b, 0x51bb, 0x6d1e, 0x515c, 0x6296, 0x6597, 0x9661,
207 0x8c46, 0x9017, 0x75d8, 0x90fd, 0x7763, 0x6bd2, 0x728a, 0x72ec,
208 0x8bfb, 0x5835, 0x7779, 0x8d4c, 0x675c, 0x9540, 0x809a, 0x5ea6,
209 0x6e21, 0x5992, 0x7aef, 0x77ed, 0x953b, 0x6bb5, 0x65ad, 0x7f0e,
210 0x5806, 0x5151, 0x961f, 0x5bf9, 0x58a9, 0x5428, 0x8e72, 0x6566,
211 0x987f, 0x56e4, 0x949d, 0x76fe, 0x9041, 0x6387, 0x54c6, 0x591a,
212 0x593a, 0x579b, 0x8eb2, 0x6735, 0x8dfa, 0x8235, 0x5241, 0x60f0,
213 0x5815, 0x86fe, 0x5ce8, 0x9e45, 0x4fc4, 0x989d, 0x8bb9, 0x5a25,
214 0x6076, 0x5384, 0x627c, 0x904f, 0x9102, 0x997f, 0x6069, 0x800c,
215 0x513f, 0x8033, 0x5c14, 0x9975, 0x6d31, 0x4e8c,
216 /* 0x37 */
217 0x8d30, 0x53d1, 0x7f5a, 0x7b4f, 0x4f10, 0x4e4f, 0x9600, 0x6cd5,
218 0x73d0, 0x85e9, 0x5e06, 0x756a, 0x7ffb, 0x6a0a, 0x77fe, 0x9492,
219 0x7e41, 0x51e1, 0x70e6, 0x53cd, 0x8fd4, 0x8303, 0x8d29, 0x72af,
220 0x996d, 0x6cdb, 0x574a, 0x82b3, 0x65b9, 0x80aa, 0x623f, 0x9632,
221 0x59a8, 0x4eff, 0x8bbf, 0x7eba, 0x653e, 0x83f2, 0x975e, 0x5561,
222 0x98de, 0x80a5, 0x532a, 0x8bfd, 0x5420, 0x80ba, 0x5e9f, 0x6cb8,
223 0x8d39, 0x82ac, 0x915a, 0x5429, 0x6c1b, 0x5206, 0x7eb7, 0x575f,
224 0x711a, 0x6c7e, 0x7c89, 0x594b, 0x4efd, 0x5fff, 0x6124, 0x7caa,
225 0x4e30, 0x5c01, 0x67ab, 0x8702, 0x5cf0, 0x950b, 0x98ce, 0x75af,
226 0x70fd, 0x9022, 0x51af, 0x7f1d, 0x8bbd, 0x5949, 0x51e4, 0x4f5b,
227 0x5426, 0x592b, 0x6577, 0x80a4, 0x5b75, 0x6276, 0x62c2, 0x8f90,
228 0x5e45, 0x6c1f, 0x7b26, 0x4f0f, 0x4fd8, 0x670d,
229 /* 0x38 */
230 0x6d6e, 0x6daa, 0x798f, 0x88b1, 0x5f17, 0x752b, 0x629a, 0x8f85,
231 0x4fef, 0x91dc, 0x65a7, 0x812f, 0x8151, 0x5e9c, 0x8150, 0x8d74,
232 0x526f, 0x8986, 0x8d4b, 0x590d, 0x5085, 0x4ed8, 0x961c, 0x7236,
233 0x8179, 0x8d1f, 0x5bcc, 0x8ba3, 0x9644, 0x5987, 0x7f1a, 0x5490,
234 0x5676, 0x560e, 0x8be5, 0x6539, 0x6982, 0x9499, 0x76d6, 0x6e89,
235 0x5e72, 0x7518, 0x6746, 0x67d1, 0x7aff, 0x809d, 0x8d76, 0x611f,
236 0x79c6, 0x6562, 0x8d63, 0x5188, 0x521a, 0x94a2, 0x7f38, 0x809b,
237 0x7eb2, 0x5c97, 0x6e2f, 0x6760, 0x7bd9, 0x768b, 0x9ad8, 0x818f,
238 0x7f94, 0x7cd5, 0x641e, 0x9550, 0x7a3f, 0x544a, 0x54e5, 0x6b4c,
239 0x6401, 0x6208, 0x9e3d, 0x80f3, 0x7599, 0x5272, 0x9769, 0x845b,
240 0x683c, 0x86e4, 0x9601, 0x9694, 0x94ec, 0x4e2a, 0x5404, 0x7ed9,
241 0x6839, 0x8ddf, 0x8015, 0x66f4, 0x5e9a, 0x7fb9,
242 /* 0x39 */
243 0x57c2, 0x803f, 0x6897, 0x5de5, 0x653b, 0x529f, 0x606d, 0x9f9a,
244 0x4f9b, 0x8eac, 0x516c, 0x5bab, 0x5f13, 0x5de9, 0x6c5e, 0x62f1,
245 0x8d21, 0x5171, 0x94a9, 0x52fe, 0x6c9f, 0x82df, 0x72d7, 0x57a2,
246 0x6784, 0x8d2d, 0x591f, 0x8f9c, 0x83c7, 0x5495, 0x7b8d, 0x4f30,
247 0x6cbd, 0x5b64, 0x59d1, 0x9f13, 0x53e4, 0x86ca, 0x9aa8, 0x8c37,
248 0x80a1, 0x6545, 0x987e, 0x56fa, 0x96c7, 0x522e, 0x74dc, 0x5250,
249 0x5be1, 0x6302, 0x8902, 0x4e56, 0x62d0, 0x602a, 0x68fa, 0x5173,
250 0x5b98, 0x51a0, 0x89c2, 0x7ba1, 0x9986, 0x7f50, 0x60ef, 0x704c,
251 0x8d2f, 0x5149, 0x5e7f, 0x901b, 0x7470, 0x89c4, 0x572d, 0x7845,
252 0x5f52, 0x9f9f, 0x95fa, 0x8f68, 0x9b3c, 0x8be1, 0x7678, 0x6842,
253 0x67dc, 0x8dea, 0x8d35, 0x523d, 0x8f8a, 0x6eda, 0x68cd, 0x9505,
254 0x90ed, 0x56fd, 0x679c, 0x88f9, 0x8fc7, 0x54c8,
255 /* 0x3a */
256 0x9ab8, 0x5b69, 0x6d77, 0x6c26, 0x4ea5, 0x5bb3, 0x9a87, 0x9163,
257 0x61a8, 0x90af, 0x97e9, 0x542b, 0x6db5, 0x5bd2, 0x51fd, 0x558a,
258 0x7f55, 0x7ff0, 0x64bc, 0x634d, 0x65f1, 0x61be, 0x608d, 0x710a,
259 0x6c57, 0x6c49, 0x592f, 0x676d, 0x822a, 0x58d5, 0x568e, 0x8c6a,
260 0x6beb, 0x90dd, 0x597d, 0x8017, 0x53f7, 0x6d69, 0x5475, 0x559d,
261 0x8377, 0x83cf, 0x6838, 0x79be, 0x548c, 0x4f55, 0x5408, 0x76d2,
262 0x8c89, 0x9602, 0x6cb3, 0x6db8, 0x8d6b, 0x8910, 0x9e64, 0x8d3a,
263 0x563f, 0x9ed1, 0x75d5, 0x5f88, 0x72e0, 0x6068, 0x54fc, 0x4ea8,
264 0x6a2a, 0x8861, 0x6052, 0x8f70, 0x54c4, 0x70d8, 0x8679, 0x9e3f,
265 0x6d2a, 0x5b8f, 0x5f18, 0x7ea2, 0x5589, 0x4faf, 0x7334, 0x543c,
266 0x539a, 0x5019, 0x540e, 0x547c, 0x4e4e, 0x5fff, 0x745a, 0x58f6,
267 0x846b, 0x80e1, 0x8774, 0x72d0, 0x7cca, 0x6e56,
268 /* 0x3b */
```

```
269 0x5f27, 0x864e, 0x552c, 0x62a4, 0x4e92, 0x6caa, 0x6237, 0x82b1,
270 0x54d7, 0x534e, 0x733e, 0x6ed1, 0x753b, 0x5212, 0x5316, 0x8bdd,
271 0x69d0, 0x5f8a, 0x6000, 0x6dee, 0x574f, 0x6b22, 0x73af, 0x6853,
272 0x8fd8, 0x7f13, 0x6362, 0x60a3, 0x5524, 0x75ea, 0x8c62, 0x7115,
273 0x6da3, 0x5ba6, 0x5e7b, 0x8352, 0x614c, 0x9ec4, 0x78fa, 0x8757,
274 0x7c27, 0x7687, 0x51f0, 0x60f6, 0x714c, 0x6643, 0x5e4c, 0x604d,
275 0x8c0e, 0x7070, 0x6325, 0x8f89, 0x5fbd, 0x6062, 0x86d4, 0x56de,
276 0x6bc1, 0x6094, 0x6167, 0x5349, 0x60e0, 0x6666, 0x8d3f, 0x79fd,
277 0x4f1a, 0x70e9, 0x6c47, 0x8bb3, 0x8bf2, 0x7ed8, 0x8364, 0x660f,
278 0x5a5a, 0x9b42, 0x6d51, 0x6df7, 0x8c41, 0x6d3b, 0x4f19, 0x706b,
279 0x83b7, 0x6216, 0x60d1, 0x970d, 0x8d27, 0x7978, 0x51fb, 0x573e,
280 0x57fa, 0x673a, 0x7578, 0x7a3d, 0x79ef, 0x7b95,
281 /* 0x3c */
282 0x808c, 0x9965, 0x8ff9, 0x6fc0, 0x8ba5, 0x9e21, 0x59ec, 0x7ee9,
283 0x7f09, 0x5409, 0x6781, 0x68d8, 0x8f91, 0x7c4d, 0x96c6, 0x53ca,
284 0x6025, 0x75be, 0x6c72, 0x5373, 0x5ac9, 0x7ea7, 0x6324, 0x51e0,
285 0x810a, 0x5df1, 0x84df, 0x6280, 0x5180, 0x5b63, 0x4f0e, 0x796d,
286 0x5242, 0x60b8, 0x6d4e, 0x5bc4, 0x5bc2, 0x8ba1, 0x8bb0, 0x6e5e,
287 0x5fcc, 0x9645, 0x5993, 0x7ee7, 0x7eaa, 0x5609, 0x67b7, 0x5939,
288 0x4f73, 0x5bb6, 0x52a0, 0x835a, 0x988a, 0x8d3e, 0x7532, 0x94be,
289 0x5047, 0x7a3c, 0x4ef7, 0x67b6, 0x9a7e, 0x5ac1, 0x6b7c, 0x76d1,
290 0x575a, 0x5c16, 0x7b3a, 0x95f4, 0x714e, 0x517c, 0x80a9, 0x8270,
291 0x5978, 0x7f04, 0x8327, 0x68c0, 0x67ec, 0x78b1, 0x7877, 0x62e3,
292 0x6361, 0x7b80, 0x4fed, 0x526a, 0x51cf, 0x8350, 0x69db, 0x9274,
293 0x8df5, 0x8d31, 0x89c1, 0x952e, 0x7bad, 0x4ef6,
294 /* 0x3d */
295 0x5065, 0x8230, 0x5251, 0x996f, 0x6e10, 0x6e85, 0x6da7, 0x5efa,
296 0x50f5, 0x59dc, 0x5c06, 0x6d46, 0x6c5f, 0x7586, 0x848b, 0x6868,
297 0x5956, 0x8bb2, 0x5320, 0x9171, 0x964d, 0x8549, 0x6912, 0x7901,
298 0x7126, 0x80f6, 0x4ea4, 0x90ca, 0x6d47, 0x9a84, 0x5a07, 0x56bc,
299 0x6405, 0x94f0, 0x77eb, 0x4fa5, 0x811a, 0x72e1, 0x89d2, 0x997a,
300 0x7f34, 0x7ede, 0x527f, 0x6559, 0x9175, 0x8f7f, 0x8f83, 0x53eb,
301 0x7a96, 0x63ed, 0x63a5, 0x7686, 0x79f8, 0x8857, 0x9636, 0x622a,
302 0x52ab, 0x8282, 0x6854, 0x6770, 0x6377, 0x776b, 0x7aed, 0x6d01,
303 0x7ed3, 0x89e3, 0x59d0, 0x6212, 0x85c9, 0x82a5, 0x754c, 0x501f,
304 0x4ecb, 0x75a5, 0x8beb, 0x5c4a, 0x5dfe, 0x7b4b, 0x65a4, 0x91d1,
305 0x4eca, 0x6d25, 0x895f, 0x7d27, 0x9526, 0x4ec5, 0x8c28, 0x8fdb,
306 0x9773, 0x664b, 0x7981, 0x8fd1, 0x70ec, 0x6d78,
307 /* 0x3e */
308 0x5c3d, 0x52b2, 0x8346, 0x5162, 0x830e, 0x775b, 0x6676, 0x9cb8,
309 0x4eac, 0x60ca, 0x7cbe, 0x7cb3, 0x7ecf, 0x4e95, 0x8b66, 0x666f,
310 0x9888, 0x9759, 0x5883, 0x656c, 0x955c, 0x5f84, 0x75c9, 0x9756,
311 0x7adf, 0x7ade, 0x51c0, 0x70af, 0x7a98, 0x63ea, 0x7a76, 0x7ea0,
312 0x7396, 0x97ed, 0x4e45, 0x7078, 0x4e5d, 0x9152, 0x53a9, 0x6551,
313 0x65e7, 0x81fc, 0x8205, 0x548e, 0x5c31, 0x759a, 0x97a0, 0x62d8,
314 0x72d9, 0x75bd, 0x5c45, 0x9a79, 0x83ca, 0x5c40, 0x5480, 0x77e9,
315 0x4e3e, 0x6cae, 0x805a, 0x62d2, 0x636e, 0x5de8, 0x5177, 0x8ddd,
316 0x8e1e, 0x952f, 0x4ff1, 0x53e5, 0x60e7, 0x70ac, 0x5267, 0x6350,
317 0x9e43, 0x5a1f, 0x5026, 0x7737, 0x5377, 0x7ee2, 0x6485, 0x652b,
318 0x6289, 0x6398, 0x5014, 0x7235, 0x89c9, 0x51b3, 0x8bc0, 0x7edd,
319 0x5747, 0x83cc, 0x94a7, 0x519b, 0x541b, 0x5cfb,
320 /* 0x3f */
321 0x4fca, 0x7ae3, 0x6d5a, 0x90e1, 0x9a8f, 0x5580, 0x5496, 0x5361,
322 0x54af, 0x5f00, 0x63e9, 0x6977, 0x51ef, 0x6168, 0x520a, 0x582a,
323 0x52d8, 0x574e, 0x780d, 0x770b, 0x5eb7, 0x6177, 0x7ce0, 0x625b,
324 0x6297, 0x4ea2, 0x7095, 0x8003, 0x62f7, 0x70e4, 0x9760, 0x5777,
325 0x82db, 0x67ef, 0x68f5, 0x78d5, 0x9897, 0x79d1, 0x58f3, 0x54b3,
326 0x53ef, 0x6e34, 0x514b, 0x523b, 0x5ba2, 0x8bfe, 0x80af, 0x5543,
327 0x57a6, 0x6073, 0x5751, 0x542d, 0x7a7a, 0x6050, 0x5b54, 0x63a7,
328 0x62a0, 0x53e3, 0x6263, 0x5bc7, 0x67af, 0x54ed, 0x7a9f, 0x82e6,
329 0x9177, 0x5e93, 0x88e4, 0x5938, 0x57ae, 0x630e, 0x8de8, 0x80ef,
330 0x5757, 0x7b77, 0x4fa9, 0x5feb, 0x5bbd, 0x6b3e, 0x5321, 0x7b50,
331 0x72c2, 0x6846, 0x77ff, 0x7736, 0x65f7, 0x51b5, 0x4e8f, 0x76d4,
332 0x5cbf, 0x7aa5, 0x8475, 0x594e, 0x9b41, 0x5080,
333 /* 0x40 */
334 0x9988, 0x6127, 0x6e83, 0x5764, 0x6606, 0x6346, 0x56f0, 0x62ec,
335 0x6269, 0x5ed3, 0x9614, 0x5783, 0x62c9, 0x5587, 0x8721, 0x814a,
336 0x8fa3, 0x5566, 0x83b1, 0x6765, 0x8d56, 0x84dd, 0x5a6a, 0x680f,
337 0x62e6, 0x7bee, 0x9611, 0x5170, 0x6f9c, 0x8c30, 0x63fd, 0x89c8,
338 0x61d2, 0x7f06, 0x70c2, 0x6ee5, 0x7405, 0x6994, 0x72fc, 0x5eca,
339 0x90ce, 0x6717, 0x6d6a, 0x635e, 0x52b3, 0x7262, 0x8001, 0x4f6c,
340 0x59e5, 0x916a, 0x70d9, 0x6d9d, 0x52d2, 0x4e50, 0x96f7, 0x956d,
341 0x857e, 0x78ca, 0x7d2f, 0x5121, 0x5792, 0x64c2, 0x808b, 0x7c7b,
342 0x6cea, 0x68f1, 0x695e, 0x51b7, 0x5398, 0x68a8, 0x7281, 0x9ece,
343 0x7bf1, 0x72f8, 0x79bb, 0x6f13, 0x7406, 0x674e, 0x91cc, 0x9ca4,
344 0x793c, 0x8389, 0x8354, 0x540f, 0x6817, 0x4e3d, 0x5389, 0x52b1,
345 0x783e, 0x5386, 0x5229, 0x5088, 0x4f8b, 0x4fd0,
346 /* 0x41 */
347 0x75e2, 0x7acb, 0x7c92, 0x6ca5, 0x96b6, 0x529b, 0x7483, 0x54e9,
348 0x4fe9, 0x8054, 0x83b2, 0x8fde, 0x9570, 0x5ec9, 0x601c, 0x6d9f,
349 0x5e18, 0x655b, 0x8138, 0x94fe, 0x604b, 0x70bc, 0x7ec3, 0x7cae,
350 0x51c9, 0x6881, 0x7cb1, 0x826f, 0x4e24, 0x8f86, 0x91cf, 0x667e,
351 0x4eae, 0x8c05, 0x64a9, 0x804a, 0x50da, 0x7597, 0x71ce, 0x5be5,
352 0x8fbd, 0x6f66, 0x4e86, 0x6482, 0x9563, 0x5ed6, 0x6599, 0x5217,
353 0x88c2, 0x70c8, 0x52a3, 0x730e, 0x7433, 0x6797, 0x78f7, 0x9716,
354 0x4e34, 0x90bb, 0x9cde, 0x6dcb, 0x51db, 0x8d41, 0x541d, 0x62ce,
355 0x73b2, 0x83f1, 0x96f6, 0x9f84, 0x94c3, 0x4f36, 0x7f9a, 0x51cc,
```

```
356 0x7075, 0x9675, 0x5cad, 0x9886, 0x53e6, 0x4ee4, 0x6e9c, 0x7409,
357 0x69b4, 0x786b, 0x998f, 0x7559, 0x5218, 0x7624, 0x6d41, 0x67f3,
358 0x516d, 0x9f99, 0x804b, 0x5499, 0x7b3c, 0x7abf,
359 /* 0x42 */
360 0x9686, 0x5784, 0x62e2, 0x9647, 0x697c, 0x5a04, 0x6402, 0x7bd3,
361 0x6f0f, 0x964b, 0x82a6, 0x5362, 0x9885, 0x5e90, 0x7089, 0x63b3,
362 0x5364, 0x864f, 0x9c81, 0x9e93, 0x788c, 0x9732, 0x8def, 0x8d42,
363 0x9e7f, 0x6f5e, 0x7984, 0x5f55, 0x9646, 0x622e, 0x9a74, 0x5415,
364 0x94dd, 0x4fa3, 0x65c5, 0x5c65, 0x5c61, 0x7f15, 0x8651, 0x6c2f,
365 0x5f8b, 0x7387, 0x6ee4, 0x7eff, 0x5ce6, 0x631b, 0x5b6a, 0x6ee6,
366 0x5375, 0x4e71, 0x63a0, 0x7565, 0x62a1, 0x8f6e, 0x4f26, 0x4ed1,
367 0x6ca6, 0x7eb6, 0x8bba, 0x841d, 0x87ba, 0x7f57, 0x903b, 0x9523,
368 0x7ba9, 0x9aa1, 0x88f8, 0x843d, 0x6d1b, 0x9a86, 0x7edc, 0x5988,
369 0x9ebb, 0x739b, 0x7801, 0x8682, 0x9a6c, 0x9a82, 0x561b, 0x5417,
370 0x57cb, 0x4e70, 0x9ea6, 0x5356, 0x8fc8, 0x8109, 0x7792, 0x9992,
371 0x86ee, 0x6ee1, 0x8513, 0x66fc, 0x6162, 0x6f2b,
372 /* 0x43 */
373 0x8c29, 0x8292, 0x832b, 0x76f2, 0x6c13, 0x5fd9, 0x83bd, 0x732b,
374 0x8305, 0x951a, 0x6bdb, 0x77db, 0x94c6, 0x536f, 0x8302, 0x5192,
375 0x5e3d, 0x8c8c, 0x8d38, 0x4e48, 0x73ab, 0x679a, 0x6885, 0x9176,
376 0x9709, 0x7164, 0x6ca1, 0x7709, 0x5a92, 0x9541, 0x6bcf, 0x7f8e,
377 0x6627, 0x5bd0, 0x59b9, 0x5a9a, 0x95e8, 0x95f7, 0x4eec, 0x840c,
378 0x8499, 0x6aac, 0x76df, 0x9530, 0x731b, 0x68a6, 0x5b5f, 0x772f,
379 0x919a, 0x9761, 0x7cdc, 0x8ff7, 0x8c1c, 0x5f25, 0x7c73, 0x79d8,
380 0x89c5, 0x6ccc, 0x871c, 0x5bc6, 0x5e42, 0x68c9, 0x7720, 0x7ef5,
381 0x5195, 0x514d, 0x52c9, 0x5a29, 0x7f05, 0x9762, 0x82d7, 0x63cf,
382 0x7784, 0x85d0, 0x79d2, 0x6e3a, 0x5e99, 0x5999, 0x8511, 0x706d,
383 0x6c11, 0x62bf, 0x76bf, 0x654f, 0x60af, 0x95fd, 0x660e, 0x879f,
384 0x9e23, 0x94ed, 0x540d, 0x547d, 0x8c2c, 0x6478,
385 /* 0x44 */
386 0x6479, 0x8611, 0x6a21, 0x819c, 0x78e8, 0x6469, 0x9b54, 0x62b9,
387 0x672b, 0x83ab, 0x58a8, 0x9ed8, 0x6cab, 0x6f20, 0x5bde, 0x964c,
388 0x8c0b, 0x725f, 0x67d0, 0x62c7, 0x7261, 0x4ea9, 0x59c6, 0x6bcd,
389 0x5893, 0x66ae, 0x5e55, 0x52df, 0x6155, 0x6728, 0x76ee, 0x7766,
390 0x7267, 0x7a46, 0x62ff, 0x54ea, 0x5450, 0x94a0, 0x90a3, 0x5a1c,
391 0x7eb3, 0x6c16, 0x4e43, 0x5976, 0x8010, 0x5948, 0x5357, 0x7537,
392 0x96be, 0x56ca, 0x6320, 0x8111, 0x607c, 0x95f9, 0x6dd6, 0x5462,
393 0x9981, 0x5185, 0x5ae9, 0x80fd, 0x59ae, 0x9713, 0x502a, 0x6ce5,
394 0x5c3c, 0x62df, 0x4f60, 0x533f, 0x817b, 0x9006, 0x6eba, 0x852d,
395 0x62c8, 0x5e74, 0x78be, 0x64b5, 0x637b, 0x5ff5, 0x5a18, 0x917f,
396 0x9e1f, 0x5c3f, 0x634f, 0x8042, 0x5b7d, 0x556e, 0x954a, 0x954d,
397 0x6d85, 0x60a8, 0x67e0, 0x72de, 0x51dd, 0x5b81,
398 /* 0x45 */
399 0x62e7, 0x6cde, 0x725b, 0x626d, 0x94ae, 0x7ebd, 0x8113, 0x6d53,
400 0x519c, 0x5f04, 0x5974, 0x52aa, 0x6012, 0x5973, 0x6696, 0x8650,
401 0x759f, 0x632a, 0x61e6, 0x7cef, 0x8bfa, 0x54e6, 0x6b27, 0x9e25,
402 0x6bb4, 0x85d5, 0x5455, 0x5076, 0x6ca4, 0x556a, 0x8db4, 0x722c,
403 0x5e15, 0x6015, 0x7436, 0x62cd, 0x6392, 0x724c, 0x5f98, 0x6e43,
404 0x6d3e, 0x6500, 0x6f58, 0x76d8, 0x78d0, 0x76fc, 0x7554, 0x5224,
405 0x53db, 0x4e53, 0x5e9e, 0x65c1, 0x802a, 0x80d6, 0x629b, 0x5486,
406 0x5228, 0x70ae, 0x888d, 0x8dd1, 0x6ce1, 0x5478, 0x80da, 0x57f9,
407 0x88f4, 0x8d54, 0x966a, 0x914d, 0x4f69, 0x6c9b, 0x55b7, 0x76c6,
408 0x7830, 0x62a8, 0x70f9, 0x6f8e, 0x5f6d, 0x84ec, 0x68da, 0x787c,
409 0x7bf7, 0x81a8, 0x670b, 0x9e4f, 0x6367, 0x78b0, 0x576f, 0x7812,
410 0x9739, 0x6279, 0x62ab, 0x5288, 0x7435, 0x6bd7,
411 /* 0x46 */
412 0x5564, 0x813e, 0x75b2, 0x76ae, 0x5339, 0x75de, 0x50fb, 0x5c41,
413 0x8b6c, 0x7bc7, 0x504f, 0x7247, 0x9a97, 0x98d8, 0x6f02, 0x74e2,
414 0x7968, 0x6487, 0x77a5, 0x62fc, 0x9891, 0x8d2b, 0x54c1, 0x8058,
415 0x4e52, 0x576a, 0x82f9, 0x840d, 0x5e73, 0x51ed, 0x74f6, 0x8bc4,
416 0x5c4f, 0x5761, 0x6cfc, 0x9887, 0x5a46, 0x7834, 0x9b44, 0x8feb,
417 0x7c95, 0x5256, 0x6251, 0x94fa, 0x4ec6, 0x8386, 0x8461, 0x83e9,
418 0x84b2, 0x57d4, 0x6734, 0x5703, 0x666e, 0x6d66, 0x8c31, 0x66dd,
419 0x7011, 0x671f, 0x6b3a, 0x6816, 0x621a, 0x59bb, 0x4e03, 0x51c4,
420 0x6f06, 0x67d2, 0x6c8f, 0x5176, 0x68cb, 0x5947, 0x6b67, 0x7566,
421 0x5d0e, 0x8110, 0x9f50, 0x65d7, 0x7948, 0x7941, 0x9a91, 0x8d77,
422 0x5c82, 0x4e5e, 0x4f01, 0x542f, 0x5951, 0x780c, 0x5668, 0x6c14,
423 0x8fc4, 0x5f03, 0x6c7d, 0x6ce3, 0x8bab, 0x6390,
424 /* 0x47 */
425 0x6070, 0x6d3d, 0x7275, 0x6266, 0x948e, 0x94c5, 0x5343, 0x8fc1,
426 0x7b7e, 0x4edf, 0x8c26, 0x4e7e, 0x9ed4, 0x94b1, 0x94b3, 0x524d,
427 0x6f5c, 0x9063, 0x6d45, 0x8c34, 0x5811, 0x5d4c, 0x6b20, 0x6b49,
428 0x67aa, 0x545b, 0x8154, 0x7f8c, 0x5899, 0x8537, 0x5f3a, 0x62a2,
429 0x6a47, 0x9539, 0x6572, 0x6084, 0x6865, 0x77a7, 0x4e54, 0x4fa8,
430 0x5de7, 0x9798, 0x64ac, 0x7fd8, 0x5ced, 0x4fcf, 0x7a8d, 0x5207,
431 0x8304, 0x4e14, 0x602f, 0x7a83, 0x94a6, 0x4fb5, 0x4eb2, 0x79e6,
432 0x7434, 0x52e4, 0x82b9, 0x64d2, 0x79bd, 0x5bdd, 0x6c81, 0x9752,
433 0x8f7b, 0x6c22, 0x503e, 0x537f, 0x6e05, 0x64ce, 0x6674, 0x6c30,
434 0x60c5, 0x9877, 0x8bf7, 0x5e86, 0x743c, 0x7a77, 0x79cb, 0x4e18,
435 0x90b1, 0x7403, 0x6c42, 0x56da, 0x914b, 0x6cc5, 0x8d8b, 0x533a,
436 0x86c6, 0x66f2, 0x8eaf, 0x5c48, 0x9a71, 0x6e20,
437 /* 0x48 */
438 0x53d6, 0x5a36, 0x9f8b, 0x8da3, 0x53bb, 0x5708, 0x98a7, 0x6743,
439 0x919b, 0x6cc9, 0x5168, 0x75ca, 0x62f3, 0x72ac, 0x5238, 0x529d,
440 0x7f3a, 0x7094, 0x7638, 0x5374, 0x9e4a, 0x69b7, 0x786e, 0x96c0,
441 0x88d9, 0x7fa4, 0x7136, 0x71c3, 0x5189, 0x67d3, 0x74e4, 0x58e4,
442 0x6518, 0x56b7, 0x8ba9, 0x9976, 0x6270, 0x7ed5, 0x60f9, 0x70ed,
```

```

443 0x58ec, 0x4ec1, 0x4eba, 0x5fcd, 0x97e7, 0x4efb, 0x8ba4, 0x5203,
444 0x598a, 0x7eab, 0x6254, 0x4ecd, 0x65e5, 0x620e, 0x8338, 0x84c9,
445 0x8363, 0x878d, 0x7194, 0x6eb6, 0x5bb9, 0x7ed2, 0x5197, 0x63c9,
446 0x67d4, 0x8089, 0x8339, 0x8815, 0x5112, 0x5b7a, 0x5982, 0x8fb1,
447 0x4e73, 0x6c5d, 0x5165, 0x8925, 0x8f6f, 0x962e, 0x854a, 0x745e,
448 0x9510, 0x95f0, 0x6da6, 0x82e5, 0x5f31, 0x6492, 0x6d12, 0x8428,
449 0x816e, 0x9cc3, 0x585e, 0x8d5b, 0x4e09, 0x53c1,
450 /* 0x49 */
451 0x4f1e, 0x6563, 0x6851, 0x55d3, 0x4e27, 0x6414, 0x9a9a, 0x626b,
452 0x5ac2, 0x745f, 0x8272, 0x6da9, 0x68ee, 0x50e7, 0x838e, 0x7802,
453 0x6740, 0x5239, 0x6c99, 0x7eb1, 0x50bb, 0x5565, 0x715e, 0x7b5b,
454 0x6652, 0x73ca, 0x82eb, 0x6749, 0x5c71, 0x5220, 0x717d, 0x886b,
455 0x95ea, 0x9655, 0x64c5, 0x8d61, 0x81b3, 0x5584, 0x6c55, 0x6247,
456 0x7f2e, 0x5892, 0x4f24, 0x5546, 0x8d4f, 0x664c, 0x4e0a, 0x5c1a,
457 0x88f3, 0x68a2, 0x634e, 0x7a0d, 0x70e7, 0x828d, 0x52fa, 0x97f6,
458 0x5c11, 0x54e8, 0x90b5, 0x7ecd, 0x5962, 0x8d4a, 0x86c7, 0x820c,
459 0x820d, 0x8d66, 0x6444, 0x5c04, 0x6151, 0x6d89, 0x793e, 0x8bbe,
460 0x7837, 0x7533, 0x547b, 0x4f38, 0x8eab, 0x6df1, 0x5a20, 0x7ec5,
461 0x795e, 0x6c88, 0x5ba1, 0x5a76, 0x751a, 0x80be, 0x614e, 0x6e17,
462 0x58f0, 0x751f, 0x7525, 0x7272, 0x5347, 0x7ef3,
463 /* 0x4a */
464 0x7701, 0x76db, 0x5269, 0x80dc, 0x5723, 0x5e08, 0x5931, 0x72ee,
465 0x65bd, 0x6e7f, 0x8bd7, 0x5c38, 0x8671, 0x5341, 0x77f3, 0x62fe,
466 0x65f6, 0x4ec0, 0x98df, 0x8680, 0x5b9e, 0x8bc6, 0x53f2, 0x77e2,
467 0x4f7f, 0x5c4e, 0x9a76, 0x59cb, 0x5f0f, 0x793a, 0x58eb, 0x4e16,
468 0x67ff, 0x4e8b, 0x62ed, 0x8a93, 0x901d, 0x52bf, 0x662f, 0x55dc,
469 0x566c, 0x9002, 0x4ed5, 0x4f8d, 0x91ca, 0x9970, 0x6c0f, 0x5e02,
470 0x6043, 0x5ba4, 0x89c6, 0x8bd5, 0x6536, 0x624b, 0x9996, 0x5b88,
471 0x5bff, 0x6388, 0x552e, 0x53d7, 0x7626, 0x517d, 0x852c, 0x67a2,
472 0x68b3, 0x6b8a, 0x6292, 0x8f93, 0x53d4, 0x8212, 0x6dd1, 0x758f,
473 0x4e66, 0x8d4e, 0x5b70, 0x719f, 0x85af, 0x6691, 0x66d9, 0x7f72,
474 0x8700, 0x9ecd, 0x9f20, 0x5c5e, 0x672f, 0x8ff0, 0x6811, 0x675f,
475 0x620d, 0x7ad6, 0x5885, 0x5eb6, 0x6570, 0x6f31,
476 /* 0x4b */
477 0x6055, 0x5237, 0x800d, 0x6454, 0x8870, 0x7529, 0x5e05, 0x6813,
478 0x62f4, 0x971c, 0x53cc, 0x723d, 0x8c01, 0x6c34, 0x7761, 0x7a0e,
479 0x542e, 0x77ac, 0x987a, 0x821c, 0x8bf4, 0x7855, 0x6714, 0x70c1,
480 0x65af, 0x6495, 0x5636, 0x601d, 0x79c1, 0x53f8, 0x4e1d, 0x6b7b,
481 0x8086, 0x5bfa, 0x55e3, 0x56db, 0x4f3a, 0x4f3c, 0x9972, 0x5df3,
482 0x677e, 0x8038, 0x6002, 0x9882, 0x9001, 0x5b8b, 0x8bbc, 0x8bf5,
483 0x641c, 0x8258, 0x64de, 0x55fd, 0x82cf, 0x9165, 0x4fd7, 0x7d20,
484 0x901f, 0x7c9f, 0x50f3, 0x5851, 0x6eaf, 0x5bbf, 0x8bc9, 0x8083,
485 0x9178, 0x849c, 0x7b97, 0x867d, 0x968b, 0x968f, 0x7ee5, 0x9ad3,
486 0x788e, 0x5c81, 0x7a57, 0x9042, 0x96a7, 0x795f, 0x5b59, 0x635f,
487 0x7b0b, 0x84d1, 0x68ad, 0x5506, 0x7f29, 0x7410, 0x7d22, 0x9501,
488 0x6240, 0x584c, 0x4ed6, 0x5b83, 0x5979, 0x5854,
489 /* 0x4c */
490 0x736d, 0x631e, 0x8e4b, 0x8e0f, 0x80ce, 0x82d4, 0x62ac, 0x53f0,
491 0x6cf0, 0x915e, 0x592a, 0x6001, 0x6c70, 0x574d, 0x644a, 0x8d2a,
492 0x762b, 0x6ee9, 0x575b, 0x6a80, 0x75f0, 0x6f6d, 0x8c2d, 0x8c08,
493 0x5766, 0x6bef, 0x8892, 0x78b3, 0x63a2, 0x53f9, 0x70ad, 0x6c64,
494 0x5858, 0x642a, 0x5802, 0x68e0, 0x819b, 0x5510, 0x7cd6, 0x5018,
495 0x8eba, 0x6dcc, 0x8d9f, 0x70eb, 0x638f, 0x6d9b, 0x6ed4, 0x7ee6,
496 0x8404, 0x6843, 0x9003, 0x6dd8, 0x9676, 0x8ba8, 0x5957, 0x7279,
497 0x85e4, 0x817e, 0x75bc, 0x8a8a, 0x68af, 0x5254, 0x8e22, 0x9511,
498 0x63d0, 0x9898, 0x8e44, 0x557c, 0x4f53, 0x66ff, 0x568f, 0x60d5,
499 0x6d95, 0x5243, 0x5c49, 0x5929, 0x6dfb, 0x586b, 0x7530, 0x751c,
500 0x606c, 0x8214, 0x8146, 0x6311, 0x6761, 0x8fe2, 0x773a, 0x8df3,
501 0x8d34, 0x94c1, 0x5e16, 0x5385, 0x542c, 0x70c3,
502 /* 0x4d */
503 0x6c40, 0x5ef7, 0x505c, 0x4ead, 0x5ead, 0x633a, 0x8247, 0x901a,
504 0x6850, 0x916e, 0x77b3, 0x540c, 0x94dc, 0x5f64, 0x7ae5, 0x6876,
505 0x6345, 0x7b52, 0x7edf, 0x75db, 0x5077, 0x6295, 0x5934, 0x900f,
506 0x51f8, 0x79c3, 0x7a81, 0x56fe, 0x5f92, 0x9014, 0x6d82, 0x5c60,
507 0x571f, 0x5410, 0x5154, 0x6e4d, 0x56e2, 0x63a8, 0x9893, 0x817f,
508 0x8715, 0x892a, 0x9000, 0x541e, 0x5c6f, 0x81c0, 0x62d6, 0x6258,
509 0x8131, 0x9e35, 0x9640, 0x9a6e, 0x9a7c, 0x692d, 0x59a5, 0x62d3,
510 0x553e, 0x6316, 0x54c7, 0x86d9, 0x6d3c, 0x5a03, 0x74e6, 0x889c,
511 0x6b6a, 0x5916, 0x8c4c, 0x5f2f, 0x6e7e, 0x73a9, 0x987d, 0x4e38,
512 0x70f7, 0x5b8c, 0x7897, 0x633d, 0x665a, 0x769e, 0x60cb, 0x5b9b,
513 0x5a49, 0x4e07, 0x8155, 0x6c6a, 0x738b, 0x4ea1, 0x6789, 0x7f51,
514 0x5f80, 0x65fa, 0x671b, 0x5fd8, 0x5984, 0x5a01,
515 /* 0x4e */
516 0x5dcd, 0x5fae, 0x5371, 0x97e6, 0x8fdd, 0x6845, 0x56f4, 0x552f,
517 0x60df, 0x4e3a, 0x6f4d, 0x7ef4, 0x82c7, 0x840e, 0x59d4, 0x4f1f,
518 0x4f2a, 0x5c3e, 0x7eac, 0x672a, 0x851a, 0x5473, 0x754f, 0x80c3,
519 0x5582, 0x9b4f, 0x4f4d, 0x6e2d, 0x8c13, 0x5c09, 0x6170, 0x536b,
520 0x761f, 0x6e29, 0x868a, 0x6587, 0x95fb, 0x7eb9, 0x543b, 0x7a33,
521 0x7d0a, 0x95ee, 0x55e1, 0x7fc1, 0x74ee, 0x631d, 0x8717, 0x6da1,
522 0x7a9d, 0x6211, 0x65a1, 0x5367, 0x63e1, 0x6c83, 0x5deb, 0x545c,
523 0x94a8, 0x4e4c, 0x6c61, 0x8bec, 0x5c4b, 0x65e0, 0x829c, 0x68a7,
524 0x543e, 0x5434, 0x6bcb, 0x6b66, 0x4e94, 0x6342, 0x5348, 0x821e,
525 0x4f0d, 0x4fae, 0x575e, 0x620a, 0x96fe, 0x6664, 0x7269, 0x52ff,
526 0x52a1, 0x609f, 0x8bef, 0x6614, 0x7199, 0x6790, 0x897f, 0x7852,
527 0x77fd, 0x6670, 0x563b, 0x5438, 0x9521, 0x727a,
528 /* 0x4f */
529 0x7a00, 0x606f, 0x5e0c, 0x6089, 0x819d, 0x5915, 0x60dc, 0x7184,

```

```
530 0x70ef, 0x6eaa, 0x6c50, 0x7280, 0x6a84, 0x88ad, 0x5e2d, 0x4e60,
531 0x5ab3, 0x559c, 0x94e3, 0x6d17, 0x7cfb, 0x9699, 0x620f, 0x7ec6,
532 0x778e, 0x867e, 0x5323, 0x971e, 0x8f96, 0x6687, 0x5ce1, 0x4fa0,
533 0x72ed, 0x4e0b, 0x53a6, 0x590f, 0x5413, 0x6380, 0x9528, 0x5148,
534 0x4ed9, 0x9c9c, 0x7ea4, 0x54b8, 0x8d24, 0x8854, 0x8237, 0x95f2,
535 0x6d8e, 0x5f26, 0x5acc, 0x663e, 0x9669, 0x73b0, 0x732e, 0x53bf,
536 0x817a, 0x9985, 0x7fa1, 0x5baa, 0x9677, 0x9650, 0x7ebf, 0x76f8,
537 0x53a2, 0x9576, 0x9999, 0x7bb1, 0x8944, 0x6e58, 0x4e61, 0x7fd4,
538 0x7965, 0x8be6, 0x60f3, 0x54cd, 0x4eab, 0x9879, 0x5df7, 0x6a61,
539 0x50cf, 0x5411, 0x8c61, 0x8427, 0x785d, 0x9704, 0x524a, 0x54ee,
540 0x56a3, 0x9500, 0x6d88, 0x5bb5, 0x6dc6, 0x6653,
541 /* 0x50 */
542 0x5c0f, 0x5b5d, 0x6821, 0x8096, 0x5578, 0x7b11, 0x6548, 0x6954,
543 0x4e9b, 0x6b47, 0x874e, 0x978b, 0x534f, 0x631f, 0x643a, 0x90aa,
544 0x659c, 0x80c1, 0x8c10, 0x5199, 0x68b0, 0x5378, 0x87f9, 0x61c8,
545 0x6cc4, 0x6cfb, 0x8c22, 0x5c51, 0x85aa, 0x82af, 0x950c, 0x6b23,
546 0x8f9b, 0x65b0, 0x5ffb, 0x5fc3, 0x4fe1, 0x8845, 0x661f, 0x8165,
547 0x7329, 0x60fa, 0x60f3, 0x5174, 0x5211, 0x578b, 0x5f62, 0x90a2, 0x884c,
548 0x9192, 0x5e78, 0x674f, 0x6027, 0x59d3, 0x5144, 0x51f6, 0x80f8,
549 0x5308, 0x6c79, 0x96c4, 0x718a, 0x4f11, 0x4fee, 0x7f9e, 0x673d,
550 0x55c5, 0x9508, 0x79c0, 0x8896, 0x7ee3, 0x589f, 0x620c, 0x9700,
551 0x865a, 0x5618, 0x987b, 0x5f90, 0x8bb8, 0x84c4, 0x9157, 0x53d9,
552 0x65ed, 0x5e8f, 0x755c, 0x6064, 0x7d6e, 0x5a7f, 0x7eea, 0x7eed,
553 0x8f69, 0x55a7, 0x5ba3, 0x60ac, 0x65cb, 0x7384,
554 /* 0x51 */
555 0x9009, 0x7663, 0x7729, 0x7eda, 0x9774, 0x859b, 0x5b66, 0x7a74,
556 0x96ea, 0x8840, 0x52cb, 0x718f, 0x5faa, 0x65ec, 0x8be2, 0x5bfb,
557 0x9a6f, 0x5de1, 0x6b89, 0x6c5b, 0x8bad, 0x8baf, 0x900a, 0x8fc5,
558 0x538b, 0x62bc, 0x9e26, 0x9e2d, 0x5440, 0x4e2b, 0x82bd, 0x7259,
559 0x869c, 0x5d16, 0x8859, 0x6daf, 0x96c5, 0x54d1, 0x4e9a, 0x8bb6,
560 0x7109, 0x54bd, 0x9609, 0x70df, 0x6df9, 0x76d0, 0x4e25, 0x7814,
561 0x8712, 0x5ca9, 0x5ef6, 0x8a00, 0x989c, 0x960e, 0x708e, 0x6cbf,
562 0x5944, 0x63a9, 0x773c, 0x884d, 0x6f14, 0x8273, 0x5830, 0x71d5,
563 0x538c, 0x781a, 0x96c1, 0x5501, 0x5f66, 0x7130, 0x5bb4, 0x8c1a,
564 0x9a8c, 0x6b83, 0x592e, 0x9e2f, 0x79e7, 0x6768, 0x626c, 0x4f6f,
565 0x75a1, 0x7f8a, 0x6d0b, 0x9633, 0x6c27, 0x4ef0, 0x75d2, 0x517b,
566 0x6837, 0x6f3e, 0x9080, 0x8170, 0x5996, 0x7476,
567 /* 0x52 */
568 0x6447, 0x5c27, 0x9065, 0x7a91, 0x8c23, 0x59da, 0x54ac, 0x8200,
569 0x836f, 0x8981, 0x8000, 0x6930, 0x564e, 0x8036, 0x7237, 0x91ce,
570 0x51b6, 0x4e5f, 0x9875, 0x6396, 0x4e1a, 0x53f6, 0x66f3, 0x814b,
571 0x591c, 0x6db2, 0x4e00, 0x58f9, 0x533b, 0x63d6, 0x94f1, 0x4f9d,
572 0x4f0a, 0x8863, 0x9890, 0x5937, 0x9057, 0x79fb, 0x4eea, 0x80f0,
573 0x7591, 0x6c82, 0x5b9c, 0x59e8, 0x5f5d, 0x6905, 0x8681, 0x501a,
574 0x5df2, 0x4e59, 0x77e3, 0x4ee5, 0x827a, 0x6291, 0x6613, 0x9091,
575 0x5c79, 0x4ebf, 0x5f79, 0x81c6, 0x9038, 0x8084, 0x75ab, 0x4ea6,
576 0x88d4, 0x610f, 0x6bc5, 0x5fc6, 0x4e49, 0x76ca, 0x6ea2, 0x8be3,
577 0x8bae, 0x8c0a, 0x8bd1, 0x5f02, 0x7ffc, 0x7fcc, 0x7ece, 0x8335,
578 0x836b, 0x56e0, 0x6bb7, 0x97f3, 0x9634, 0x59fb, 0x541f, 0x94f6,
579 0x6deb, 0x5bc5, 0x996e, 0x5c39, 0x5f15, 0x9690,
580 /* 0x53 */
581 0x5370, 0x82f1, 0x6a31, 0x5a74, 0x9e70, 0x5e94, 0x7f28, 0x83b9,
582 0x8424, 0x8425, 0x8367, 0x8747, 0x8fce, 0x8d62, 0x76c8, 0x5f71,
583 0x9896, 0x786c, 0x6620, 0x54df, 0x62e5, 0x4f63, 0x81c3, 0x75c8,
584 0x5eb8, 0x96cd, 0x8e0a, 0x86f9, 0x548f, 0x6cf3, 0x6d8c, 0x6c38,
585 0x607f, 0x52c7, 0x7528, 0x5e7d, 0x4f18, 0x60a0, 0x5fe7, 0x5c24,
586 0x7531, 0x90ae, 0x94c0, 0x72b9, 0x6cb9, 0x6e38, 0x9149, 0x6709,
587 0x53cb, 0x53f3, 0x4f51, 0x91c9, 0x8bf1, 0x53c8, 0x5e7c, 0x8fc2,
588 0x6de4, 0x4e8e, 0x76c2, 0x6986, 0x865e, 0x611a, 0x8206, 0x4f59,
589 0x4fde, 0x903e, 0x9c7c, 0x6109, 0x6e1d, 0x6e14, 0x9685, 0x4e88,
590 0x5a31, 0x96e8, 0x4e0e, 0x5c7f, 0x79b9, 0x5b87, 0x8bed, 0x7fbd,
591 0x7389, 0x57df, 0x828b, 0x90c1, 0x5401, 0x9047, 0x55bb, 0x5cea,
592 0x5fa1, 0x6108, 0x6b32, 0x72f1, 0x80b2, 0x8a89,
593 /* 0x54 */
594 0x6d74, 0x5bd3, 0x88d5, 0x9884, 0x8c6b, 0x9a6d, 0x9e33, 0x6e0a,
595 0x51a4, 0x5143, 0x57a3, 0x8881, 0x539f, 0x63f4, 0x8f95, 0x56ed,
596 0x5458, 0x5706, 0x733f, 0x6e90, 0x7f18, 0x8fdc, 0x82d1, 0x613f,
597 0x6028, 0x9662, 0x66f0, 0x7ea6, 0x8d8a, 0x8dc3, 0x94a5, 0x5cb3,
598 0x7ca4, 0x6708, 0x60a6, 0x9605, 0x8018, 0x4e91, 0x90e7, 0x5300,
599 0x9668, 0x5141, 0x8fd0, 0x8574, 0x915d, 0x6655, 0x97f5, 0x5b55,
600 0x531d, 0x7838, 0x6742, 0x683d, 0x54c9, 0x707e, 0x5bb0, 0x8f7d,
601 0x518d, 0x5728, 0x54b1, 0x6512, 0x6682, 0x8d5e, 0x8d43, 0x810f,
602 0x846c, 0x906d, 0x7cdf, 0x51ff, 0x85fb, 0x67a3, 0x65e9, 0x6fa1,
603 0x86a4, 0x8e81, 0x566a, 0x9020, 0x7682, 0x7076, 0x71e5, 0x8d23,
604 0x62e9, 0x5219, 0x6cfd, 0x8d3c, 0x600e, 0x589e, 0x618e, 0x66fe,
605 0x8d60, 0x624e, 0x55b3, 0x6e23, 0x672d, 0x8f67,
606 /* 0x55 */
607 0x94e1, 0x95f8, 0x7728, 0x6805, 0x69a8, 0x548b, 0x4e4d, 0x70b8,
608 0x8bc8, 0x6458, 0x658b, 0x5b85, 0x7a84, 0x503a, 0x5be8, 0x77bb,
609 0x6be1, 0x8a79, 0x7c98, 0x6cbe, 0x76cf, 0x65a9, 0x8f97, 0x5d2d,
610 0x5c55, 0x8638, 0x6808, 0x5360, 0x6218, 0x7ad9, 0x6e5b, 0x7efd,
611 0x6a1f, 0x7ae0, 0x5f70, 0x6f33, 0x5f20, 0x638c, 0x6da8, 0x6756,
612 0x4e08, 0x5e10, 0x8d26, 0x4ed7, 0x80c0, 0x7634, 0x969c, 0x62db,
613 0x662d, 0x627e, 0x6cbc, 0x8d75, 0x7167, 0x7f69, 0x5146, 0x8087,
614 0x53ec, 0x906e, 0x6298, 0x54f2, 0x86f0, 0x8f99, 0x8005, 0x9517,
615 0x8517, 0x8fd9, 0x6d59, 0x73cd, 0x659f, 0x771f, 0x7504, 0x7827,
616 0x81fb, 0x8d1e, 0x9488, 0x4fa6, 0x6795, 0x75b9, 0x8bca, 0x9707,
```



```

617 0x632f, 0x9547, 0x9635, 0x84b8, 0x6323, 0x7741, 0x5f81, 0x72f0,
618 0x4e89, 0x6014, 0x6574, 0x62ef, 0x6b63, 0x653f,
619 /* 0x56 */
620 0x5e27, 0x75c7, 0x90d1, 0x8bc1, 0x829d, 0x679d, 0x652f, 0x5431,
621 0x8718, 0x77e5, 0x80a2, 0x8102, 0x6c41, 0x4e4b, 0x7ec7, 0x804c,
622 0x76f4, 0x690d, 0x6b96, 0x6267, 0x503c, 0x4f84, 0x5740, 0x6307,
623 0x6b62, 0x8dbe, 0x53ea, 0x65e8, 0x7eb8, 0x5fd7, 0x631a, 0x63b7,
624 0x81f3, 0x81f4, 0x7f6e, 0x5e1c, 0x5cd9, 0x5236, 0x667a, 0x79e9,
625 0x7a1a, 0x8d28, 0x7099, 0x75d4, 0x6ede, 0x6cbb, 0x7a92, 0x4e2d,
626 0x76c5, 0x5fe0, 0x949f, 0x8877, 0x7ec8, 0x79cd, 0x80bf, 0x91cd,
627 0x4ef2, 0x4f17, 0x821f, 0x5468, 0x5dde, 0x6d32, 0x8bcc, 0x7ca5,
628 0x8f74, 0x8098, 0x5e1a, 0x5492, 0x76b1, 0x5b99, 0x663c, 0x9aa4,
629 0x73e0, 0x682a, 0x86db, 0x6731, 0x732a, 0x8bf8, 0x8bdb, 0x9010,
630 0x7af9, 0x70db, 0x716e, 0x62c4, 0x77a9, 0x5631, 0x4e3b, 0x8457,
631 0x67f1, 0x52a9, 0x86c0, 0x8d2e, 0x94f8, 0x7b51,
632 /* 0x57 */
633 0x4f4f, 0x6ce8, 0x795d, 0x9a7b, 0x6293, 0x722a, 0x62fd, 0x4e13,
634 0x7816, 0x8f6c, 0x64b0, 0x8d5a, 0x7bc6, 0x6869, 0x5e84, 0x88c5,
635 0x5986, 0x649e, 0x58ee, 0x72b6, 0x690e, 0x9525, 0x8ffd, 0x8d58,
636 0x5760, 0x7f00, 0x8c06, 0x51c6, 0x6349, 0x62d9, 0x5353, 0x684c,
637 0x7422, 0x8301, 0x914c, 0x5544, 0x7740, 0x707c, 0x6d4a, 0x5179,
638 0x54a8, 0x8d44, 0x59ff, 0x6ecb, 0x6dc4, 0x5b5c, 0x7d2b, 0x4ed4,
639 0x7c7d, 0x6ed3, 0x5b50, 0x81ea, 0x6e0d, 0x5b57, 0x9b03, 0x68d5,
640 0x8e2a, 0x5b97, 0x9f17, 0x7efc, 0x603b, 0x7eb5, 0x90b9, 0x8d70, 0x594f,
641 0x63cd, 0x79df, 0x8db3, 0x5352, 0x65cf, 0x7956, 0x8bc5, 0x963b,
642 0x7ec4, 0x94bb, 0x7e82, 0x5634, 0x9189, 0x6700, 0x7f6a, 0x5c0a,
643 0x9075, 0x6628, 0x5de6, 0x4f50, 0x67de, 0x505a, 0x4f5c, 0x5750,
644 0x5ea7, 0xfffd, 0xfffd, 0xfffd, 0xfffd, 0xfffd,
645 /* 0x58 */
646 0x4e8d, 0x4e0c, 0x5140, 0x4e10, 0x5eff, 0x5345, 0x4e15, 0x4e98,
647 0x4e1e, 0x9b32, 0x5b6c, 0x5669, 0x4e28, 0x79ba, 0x4e3f, 0x5315,
648 0x4e47, 0x592d, 0x723b, 0x536e, 0x6c10, 0x56d4, 0x80e4, 0x9997,
649 0x6bd3, 0x777e, 0x9f17, 0x4e36, 0x4e9f, 0x9f10, 0x4e5c, 0x4e69,
650 0x4e93, 0x8288, 0x5b5b, 0x556c, 0x560f, 0x4ec4, 0x538d, 0x539d,
651 0x53a3, 0x53a5, 0x53ae, 0x9765, 0x8d5d, 0x531a, 0x53f5, 0x5326,
652 0x532e, 0x533e, 0x8d5c, 0x5366, 0x5363, 0x5202, 0x5208, 0x520e,
653 0x522d, 0x5233, 0x523f, 0x5240, 0x524c, 0x525e, 0x5261, 0x525c,
654 0x84af, 0x527d, 0x5282, 0x5281, 0x5290, 0x5293, 0x5182, 0x7f54,
655 0x4ebb, 0x4ec3, 0x4ec9, 0x4ec2, 0x4ee8, 0x4ee1, 0x4eeb, 0x4ede,
656 0x4f1b, 0x4ef3, 0x4f22, 0x4f64, 0x4ef5, 0x4f25, 0x4f27, 0x4f09,
657 0x4f2b, 0x4f5e, 0x4f67, 0x6538, 0x4f5a, 0x4f5d,
658 /* 0x59 */
659 0x4f5f, 0x4f57, 0x4f32, 0x4f3d, 0x4f76, 0x4f74, 0x4f91, 0x4f89,
660 0x4f83, 0x4f8f, 0x4f7e, 0x4f7b, 0x4faa, 0x4f7c, 0x4fac, 0x4f94,
661 0x4fe6, 0x4fe8, 0x4fea, 0x4fc5, 0x4fda, 0x4fe3, 0x4fdc, 0x4fd1,
662 0x4fdf, 0x4ff8, 0x5029, 0x504c, 0x4ff3, 0x502c, 0x500f, 0x502e,
663 0x502d, 0x4ffe, 0x501c, 0x500c, 0x5025, 0x5028, 0x507e, 0x5043,
664 0x5055, 0x5048, 0x504e, 0x506c, 0x507b, 0x50a5, 0x50a7, 0x50a9,
665 0x50ba, 0x50d6, 0x5106, 0x50ed, 0x50ec, 0x50e6, 0x50ee, 0x5107,
666 0x510b, 0x4edd, 0x6c3d, 0x4f58, 0x4f65, 0x4fce, 0x9fa0, 0x6c46,
667 0x7c74, 0x516e, 0x5dfd, 0x9998, 0x5181, 0x5914, 0x52f9,
668 0x530d, 0x8a07, 0x5310, 0x51eb, 0x5919, 0x5155, 0x4ea0, 0x5156,
669 0x4eb3, 0x886e, 0x88a4, 0x4eb5, 0x8114, 0x88d2, 0x7980, 0x5b34,
670 0x8803, 0x7fb8, 0x51ab, 0x51b1, 0x51bd, 0x51bc,
671 /* 0x5a */
672 0x51c7, 0x5196, 0x51a2, 0x51a5, 0x8ba0, 0x8ba6, 0x8ba7, 0x8baa,
673 0x8bb4, 0x8bb5, 0x8bb7, 0x8bc2, 0x8bc3, 0x8bcb, 0x8bcf, 0x8bce,
674 0x8bd2, 0x8bd3, 0x8bd4, 0x8bd6, 0x8bd8, 0x8bd9, 0x8bdc, 0x8bdf,
675 0x8be0, 0x8be4, 0x8be8, 0x8be9, 0x8bee, 0x8bf0, 0x8bf3, 0x8bf6,
676 0x8bf9, 0x8bfc, 0x8bff, 0x8c00, 0x8c02, 0x8c04, 0x8c07, 0x8c0c,
677 0x8c0f, 0x8c11, 0x8c12, 0x8c14, 0x8c15, 0x8c16, 0x8c19, 0x8c1b,
678 0x8c18, 0x8c1d, 0x8c1f, 0x8c20, 0x8c21, 0x8c25, 0x8c27, 0x8c2a,
679 0x8c2b, 0x8c2e, 0x8c2f, 0x8c32, 0x8c33, 0x8c35, 0x8c36, 0x5369,
680 0x537a, 0x961d, 0x9622, 0x9621, 0x9631, 0x962a, 0x963d, 0x963c,
681 0x9642, 0x9649, 0x9654, 0x965f, 0x9667, 0x966c, 0x9672, 0x9674,
682 0x9688, 0x968d, 0x9697, 0x96b0, 0x9097, 0x909b, 0x909d, 0x9099,
683 0x90ac, 0x90a1, 0x90b4, 0x90b3, 0x90b6, 0x90ba,
684 /* 0x5b */
685 0x90b8, 0x90b0, 0x90cf, 0x90c5, 0x90be, 0x90d0, 0x90c4, 0x90c7,
686 0x90d3, 0x90e6, 0x90e2, 0x90dc, 0x90d7, 0x90db, 0x90eb, 0x90ef,
687 0x90fe, 0x9104, 0x9122, 0x911e, 0x9123, 0x9131, 0x912f, 0x9139,
688 0x9143, 0x9146, 0x520d, 0x5942, 0x52a2, 0x52ac, 0x52ad, 0x52be,
689 0x54ff, 0x52d0, 0x52d6, 0x52f0, 0x53df, 0x71ee, 0x77cd, 0x5ef4,
690 0x51f5, 0x51fc, 0x9b2f, 0x53b6, 0x5f01, 0x755a, 0x5def, 0x574c,
691 0x57a9, 0x57a1, 0x587e, 0x58bc, 0x58c5, 0x58d1, 0x5729, 0x572c,
692 0x572a, 0x5733, 0x5739, 0x572e, 0x572f, 0x575c, 0x573b, 0x5742,
693 0x5769, 0x5785, 0x576b, 0x578e, 0x577c, 0x577b, 0x5768, 0x576d,
694 0x5776, 0x5773, 0x57ad, 0x57a4, 0x578c, 0x57b2, 0x57cf, 0x57a7,
695 0x57b4, 0x5793, 0x57a0, 0x57d5, 0x57d8, 0x57da, 0x57d9, 0x57d2,
696 0x57b8, 0x57f4, 0x57ef, 0x57f8, 0x57e4, 0x57dd,
697 /* 0x5c */
698 0x580b, 0x580d, 0x57fd, 0x57ed, 0x5800, 0x581e, 0x5819, 0x5844,
699 0x5820, 0x5865, 0x586c, 0x5881, 0x5889, 0x589a, 0x5880, 0x99a8,
700 0x9f19, 0x61ff, 0x8279, 0x827d, 0x827f, 0x828f, 0x828a, 0x82a8,
701 0x8284, 0x828e, 0x8291, 0x8297, 0x8299, 0x82ab, 0x82b8, 0x82be,
702 0x82b0, 0x82c8, 0x82ca, 0x82e3, 0x8298, 0x82b7, 0x82ae, 0x82cb,
703 0x82cc, 0x82c1, 0x82a9, 0x82b4, 0x82a1, 0x82aa, 0x829f, 0x82c4,

```

```
704 0x82ce, 0x82a4, 0x82e1, 0x8309, 0x82f7, 0x82e4, 0x830f, 0x8307,
705 0x82dc, 0x82f4, 0x82d2, 0x82d8, 0x830c, 0x82fb, 0x82d3, 0x8311,
706 0x831a, 0x8306, 0x8306, 0x8314, 0x8315, 0x82e0, 0x82d5, 0x831c, 0x8351,
707 0x835b, 0x835c, 0x8308, 0x8392, 0x833c, 0x8334, 0x8331, 0x839b,
708 0x835e, 0x832f, 0x834f, 0x8347, 0x8343, 0x835f, 0x8340, 0x8317,
709 0x8360, 0x832d, 0x833a, 0x8333, 0x8366, 0x8365,
710 /* 0x5d */
711 0x8368, 0x831b, 0x8369, 0x836c, 0x836a, 0x836d, 0x836e, 0x83b0,
712 0x8378, 0x83b3, 0x83b4, 0x83a0, 0x83aa, 0x8393, 0x839c, 0x8385,
713 0x837c, 0x83b6, 0x83a9, 0x837d, 0x83b8, 0x837b, 0x8398, 0x839e,
714 0x83a8, 0x83ba, 0x83bc, 0x83c1, 0x8401, 0x83e5, 0x83d8, 0x5807,
715 0x8418, 0x840b, 0x83dd, 0x83fd, 0x83d6, 0x841c, 0x8438, 0x8411,
716 0x8406, 0x83d4, 0x83df, 0x840f, 0x8403, 0x83f8, 0x83f9, 0x83ea,
717 0x83c5, 0x83c0, 0x8426, 0x83f0, 0x83e1, 0x845c, 0x8451, 0x845a,
718 0x8459, 0x8473, 0x8487, 0x8488, 0x847a, 0x8489, 0x8478, 0x843c,
719 0x8446, 0x8469, 0x8476, 0x848c, 0x848e, 0x8431, 0x846d, 0x84c1,
720 0x84cd, 0x84d0, 0x84e6, 0x84bd, 0x84d3, 0x84ca, 0x84bf, 0x84ba,
721 0x84e0, 0x84a1, 0x84b9, 0x84b4, 0x8497, 0x84e5, 0x84e3, 0x850c,
722 0x750d, 0x8538, 0x84f0, 0x8539, 0x851f, 0x853a,
723 /* 0x5e */
724 0x8556, 0x853b, 0x84ff, 0x84fc, 0x8559, 0x8548, 0x8568, 0x8564,
725 0x855e, 0x857a, 0x77a2, 0x8543, 0x8572, 0x857b, 0x85a4, 0x85a8,
726 0x8587, 0x858f, 0x8579, 0x85ae, 0x859c, 0x8585, 0x85b9, 0x85b7,
727 0x85b0, 0x85d3, 0x85c1, 0x85dc, 0x85ff, 0x8627, 0x8605, 0x8629,
728 0x8616, 0x863c, 0x5efe, 0x5f08, 0x593c, 0x5941, 0x8037, 0x5955,
729 0x595a, 0x5958, 0x530f, 0x5c22, 0x5c25, 0x5c2c, 0x5c34, 0x624c,
730 0x626a, 0x629f, 0x62bb, 0x62ca, 0x62da, 0x62d7, 0x62ee, 0x6322,
731 0x62f6, 0x6339, 0x634b, 0x6343, 0x63ad, 0x63f6, 0x6371, 0x637a,
732 0x638e, 0x63b4, 0x636d, 0x63ac, 0x638a, 0x6369, 0x63ae, 0x63bc,
733 0x63f2, 0x63f8, 0x63e0, 0x63ff, 0x63c4, 0x63de, 0x63ce, 0x6452,
734 0x63c6, 0x63be, 0x6445, 0x6441, 0x640b, 0x641b, 0x6420, 0x640c,
735 0x6426, 0x6421, 0x645e, 0x6484, 0x646d, 0x6496,
736 /* 0x5f */
737 0x647a, 0x64b7, 0x64b8, 0x6499, 0x64ba, 0x64c0, 0x64d0, 0x64d7,
738 0x64e4, 0x64e2, 0x6509, 0x6525, 0x652e, 0x5f0b, 0x5fd2, 0x7519,
739 0x5f11, 0x535f, 0x53f1, 0x53fd, 0x53e9, 0x53e8, 0x53fb, 0x5412,
740 0x5416, 0x5406, 0x544b, 0x5452, 0x5453, 0x5454, 0x5456, 0x5443,
741 0x5421, 0x5457, 0x5459, 0x5423, 0x5432, 0x5482, 0x5494, 0x5477,
742 0x5471, 0x5464, 0x549a, 0x549b, 0x5484, 0x5476, 0x5466, 0x549d,
743 0x54d0, 0x54ad, 0x54c2, 0x54b4, 0x54d2, 0x54a7, 0x54a6, 0x54d3,
744 0x54d4, 0x5472, 0x54a3, 0x54d5, 0x54bb, 0x54bf, 0x54cc, 0x54d9,
745 0x54da, 0x54dc, 0x54a9, 0x54aa, 0x54a4, 0x54dd, 0x54cf, 0x54de,
746 0x551b, 0x54e7, 0x5520, 0x54fd, 0x5514, 0x54f3, 0x5522, 0x5523,
747 0x550f, 0x5511, 0x5527, 0x552a, 0x5567, 0x558f, 0x55b5, 0x5549,
748 0x556d, 0x5541, 0x5555, 0x553f, 0x5550, 0x553c,
749 /* 0x60 */
750 0x5537, 0x5556, 0x5575, 0x5576, 0x5577, 0x5533, 0x5530, 0x555c,
751 0x558b, 0x55d2, 0x5583, 0x55b1, 0x55b9, 0x5588, 0x5581, 0x559f,
752 0x557e, 0x55d6, 0x5591, 0x557b, 0x55df, 0x55bd, 0x55be, 0x5594,
753 0x5599, 0x55ea, 0x55f7, 0x55c9, 0x561f, 0x55d1, 0x55eb, 0x55ec,
754 0x55d4, 0x55e6, 0x55dd, 0x55c4, 0x55ef, 0x55e5, 0x55f2, 0x55f3,
755 0x55cc, 0x55cd, 0x55e8, 0x55f5, 0x55e4, 0x8f94, 0x561e, 0x5608,
756 0x560c, 0x5601, 0x5624, 0x5623, 0x55fe, 0x5600, 0x5627, 0x562d,
757 0x5658, 0x5639, 0x5657, 0x562c, 0x564d, 0x5662, 0x5659, 0x565c,
758 0x564c, 0x5654, 0x5686, 0x5664, 0x5671, 0x566b, 0x567b, 0x567c,
759 0x5685, 0x5693, 0x56af, 0x56d4, 0x56d7, 0x56dd, 0x56e1, 0x56f5,
760 0x56eb, 0x56f9, 0x56ff, 0x5704, 0x570a, 0x5709, 0x571c, 0x5e0f,
761 0x5e19, 0x5e14, 0x5e11, 0x5e31, 0x5e3b, 0x5e3c,
762 /* 0x61 */
763 0x5e37, 0x5e44, 0x5e54, 0x5e5b, 0x5e5e, 0x5e61, 0x5c8c, 0x5c7a,
764 0x5c8d, 0x5c90, 0x5c96, 0x5c88, 0x5c98, 0x5c99, 0x5c91, 0x5c9a,
765 0x5c9c, 0x5cb5, 0x5ca2, 0x5cbd, 0x5cac, 0x5cab, 0x5cb1, 0x5ca3,
766 0x5cc1, 0x5cb7, 0x5ccc4, 0x5cd2, 0x5ce4, 0x5ccb, 0x5cee5, 0x5cd02,
767 0x5d03, 0x5d27, 0x5d26, 0x5d2e, 0x5d24, 0x5d1e, 0x5d06, 0x5d1b,
768 0x5d58, 0x5d3e, 0x5d34, 0x5d3d, 0x5d6c, 0x5d5b, 0x5d6f, 0x5d5d,
769 0x5d6b, 0x5d4b, 0x5d4a, 0x5d69, 0x5d74, 0x5d82, 0x5d99, 0x5d9d,
770 0x8c73, 0x5db7, 0x5dc5, 0x5f73, 0x5f77, 0x5f82, 0x5f87, 0x5f89,
771 0x5f8c, 0x5f95, 0x5f99, 0x5f9c, 0x5fa8, 0x5fad, 0x5fb5, 0x5fbc,
772 0x8862, 0x5f61, 0x72ad, 0x72b0, 0x72b4, 0x72b7, 0x72b8, 0x72c3,
773 0x72c1, 0x72ce, 0x72cd, 0x72d2, 0x72e8, 0x72ef, 0x72e9, 0x72f2,
774 0x72f4, 0x72f7, 0x7301, 0x72f3, 0x7303, 0x72fa,
775 /* 0x62 */
776 0x72fb, 0x7317, 0x7313, 0x7321, 0x730a, 0x731e, 0x731d, 0x7315,
777 0x7322, 0x7339, 0x7325, 0x732c, 0x7338, 0x7331, 0x7350, 0x734d,
778 0x7357, 0x7360, 0x736c, 0x736f, 0x737e, 0x821b, 0x5925, 0x98e7,
779 0x5924, 0x5902, 0x9963, 0x9967, 0x9968, 0x9969, 0x996a, 0x996b,
780 0x996c, 0x9974, 0x9977, 0x997d, 0x9980, 0x9984, 0x9987, 0x998a,
781 0x998d, 0x9990, 0x9991, 0x9993, 0x9994, 0x9995, 0x5e80, 0x5e91,
782 0x5e8b, 0x5e96, 0x5ea5, 0x5ea0, 0x5eb9, 0x5eb5, 0x5ebe, 0x5eb3,
783 0x8d53, 0x5ed2, 0x5ed1, 0x5edb, 0x5ee8, 0x5eea, 0x81ba, 0x5fc4,
784 0x5fc9, 0x5fd6, 0x5fcf, 0x6003, 0x5fee, 0x6004, 0x5fe1, 0x5fe4,
785 0x5ffe, 0x6005, 0x6006, 0x5fea, 0x5fed, 0x5fff8, 0x6019, 0x6035,
786 0x6026, 0x601b, 0x600f, 0x600d, 0x6029, 0x602b, 0x600a, 0x603f,
787 0x6021, 0x6078, 0x6079, 0x607b, 0x607a, 0x6042,
788 /* 0x63 */
789 0x606a, 0x607d, 0x6096, 0x609a, 0x60ad, 0x609d, 0x6083, 0x6092,
790 0x608c, 0x609b, 0x60ec, 0x60bb, 0x60b1, 0x60dd, 0x60d8, 0x60c6,
```

791 0x60da, 0x60b4, 0x6120, 0x6126, 0x6115, 0x6123, 0x60f4, 0x6100,
792 0x610e, 0x612b, 0x614a, 0x6175, 0x61ac, 0x6194, 0x61a7, 0x61b7,
793 0x61d4, 0x61f5, 0x5fdd, 0x96b3, 0x95e9, 0x95eb, 0x95f1, 0x95f3,
794 0x95f5, 0x95f6, 0x95fc, 0x95fe, 0x9603, 0x9604, 0x9606, 0x9608,
795 0x960a, 0x960b, 0x960c, 0x960d, 0x960f, 0x9612, 0x9615, 0x9616,
796 0x9617, 0x9619, 0x961a, 0x4e2c, 0x723f, 0x6215, 0x6c35, 0x6c54,
797 0x6c5c, 0x6c4a, 0x6ca3, 0x6c85, 0x6c90, 0x6c94, 0x6c8c, 0x6c68,
798 0x6c69, 0x6c74, 0x6c76, 0x6c86, 0x6ca9, 0x6cd0, 0x6cd4, 0x6cad,
799 0x6cf7, 0x6cf8, 0x6cf1, 0x6cd7, 0x6cb2, 0x6ce0, 0x6cd6, 0x6cfa,
800 0x6ceb, 0x6cee, 0x6cb1, 0x6cd3, 0x6cef, 0x6cfe,
801 /* 0x64 */
802 0x6d39, 0x6d27, 0x6d0c, 0x6d43, 0x6d48, 0x6d07, 0x6d04, 0x6d19,
803 0x6d0e, 0x6d2b, 0x6d4d, 0x6d2e, 0x6d35, 0x6d1a, 0x6d4f, 0x6d52,
804 0x6d54, 0x6d33, 0x6d91, 0x6d6f, 0x6d9e, 0x6da0, 0x6d5e, 0x6d93,
805 0x6d94, 0x6d5c, 0x6d60, 0x6d7c, 0x6d63, 0x6e1a, 0x6dc7, 0x6dc5,
806 0x6dde, 0x6e0e, 0x6dbf, 0x6de0, 0x6e11, 0x6de6, 0x6ddd, 0x6dd9,
807 0x6e16, 0x6dab, 0x6e0c, 0x6dae, 0x6e2b, 0x6e6e, 0x6e4e, 0x6e6b,
808 0x6eb2, 0x6e5f, 0x6e86, 0x6e53, 0x6e54, 0x6e32, 0x6e25, 0x6e44,
809 0x6edf, 0x6eb1, 0x6e98, 0x6ee0, 0x6f2d, 0x6ee2, 0x6ea5, 0x6ea7,
810 0x6ebd, 0x6ebb, 0x6eb7, 0x6ed7, 0x6eb4, 0x6ecf, 0x6e8f, 0x6ec2,
811 0x6ef9, 0x6f62, 0x6f46, 0x6f47, 0x6f24, 0x6f15, 0x6ef9, 0x6f2f,
812 0x6f36, 0x6f4b, 0x6f74, 0x6f2a, 0x6f09, 0x6f29, 0x6f89, 0x6f8d,
813 0x6f8c, 0x6f78, 0x6f72, 0x6f7c, 0x6f7a, 0x6fd1,
814 /* 0x65 */
815 0x6fc9, 0x6fa7, 0x6fb9, 0x6fb6, 0x6fc2, 0x6fe1, 0x6fee, 0x6fde,
816 0x6fe0, 0x6fef, 0x701a, 0x7023, 0x701b, 0x7039, 0x7035, 0x704f,
817 0x705e, 0x5b80, 0x5b84, 0x5b95, 0x5b93, 0x5ba5, 0x5bb8, 0x752f,
818 0x9a9e, 0x6434, 0x5be4, 0x5bee, 0x8930, 0x5bf0, 0x8e47, 0x8b07,
819 0x8fb6, 0x8fd3, 0x8fd5, 0x8fe5, 0x8fee, 0x8fe4, 0x8fe9, 0x8fe6,
820 0x8ff3, 0x8fe8, 0x9005, 0x9004, 0x900b, 0x9026, 0x9011, 0x900d,
821 0x9016, 0x9021, 0x9035, 0x9036, 0x902d, 0x902f, 0x9044, 0x9051,
822 0x9052, 0x9050, 0x9068, 0x9058, 0x9062, 0x905b, 0x66b9, 0x9074,
823 0x907d, 0x9082, 0x9088, 0x9083, 0x908b, 0x5f50, 0x5f57, 0x5f56,
824 0x5f58, 0x5c3b, 0x54ab, 0x5c50, 0x5c59, 0x5b71, 0x5c63, 0x5c66,
825 0x7fbc, 0x5f2a, 0x5f29, 0x5f2d, 0x8274, 0x5f3c, 0x9b3b, 0x5c6e,
826 0x5981, 0x5983, 0x598d, 0x59a9, 0x59aa, 0x59a3,
827 /* 0x66 */
828 0x5997, 0x59ca, 0x59ab, 0x599e, 0x59a4, 0x59d2, 0x59b2, 0x59af,
829 0x59d7, 0x59be, 0x59a0, 0x5a05, 0x5a06, 0x59dd, 0x5a08, 0x593c, 0x59d8,
830 0x59f9, 0x5a0c, 0x5a09, 0x5a32, 0x5a34, 0x5a11, 0x5a23, 0x5a13,
831 0x5a40, 0x5a67, 0x5a4a, 0x5a55, 0x5a3c, 0x5a62, 0x5a75, 0x80ec,
832 0x5aaa, 0x5a9b, 0x5a77, 0x5a7a, 0x5abe, 0x5aeb, 0x5ab2, 0x5ad2,
833 0x5ad4, 0x5ab8, 0x5ae0, 0x5ae3, 0x5af1, 0x5ad6, 0x5ae6, 0x5ad8,
834 0x5adc, 0x5b09, 0x5b17, 0x5b16, 0x5b32, 0x5b37, 0x5b40, 0x5c15,
835 0x5c1c, 0x5b5a, 0x5b65, 0x5b73, 0x5b51, 0x5b53, 0x5b62, 0x9a75,
836 0x9a77, 0x9a78, 0x9a7a, 0x9a7e, 0x9a7d, 0x9a80, 0x9a81, 0x9a85,
837 0x9a88, 0x9a8a, 0x9a90, 0x9a92, 0x9a93, 0x9a96, 0x9a98, 0x9a9b,
838 0x9a9c, 0x9a9d, 0x9a9f, 0x9aa0, 0x9aa2, 0x9aa3, 0x9aa5, 0x9aa7,
839 0x7e9f, 0x7ea1, 0x7ea3, 0x7ea5, 0x7ea8, 0x7ea9,
840 /* 0x67 */
841 0x7ead, 0x7eb0, 0x7eb6, 0x7ec0, 0x7ec1, 0x7ec2, 0x7ec9, 0x7ecb,
842 0x7ecc, 0x7ed0, 0x7ed4, 0x7ed7, 0x7edb, 0x7ee0, 0x7ee1, 0x7ee8,
843 0x7eeb, 0x7eee, 0x7eef, 0x7ef1, 0x7ef2, 0x7f0d, 0x7ef6, 0x7efa,
844 0x7efb, 0x7efe, 0x7f01, 0x7f02, 0x7f03, 0x7f07, 0x7f08, 0x7f0b,
845 0x7f0c, 0x7f0f, 0x7f11, 0x7f12, 0x7f17, 0x7f19, 0x7f1c, 0x7f1b,
846 0x7f1f, 0x7f21, 0x7f22, 0x7f23, 0x7f24, 0x7f25, 0x7f26, 0x7f27,
847 0x7f2a, 0x7f2b, 0x7f2c, 0x7f2d, 0x7f2f, 0x7f30, 0x7f31, 0x7f32,
848 0x7f33, 0x7f35, 0x5e7a, 0x757e, 0x5ddb, 0x753e, 0x9095, 0x738e,
849 0x7391, 0x73ae, 0x73a2, 0x739f, 0x73cf, 0x73c2, 0x73d1, 0x73b7,
850 0x73b3, 0x73c0, 0x73c9, 0x73c8, 0x73e5, 0x73d9, 0x987c, 0x740a,
851 0x73e9, 0x73e7, 0x73de, 0x73ba, 0x73f2, 0x740f, 0x742a, 0x745b,
852 0x7426, 0x7425, 0x7428, 0x7430, 0x742e, 0x742c,
853 /* 0x68 */
854 0x741b, 0x741a, 0x7441, 0x745c, 0x7457, 0x7455, 0x7459, 0x7477,
855 0x746d, 0x747e, 0x749c, 0x748e, 0x7480, 0x7481, 0x7487, 0x748b,
856 0x749e, 0x74a8, 0x74a9, 0x7490, 0x74a7, 0x74d2, 0x74ba, 0x97ea,
857 0x97eb, 0x97ec, 0x674c, 0x6753, 0x675e, 0x6748, 0x6769, 0x67a5,
858 0x6787, 0x676a, 0x6773, 0x6798, 0x67a7, 0x6775, 0x67a8, 0x679e,
859 0x67ad, 0x678b, 0x6777, 0x677c, 0x67f0, 0x6809, 0x67d8, 0x680a,
860 0x67e9, 0x67b0, 0x6800, 0x67d9, 0x67b5, 0x67da, 0x67b3, 0x67dd,
861 0x6800, 0x67c3, 0x67b8, 0x67e2, 0x680e, 0x67c1, 0x67fd, 0x6832,
862 0x6833, 0x6860, 0x6861, 0x684e, 0x6862, 0x6844, 0x6864, 0x6883,
863 0x681d, 0x6855, 0x6866, 0x6841, 0x6867, 0x6840, 0x683e, 0x684a,
864 0x6849, 0x6829, 0x68b5, 0x688f, 0x6874, 0x6877, 0x6893, 0x686b,
865 0x68c2, 0x696e, 0x68fc, 0x691f, 0x6920, 0x68f9,
866 /* 0x69 */
867 0x6924, 0x68f0, 0x690b, 0x6901, 0x6957, 0x68e3, 0x6910, 0x6971,
868 0x6939, 0x6960, 0x6942, 0x695d, 0x6984, 0x696b, 0x6980, 0x6998,
869 0x6978, 0x6934, 0x69cc, 0x6987, 0x6988, 0x69ce, 0x6989, 0x6966,
870 0x6963, 0x6979, 0x699b, 0x69a7, 0x69bb, 0x69ab, 0x69ad, 0x69d4,
871 0x69b1, 0x69c1, 0x69ca, 0x69df, 0x6995, 0x69e0, 0x698d, 0x69ff,
872 0x6a2f, 0x69ed, 0x6a17, 0x6a18, 0x6a65, 0x69f2, 0x6a44, 0x6a3e,
873 0x6aa0, 0x6aa5, 0x6a5b, 0x6a35, 0x6a8e, 0x6a79, 0x6a3d, 0x6a28,
874 0x6a58, 0x6a7c, 0x6a91, 0x6a90, 0x6aa9, 0x6a97, 0x6aab, 0x7337,
875 0x7352, 0x6b81, 0x6b82, 0x6b87, 0x6b84, 0x6b92, 0x6b93, 0x6b8d,
876 0x6b9a, 0x6b9b, 0x6ba1, 0x6baa, 0x8f6b, 0x8f6d, 0x8f71, 0x8f72,
877 0x8f73, 0x8f75, 0x8f76, 0x8f78, 0x8f77, 0x8f79, 0x8f7a, 0x8f7c,

```

878 0x8f7e, 0x8f81, 0x8f82, 0x8f84, 0x8f87, 0x8f8b,
879 /* 0x6a */
880 0x8f8d, 0x8f8e, 0x8f8f, 0x8f98, 0x8f9a, 0x8ece, 0x620b, 0x6217,
881 0x621b, 0x621f, 0x6222, 0x6221, 0x6225, 0x6224, 0x622c, 0x81e7,
882 0x74ef, 0x74f4, 0x74ff, 0x750f, 0x7511, 0x7513, 0x6534, 0x65ee,
883 0x65ef, 0x65f0, 0x660a, 0x6619, 0x6772, 0x6603, 0x6615, 0x6600,
884 0x7085, 0x66f7, 0x661d, 0x6634, 0x6631, 0x6636, 0x6635, 0x8006,
885 0x665f, 0x6654, 0x6641, 0x664f, 0x6656, 0x6661, 0x6657, 0x6677,
886 0x6684, 0x668c, 0x66a7, 0x669d, 0x66be, 0x66db, 0x66dc, 0x66e6,
887 0x66e9, 0x8d32, 0x8d33, 0x8d36, 0x8d3b, 0x8d3d, 0x8d40, 0x8d45,
888 0x8d46, 0x8d48, 0x8d49, 0x8d47, 0x8d4d, 0x8d55, 0x8d59, 0x89c7,
889 0x89ca, 0x89cb, 0x89cc, 0x89ce, 0x89cf, 0x89d0, 0x89d1, 0x726e,
890 0x729f, 0x725d, 0x7266, 0x726f, 0x727e, 0x727f, 0x7284, 0x728b,
891 0x728d, 0x728f, 0x7292, 0x6308, 0x6332, 0x63b0,
892 /* 0x6b */
893 0x643f, 0x64d8, 0x8004, 0x6bea, 0x6bf3, 0x6bfd, 0x6bf5, 0x6bf9,
894 0x6c05, 0x6c07, 0x6c06, 0x6c0d, 0x6c15, 0x6c18, 0x6c19, 0x6c1a,
895 0x6c21, 0x6c29, 0x6c29, 0x6c24, 0x6c2a, 0x6c32, 0x6535, 0x6555, 0x656b,
896 0x724d, 0x7252, 0x7256, 0x7230, 0x8662, 0x5216, 0x809f, 0x809c,
897 0x8093, 0x80bc, 0x670a, 0x80bd, 0x80b1, 0x80ab, 0x80ad, 0x80b4,
898 0x80b7, 0x80e7, 0x80e8, 0x80e9, 0x80ea, 0x80db, 0x80c2, 0x80c4,
899 0x80d9, 0x80cd, 0x80d7, 0x6710, 0x80dd, 0x80eb, 0x80f1, 0x80f4,
900 0x80ed, 0x810d, 0x810e, 0x80f2, 0x80fc, 0x6715, 0x8112, 0x8c5a,
901 0x8136, 0x811e, 0x812c, 0x8118, 0x8132, 0x8148, 0x814c, 0x8153,
902 0x8174, 0x8159, 0x815a, 0x8171, 0x8160, 0x8169, 0x817c, 0x817d,
903 0x816d, 0x8167, 0x584d, 0x5ab5, 0x8188, 0x8182, 0x8191, 0x6ed5,
904 0x81a3, 0x81aa, 0x81cc, 0x6726, 0x81ca, 0x81bb,
905 /* 0x6c */
906 0x81c1, 0x81a6, 0x6b24, 0x6b37, 0x6b39, 0x6b43, 0x6b46, 0x6b59,
907 0x98d1, 0x98d2, 0x98d3, 0x98d5, 0x98d9, 0x98da, 0x6bb3, 0x5f40,
908 0x6bc2, 0x89f3, 0x6590, 0x9f51, 0x6593, 0x65bc, 0x65c6, 0x65c4,
909 0x65c3, 0x65cc, 0x65ce, 0x65d2, 0x65d6, 0x7080, 0x709c, 0x7096,
910 0x709d, 0x70bb, 0x70c0, 0x70b7, 0x70ab, 0x70b1, 0x70e8, 0x70ca,
911 0x7110, 0x7113, 0x7116, 0x712f, 0x7131, 0x7173, 0x715c, 0x7168,
912 0x7145, 0x7172, 0x714a, 0x7178, 0x717a, 0x7198, 0x71b3, 0x71b5,
913 0x71a8, 0x71a0, 0x71e0, 0x71d4, 0x71e7, 0x71f9, 0x721d, 0x7228,
914 0x706c, 0x7118, 0x7166, 0x71b9, 0x623e, 0x623d, 0x6243, 0x6248,
915 0x6249, 0x793b, 0x7940, 0x7946, 0x7949, 0x795b, 0x795c, 0x7953,
916 0x795a, 0x7962, 0x7957, 0x7960, 0x796f, 0x7967, 0x797a, 0x7985,
917 0x798a, 0x799a, 0x79a7, 0x79b3, 0x5fd1, 0x5fd0,
918 /* 0x6d */
919 0x603c, 0x605d, 0x605a, 0x6067, 0x6041, 0x6059, 0x6063, 0x60ab,
920 0x6106, 0x610d, 0x615d, 0x61a9, 0x619d, 0x61cb, 0x61d1, 0x6206,
921 0x8080, 0x807f, 0x6c93, 0x6cf6, 0x6dfc, 0x77f6, 0x77f8, 0x7800,
922 0x7809, 0x7817, 0x7818, 0x7818, 0x65ab, 0x782d, 0x781c, 0x781d,
923 0x7839, 0x783a, 0x783b, 0x781f, 0x783c, 0x7825, 0x782c, 0x7823,
924 0x7829, 0x784e, 0x786d, 0x7856, 0x7857, 0x7826, 0x7850, 0x7847,
925 0x784c, 0x786a, 0x789b, 0x7893, 0x789a, 0x7887, 0x789c, 0x78a1,
926 0x78a3, 0x78b2, 0x78b9, 0x78a5, 0x78d4, 0x78d9, 0x78c9, 0x78ec,
927 0x78f2, 0x7905, 0x78f4, 0x7913, 0x7924, 0x791e, 0x7934, 0x9f9b,
928 0x9ef9, 0x9efb, 0x9efc, 0x76f1, 0x7704, 0x770d, 0x76f9, 0x7707,
929 0x7708, 0x771a, 0x7722, 0x7719, 0x772d, 0x7726, 0x7735, 0x7738,
930 0x7750, 0x7751, 0x7747, 0x7743, 0x775a, 0x7768,
931 /* 0x6e */
932 0x7762, 0x7765, 0x777f, 0x778d, 0x777d, 0x7780, 0x778c, 0x7791,
933 0x779f, 0x77a0, 0x77b0, 0x77b5, 0x77bd, 0x753a, 0x7540, 0x754e,
934 0x754b, 0x7548, 0x755b, 0x7572, 0x7579, 0x7583, 0x7f58, 0x7f61,
935 0x7f5f, 0x8a48, 0x7f68, 0x7f74, 0x7f71, 0x7f79, 0x7f81, 0x7f7e,
936 0x76cd, 0x76e5, 0x8832, 0x9485, 0x9486, 0x9487, 0x948b, 0x948a,
937 0x948c, 0x948d, 0x948f, 0x9490, 0x9494, 0x9497, 0x9495, 0x949a,
938 0x949b, 0x949c, 0x94a3, 0x94a4, 0x94ab, 0x94aa, 0x94ad, 0x94ac,
939 0x94af, 0x94b0, 0x94b2, 0x94b4, 0x94b6, 0x94b7, 0x94b8, 0x94b9,
940 0x94ba, 0x94bc, 0x94bd, 0x94bf, 0x94c4, 0x94c8, 0x94c9, 0x94ca,
941 0x94cb, 0x94cc, 0x94cd, 0x94ce, 0x94d0, 0x94d1, 0x94d2, 0x94d5,
942 0x94d6, 0x94d7, 0x94d9, 0x94d8, 0x94db, 0x94de, 0x94df, 0x94e0,
943 0x94e2, 0x94e4, 0x94e5, 0x94e7, 0x94e8, 0x94ea,
944 /* 0x6f */
945 0x94e9, 0x94eb, 0x94ee, 0x94ef, 0x94f3, 0x94f4, 0x94f5, 0x94f7,
946 0x94f9, 0x94fc, 0x94fd, 0x94ff, 0x9503, 0x9502, 0x9506, 0x9507,
947 0x9509, 0x950a, 0x950d, 0x950e, 0x950f, 0x9512, 0x9513, 0x9514,
948 0x9515, 0x9516, 0x9518, 0x9518, 0x951b, 0x951d, 0x951e, 0x951f, 0x9522,
949 0x952a, 0x952b, 0x9529, 0x952c, 0x9531, 0x9532, 0x9534, 0x9536,
950 0x9537, 0x9538, 0x953c, 0x953e, 0x953f, 0x9542, 0x9535, 0x9544,
951 0x9545, 0x9546, 0x9549, 0x954c, 0x954e, 0x954f, 0x9552, 0x9553,
952 0x9554, 0x9556, 0x9557, 0x9558, 0x9559, 0x955b, 0x955e, 0x955f,
953 0x955d, 0x9561, 0x9562, 0x9564, 0x9565, 0x9566, 0x9567, 0x9568,
954 0x9569, 0x956a, 0x956b, 0x956c, 0x956f, 0x9571, 0x9572, 0x9573,
955 0x953a, 0x77e7, 0x77ec, 0x96c9, 0x79d5, 0x79ed, 0x79e3, 0x79eb,
956 0x7a06, 0x5d47, 0x7a03, 0x7a02, 0x7a1e, 0x7a14,
957 /* 0x70 */
958 0x7a39, 0x7a37, 0x7a51, 0x9ecf, 0x99a5, 0x7a70, 0x7688, 0x768e,
959 0x7693, 0x7699, 0x76a4, 0x74de, 0x74e0, 0x752c, 0x9e20, 0x9e22,
960 0x9e28, 0x9e29, 0x9e2a, 0x9e2b, 0x9e2c, 0x9e32, 0x9e31, 0x9e36,
961 0x9e38, 0x9e37, 0x9e39, 0x9e3a, 0x9e3e, 0x9e41, 0x9e42, 0x9e44,
962 0x9e46, 0x9e47, 0x9e48, 0x9e49, 0x9e4b, 0x9e4c, 0x9e4e, 0x9e51,
963 0x9e55, 0x9e57, 0x9e5a, 0x9e5b, 0x9e5c, 0x9e5e, 0x9e63, 0x9e66,
964 0x9e67, 0x9e68, 0x9e69, 0x9e6a, 0x9e6b, 0x9e6c, 0x9e71, 0x9e6d,

```

```

965  0x9e73, 0x7592, 0x7594, 0x7596, 0x75a0, 0x759d, 0x75ac, 0x75a3,
966  0x75b3, 0x75b4, 0x75b8, 0x75c4, 0x75b1, 0x75b0, 0x75c3, 0x75c2,
967  0x75d6, 0x75cd, 0x75e3, 0x75e8, 0x75e6, 0x75e4, 0x75eb, 0x75e7,
968  0x7603, 0x75f1, 0x75fc, 0x75ff, 0x7610, 0x7600, 0x7605, 0x760c,
969  0x7617, 0x760a, 0x7625, 0x7618, 0x7615, 0x7619,
970  /* 0x71 */
971  0x761b, 0x763c, 0x7622, 0x7620, 0x7640, 0x762d, 0x7630, 0x763f,
972  0x7635, 0x7643, 0x763e, 0x7633, 0x764d, 0x765e, 0x7654, 0x765c,
973  0x7656, 0x766b, 0x766f, 0x7fca, 0x7ae6, 0x7a78, 0x7a79, 0x7a80,
974  0x7a86, 0x7a88, 0x7a95, 0x7aa6, 0x7aa0, 0x7aac, 0x7aa8, 0x7aad,
975  0x7ab3, 0x8864, 0x8869, 0x8872, 0x887d, 0x887f, 0x8882, 0x88a2,
976  0x88c6, 0x88b7, 0x88bc, 0x88c9, 0x88e2, 0x88ce, 0x88e3, 0x88e5,
977  0x88f1, 0x891a, 0x88fc, 0x88e8, 0x88fe, 0x88f0, 0x8921, 0x8919,
978  0x8913, 0x891b, 0x890a, 0x8934, 0x892b, 0x8936, 0x8941, 0x8966,
979  0x897b, 0x758b, 0x80e5, 0x76b2, 0x76b4, 0x77dc, 0x8012, 0x8014,
980  0x8016, 0x801c, 0x8020, 0x8022, 0x8025, 0x8026, 0x8027, 0x8029,
981  0x8028, 0x8031, 0x800b, 0x8035, 0x8043, 0x8046, 0x804d, 0x8052,
982  0x8069, 0x8071, 0x8983, 0x9878, 0x9880, 0x9883,
983  /* 0x72 */
984  0x9889, 0x988c, 0x988d, 0x988f, 0x9894, 0x989a, 0x989b, 0x989e,
985  0x989f, 0x98a1, 0x98a2, 0x98a5, 0x98a6, 0x864d, 0x8654, 0x866c,
986  0x866e, 0x867f, 0x867a, 0x867c, 0x867b, 0x86a8, 0x868d, 0x868b,
987  0x86ac, 0x869d, 0x86a7, 0x86a3, 0x86aa, 0x8693, 0x86a9, 0x86b6,
988  0x86c4, 0x86b5, 0x86ce, 0x86b0, 0x86ba, 0x86b1, 0x86af, 0x86c9,
989  0x86cf, 0x86b4, 0x86e9, 0x86f1, 0x86f2, 0x86ed, 0x86f3, 0x86d0,
990  0x8713, 0x86de, 0x86f4, 0x86df, 0x86d8, 0x86d1, 0x8703, 0x8707,
991  0x86f8, 0x8708, 0x870a, 0x870d, 0x8709, 0x8723, 0x873b, 0x871e,
992  0x8725, 0x872e, 0x871a, 0x873e, 0x8748, 0x8734, 0x8731, 0x8729,
993  0x8737, 0x873f, 0x8782, 0x8722, 0x877d, 0x877e, 0x877b, 0x8760,
994  0x8770, 0x874c, 0x876e, 0x878b, 0x8753, 0x8763, 0x877c, 0x8764,
995  0x8759, 0x8765, 0x8793, 0x87af, 0x87a8, 0x87d2,
996  /* 0x73 */
997  0x87c6, 0x8788, 0x8785, 0x87ad, 0x8797, 0x8783, 0x87ab, 0x87e5,
998  0x87ac, 0x87b5, 0x87b3, 0x87cb, 0x87d3, 0x87bd, 0x87d1, 0x87c0,
999  0x87ca, 0x87db, 0x87ea, 0x87e0, 0x87ee, 0x8816, 0x8813, 0x87fe,
1000  0x880a, 0x881b, 0x8821, 0x8839, 0x883c, 0x7f36, 0x7f42, 0x7f44,
1001  0x7f45, 0x8210, 0x7afa, 0x7afd, 0x7b08, 0x7b03, 0x7b04, 0x7b15,
1002  0x7b0a, 0x7b2b, 0x7b0f, 0x7b47, 0x7b38, 0x7b2a, 0x7b19, 0x7b2e,
1003  0x7b31, 0x7b20, 0x7b25, 0x7b24, 0x7b33, 0x7b3e, 0x7b1e, 0x7b58,
1004  0x7b5a, 0x7b45, 0x7b75, 0x7b4c, 0x7b5d, 0x7b60, 0x7b6e, 0x7b7b,
1005  0x7b62, 0x7b72, 0x7b71, 0x7b90, 0x7ba6, 0x7ba7, 0x7bb8, 0x7bac,
1006  0x7b9d, 0x7ba8, 0x7b85, 0x7baa, 0x7b9c, 0x7ba2, 0x7bab, 0x7bb4,
1007  0x7bd1, 0x7bc1, 0x7bcc, 0x7bdd, 0x7bda, 0x7be5, 0x7be6, 0x7bea,
1008  0x7c0c, 0x7bfe, 0x7bfc, 0x7c0f, 0x7c16, 0x7c0b,
1009  /* 0x74 */
1010  0x7c1f, 0x7c2a, 0x7c26, 0x7c38, 0x7c41, 0x7c40, 0x81fe, 0x8201,
1011  0x8202, 0x8204, 0x81ec, 0x8844, 0x8221, 0x8222, 0x8223, 0x822d,
1012  0x822f, 0x8228, 0x822b, 0x8238, 0x823b, 0x8233, 0x8234, 0x823e,
1013  0x8244, 0x8249, 0x824b, 0x824f, 0x825a, 0x825f, 0x8268, 0x887e,
1014  0x8885, 0x8888, 0x88d8, 0x88df, 0x895e, 0x7f9d, 0x7f9f, 0x7fa7,
1015  0x7faf, 0x7fb0, 0x7fb2, 0x7c7c, 0x6549, 0x7c91, 0x7c9d, 0x7c9c,
1016  0x7c9e, 0x7ca2, 0x7cb2, 0x7cbc, 0x7cbd, 0x7cc1, 0x7cc7, 0x7ccc,
1017  0x7ccd, 0x7cc8, 0x7cc5, 0x7cd7, 0x7ce8, 0x826e, 0x66a8, 0x7fbf,
1018  0x7fce, 0x7fd5, 0x7fe5, 0x7fe1, 0x7fe6, 0x7fe9, 0x7fee, 0x7ff3,
1019  0x7cf8, 0x7d77, 0x7da6, 0x7dae, 0x7e47, 0x7e9b, 0x9eb8, 0x9eb4,
1020  0x8d73, 0x8d84, 0x8d94, 0x8d91, 0x8db1, 0x8d67, 0x8d6d, 0x8c47,
1021  0x8c49, 0x914a, 0x9150, 0x914e, 0x914f, 0x9164,
1022  /* 0x75 */
1023  0x9162, 0x9161, 0x9170, 0x9169, 0x916f, 0x917d, 0x917e, 0x9172,
1024  0x9174, 0x9179, 0x918c, 0x9185, 0x9190, 0x918d, 0x9191, 0x91a2,
1025  0x91a3, 0x91aa, 0x91ad, 0x91ae, 0x91af, 0x91b5, 0x91b4, 0x91ba,
1026  0x8c55, 0x9e7e, 0x8db8, 0x8deb, 0x8e05, 0x8e59, 0x8e69, 0x8db5,
1027  0x8dbf, 0x8dbc, 0x8dba, 0x8dd4, 0x8dd7, 0x8dda, 0x8dde,
1028  0x8dce, 0x8dcf, 0x8ddb, 0x8dc6, 0x8dec, 0x8df7, 0x8df8, 0x8de3,
1029  0x8df9, 0x8dfb, 0x8de4, 0x8e09, 0x8dfd, 0x8e14, 0x8e1d, 0x8e1f,
1030  0x8e2c, 0x8e2e, 0x8e23, 0x8e2f, 0x8e3a, 0x8e40, 0x8e39, 0x8e35,
1031  0x8e3d, 0x8e31, 0x8e49, 0x8e41, 0x8e42, 0x8e51, 0x8e52, 0x8e4a,
1032  0x8e70, 0x8e76, 0x8e7c, 0x8e6f, 0x8e74, 0x8e85, 0x8e8f, 0x8e94,
1033  0x8e90, 0x8e9c, 0x8e9e, 0x8c78, 0x8c82, 0x8c8a, 0x8c85, 0x8c98,
1034  0x8c94, 0x659b, 0x89d6, 0x89de, 0x89da, 0x89dc,
1035  /* 0x76 */
1036  0x89e5, 0x89eb, 0x89ef, 0x8a3e, 0x8b26, 0x9753, 0x96e9, 0x96f3,
1037  0x96ef, 0x9706, 0x9701, 0x9708, 0x970f, 0x970e, 0x972a, 0x972d,
1038  0x9730, 0x973e, 0x9f80, 0x9f83, 0x9f85, 0x9f86, 0x9f87, 0x9f88,
1039  0x9f89, 0x9f8a, 0x9f8c, 0x9efe, 0x9f0b, 0x9f0d, 0x96b9, 0x96bc,
1040  0x96bd, 0x96ce, 0x96d2, 0x77bf, 0x96e0, 0x928e, 0x92ae, 0x92c8,
1041  0x933e, 0x936a, 0x93ca, 0x938f, 0x943e, 0x946b, 0x9c7f, 0x9c82,
1042  0x9c85, 0x9c86, 0x9c87, 0x9c88, 0x7a23, 0x9c8b, 0x9c8e, 0x9c90,
1043  0x9c91, 0x9c92, 0x9c94, 0x9c95, 0x9c9a, 0x9c9b, 0x9c9e, 0x9c9f,
1044  0x9ca0, 0x9ca1, 0x9ca2, 0x9ca3, 0x9ca5, 0x9ca6, 0x9ca7, 0x9ca8,
1045  0x9ca9, 0x9cab, 0x9cad, 0x9cae, 0x9cb0, 0x9cb1, 0x9cb2, 0x9cb3,
1046  0x9cb4, 0x9cb5, 0x9cb6, 0x9cb7, 0x9cba, 0x9cbb, 0x9cbc, 0x9cbd,
1047  0x9cc4, 0x9cc5, 0x9cc6, 0x9cc7, 0x9cca, 0x9ccb,
1048  /* 0x77 */
1049  0x9ccc, 0x9ccd, 0x9cce, 0x9ccf, 0x9cd0, 0x9cd3, 0x9cd4, 0x9cd5,
1050  0x9cd7, 0x9cd8, 0x9cd9, 0x9cdc, 0x9cdd, 0x9cdf, 0x9ce2, 0x977c,
1051  0x9785, 0x9791, 0x9792, 0x9794, 0x97af, 0x97ab, 0x97a3, 0x97b2,

```

```
1052 0x97b4, 0x9ab1, 0x9ab0, 0x9ab7, 0x9e58, 0x9ab6, 0x9aba, 0x9abc,
1053 0x9ac1, 0x9ac0, 0x9ac5, 0x9ac2, 0x9acb, 0x9acc, 0x9ad1, 0x9b45,
1054 0x9b43, 0x9b47, 0x9b49, 0x9b48, 0x9b4d, 0x9b51, 0x9e88, 0x990d,
1055 0x992e, 0x9955, 0x9954, 0x9adf, 0x9aef, 0x9ae6, 0x9aef, 0x9aeb,
1056 0x9afb, 0x9aed, 0x9af9, 0x9b08, 0x9b0f, 0x9b13, 0x9b1f, 0x9b23,
1057 0x9ebd, 0x9ebe, 0x9e3b, 0x9e82, 0x9e87, 0x9e88, 0x9e8b, 0x9e92,
1058 0x93d6, 0x9e9d, 0x9e9f, 0x9edb, 0x9edc, 0x9edd, 0x9ee0, 0x9edf,
1059 0x9ee2, 0x9ee9, 0x9ee7, 0x9ee5, 0x9eea, 0x9eef, 0x9f22, 0x9f2c,
1060 0x9f2f, 0x9f39, 0x9f37, 0x9f3d, 0x9f3e, 0x9f44,
1061 };
1062
1063 static int
1064 gb2312_mbtowc (conv_t conv, ucs4_t *pwc, const unsigned char *s, int n)
1065 {
1066     unsigned char c1 = (s[0] & 0x7f);
1067     if ((c1 >= 0x21 && c1 <= 0x29) || (c1 >= 0x30 && c1 <= 0x77)) {
1068         if (n >= 2) {
1069             unsigned char c2 = (s[1] & 0x7f);
1070             if (c2 >= 0x21 && c2 < 0x7f) {
1071                 unsigned int i = 94 * (c1 - 0x21) + (c2 - 0x21);
1072                 unsigned short wc = 0xffffd;
1073                 if (i < 1410) {
1074                     if (i < 831)
1075                         wc = gb2312_2uni_page21[i];
1076                 } else {
1077                     if (i < 8178)
1078                         wc = gb2312_2uni_page30[i-1410];
1079                 }
1080                 if (wc != 0xffffd) {
1081                     *pwc = (ucs4_t) wc;
1082                     return 2;
1083                 }
1084             }
1085             return RET_ILSEQ;
1086         }
1087         return RET_TOOFEW(0);
1088     }
1089     return RET_ILSEQ;
1090 }
1091 #endif /* NEED_TOWC */
1092
1093 #ifdef NEED_TOMB
1094 static const unsigned short gb2312_2charset[7445] = {
1095 0x2168, 0x216c, 0x2127, 0x2163, 0x2140, 0x2141, 0x2824, 0x2822,
1096 0x2828, 0x2826, 0x283a, 0x282c, 0x282a, 0x2830, 0x282e, 0x2142,
1097 0x2834, 0x2832, 0x2839, 0x2821, 0x2825, 0x2827, 0x2829, 0x282d,
1098 0x2831, 0x2823, 0x282b, 0x282f, 0x2833, 0x2835, 0x2836, 0x2837,
1099 0x2838, 0x2126, 0x2125, 0x2621, 0x2622, 0x2623, 0x2624, 0x2625,
1100 0x2626, 0x2627, 0x2628, 0x2629, 0x262a, 0x262b, 0x262c, 0x262d,
1101 0x262e, 0x262f, 0x2630, 0x2631, 0x2632, 0x2633, 0x2634, 0x2635,
1102 0x2636, 0x2637, 0x2638, 0x2639, 0x2641, 0x2642, 0x2643, 0x2644, 0x2645,
1103 0x2646, 0x2647, 0x2648, 0x2649, 0x264a, 0x264b, 0x264c, 0x264d,
1104 0x264e, 0x264f, 0x2650, 0x2651, 0x2652, 0x2653, 0x2654, 0x2655,
1105 0x2656, 0x2657, 0x2658, 0x2727, 0x2721, 0x2722, 0x2723, 0x2724,
1106 0x2725, 0x2726, 0x2728, 0x2729, 0x272a, 0x272b, 0x272c, 0x272d,
1107 0x272e, 0x272f, 0x2730, 0x2731, 0x2732, 0x2733, 0x2734, 0x2735,
1108 0x2736, 0x2737, 0x2738, 0x2739, 0x273a, 0x273b, 0x273c, 0x273d,
1109 0x273e, 0x273f, 0x2740, 0x2741, 0x2751, 0x2752, 0x2753, 0x2754,
1110 0x2755, 0x2756, 0x2758, 0x2759, 0x275a, 0x275b, 0x275c, 0x275d,
1111 0x275e, 0x275f, 0x2760, 0x2761, 0x2762, 0x2763, 0x2764, 0x2765,
1112 0x2766, 0x2767, 0x2768, 0x2769, 0x276a, 0x276b, 0x276c, 0x276d,
1113 0x276e, 0x276f, 0x2770, 0x2771, 0x2757, 0x212a, 0x212c, 0x212e,
1114 0x212f, 0x2130, 0x2131, 0x2132, 0x216b, 0x2164, 0x2165, 0x2179,
1115 0x2166, 0x216d, 0x2271, 0x2272, 0x2273, 0x2274, 0x2275, 0x2276,
1116 0x2277, 0x2278, 0x2279, 0x227a, 0x227b, 0x227c, 0x217b, 0x217c,
1117 0x217a, 0x217d, 0x214a, 0x2147, 0x2146, 0x214c, 0x2158, 0x215e,
1118 0x214f, 0x214e, 0x2144, 0x2145, 0x2149, 0x2148, 0x2152, 0x2153,
1119 0x2160, 0x215f, 0x2143, 0x214b, 0x2157, 0x2156, 0x2155, 0x2159,
1120 0x2154, 0x215c, 0x215d, 0x215a, 0x215b, 0x2151, 0x214d, 0x2150,
1121 0x2259, 0x225a, 0x225b, 0x225c, 0x225d, 0x225e, 0x225f, 0x2260,
1122 0x2261, 0x2262, 0x2245, 0x2246, 0x2247, 0x2248, 0x2249, 0x224a,
1123 0x224b, 0x224c, 0x224d, 0x224e, 0x224f, 0x2250, 0x2251, 0x2252,
1124 0x2253, 0x2254, 0x2255, 0x2256, 0x2257, 0x2258, 0x2231, 0x2232,
1125 0x2233, 0x2234, 0x2235, 0x2236, 0x2237, 0x2238, 0x2239, 0x223a,
1126 0x223b, 0x223c, 0x223d, 0x223e, 0x223f, 0x2240, 0x2241, 0x2242,
1127 0x2243, 0x2244, 0x2924, 0x2925, 0x2926, 0x2927, 0x2928, 0x2929,
1128 0x292a, 0x292b, 0x292c, 0x292d, 0x292e, 0x292f, 0x2930, 0x2931,
1129 0x2932, 0x2933, 0x2934, 0x2935, 0x2936, 0x2937, 0x2938, 0x2939,
1130 0x293a, 0x293b, 0x293c, 0x293d, 0x293e, 0x293f, 0x2940, 0x2941,
1131 0x2942, 0x2943, 0x2944, 0x2945, 0x2946, 0x2947, 0x2948, 0x2949,
1132 0x294a, 0x294b, 0x294c, 0x294d, 0x294e, 0x294f, 0x2950, 0x2951,
1133 0x2952, 0x2953, 0x2954, 0x2955, 0x2956, 0x2957, 0x2958, 0x2959,
1134 0x295a, 0x295b, 0x295c, 0x295d, 0x295e, 0x295f, 0x2960, 0x2961,
1135 0x2962, 0x2963, 0x2964, 0x2965, 0x2966, 0x2967, 0x2968, 0x2969,
1136 0x296a, 0x296b, 0x296c, 0x296d, 0x296e, 0x296f, 0x2176, 0x2175,
1137 0x2178, 0x2177, 0x2174, 0x2173, 0x2170, 0x2172, 0x2171, 0x216f,
1138 0x216e, 0x2162, 0x2161, 0x2121, 0x2122, 0x2123, 0x2128, 0x2129,
```



```

1400 0x632f, 0x6331, 0x4f27, 0x632e, 0x4e29, 0x3b5d, 0x356b, 0x3e65,
1401 0x3252, 0x334d, 0x3139, 0x632b, 0x3251, 0x352c, 0x395f, 0x3668,
1402 0x4f6b, 0x6337, 0x3b4c, 0x4847, 0x504a, 0x6338, 0x336e, 0x6d29,
1403 0x537a, 0x5364, 0x6d2a, 0x6339, 0x5262, 0x6335, 0x535e, 0x3850,
1404 0x6333, 0x6336, 0x375f, 0x6334, 0x4022, 0x633a, 0x5438, 0x3448,
1405 0x633b, 0x3b45, 0x4977, 0x4965, 0x443d, 0x6d2b, 0x427d, 0x3b5b,
1406 0x3f2e, 0x4e3f, 0x633c, 0x3f36, 0x316f, 0x5477, 0x633e, 0x6d2d,
1407 0x633f, 0x3a29, 0x6d2c, 0x633d, 0x6340, 0x3a36, 0x362e, 0x5038,
1408 0x3043, 0x6d2e, 0x6d2f, 0x4041, 0x6341, 0x4533, 0x6342, 0x5c32,
1409 0x6d30, 0x386a, 0x4e6c, 0x6a27, 0x5067, 0x4a79, 0x4856, 0x4f37,
1410 0x3349, 0x4e52, 0x3d64, 0x635e, 0x3b72, 0x6a28, 0x553d, 0x465d,
1411 0x6a29, 0x6a2a, 0x6a2c, 0x6a2b, 0x6a2e, 0x6a2d, 0x3d58, 0x6a2f,
1412 0x423e, 0x3441, 0x3477, 0x3b27, 0x6c66, 0x6c65, 0x373f, 0x4b79,
1413 0x3162, 0x6c67, 0x4948, 0x6c68, 0x6c69, 0x4a56, 0x5e50, 0x3245,
1414 0x547a, 0x464b, 0x464b, 0x3047, 0x3472, 0x4853, 0x4d50, 0x3f38, 0x3f5b,
1415 0x4724, 0x5634, 0x4029, 0x5e51, 0x4928, 0x516f, 0x4524, 0x3067,
1416 0x3336, 0x4845, 0x3062, 0x3776, 0x457a, 0x3673, 0x5552, 0x3350,
1417 0x3c3c, 0x332d, 0x3e71, 0x3051, 0x5256, 0x4a63, 0x5725, 0x4d36,
1418 0x3636, 0x3f39, 0x555b, 0x3827, 0x4557, 0x5e52, 0x3f59, 0x4255,
1419 0x4740, 0x3b24, 0x3128, 0x456a, 0x457b, 0x4c27, 0x3127, 0x3556,
1420 0x4428, 0x5e53, 0x513a, 0x3369, 0x4372, 0x3777, 0x5674, 0x3523,
1421 0x3270, 0x4434, 0x4469, 0x402d, 0x5e54, 0x3068, 0x4544, 0x4160,
1422 0x3955, 0x3e5c, 0x4d58, 0x304e, 0x4d4f, 0x5e56, 0x3e50, 0x573e,
1423 0x5e55, 0x5550, 0x305d, 0x4462, 0x4223, 0x3c70, 0x5335, 0x4039,
1424 0x4521, 0x3226, 0x5471, 0x4028, 0x4a43, 0x5e57, 0x557c, 0x3930,
1425 0x482d, 0x4b29, 0x5e59, 0x3f3d, 0x4634, 0x5727, 0x4a30, 0x4443,
1426 0x3356, 0x3952, 0x5638, 0x6a7c, 0x3034, 0x3f66, 0x4c74, 0x4d5a,
1427 0x563f, 0x424e, 0x4e4e, 0x4c22, 0x502e, 0x4453, 0x3532, 0x5e58,
1428 0x5575, 0x3c37, 0x3b53, 0x3024, 0x4532, 0x346c, 0x5571, 0x6a7d,
1429 0x5e5a, 0x4d26, 0x4d6c, 0x4e66, 0x4e66, 0x5e5c, 0x4d31, 0x4026, 0x573d,
1430 0x5e5b, 0x3046, 0x3a34, 0x4953, 0x4473, 0x3e68, 0x3236, 0x404c,
1431 0x4b70, 0x3c71, 0x3b3b, 0x3537, 0x4575, 0x5e66, 0x5e63, 0x3e5d,
1432 0x5e5f, 0x3437, 0x3d5d, 0x5e60, 0x446d, 0x4f46, 0x3560, 0x365e,
1433 0x4a5a, 0x3574, 0x5e65, 0x5546, 0x5e61, 0x4c4d, 0x467e, 0x4545,
1434 0x5234, 0x3e72, 0x4253, 0x4c3d, 0x3338, 0x3d53, 0x3f58, 0x4d46,
1435 0x515a, 0x346b, 0x5e64, 0x5e64, 0x5e5d, 0x5e67, 0x6a7e, 0x4230, 0x5e62,
1436 0x5640, 0x3527, 0x3274, 0x5e68, 0x5e72, 0x5e6d, 0x5e71, 0x4860,
1437 0x5761, 0x5e6f, 0x4368, 0x4c61, 0x3265, 0x523e, 0x5e6e, 0x5e6b,
1438 0x4e55, 0x3427, 0x3f2b, 0x3e3e, 0x3d52, 0x5e69, 0x542e, 0x5e5e,
1439 0x5e6a, 0x403f, 0x5e6c, 0x3273, 0x3869, 0x4227, 0x3d41, 0x5e75,
1440 0x5e78, 0x322b, 0x3424, 0x346a, 0x4926, 0x5e76, 0x4b51, 0x3863,
1441 0x5e77, 0x5e7a, 0x5e79, 0x4c42, 0x3061, 0x346e, 0x653a, 0x502f,
1442 0x326b, 0x6b21, 0x5e74, 0x4963, 0x5e73, 0x305a, 0x5221, 0x3177,
1443 0x4c2f, 0x5e70, 0x4b24, 0x552a, 0x5e7b, 0x345d, 0x4426, 0x5e7d,
1444 0x437e, 0x4421, 0x5f21, 0x414c, 0x5e7c, 0x3e6f, 0x4632, 0x3345,
1445 0x4876, 0x4b3a, 0x5e7e, 0x5f24, 0x5732, 0x3337, 0x4143, 0x474b,
1446 0x3225, 0x3469, 0x572b, 0x446c, 0x5f22, 0x5f23, 0x5f25, 0x3a33,
1447 0x5f26, 0x405e, 0x4943, 0x3259, 0x4766, 0x5f27, 0x475c, 0x5f28,
1448 0x6b22, 0x4b53, 0x5f2a, 0x5f29, 0x3241, 0x454a, 0x5f2b, 0x545c,
1449 0x4841, 0x5f2c, 0x3e70, 0x5f2d, 0x5627, 0x6a37, 0x6b36, 0x4a55,
1450 0x587c, 0x3844, 0x3925, 0x3745, 0x557e, 0x394a, 0x5027, 0x744d,
1451 0x3550, 0x4374, 0x3e48, 0x6b37, 0x303d, 0x3d4c, 0x4132, 0x3156,
1452 0x3328, 0x3852, 0x4922, 0x3658, 0x6b38, 0x3e34, 0x4a7d, 0x4743,
1453 0x557b, 0x3773, 0x4e44, 0x552b, 0x3173, 0x6c33, 0x305f, 0x6c35,
1454 0x3637, 0x414f, 0x757a, 0x5031, 0x5565, 0x4e53, 0x3d6f, 0x3362,
1455 0x382b, 0x5536, 0x6d3d, 0x364f, 0x4b39, 0x5042, 0x373d, 0x6c36,
1456 0x4a29, 0x4554, 0x6c39, 0x6c38, 0x4243, 0x6c37, 0x507d, 0x6c3a,
1457 0x6c3b, 0x5765, 0x6c3c, 0x6c3d, 0x466c, 0x4e5e, 0x3c48, 0x4855,
1458 0x3529, 0x3e49, 0x563c, 0x5467, 0x512e, 0x5071, 0x6a38, 0x6a39,
1459 0x6a3a, 0x3a35, 0x4a31, 0x3f75, 0x4d7a, 0x6a40, 0x303a, 0x6a3e,
1460 0x4025, 0x6a3b, 0x327d, 0x4377, 0x3b68, 0x5257, 0x4e74, 0x6a3f,
1461 0x6a3c, 0x6a43, 0x5047, 0x5333, 0x343a, 0x4341, 0x5772, 0x5551,
1462 0x4a47, 0x6a45, 0x6a44, 0x6a47, 0x6a46, 0x5667, 0x4f54, 0x6a4b,
1463 0x3b4e, 0x3d7a, 0x494e, 0x6a4c, 0x4939, 0x4f7e, 0x6a4a, 0x544e,
1464 0x6a4d, 0x6a4f, 0x4d6d, 0x6a49, 0x6a4e, 0x4e6e, 0x3b5e, 0x333f,
1465 0x4655, 0x3e30, 0x4e7a, 0x4767, 0x3e27, 0x6a50, 0x5647, 0x4140,
1466 0x545d, 0x6a51, 0x4f3e, 0x6a52, 0x4a6e, 0x452f, 0x3035, 0x6a54,
1467 0x6a53, 0x745f, 0x443a, 0x3129, 0x655f, 0x6a55, 0x4a6f, 0x6a56,
1468 0x6a57, 0x4658, 0x6a58, 0x6a59, 0x543b, 0x477a, 0x5237, 0x387c,
1469 0x6a42, 0x325c, 0x427c, 0x5478, 0x4c66, 0x576e, 0x5442, 0x5350,
1470 0x6b43, 0x4573, 0x377e, 0x6b54, 0x4b37, 0x6b5e, 0x404a, 0x4d7b,
1471 0x332f, 0x465a, 0x6b7c, 0x443e, 0x4e34, 0x4429, 0x313e, 0x547d,
1472 0x4a75, 0x566c, 0x4653, 0x3664, 0x3b7a, 0x5060, 0x4931, 0x5453,
1473 0x4828, 0x384b, 0x683e, 0x493c, 0x683b, 0x406e, 0x5053, 0x3244,
1474 0x3465, 0x683c, 0x5548, 0x3645, 0x683d, 0x4a78, 0x385c, 0x4c75,
1475 0x4034, 0x516e, 0x683f, 0x6842, 0x3a3c, 0x312d, 0x3d5c, 0x6a3d,
1476 0x6843, 0x6846, 0x684b, 0x684c, 0x684c, 0x4b49, 0x3065, 0x3c2b, 0x3939,
1477 0x6841, 0x4d77, 0x684a, 0x4e76, 0x556d, 0x4156, 0x6844, 0x4336,
1478 0x397b, 0x5626, 0x6848, 0x4a60, 0x5466, 0x6840, 0x6845, 0x6847,
1479 0x4739, 0x3763, 0x6849, 0x3f5d, 0x6852, 0x6857, 0x6855, 0x3c5c,
1480 0x3c4f, 0x685b, 0x685e, 0x685a, 0x317a, 0x3058, 0x4433, 0x384c,
1481 0x4662, 0x483e, 0x4861, 0x684f, 0x6854, 0x6856, 0x3971, 0x6858,
1482 0x5775, 0x447b, 0x685c, 0x3269, 0x6851, 0x3c6d, 0x3f42, 0x684d,
1483 0x5679, 0x4178, 0x3271, 0x685f, 0x4a41, 0x6859, 0x5524, 0x316a,
1484 0x553b, 0x684e, 0x6850, 0x3630, 0x6853, 0x685d, 0x4038, 0x4a77,
1485 0x4b28, 0x465c, 0x4075, 0x6869, 0x5023, 0x6872, 0x566a, 0x6860,
1486 0x6861, 0x5179, 0x3a4b, 0x3879, 0x3871, 0x5454, 0x686f, 0x686e,

```



```

1574 0x5166, 0x6c4d, 0x483b, 0x6c51, 0x6c53, 0x3b4d, 0x3c65, 0x6c4f,
1575 0x4937, 0x433a, 0x6c63, 0x5555, 0x6c50, 0x5673, 0x6c52, 0x6c4e,
1576 0x6c54, 0x6c55, 0x493f, 0x4f28, 0x505c, 0x512c, 0x485b, 0x6c56,
1577 0x4e75, 0x4a6c, 0x6c5a, 0x6c59, 0x303e, 0x6c57, 0x6c58, 0x6c64,
1578 0x483c, 0x4147, 0x6c5c, 0x5160, 0x6c5b, 0x546f, 0x6c5d, 0x5b46,
1579 0x6c5e, 0x312c, 0x6c5f, 0x6c60, 0x5726, 0x4540, 0x6b3c, 0x302e,
1580 0x3e74, 0x3838, 0x522f, 0x3056, 0x3579, 0x5833, 0x4b2c, 0x635d,
1581 0x462c, 0x3066, 0x4546, 0x6b39, 0x6b3a, 0x6b3b, 0x5140, 0x4523,
1582 0x6a72, 0x4432, 0x4435, 0x404e, 0x6a73, 0x4441, 0x4e6f, 0x6a70,
1583 0x6a74, 0x497c, 0x4723, 0x4c58, 0x4e7e, 0x6a75, 0x6a76, 0x4f2c,
1584 0x4067, 0x6a77, 0x363f, 0x6a78, 0x6a79, 0x6a7a, 0x6a7b, 0x6a71,
1585 0x482e, 0x616b, 0x3738, 0x616c, 0x616d, 0x5734, 0x616e, 0x616f,
1586 0x534c, 0x6171, 0x3f71, 0x6170, 0x3552, 0x3137, 0x6173, 0x6172,
1587 0x3a7c, 0x6174, 0x3937, 0x3e51, 0x447c, 0x3a5d, 0x3d46, 0x6175,
1588 0x6177, 0x3640, 0x4f41, 0x4a28, 0x6176, 0x5578, 0x537c, 0x6178,
1589 0x617c, 0x6179, 0x617a, 0x406a, 0x617e, 0x6221, 0x4047, 0x617b,
1590 0x617d, 0x6225, 0x4154, 0x6223, 0x6228, 0x327e, 0x6222, 0x434d,
1591 0x3242, 0x6227, 0x6226, 0x6224, 0x6229, 0x622b, 0x5049, 0x566d,
1592 0x4328, 0x622c, 0x4f57, 0x622e, 0x3a6f, 0x6960, 0x622d, 0x622a,
1593 0x3b2b, 0x5433, 0x6230, 0x622f, 0x6961, 0x6231, 0x6232, 0x6233,
1594 0x4c21, 0x6234, 0x6235, 0x507e, 0x424a, 0x5371, 0x4d75, 0x6760,
1595 0x6761, 0x3e41, 0x426a, 0x6764, 0x6763, 0x4d66, 0x4335, 0x6762,
1596 0x3b37, 0x4f56, 0x4161, 0x6769, 0x6768, 0x6774, 0x3223, 0x676a,
1597 0x6766, 0x676c, 0x676b, 0x493a, 0x5564, 0x6765, 0x3729, 0x6767,
1598 0x676e, 0x6773, 0x5669, 0x676d, 0x6772, 0x6771, 0x3060, 0x6775,
1599 0x4772, 0x4045, 0x406d, 0x4170, 0x6770, 0x6776, 0x4b76, 0x6822,
1600 0x6821, 0x5741, 0x677a, 0x6779, 0x6779, 0x677b, 0x6777, 0x677e, 0x677d,
1601 0x677c, 0x4155, 0x4759, 0x457d, 0x4543, 0x476d, 0x6823, 0x6826,
1602 0x6825, 0x6827, 0x3a77, 0x6778, 0x6824, 0x4870, 0x492a, 0x6829,
1603 0x3965, 0x517e, 0x6828, 0x6828, 0x682a, 0x682d, 0x682e, 0x4127, 0x682f,
1604 0x6830, 0x682c, 0x6834, 0x682b, 0x6831, 0x6835, 0x6832, 0x6833,
1605 0x6837, 0x6836, 0x394f, 0x702c, 0x702d, 0x4630, 0x306a, 0x483f,
1606 0x4d5f, 0x4e4d, 0x6a31, 0x6a32, 0x463f, 0x3449, 0x6a33, 0x5567,
1607 0x5d79, 0x6a34, 0x6a35, 0x6a36, 0x384a, 0x5f30, 0x4975, 0x4c70,
1608 0x497a, 0x497b, 0x5343, 0x4b26, 0x3826, 0x702e, 0x3142, 0x6538,
1609 0x4c6f, 0x5349, 0x3c57, 0x496a, 0x3567, 0x4450, 0x3569, 0x6e2e,
1610 0x3b2d, 0x675e, 0x6e2f, 0x3329, 0x6e32, 0x6e31, 0x3d67, 0x6e30,
1611 0x4e37, 0x454f, 0x4174, 0x5b4e, 0x6e33, 0x5073, 0x4254, 0x4668,
1612 0x372c, 0x6e34, 0x336b, 0x3b7b, 0x6e35, 0x675c, 0x6e36, 0x3d2e,
1613 0x7162, 0x4a68, 0x5249, 0x705a, 0x705b, 0x705c, 0x4146, 0x386d,
1614 0x3e4e, 0x705e, 0x4531, 0x705d, 0x5171, 0x7060, 0x304c, 0x3d6a,
1615 0x525f, 0x705f, 0x342f, 0x3768, 0x7066, 0x7065, 0x4623, 0x7061,
1616 0x7062, 0x3443, 0x7063, 0x556e, 0x4c5b, 0x3e52, 0x3c32, 0x7068,
1617 0x7067, 0x7064, 0x3221, 0x5622, 0x5338, 0x3e37, 0x482c, 0x706a,
1618 0x5177, 0x564c, 0x3a5b, 0x7069, 0x363b, 0x4d34, 0x4626, 0x4121,
1619 0x706b, 0x706e, 0x706d, 0x7070, 0x706c, 0x3b3e, 0x706f, 0x4c35,
1620 0x7072, 0x3355, 0x3154, 0x7073, 0x7074, 0x7076, 0x3461, 0x7071,
1621 0x7077, 0x707a, 0x7078, 0x7075, 0x707d, 0x7079, 0x707c, 0x707e,
1622 0x7121, 0x4e41, 0x7124, 0x7123, 0x4176, 0x707b, 0x4a5d, 0x3471,
1623 0x3171, 0x4c31, 0x7126, 0x7127, 0x712c, 0x554e, 0x7129, 0x4833,
1624 0x7122, 0x712b, 0x7128, 0x7125, 0x712a, 0x3029, 0x712d, 0x712f,
1625 0x7131, 0x7130, 0x712e, 0x5122, 0x7132, 0x7133, 0x396f, 0x3547,
1626 0x3057, 0x3059, 0x546d, 0x3544, 0x3d54, 0x3b4a, 0x7027, 0x385e,
1627 0x7028, 0x3028, 0x7029, 0x4d6e, 0x702a, 0x702b, 0x4624, 0x5665,
1628 0x7164, 0x7165, 0x4373, 0x535b, 0x5651, 0x4568, 0x532f, 0x5266,
1629 0x6e41, 0x303b, 0x5535, 0x514e, 0x3c60, 0x3a50, 0x3f78, 0x3847,
1630 0x3541, 0x454c, 0x4a22, 0x434b, 0x6e42, 0x443f, 0x3622, 0x6d6c,
1631 0x4324, 0x5631, 0x4f60, 0x6d6f, 0x454e, 0x365c, 0x4a21, 0x6d6d,
1632 0x6d70, 0x6d71, 0x433c, 0x3f34, 0x6d6e, 0x6d74, 0x6d72, 0x5566,
1633 0x435f, 0x6d73, 0x6d76, 0x5523, 0x5123, 0x6d75, 0x4350, 0x6d77,
1634 0x3f74, 0x3e6c, 0x6d78, 0x4c77, 0x515b, 0x5745, 0x5576, 0x6d7c,
1635 0x6d7b, 0x6d79, 0x6d7a, 0x6d7d, 0x3e26, 0x4b2f, 0x6e21, 0x363d,
1636 0x6e22, 0x4440, 0x6d7e, 0x3d5e, 0x3247, 0x3643, 0x6e25, 0x583a,
1637 0x6e23, 0x6e26, 0x4369, 0x3372, 0x6e27, 0x6e24, 0x4f39, 0x6e28,
1638 0x4277, 0x6e29, 0x6e2a, 0x5e2b, 0x4633, 0x4746, 0x5675, 0x3549,
1639 0x4b32, 0x6e2b, 0x4d2b, 0x6e2c, 0x5530, 0x6e2d, 0x7644, 0x5b47,
1640 0x3423, 0x432c, 0x7166, 0x4a38, 0x5253, 0x562a, 0x6f72, 0x3e58,
1641 0x3d43, 0x6f73, 0x364c, 0x302b, 0x4a2f, 0x6d36, 0x6d37, 0x4e79,
1642 0x372f, 0x3f73, 0x6d38, 0x426b, 0x4930, 0x6d39, 0x4676, 0x3f33,
1643 0x6d3c, 0x4578, 0x5150, 0x5729, 0x6d3a, 0x6d3b, 0x5162, 0x6d3f,
1644 0x6d40, 0x6d44, 0x6d48, 0x6d46, 0x6d4e, 0x5568, 0x6d49, 0x6d47,
1645 0x6d3e, 0x4569, 0x4646, 0x4969, 0x5452, 0x6d41, 0x6d42, 0x6d43,
1646 0x6d45, 0x4079, 0x3421, 0x3968, 0x6d50, 0x6d51, 0x6d4a, 0x6d4f,
1647 0x4e78, 0x4b36, 0x6d4c, 0x6d4d, 0x4f75, 0x6d52, 0x4172, 0x5332,
1648 0x6d4b, 0x4837, 0x3c6f, 0x4570, 0x6d56, 0x356f, 0x4235, 0x302d,
1649 0x4b69, 0x312e, 0x6d54, 0x4d6b, 0x3562, 0x6d55, 0x6d53, 0x6d57,
1650 0x357a, 0x6d58, 0x6d59, 0x6d5c, 0x314c, 0x4576, 0x3c6e, 0x6d5a,
1651 0x4c3c, 0x326a, 0x6d5b, 0x446b, 0x3445, 0x3075, 0x6d5f, 0x405a,
1652 0x3468, 0x454d, 0x6d5d, 0x3f44, 0x6d5e, 0x4425, 0x6d60, 0x6d61,
1653 0x6d63, 0x4157, 0x3b47, 0x3d38, 0x6d62, 0x6d64, 0x6d66, 0x6d65,
1654 0x6d67, 0x4a3e, 0x6c6a, 0x4071, 0x4967, 0x6c6b, 0x466e, 0x6c6c,
1655 0x466d, 0x6c6d, 0x6c70, 0x5766, 0x6c73, 0x6c71, 0x6c6e, 0x6c6f,
1656 0x5723, 0x4971, 0x4b6e, 0x6c74, 0x6c72, 0x4f69, 0x6c76, 0x4631,
1657 0x3c40, 0x6c75, 0x353b, 0x3b76, 0x6c77, 0x5977, 0x3d7b, 0x423b,
1658 0x6c78, 0x6c79, 0x3823, 0x6c7a, 0x6c7b, 0x6c7c, 0x536d, 0x582e,
1659 0x406b, 0x475d, 0x3a4c, 0x5063, 0x4b3d, 0x4d3a, 0x3851, 0x317c,
1660 0x476f, 0x5656, 0x3f46, 0x436b, 0x6f75, 0x4358, 0x5762, 0x6f77,

```


1748 0x7438, 0x7439, 0x4d27, 0x743a, 0x743b, 0x743c, 0x4b52, 0x743d,
1749 0x743e, 0x743f, 0x745e, 0x413c, 0x3c68, 0x492b, 0x515e, 0x6575,
1750 0x5c33, 0x5255, 0x5c34, 0x302c, 0x5c35, 0x3d5a, 0x5c39, 0x5842,
1751 0x5c37, 0x5373, 0x4956, 0x5c3a, 0x5c36, 0x5c3b, 0x4322, 0x5c3c,
1752 0x5c45, 0x5c3d, 0x4e5f, 0x5625, 0x5c4f, 0x5c4d, 0x5c52, 0x3d66,
1753 0x422b, 0x5c38, 0x5c4b, 0x5c4e, 0x5c3e, 0x3752, 0x3045, 0x5c47,
1754 0x503e, 0x5c41, 0x3b28, 0x373c, 0x5c4c, 0x5c46, 0x5c3f, 0x475b,
1755 0x513f, 0x5c40, 0x5c4a, 0x5c50, 0x4e2d, 0x5c42, 0x5c43, 0x5c48,
1756 0x5c49, 0x3254, 0x5c51, 0x4b55, 0x5437, 0x5c5b, 0x5c5f, 0x4c26,
1757 0x5c66, 0x4367, 0x5c5c, 0x3f41, 0x5c59, 0x307a, 0x3936, 0x5c65,
1758 0x5c53, 0x5c44, 0x5c56, 0x4874, 0x3f60, 0x493b, 0x313d, 0x5322,
1759 0x5c5a, 0x5c55, 0x463b, 0x5c5e, 0x5742, 0x432f, 0x3736, 0x4751,
1760 0x4329, 0x5c62, 0x5c58, 0x5c6b, 0x5c54, 0x5c5d, 0x3e25, 0x5c57,
1761 0x5c60, 0x5c63, 0x5c64, 0x5c78, 0x5c61, 0x5d22, 0x5c67, 0x3c6b,
1762 0x3444, 0x4323, 0x3267, 0x5c7a, 0x5c72, 0x5c6f, 0x5c7c, 0x5c6e,
1763 0x5270, 0x3268, 0x4857, 0x4863, 0x5c7b, 0x5c6d, 0x5c77, 0x5c75,
1764 0x3e23, 0x5c74, 0x325d, 0x5c73, 0x3c76, 0x5c68, 0x3b44, 0x4073,
1765 0x3c54, 0x5c69, 0x5c6a, 0x5c71, 0x5c76, 0x5c79, 0x3534, 0x4859,
1766 0x3b67, 0x5c7e, 0x5c7d, 0x532b, 0x5d21, 0x5d23, 0x5d25, 0x5271,
1767 0x5d24, 0x5d26, 0x5d27, 0x5229, 0x3a49, 0x5d29, 0x5d36, 0x5d31,
1768 0x5d34, 0x5d30, 0x464e, 0x4072, 0x492f, 0x5c6c, 0x5d2e, 0x5d37,
1769 0x5c70, 0x5d2f, 0x5d38, 0x5d2c, 0x5d39, 0x5d33, 0x5d2d, 0x442a,
1770 0x5d28, 0x4033, 0x412b, 0x5d2a, 0x5d2b, 0x5d32, 0x3b71, 0x5d35,
1771 0x5328, 0x5d3a, 0x5d3b, 0x4327, 0x5d52, 0x5d3c, 0x5d51, 0x393d,
1772 0x3e55, 0x3e7a, 0x3a4a, 0x5d4a, 0x5d45, 0x5d3f, 0x324b, 0x5d43,
1773 0x5d4b, 0x3224, 0x5d55, 0x5d3e, 0x4650, 0x5d50, 0x5d54, 0x4162,
1774 0x3746, 0x5d4e, 0x5d4f, 0x5d44, 0x5d3d, 0x5d4d, 0x4c51, 0x5d49,
1775 0x5d42, 0x4348, 0x463c, 0x4e2e, 0x5d4c, 0x5d48, 0x5d41, 0x5d46,
1776 0x425c, 0x5329, 0x532a, 0x5d53, 0x4f74, 0x4878, 0x5d66, 0x5d47,
1777 0x5d60, 0x4264, 0x5d61, 0x5d57, 0x5678, 0x5d59, 0x5d58, 0x3870,
1778 0x5d56, 0x464f, 0x362d, 0x5d62, 0x3a79, 0x5461, 0x5d67, 0x3450,
1779 0x5d5a, 0x3f7b, 0x5d63, 0x5d5f, 0x5d5d, 0x3559, 0x5d5b, 0x5d5c,
1780 0x5d5e, 0x3d2f, 0x5d64, 0x5d65, 0x5d75, 0x4349, 0x4b62, 0x5d72,
1781 0x5861, 0x4651, 0x5d74, 0x5574, 0x5d73, 0x5d70, 0x5d6c, 0x5d6f,
1782 0x5d68, 0x506e, 0x4858, 0x5d6e, 0x5d69, 0x5d6a, 0x4b72, 0x5d6d,
1783 0x314d, 0x4036, 0x3c3b, 0x5d71, 0x5d77, 0x5d76, 0x5d6b, 0x456e,
1784 0x5d7b, 0x5e24, 0x5e23, 0x5d78, 0x436f, 0x427b, 0x5561, 0x4e35,
1785 0x5d7d, 0x324c, 0x4468, 0x4a5f, 0x473e, 0x5d7a, 0x5d7c, 0x5d7e,
1786 0x5e22, 0x302a, 0x314e, 0x5e2c, 0x5e26, 0x3d36, 0x486f, 0x5e21,
1787 0x5e25, 0x5e29, 0x5e28, 0x5e27, 0x5e2d, 0x544c, 0x5e33, 0x5e2a,
1788 0x5e2e, 0x4059, 0x3121, 0x5e36, 0x5e31, 0x5e32, 0x5126, 0x5e35,
1789 0x5e2f, 0x5e30, 0x503d, 0x5e34, 0x4a6d, 0x5e39, 0x5e38, 0x5e37,
1790 0x5e3b, 0x3d65, 0x3258, 0x436a, 0x5e3a, 0x453a, 0x5e3c, 0x4c59,
1791 0x372a, 0x5465, 0x5e3d, 0x5e3f, 0x4422, 0x5e41, 0x5e3e, 0x5e40,
1792 0x553a, 0x5e42, 0x722e, 0x3b22, 0x4232, 0x4530, 0x4247, 0x722f,
1793 0x5069, 0x535d, 0x6b3d, 0x3366, 0x7230, 0x7231, 0x4a2d, 0x3a67,
1794 0x7233, 0x7235, 0x7234, 0x4b64, 0x4f3a, 0x7232, 0x4a34, 0x524f,
1795 0x426c, 0x4e43, 0x7238, 0x3076, 0x7237, 0x723e, 0x324f, 0x5141,
1796 0x723a, 0x723c, 0x5469, 0x723b, 0x7236, 0x723f, 0x723d, 0x7239,
1797 0x7247, 0x7244, 0x7246, 0x724a, 0x7242, 0x7240, 0x7245, 0x567b,
1798 0x7241, 0x4779, 0x495f, 0x7248, 0x3946, 0x3530, 0x7243, 0x7249,
1799 0x7250, 0x7256, 0x3b57, 0x7255, 0x4d5c, 0x566b, 0x7252, 0x7254,
1800 0x3872, 0x724b, 0x724e, 0x4279, 0x555d, 0x724c, 0x724d, 0x724f,
1801 0x7253, 0x7259, 0x533c, 0x366a, 0x4a71, 0x3764, 0x7257, 0x7258,
1802 0x725a, 0x725d, 0x725b, 0x725c, 0x5151, 0x7251, 0x4d49, 0x4e4f,
1803 0x5629, 0x7263, 0x435b, 0x7260, 0x402f, 0x726c, 0x725e, 0x7261,
1804 0x7268, 0x7262, 0x7267, 0x7266, 0x7269, 0x725f, 0x7264, 0x726a,
1805 0x532c, 0x7265, 0x3275, 0x7272, 0x502b, 0x7275, 0x3b48, 0x7279,
1806 0x7270, 0x7276, 0x7278, 0x727a, 0x7273, 0x7271, 0x3a7b, 0x357b,
1807 0x726f, 0x7277, 0x726d, 0x726e, 0x726b, 0x7326, 0x7323, 0x7322,
1808 0x7274, 0x485a, 0x727b, 0x7325, 0x4378, 0x727d, 0x7327, 0x7329,
1809 0x7324, 0x727c, 0x732b, 0x732a, 0x425d, 0x732e, 0x7330, 0x7321,
1810 0x7331, 0x732c, 0x732f, 0x732d, 0x732d, 0x7332, 0x7334, 0x7328,
1811 0x7333, 0x7335, 0x5037, 0x7338, 0x5979, 0x7339, 0x7337, 0x4864,
1812 0x7336, 0x733a, 0x733b, 0x3440, 0x6e43, 0x733c, 0x733d, 0x512a,
1813 0x742c, 0x5046, 0x5050, 0x515c, 0x4f4e, 0x3d56, 0x5143, 0x3a62,
1814 0x6169, 0x5242, 0x7142, 0x3239, 0x316d, 0x7143, 0x4940, 0x3344,
1815 0x5972, 0x4b25, 0x7144, 0x5654, 0x7145, 0x7440, 0x7146, 0x542c,
1816 0x7147, 0x3040, 0x7441, 0x7442, 0x347c, 0x455b, 0x4c3b, 0x5064,
1817 0x4d60, 0x7148, 0x5973, 0x313b, 0x4f2e, 0x3824, 0x714a, 0x714b,
1818 0x3243, 0x4151, 0x5730, 0x7149, 0x714c, 0x714e, 0x5976, 0x5261,
1819 0x5423, 0x7443, 0x4839, 0x7444, 0x714d, 0x714f, 0x3f63, 0x7150,
1820 0x7154, 0x7156, 0x7151, 0x4951, 0x4561, 0x4263, 0x397c, 0x7153,
1821 0x7155, 0x3953, 0x715b, 0x3a56, 0x307d, 0x7159, 0x7158, 0x7152,
1822 0x715a, 0x7157, 0x486c, 0x4d4a, 0x715d, 0x653d, 0x715c, 0x715e,
1823 0x715f, 0x4f65, 0x7445, 0x3d73, 0x7160, 0x7161, 0x4e77, 0x522a,
1824 0x717b, 0x3832, 0x3c7b, 0x395b, 0x3966, 0x4359, 0x4a53, 0x6a68,
1825 0x4040, 0x3e75, 0x6a69, 0x6a6a, 0x6a6b, 0x6a6c, 0x6a6d, 0x6a6e,
1826 0x6a6f, 0x3d47, 0x757b, 0x757d, 0x757e, 0x757c, 0x3d62, 0x7621,
1827 0x3425, 0x7622, 0x7623, 0x6c32, 0x5154, 0x596a, 0x7624, 0x6e3a,
1828 0x5532, 0x537e, 0x4c5c, 0x4a44, 0x6540, 0x7625, 0x3e2f, 0x4629,
1829 0x5a25, 0x3c46, 0x3629, 0x383c, 0x484f, 0x3c25, 0x5a26, 0x5a27,
1830 0x4c56, 0x4843, 0x5a28, 0x467d, 0x5135, 0x5269, 0x5136, 0x3c47,
1831 0x3d32, 0x3b64, 0x5a29, 0x5a2a, 0x5148, 0x5a2b, 0x506d, 0x366f,
1832 0x425b, 0x4b4f, 0x376d, 0x4968, 0x3743, 0x3e77, 0x5624, 0x5a2c,
1833 0x5a2d, 0x4640, 0x5767, 0x4a36, 0x5529, 0x4b5f, 0x556f, 0x5a2e,
1834 0x565f, 0x344a, 0x5a30, 0x5a2f, 0x526b, 0x5a31, 0x5a32, 0x5a33,

```
1835 0x4a54, 0x5a34, 0x4a2b, 0x5a35, 0x5a36, 0x334f, 0x566f, 0x5a37,
1836 0x3b30, 0x352e, 0x5a38, 0x5a39, 0x396e, 0x512f, 0x5268, 0x5a3a,
1837 0x3843, 0x4f6a, 0x326f, 0x5a3b, 0x5a3c, 0x3d6b, 0x4e5c, 0x536f,
1838 0x5a3d, 0x4e73, 0x5a3e, 0x5355, 0x3b65, 0x5a3f, 0x4b35, 0x4b50,
1839 0x5a40, 0x476b, 0x566e, 0x5a41, 0x4535, 0x3641, 0x5a42, 0x374c,
1840 0x3f4e, 0x5a43, 0x5a44, 0x5a44, 0x4b2d, 0x5a45, 0x3577, 0x5a46, 0x4142,
1841 0x573b, 0x5a47, 0x4c38, 0x526a, 0x4431, 0x5a48, 0x357d, 0x3b51,
1842 0x5a49, 0x5033, 0x5a4a, 0x5a4b, 0x4e3d, 0x5a4c, 0x5a4d, 0x5a4e,
1843 0x3277, 0x5a51, 0x5a4f, 0x5168, 0x5a50, 0x4355, 0x5a52, 0x5a53,
1844 0x5a54, 0x5a55, 0x503b, 0x5225, 0x3079, 0x5a56, 0x472b, 0x5a57,
1845 0x3d77, 0x4321, 0x5a58, 0x5a59, 0x437d, 0x4c37, 0x5a5a, 0x5a5b,
1846 0x403e, 0x4657, 0x5a5c, 0x5a5d, 0x4734, 0x5a5e, 0x5a5f, 0x3948,
1847 0x3b6d, 0x3639, 0x7478, 0x7479, 0x4d63, 0x7539, 0x6b60, 0x4f73,
1848 0x3b3f, 0x3a40, 0x5425, 0x6159, 0x7574, 0x312a, 0x3272, 0x7575,
1849 0x7577, 0x3a51, 0x7576, 0x7576, 0x4332, 0x7579, 0x7578, 0x3134, 0x556a,
1850 0x383a, 0x3931, 0x3246, 0x5470, 0x4f4d, 0x305c, 0x554b, 0x3b75,
1851 0x564a, 0x3737, 0x4c30, 0x4636, 0x3161, 0x393a, 0x567c, 0x3961,
1852 0x3721, 0x3c7a, 0x6a5a, 0x6a5a, 0x4c79, 0x3973, 0x6a5c, 0x347b,
1853 0x4333, 0x3751, 0x3a58, 0x6a5d, 0x5474, 0x6a5e, 0x3c56, 0x3b5f,
1854 0x6a5f, 0x415e, 0x4238, 0x545f, 0x574a, 0x6a60, 0x6a61, 0x6a64,
1855 0x6a62, 0x6a63, 0x495e, 0x495e, 0x3833, 0x3644, 0x6a65, 0x4a6a, 0x494d,
1856 0x344d, 0x6259, 0x4562, 0x6a66, 0x4035, 0x5738, 0x6a67, 0x572c,
1857 0x487c, 0x5853, 0x584d, 0x545e, 0x5479, 0x4944, 0x532e, 0x3853,
1858 0x3360, 0x4962, 0x7476, 0x3a55, 0x7477, 0x575f, 0x7471, 0x3830,
1859 0x5554, 0x384f, 0x4670, 0x3343, 0x7472, 0x332c, 0x543d, 0x4777,
1860 0x7474, 0x7473, 0x4c4b, 0x4824, 0x7475, 0x5763, 0x453f, 0x7540,
1861 0x753b, 0x7543, 0x7542, 0x563a, 0x7541, 0x543e, 0x7544, 0x754c,
1862 0x304f, 0x3578, 0x7549, 0x754a, 0x455c, 0x7545, 0x7546, 0x7547,
1863 0x754b, 0x3e60, 0x7548, 0x387a, 0x7550, 0x7553, 0x3f67, 0x3972,
1864 0x753c, 0x754d, 0x4237, 0x4c78, 0x4c79, 0x3c79, 0x754e, 0x754f, 0x7551,
1865 0x3665, 0x7552, 0x7555, 0x753d, 0x7554, 0x533b, 0x336c, 0x4c24,
1866 0x7556, 0x7557, 0x3e61, 0x7558, 0x4c5f, 0x755b, 0x3248, 0x5759,
1867 0x7559, 0x755a, 0x755c, 0x7562, 0x7560, 0x755f, 0x755d, 0x7561,
1868 0x755e, 0x7564, 0x7565, 0x4c63, 0x653f, 0x3538, 0x7563, 0x7568,
1869 0x4c23, 0x7566, 0x7567, 0x753e, 0x3144, 0x753f, 0x3545, 0x3264,
1870 0x756c, 0x7569, 0x3657, 0x756d, 0x756a, 0x756b, 0x345a, 0x546a,
1871 0x756e, 0x3379, 0x756f, 0x7571, 0x7570, 0x7572, 0x7573, 0x496d,
1872 0x392a, 0x477b, 0x3663, 0x4c49, 0x6a26, 0x3335, 0x547e, 0x396c,
1873 0x5079, 0x696d, 0x572a, 0x696e, 0x4256, 0x486d, 0x3a64, 0x696f,
1874 0x6970, 0x6971, 0x5661, 0x6972, 0x6973, 0x6975, 0x6974, 0x6976,
1875 0x6977, 0x4761, 0x6978, 0x5458, 0x6979, 0x3d4e, 0x697a, 0x697b,
1876 0x3d4f, 0x697c, 0x3828, 0x413e, 0x697d, 0x3132, 0x3b54, 0x3975,
1877 0x697e, 0x6a21, 0x6a22, 0x6a23, 0x3778, 0x3c2d, 0x4a64, 0x604e,
1878 0x542f, 0x4f3d, 0x5537, 0x6a24, 0x555e, 0x6a25, 0x5041, 0x393c,
1879 0x3447, 0x3159, 0x4031, 0x3166, 0x3167, 0x3168, 0x333d, 0x4868,
1880 0x6541, 0x315f, 0x4149, 0x346f, 0x4728, 0x5358, 0x4679, 0x5138,
1881 0x397d, 0x4275, 0x532d, 0x544b, 0x3d7c, 0x6542, 0x3735, 0x6543,
1882 0x3b39, 0x5562, 0x3d78, 0x5436, 0x4e25, 0x412c, 0x3359, 0x4c7e,
1883 0x6546, 0x6544, 0x6548, 0x654a, 0x6547, 0x354f, 0x4648, 0x357c,
1884 0x6545, 0x4a76, 0x6549, 0x4354, 0x3145, 0x3c23, 0x5737, 0x4d4b,
1885 0x4b4d, 0x4a4a, 0x4c53, 0x654c, 0x654b, 0x4466, 0x5121, 0x5137,
1886 0x654d, 0x6550, 0x4d38, 0x5670, 0x654f, 0x355d, 0x4d3e, 0x6551,
1887 0x363a, 0x4d28, 0x3964, 0x4a45, 0x3351, 0x4b59, 0x546c, 0x6552,
1888 0x376a, 0x654e, 0x6555, 0x347e, 0x6556, 0x6553, 0x6554, 0x525d,
1889 0x425f, 0x3146, 0x5362, 0x365d, 0x4b6c, 0x6557, 0x5376, 0x3169,
1890 0x3674, 0x655a, 0x6558, 0x6559, 0x3540, 0x5245, 0x655c, 0x655e,
1891 0x655d, 0x4732, 0x5223, 0x655b, 0x5462, 0x555a, 0x6560, 0x5771,
1892 0x6561, 0x315c, 0x517b, 0x6562, 0x6564, 0x6563, 0x6565, 0x5258,
1893 0x354b, 0x675f, 0x5a75, 0x5a78, 0x5a76, 0x5a77, 0x5a7a, 0x504f,
1894 0x4447, 0x306e, 0x5030, 0x5a79, 0x534a, 0x3a2a, 0x5b22, 0x4771,
1895 0x5a7c, 0x5a7b, 0x495b, 0x5a7d, 0x5b21, 0x575e, 0x5a7e, 0x415a,
1896 0x5b25, 0x5374, 0x5b27, 0x5b24, 0x5b28, 0x3d3c, 0x4049, 0x5b23,
1897 0x5b26, 0x5623, 0x5b29, 0x5b2d, 0x5b2e, 0x5b2c, 0x3a42, 0x3f24,
1898 0x5b2b, 0x5b2a, 0x5447, 0x323f, 0x5b2f, 0x3979, 0x5b30, 0x333b,
1899 0x3526, 0x363c, 0x5b31, 0x3675, 0x5b32, 0x3149, 0x5b34, 0x5b33,
1900 0x5b35, 0x5b37, 0x5b36, 0x5b38, 0x5b39, 0x5b3a, 0x534f, 0x747a,
1901 0x4775, 0x5743, 0x4564, 0x747c, 0x747d, 0x747b, 0x3e46, 0x506f,
1902 0x3753, 0x544d, 0x4c2a, 0x7522, 0x7521, 0x3a28, 0x747e, 0x4b56,
1903 0x7524, 0x4052, 0x336a, 0x4d2a, 0x7525, 0x7523, 0x3d34, 0x7528,
1904 0x7529, 0x3d4d, 0x4338, 0x3f61, 0x4b61, 0x752a, 0x7526, 0x7527,
1905 0x4470, 0x752c, 0x343c, 0x343c, 0x576d, 0x3457, 0x752b, 0x752e, 0x752d,
1906 0x752f, 0x5051, 0x4351, 0x4829, 0x7530, 0x7531, 0x7532, 0x7533,
1907 0x7534, 0x7535, 0x7537, 0x7536, 0x7538, 0x3249, 0x5354, 0x4a4d,
1908 0x406f, 0x5658, 0x5230, 0x413f, 0x3d70, 0x382a, 0x3c78, 0x7646,
1909 0x7647, 0x7648, 0x7649, 0x764a, 0x764c, 0x764b, 0x7769, 0x764d,
1910 0x764e, 0x6e44, 0x6e45, 0x6e46, 0x556b, 0x3624, 0x6e48, 0x6e47,
1911 0x6e49, 0x6e4a, 0x4725, 0x6e4b, 0x6e4c, 0x3730, 0x3576, 0x6e4d,
1912 0x6e4f, 0x6e4e, 0x3846, 0x6e50, 0x6e51, 0x6e52, 0x365b, 0x332e,
1913 0x5653, 0x4446, 0x3135, 0x3856, 0x6e53, 0x6e54, 0x543f, 0x4755,
1914 0x3e7b, 0x4e59, 0x3933, 0x6e56, 0x6e55, 0x6e58, 0x6e57, 0x4525,
1915 0x6e59, 0x6e5a, 0x472e, 0x6e5b, 0x472f, 0x6e5c, 0x3227, 0x6e5d,
1916 0x6e5e, 0x6e5f, 0x6e60, 0x6e61, 0x576a, 0x6e62, 0x6e63, 0x3c58,
1917 0x6e64, 0x534b, 0x4c7a, 0x322c, 0x4165, 0x6e65, 0x4726, 0x432d,
1918 0x6e66, 0x6e67, 0x6e68, 0x6e69, 0x6e6a, 0x6e6b, 0x6e6c, 0x6e6d,
1919 0x6e6e, 0x6e6f, 0x6e70, 0x6e71, 0x6e72, 0x6e74, 0x6e73, 0x6e75,
1920 0x4d2d, 0x4241, 0x6e76, 0x6e77, 0x6e78, 0x5521, 0x6e79, 0x4f33,
1921 0x6e7a, 0x6e7b, 0x6e7c, 0x6e7d, 0x6f21, 0x6e7e, 0x6f22, 0x3875,
```

1922 0x437a, 0x6f23, 0x6f24, 0x3d42, 0x523f, 0x3279, 0x6f25, 0x6f26,
1923 0x6f27, 0x5278, 0x6f28, 0x567d, 0x6f29, 0x464c, 0x6f2a, 0x6f2b,
1924 0x4134, 0x6f2c, 0x4f7a, 0x4b78, 0x6f2e, 0x6f2d, 0x337a, 0x3978,
1925 0x6f2f, 0x6f30, 0x5062, 0x6f31, 0x6f32, 0x3766, 0x503f, 0x6f33,
1926 0x6f34, 0x6f35, 0x4871, 0x4c60, 0x6f36, 0x6f37, 0x6f38, 0x6f39,
1927 0x6f3a, 0x5560, 0x6f3b, 0x346d, 0x432a, 0x6f3c, 0x6f3d, 0x6f3e,
1928 0x6f3f, 0x4e7d, 0x6f40, 0x4260, 0x3438, 0x5736, 0x3d75, 0x4f47,
1929 0x6f43, 0x6f41, 0x6f42, 0x6f44, 0x3627, 0x3c7c, 0x3e62, 0x434c,
1930 0x6f45, 0x6f46, 0x6f47, 0x6f4f, 0x6f48, 0x6f49, 0x6f4a, 0x4742,
1931 0x6f71, 0x364d, 0x6f4b, 0x6f4c, 0x6f4d, 0x3646, 0x433e, 0x6f4e,
1932 0x6f50, 0x6f51, 0x6f52, 0x5572, 0x6f53, 0x4477, 0x6f54, 0x4478,
1933 0x6f55, 0x6f56, 0x3864, 0x3077, 0x6f57, 0x6f58, 0x6f59, 0x6f5a,
1934 0x6f5b, 0x6f5c, 0x6f5d, 0x6f5e, 0x3e35, 0x6f61, 0x6f5f, 0x6f60,
1935 0x6f62, 0x6f63, 0x414d, 0x6f64, 0x6f65, 0x6f66, 0x6f67, 0x6f68,
1936 0x6f69, 0x6f6a, 0x6f6b, 0x6f6c, 0x4058, 0x6f6d, 0x412d, 0x6f6e,
1937 0x6f6f, 0x6f70, 0x4f62, 0x3324, 0x4345, 0x6345, 0x4941, 0x6346,
1938 0x3155, 0x4e4a, 0x3433, 0x4872, 0x6347, 0x4f50, 0x6348, 0x3c64,
1939 0x6349, 0x6346, 0x4346, 0x4346, 0x5522, 0x4456, 0x396b, 0x4e45, 0x634b,
1940 0x4376, 0x634c, 0x3727, 0x3873, 0x3a52, 0x634d, 0x634e, 0x5444,
1941 0x634f, 0x6350, 0x514b, 0x6351, 0x6352, 0x6353, 0x6354, 0x5156,
1942 0x6355, 0x327b, 0x403b, 0x6356, 0x402b, 0x6357, 0x6358, 0x6359,
1943 0x635a, 0x635b, 0x3837, 0x5a62, 0x3653, 0x5a64, 0x5a63, 0x5a66,
1944 0x486e, 0x5a65, 0x3740, 0x5174, 0x5275, 0x5573, 0x3d57, 0x5768,
1945 0x5a68, 0x5a67, 0x3022, 0x4d53, 0x5a69, 0x383d, 0x3c4a, 0x423d,
1946 0x4224, 0x3342, 0x5a6a, 0x422a, 0x4430, 0x3d35, 0x4f5e, 0x5a6b,
1947 0x4942, 0x315d, 0x5a6c, 0x3638, 0x543a, 0x337d, 0x5a6d, 0x5449,
1948 0x4f55, 0x4563, 0x5a6e, 0x5a6f, 0x5a70, 0x416a, 0x4c55, 0x4f5d,
1949 0x5367, 0x4221, 0x5a71, 0x4b65, 0x5a72, 0x4b66, 0x527e, 0x3874,
1950 0x5a73, 0x302f, 0x4f36, 0x554f, 0x4b6d, 0x5a74, 0x6344, 0x4125,
1951 0x763f, 0x7640, 0x7641, 0x4451, 0x4838, 0x5163, 0x505b, 0x5145,
1952 0x3c2f, 0x394d, 0x6f74, 0x3446, 0x533a, 0x7642, 0x337b, 0x7643,
1953 0x3571, 0x7645, 0x536a, 0x7627, 0x5129, 0x7629, 0x7628, 0x4163,
1954 0x4057, 0x3122, 0x4e6d, 0x5068, 0x762b, 0x4f76, 0x762a, 0x5570,
1955 0x762c, 0x4339, 0x3b74, 0x762e, 0x762d, 0x445e, 0x4158, 0x4b2a,
1956 0x4f3c, 0x762f, 0x7630, 0x7631, 0x4236, 0x3054, 0x4579, 0x7632,
1957 0x4760, 0x762e, 0x3e38, 0x3e32, 0x3565, 0x3747, 0x3f3f, 0x4352,
1958 0x4366, 0x584c, 0x386f, 0x3d79, 0x5125, 0x3050, 0x7730, 0x7731,
1959 0x502c, 0x3030, 0x7732, 0x7733, 0x7734, 0x474a, 0x3e4f, 0x7737,
1960 0x7736, 0x315e, 0x7735, 0x7738, 0x7739, 0x4e24, 0x484d, 0x3a2b,
1961 0x6838, 0x6839, 0x683a, 0x3e42, 0x5274, 0x544f, 0x4958, 0x5233,
1962 0x3625, 0x476a, 0x717c, 0x4f6e, 0x4b33, 0x506b, 0x676f, 0x4d67,
1963 0x394b, 0x3659, 0x717d, 0x3064, 0x4b4c, 0x717e, 0x5424, 0x422d,
1964 0x416c, 0x4644, 0x3e31, 0x7221, 0x3c55, 0x7222, 0x7223, 0x7224,
1965 0x5243, 0x4635, 0x4d47, 0x7225, 0x5331, 0x3f45, 0x4c62, 0x7226,
1966 0x7227, 0x5155, 0x366e, 0x7228, 0x7229, 0x355f, 0x722a, 0x722b,
1967 0x327c, 0x722c, 0x722d, 0x4827, 0x3767, 0x6c29, 0x6c2a, 0x6c2b,
1968 0x6c2c, 0x462e, 0x6c2d, 0x6c2e, 0x3749, 0x4a33, 0x6238, 0x774f,
1969 0x7750, 0x324d, 0x7751, 0x7753, 0x7752, 0x623b, 0x3c22, 0x623c,
1970 0x623d, 0x623e, 0x623f, 0x6240, 0x6241, 0x3739, 0x527b, 0x3d24,
1971 0x4a4e, 0x3125, 0x4b47, 0x6242, 0x367c, 0x4844, 0x6243, 0x3d48,
1972 0x317d, 0x6244, 0x3676, 0x6245, 0x4459, 0x6246, 0x4f5a, 0x395d,
1973 0x6247, 0x4021, 0x6248, 0x3276, 0x6249, 0x4173, 0x624a, 0x624b,
1974 0x4278, 0x624c, 0x624d, 0x624e, 0x4a57, 0x5838, 0x5965, 0x4f63,
1975 0x7025, 0x5c30, 0x426d, 0x5426, 0x4d54, 0x5131, 0x335b, 0x477d,
1976 0x3235, 0x423f, 0x6660, 0x4a3b, 0x6661, 0x6662, 0x3e54, 0x6663,
1977 0x5724, 0x4d55, 0x6665, 0x3c5d, 0x6664, 0x6666, 0x6667, 0x426e,
1978 0x3d3e, 0x6668, 0x4266, 0x3a27, 0x6669, 0x666a, 0x3352, 0x5169,
1979 0x3f25, 0x666b, 0x466f, 0x666c, 0x666d, 0x666e, 0x462d, 0x666f,
1980 0x4927, 0x6670, 0x6671, 0x6672, 0x6539, 0x6673, 0x6674, 0x4262,
1981 0x6675, 0x6676, 0x5668, 0x6677, 0x6678, 0x3947, 0x773b, 0x773a,
1982 0x773e, 0x773c, 0x3a21, 0x773f, 0x7740, 0x7742, 0x7741, 0x7744,
1983 0x7743, 0x7745, 0x7746, 0x7747, 0x4b68, 0x385f, 0x7754, 0x7755,
1984 0x7756, 0x7758, 0x775a, 0x7757, 0x775b, 0x7759, 0x5757, 0x775c,
1985 0x775d, 0x775e, 0x775f, 0x7760, 0x5b4b, 0x582a, 0x6577, 0x396d,
1986 0x3f7d, 0x3b6a, 0x7749, 0x4647, 0x7748, 0x774a, 0x774c, 0x774b,
1987 0x774d, 0x4e3a, 0x774e, 0x4427, 0x5363, 0x764f, 0x4233, 0x7650,
1988 0x7651, 0x7652, 0x7653, 0x7654, 0x7656, 0x312b, 0x7657, 0x7658,
1989 0x7659, 0x765a, 0x765b, 0x765c, 0x765d, 0x765e, 0x4f4a, 0x765f,
1990 0x7660, 0x7661, 0x7662, 0x7663, 0x7664, 0x4070, 0x7665, 0x7666,
1991 0x7667, 0x7668, 0x7669, 0x766a, 0x766b, 0x766c, 0x766d, 0x766e,
1992 0x766f, 0x7670, 0x7671, 0x7672, 0x7673, 0x7674, 0x3e28, 0x7675,
1993 0x7676, 0x7677, 0x7678, 0x487a, 0x7679, 0x767a, 0x767b, 0x767c,
1994 0x767d, 0x767e, 0x7721, 0x7722, 0x7723, 0x7724, 0x7725, 0x7726,
1995 0x7727, 0x7728, 0x316e, 0x7729, 0x772a, 0x772b, 0x772c, 0x772d,
1996 0x415b, 0x772e, 0x772f, 0x4471, 0x702f, 0x3c26, 0x7030, 0x4379,
1997 0x4538, 0x513b, 0x7031, 0x7032, 0x7033, 0x7034, 0x7035, 0x513c,
1998 0x516c, 0x7037, 0x7036, 0x5427, 0x4d52, 0x7038, 0x703a, 0x7039,
1999 0x703b, 0x703c, 0x386b, 0x703d, 0x3a68, 0x703e, 0x703f, 0x3e69,
2000 0x7040, 0x366c, 0x7041, 0x7042, 0x7043, 0x7044, 0x4835, 0x7045,
2001 0x7046, 0x7047, 0x4574, 0x7048, 0x7049, 0x704a, 0x773d, 0x704b,
2002 0x704c, 0x704d, 0x704e, 0x704f, 0x3a57, 0x7050, 0x7051, 0x7052,
2003 0x7053, 0x7054, 0x7055, 0x7056, 0x7058, 0x5325, 0x7057, 0x7059,
2004 0x753a, 0x4239, 0x7764, 0x7765, 0x7766, 0x7767, 0x7768, 0x4234,
2005 0x776a, 0x776b, 0x4273, 0x7470, 0x746f, 0x4269, 0x7761, 0x7762,
2006 0x3b46, 0x5964, 0x4a72, 0x4068, 0x7024, 0x3a5a, 0x472d, 0x442c,
2007 0x776c, 0x776d, 0x776e, 0x7770, 0x776f, 0x7771, 0x7774, 0x7773,
2008 0x7772, 0x7775, 0x7776, 0x6d69, 0x6d6a, 0x6d6b, 0x763c, 0x763d,


```
2009 0x763e, 0x3626, 0x583e, 0x3944, 0x583b, 0x5c31, 0x4a73, 0x7777,
2010 0x7778, 0x7779, 0x777b, 0x777a, 0x3147, 0x777c, 0x777d, 0x777e,
2011 0x466b, 0x6c34, 0x335d, 0x7633, 0x7634, 0x4164, 0x7635, 0x7636,
2012 0x7637, 0x7638, 0x7639, 0x763a, 0x4823, 0x763b, 0x417a, 0x3928,
2013 0x6d68, 0x396a, 0x595f, 0x2321, 0x2322, 0x2323, 0x2167, 0x2325,
2014 0x2326, 0x2327, 0x2328, 0x2329, 0x232a, 0x232b, 0x232c, 0x232d,
2015 0x232e, 0x232f, 0x2330, 0x2331, 0x2332, 0x2333, 0x2334, 0x2335,
2016 0x2336, 0x2337, 0x2338, 0x2339, 0x233a, 0x233b, 0x233c, 0x233d,
2017 0x233e, 0x233f, 0x2340, 0x2341, 0x2342, 0x2343, 0x2344, 0x2345,
2018 0x2346, 0x2347, 0x2348, 0x2349, 0x234a, 0x234b, 0x234c, 0x234d,
2019 0x234e, 0x234f, 0x2350, 0x2351, 0x2352, 0x2353, 0x2354, 0x2355,
2020 0x2356, 0x2357, 0x2358, 0x2359, 0x235a, 0x235b, 0x235c, 0x235d,
2021 0x235e, 0x235f, 0x2360, 0x2361, 0x2362, 0x2363, 0x2364, 0x2365,
2022 0x2366, 0x2367, 0x2368, 0x2369, 0x236a, 0x236b, 0x236c, 0x236d,
2023 0x236e, 0x236f, 0x2370, 0x2371, 0x2372, 0x2373, 0x2374, 0x2375,
2024 0x2376, 0x2377, 0x2378, 0x2379, 0x237a, 0x237b, 0x237c, 0x237d,
2025 0x212b, 0x2169, 0x216a, 0x237e, 0x2324,
2026 };
2027
2028 static const Summary16 gb2312_uni2indx_page00[70] = {
2029 /* 0x0000 */
2030 { 0, 0x0000 }, { 0, 0x0000 }, { 0, 0x0000 }, { 0, 0x0000 },
2031 { 0, 0x0000 }, { 0, 0x0000 }, { 0, 0x0000 }, { 0, 0x0000 },
2032 { 0, 0x0000 }, { 0, 0x0000 }, { 0, 0x0190 }, { 3, 0x0003 },
2033 { 5, 0x0000 }, { 5, 0x0080 }, { 6, 0x3703 }, { 13, 0x168c },
2034 /* 0x0100 */
2035 { 19, 0x0002 }, { 20, 0x0808 }, { 22, 0x0800 }, { 23, 0x0000 },
2036 { 23, 0x2000 }, { 24, 0x0000 }, { 24, 0x0800 }, { 25, 0x0000 },
2037 { 25, 0x0000 }, { 25, 0x0000 }, { 25, 0x0000 }, { 25, 0x0000 },
2038 { 25, 0x4000 }, { 26, 0x1555 }, { 33, 0x0000 }, { 33, 0x0000 },
2039 /* 0x0200 */
2040 { 33, 0x0000 }, { 33, 0x0000 }, { 33, 0x0000 }, { 33, 0x0000 },
2041 { 33, 0x0000 }, { 33, 0x0000 }, { 33, 0x0000 }, { 33, 0x0000 },
2042 { 33, 0x0000 }, { 33, 0x0000 }, { 33, 0x0000 }, { 33, 0x0000 },
2043 { 33, 0x0280 }, { 35, 0x0000 }, { 35, 0x0000 }, { 35, 0x0000 },
2044 /* 0x0300 */
2045 { 35, 0x0000 }, { 35, 0x0000 }, { 35, 0x0000 }, { 35, 0x0000 },
2046 { 35, 0x0000 }, { 35, 0x0000 }, { 35, 0x0000 }, { 35, 0x0000 },
2047 { 35, 0x0000 }, { 35, 0xffff }, { 50, 0x03fb }, { 59, 0xffff },
2048 { 74, 0x03fb }, { 83, 0x0000 }, { 83, 0x0000 }, { 83, 0x0000 },
2049 /* 0x0400 */
2050 { 83, 0x0002 }, { 84, 0xffff }, { 100, 0xffff }, { 116, 0xffff },
2051 { 132, 0xffff }, { 148, 0x0002 },
2052 };
2053 static const Summary16 gb2312_uni2indx_page20[101] = {
2054 /* 0x2000 */
2055 { 149, 0x0000 }, { 149, 0x3360 }, { 155, 0x0040 }, { 156, 0x080d },
2056 { 160, 0x0000 }, { 160, 0x0000 }, { 160, 0x0000 }, { 160, 0x0000 },
2057 { 160, 0x0000 }, { 160, 0x0000 }, { 160, 0x0000 }, { 160, 0x0000 },
2058 { 160, 0x0000 }, { 160, 0x0000 }, { 160, 0x0000 }, { 160, 0x0000 },
2059 /* 0x2100 */
2060 { 160, 0x0008 }, { 161, 0x0040 }, { 162, 0x0000 }, { 162, 0x0000 },
2061 { 162, 0x0000 }, { 162, 0x0000 }, { 162, 0x0fff }, { 174, 0x0000 },
2062 { 174, 0x0000 }, { 174, 0x000f }, { 178, 0x0000 }, { 178, 0x0000 },
2063 { 178, 0x0000 }, { 178, 0x0000 }, { 178, 0x0000 }, { 178, 0x0000 },
2064 /* 0x2200 */
2065 { 178, 0x8100 }, { 180, 0x6402 }, { 184, 0x4fa1 }, { 192, 0x20f0 },
2066 { 197, 0x1100 }, { 199, 0x0000 }, { 199, 0xc033 }, { 205, 0x0000 },
2067 { 205, 0x0000 }, { 205, 0x0200 }, { 206, 0x0020 }, { 207, 0x0000 },
2068 { 207, 0x0000 }, { 207, 0x0000 }, { 207, 0x0000 }, { 207, 0x0000 },
2069 /* 0x2300 */
2070 { 207, 0x0000 }, { 207, 0x0004 }, { 208, 0x0000 }, { 208, 0x0000 },
2071 { 208, 0x0000 }, { 208, 0x0000 }, { 208, 0x0000 }, { 208, 0x0000 },
2072 { 208, 0x0000 }, { 208, 0x0000 }, { 208, 0x0000 }, { 208, 0x0000 },
2073 { 208, 0x0000 }, { 208, 0x0000 }, { 208, 0x0000 }, { 208, 0x0000 },
2074 /* 0x2400 */
2075 { 208, 0x0000 }, { 208, 0x0000 }, { 208, 0x0000 }, { 208, 0x0000 },
2076 { 208, 0x0000 }, { 208, 0x0000 }, { 208, 0x03ff }, { 218, 0xffff },
2077 { 230, 0xffff }, { 246, 0x0fff }, { 258, 0x0000 }, { 258, 0x0000 },
2078 { 258, 0x0000 }, { 258, 0x0000 }, { 258, 0x0000 }, { 258, 0x0000 },
2079 /* 0x2500 */
2080 { 258, 0xffff }, { 274, 0xffff }, { 290, 0xffff }, { 306, 0xffff },
2081 { 322, 0x0fff }, { 334, 0x0000 }, { 334, 0x0000 }, { 334, 0x0000 },
2082 { 334, 0x0000 }, { 334, 0x0000 }, { 334, 0x0003 }, { 336, 0x000c },
2083 { 338, 0xc8c0 }, { 343, 0x0000 }, { 343, 0x0000 }, { 343, 0x0000 },
2084 /* 0x2600 */
2085 { 343, 0x0060 }, { 345, 0x0000 }, { 345, 0x0000 }, { 345, 0x0000 },
2086 { 345, 0x0005 },
2087 };
2088 static const Summary16 gb2312_uni2indx_page30[35] = {
2089 /* 0x3000 */
2090 { 347, 0xff2f }, { 360, 0x00fb }, { 367, 0x0000 }, { 367, 0x0000 },
2091 { 367, 0xffff }, { 382, 0xffff }, { 398, 0xffff }, { 414, 0xffff },
2092 { 430, 0xffff }, { 446, 0x000f }, { 450, 0xffff }, { 465, 0xffff },
2093 { 481, 0xffff }, { 497, 0xffff }, { 513, 0xffff }, { 529, 0x087f },
2094 /* 0x3100 */
2095 { 537, 0xffe0 }, { 548, 0xffff }, { 564, 0x03ff }, { 574, 0x0000 },
```

```

2096 { 574, 0x0000 }, { 574, 0x0000 }, { 574, 0x0000 }, { 574, 0x0000 },
2097 { 574, 0x0000 }, { 574, 0x0000 }, { 574, 0x0000 }, { 574, 0x0000 },
2098 { 574, 0x0000 }, { 574, 0x0000 }, { 574, 0x0000 }, { 574, 0x0000 },
2099 /* 0x3200 */
2100 { 574, 0x0000 }, { 574, 0x0000 }, { 574, 0x03ff },
2101 };
2102 static const Summary16 gb2312_uni2indx_page4e[1263] = {
2103 /* 0x4e00 */
2104 { 584, 0x7f8b }, { 595, 0x7f7b }, { 608, 0x3db4 }, { 617, 0xef55 },
2105 { 628, 0xfba8 }, { 638, 0xf35d }, { 649, 0x0243 }, { 653, 0x400b },
2106 { 657, 0xfb40 }, { 665, 0x8d3e }, { 674, 0x7bf7 }, { 687, 0x8c2c },
2107 { 693, 0x6eff }, { 706, 0xe3fa }, { 717, 0x1d3a }, { 725, 0xa8ed },
2108 /* 0x4f00 */
2109 { 734, 0xe602 }, { 740, 0xcf83 }, { 749, 0x8cf5 }, { 758, 0x3555 },
2110 { 766, 0xe048 }, { 771, 0xffab }, { 784, 0x92b9 }, { 792, 0xd859 },
2111 { 800, 0xab18 }, { 807, 0x2892 }, { 812, 0xd7e9 }, { 823, 0x8020 },
2112 { 825, 0xc438 }, { 831, 0xf583 }, { 840, 0xe74a }, { 849, 0x450a },
2113 /* 0x5000 */
2114 { 854, 0xb000 }, { 857, 0x9714 }, { 864, 0x7762 }, { 873, 0x5400 },
2115 { 876, 0xd188 }, { 882, 0x1420 }, { 885, 0x1020 }, { 887, 0xc8c0 },
2116 { 892, 0x2121 }, { 896, 0x0000 }, { 896, 0x13a8 }, { 902, 0x0c04 },
2117 { 905, 0x8000 }, { 906, 0x0440 }, { 908, 0x70c0 }, { 913, 0x0828 },
2118 /* 0x5100 */
2119 { 916, 0x08c0 }, { 919, 0x0004 }, { 920, 0x0002 }, { 921, 0x8000 },
2120 { 922, 0x2b7b }, { 932, 0x1472 }, { 938, 0x7924 }, { 945, 0x3bfb },
2121 { 957, 0x3327 }, { 965, 0x1ae4 }, { 972, 0x9835 }, { 979, 0x38ef },
2122 { 989, 0x9ad1 }, { 997, 0x2802 }, { 1000, 0xa813 }, { 1006, 0xbf69 },
2123 /* 0x5200 */
2124 { 1017, 0x65cf }, { 1027, 0x2fc6 }, { 1036, 0x6b11 }, { 1043, 0xafc9 },
2125 { 1053, 0x340f }, { 1060, 0x5053 }, { 1066, 0x86a2 }, { 1072, 0xa004 },
2126 { 1075, 0x0106 }, { 1078, 0xe809 }, { 1084, 0x3f0f }, { 1094, 0xc00e },
2127 { 1099, 0x0a88 }, { 1103, 0x8145 }, { 1108, 0x0010 }, { 1109, 0xc601 },
2128 /* 0x5300 */
2129 { 1114, 0xa161 }, { 1120, 0x26e1 }, { 1127, 0x444b }, { 1133, 0xce00 },
2130 { 1138, 0xc7aa }, { 1147, 0xd4ee }, { 1157, 0xcadf }, { 1168, 0x85bb },
2131 { 1177, 0x3a74 }, { 1185, 0xa520 }, { 1190, 0x436c }, { 1197, 0x8840 },
2132 { 1200, 0x3f06 }, { 1208, 0x8bd2 }, { 1216, 0xff79 }, { 1229, 0x3bef },
2133 /* 0x5400 */
2134 { 1241, 0xf75a }, { 1252, 0xe8ef }, { 1263, 0xfbc9 }, { 1275, 0x5b36 },
2135 { 1284, 0x0d49 }, { 1290, 0x1bfd }, { 1301, 0x0154 }, { 1305, 0x39ee },
2136 { 1315, 0xd855 }, { 1323, 0x2e75 }, { 1332, 0xbfd8 }, { 1343, 0xa91a },
2137 { 1350, 0xf3d7 }, { 1362, 0xf6bf }, { 1375, 0x67e0 }, { 1383, 0xb40c },
2138 /* 0x5500 */
2139 { 1389, 0x82c2 }, { 1394, 0x0813 }, { 1398, 0xd49d }, { 1407, 0xd08b },
2140 { 1414, 0x065a }, { 1420, 0x1061 }, { 1424, 0x74f2 }, { 1433, 0x59e0 },
2141 { 1440, 0x8f9f }, { 1451, 0xb312 }, { 1458, 0x0080 }, { 1459, 0x6aaa },
2142 { 1467, 0x3230 }, { 1472, 0xb05e }, { 1480, 0x9d7a }, { 1490, 0x60ac },
2143 /* 0x5600 */
2144 { 1496, 0xd303 }, { 1503, 0xc900 }, { 1507, 0x3098 }, { 1512, 0x8a56 },
2145 { 1519, 0x7000 }, { 1522, 0x1390 }, { 1527, 0x1f14 }, { 1534, 0x1842 },
2146 { 1538, 0xc060 }, { 1542, 0x0008 }, { 1543, 0x8008 }, { 1545, 0x1080 },
2147 { 1547, 0x0400 }, { 1548, 0xec90 }, { 1555, 0x2817 }, { 1561, 0xe633 },
2148 /* 0x5700 */
2149 { 1570, 0x0758 }, { 1576, 0x9000 }, { 1578, 0xf708 }, { 1586, 0x4e09 },
2150 { 1592, 0xf485 }, { 1600, 0xfc83 }, { 1609, 0xaf53 }, { 1619, 0x18c8 },
2151 { 1624, 0x187c }, { 1631, 0x080c }, { 1634, 0x6adf }, { 1645, 0x0114 },
2152 { 1648, 0xc80c }, { 1653, 0xa734 }, { 1661, 0xa011 }, { 1665, 0x2710 },
2153 /* 0x5800 */
2154 { 1670, 0x28c5 }, { 1676, 0x4222 }, { 1680, 0x0413 }, { 1684, 0x0021 },
2155 { 1686, 0x3010 }, { 1689, 0x4112 }, { 1693, 0x1820 }, { 1696, 0x4000 },
2156 { 1697, 0x022b }, { 1702, 0xc60c }, { 1708, 0x0300 }, { 1710, 0x1000 },
2157 { 1711, 0x0022 }, { 1713, 0x0022 }, { 1715, 0x5810 }, { 1719, 0x0249 },
2158 /* 0x5900 */
2159 { 1723, 0xa094 }, { 1728, 0x9670 }, { 1735, 0xeeb0 }, { 1744, 0x1792 },
2160 { 1751, 0xcb96 }, { 1760, 0x05f2 }, { 1767, 0x0025 }, { 1770, 0x2358 },
2161 { 1776, 0x25de }, { 1785, 0x42cc }, { 1791, 0xcf38 }, { 1800, 0x4a04 },
2162 { 1804, 0x0c40 }, { 1807, 0x359f }, { 1817, 0x1128 }, { 1821, 0x8a00 },
2163 /* 0x5a00 */
2164 { 1824, 0x13fa }, { 1833, 0x910a }, { 1838, 0x0229 }, { 1842, 0x1056 },
2165 { 1847, 0x0641 }, { 1851, 0x0420 }, { 1853, 0x0484 }, { 1856, 0x84f0 },
2166 { 1862, 0x0000 }, { 1862, 0x0c04 }, { 1865, 0x0400 }, { 1866, 0x412c },
2167 { 1871, 0x1206 }, { 1875, 0x1154 }, { 1880, 0x0a4b }, { 1886, 0x0002 },
2168 /* 0x5b00 */
2169 { 1887, 0x0200 }, { 1888, 0x00c0 }, { 1890, 0x0000 }, { 1890, 0x0094 },
2170 { 1893, 0x0001 }, { 1894, 0xbfb9 }, { 1907, 0x167c }, { 1915, 0x242b },
2171 { 1921, 0x9bbb }, { 1932, 0x7fa8 }, { 1942, 0x0c7f }, { 1951, 0xe379 },
2172 { 1961, 0x10f4 }, { 1967, 0xe00d }, { 1973, 0x4132 }, { 1978, 0x9f01 },
2173 /* 0x5c00 */
2174 { 1985, 0x8652 }, { 1991, 0x3572 }, { 1999, 0x10b4 }, { 2004, 0xff12 },
2175 { 2014, 0xc2f7 }, { 2024, 0x4223 }, { 2029, 0xc06b }, { 2036, 0x8602 },
2176 { 2040, 0x3106 }, { 2045, 0x1fd3 }, { 2055, 0x3a0c }, { 2061, 0xa1aa },
2177 { 2068, 0x0812 }, { 2071, 0x0204 }, { 2073, 0x2572 }, { 2080, 0x0801 },
2178 /* 0x5d00 */
2179 { 2082, 0x40cc }, { 2087, 0x4850 }, { 2091, 0x62d0 }, { 2097, 0x6010 },
2180 { 2100, 0x1c80 }, { 2104, 0x2900 }, { 2107, 0x9a00 }, { 2111, 0x0010 },
2181 { 2112, 0x0004 }, { 2113, 0x2200 }, { 2115, 0x0000 }, { 2115, 0x0080 },
2182 { 2116, 0x2020 }, { 2118, 0x6800 }, { 2121, 0xcbe6 }, { 2131, 0x609e },

```

```
2183 /* 0x5e00 */
2184 { 2138, 0x916e }, { 2146, 0x3f73 }, { 2157, 0x60c0 }, { 2161, 0x3982 },
2185 { 2167, 0x1034 }, { 2171, 0x4830 }, { 2175, 0x0006 }, { 2177, 0xbd5c },
2186 { 2187, 0x8cd1 }, { 2194, 0xd6fb }, { 2206, 0x20e1 }, { 2211, 0x43e8 },
2187 { 2218, 0x0600 }, { 2220, 0x084e }, { 2225, 0x0500 }, { 2227, 0xc4d0 },
2188 /* 0x5f00 */
2189 { 2233, 0x8d1f }, { 2242, 0x89aa }, { 2249, 0xa6e1 }, { 2257, 0x1602 },
2190 { 2261, 0x0001 }, { 2262, 0x21ed }, { 2270, 0x3656 }, { 2278, 0x1a8b },
2191 { 2285, 0x1fb7 }, { 2296, 0x13a5 }, { 2303, 0x6502 }, { 2308, 0x30a0 },
2192 { 2312, 0xb278 }, { 2320, 0x23c7 }, { 2328, 0x6c93 }, { 2336, 0xe922 },
2193 /* 0x6000 */
2194 { 2343, 0xe47f }, { 2354, 0x3a74 }, { 2362, 0x8fe3 }, { 2372, 0x9820 },
2195 { 2376, 0x280e }, { 2381, 0x2625 }, { 2387, 0xbf9c }, { 2398, 0xbf49 },
2196 { 2408, 0x3218 }, { 2413, 0xac54 }, { 2420, 0xb949 }, { 2428, 0x1916 },
2197 { 2434, 0x0c60 }, { 2438, 0xb522 }, { 2445, 0xfbc1 }, { 2455, 0x0659 },
2198 /* 0x6100 */
2199 { 2461, 0xe343 }, { 2469, 0x8420 }, { 2472, 0x08d9 }, { 2478, 0x8000 },
2200 { 2479, 0x5500 }, { 2483, 0x2022 }, { 2486, 0x0184 }, { 2489, 0x00a1 },
2201 { 2492, 0x4800 }, { 2494, 0x2010 }, { 2496, 0x1380 }, { 2500, 0x4080 },
2202 { 2502, 0x0d04 }, { 2506, 0x0016 }, { 2509, 0x0040 }, { 2510, 0x8020 },
2203 /* 0x6200 */
2204 { 2512, 0xfd40 }, { 2520, 0x8de7 }, { 2530, 0x5436 }, { 2537, 0xe098 },
2205 { 2543, 0x7b8b }, { 2553, 0x091e }, { 2559, 0xfec8 }, { 2569, 0xd249 },
2206 { 2576, 0x0611 }, { 2580, 0x8dee }, { 2590, 0x1937 }, { 2598, 0xba22 },
2207 { 2605, 0x77f4 }, { 2616, 0x9fdd }, { 2628, 0xf3ec }, { 2639, 0xf0da },
2208 /* 0x6300 */
2209 { 2648, 0x4386 }, { 2654, 0xec42 }, { 2661, 0x8d3f }, { 2671, 0x2604 },
2210 { 2675, 0xfa6c }, { 2685, 0xc021 }, { 2689, 0x628e }, { 2696, 0x0cc2 },
2211 { 2701, 0xd785 }, { 2710, 0x0145 }, { 2714, 0x77ad }, { 2725, 0x5599 },
2212 { 2733, 0xe250 }, { 2739, 0x4045 }, { 2743, 0x260b }, { 2749, 0xa154 },
2213 /* 0x6400 */
2214 { 2755, 0x9827 }, { 2762, 0x5819 }, { 2768, 0x3443 }, { 2774, 0xa410 },
2215 { 2778, 0x05f2 }, { 2785, 0x4114 }, { 2789, 0x2280 }, { 2792, 0x0700 },
2216 { 2795, 0x00b4 }, { 2799, 0x4266 }, { 2805, 0x7210 }, { 2810, 0x15a1 },
2217 { 2816, 0x6025 }, { 2821, 0x4185 }, { 2826, 0x0054 }, { 2829, 0x0000 },
2218 /* 0x6500 */
2219 { 2829, 0x0201 }, { 2831, 0x0104 }, { 2833, 0xc820 }, { 2837, 0xcb70 },
2220 { 2845, 0x9320 }, { 2850, 0x6a62 }, { 2857, 0x184c }, { 2862, 0x0095 },
2221 { 2866, 0x1880 }, { 2869, 0x9a8b }, { 2877, 0xaaab }, { 2885, 0x3201 },
2222 { 2889, 0xd87a }, { 2898, 0x00c4 }, { 2901, 0xf3e5 }, { 2912, 0x04c3 },
2223 /* 0x6600 */
2224 { 2917, 0xd44d }, { 2925, 0xa238 }, { 2931, 0xa1a1 }, { 2937, 0x5072 },
2225 { 2943, 0x980a }, { 2948, 0x84fc }, { 2956, 0xc152 }, { 2962, 0x44d1 },
2226 { 2968, 0x1094 }, { 2972, 0x20c2 }, { 2976, 0x4180 }, { 2979, 0x4210 },
2227 { 2982, 0x0000 }, { 2982, 0x3a00 }, { 2986, 0x0240 }, { 2988, 0xd29d },
2228 /* 0x6700 */
2229 { 2997, 0x2f01 }, { 3003, 0xa8b1 }, { 3010, 0xbd40 }, { 3017, 0x2432 },
2230 { 3022, 0xd34d }, { 3031, 0xd04b }, { 3038, 0xa723 }, { 3046, 0xd0ad },
2231 { 3054, 0x0a92 }, { 3059, 0x75a1 }, { 3067, 0xadac }, { 3076, 0x01e9 },
2232 { 3082, 0x801a }, { 3086, 0x771f }, { 3097, 0x9225 }, { 3103, 0xa01b },
2233 /* 0x6800 */
2234 { 3109, 0xdfa1 }, { 3119, 0x20ca }, { 3124, 0x0602 }, { 3127, 0x738c },
2235 { 3135, 0x577f }, { 3147, 0x003b }, { 3152, 0x0bff }, { 3163, 0x00d0 },
2236 { 3166, 0x806a }, { 3171, 0x0088 }, { 3173, 0xa1c4 }, { 3179, 0x0029 },
2237 { 3182, 0x2a05 }, { 3187, 0x0524 }, { 3191, 0x4009 }, { 3194, 0x1623 },
2238 /* 0x6900 */
2239 { 3200, 0x6822 }, { 3205, 0x8005 }, { 3208, 0x2011 }, { 3211, 0xa211 },
2240 { 3216, 0x0004 }, { 3217, 0x6490 }, { 3222, 0x4849 }, { 3227, 0x1382 },
2241 { 3232, 0x23d5 }, { 3240, 0x1930 }, { 3245, 0x2980 }, { 3249, 0x0892 },
2242 { 3253, 0x5402 }, { 3257, 0x8811 }, { 3261, 0x2001 }, { 3263, 0xa004 },
2243 /* 0x6a00 */
2244 { 3266, 0x0400 }, { 3267, 0x8180 }, { 3270, 0x8502 }, { 3274, 0x6022 },
2245 { 3278, 0x0090 }, { 3280, 0x0b01 }, { 3284, 0x0022 }, { 3286, 0x1202 },
2246 { 3289, 0x4011 }, { 3292, 0x0083 }, { 3295, 0x1a01 }, { 3299, 0x0000 },
2247 { 3299, 0x0000 }, { 3299, 0x0000 }, { 3299, 0x0000 }, { 3299, 0x0000 },
2248 /* 0x6b00 */
2249 { 3299, 0x0000 }, { 3299, 0x0000 }, { 3299, 0x009f }, { 3305, 0x4684 },
2250 { 3310, 0x12c8 }, { 3315, 0x0200 }, { 3316, 0x04fc }, { 3323, 0x1a00 },
2251 { 3326, 0x2ede }, { 3336, 0x0c4c }, { 3341, 0x0402 }, { 3343, 0x80b8 },
2252 { 3348, 0xa826 }, { 3354, 0x0afc }, { 3362, 0x8c02 }, { 3366, 0x2228 },
2253 /* 0x6c00 */
2254 { 3370, 0xa0e0 }, { 3375, 0x8f7b }, { 3386, 0xc7d6 }, { 3396, 0x2135 },
2255 { 3402, 0x06c7 }, { 3409, 0xf8b1 }, { 3418, 0x0713 }, { 3424, 0x6255 },
2256 { 3431, 0x936e }, { 3440, 0x8a19 }, { 3446, 0x6efa }, { 3457, 0xfb0e },
2257 { 3467, 0x1630 }, { 3472, 0x48f9 }, { 3480, 0xcd2f }, { 3490, 0x7deb },
2258 /* 0x6d00 */
2259 { 3502, 0x5892 }, { 3508, 0x4e84 }, { 3514, 0x4ca0 }, { 3519, 0x7a2e },
2260 { 3528, 0xedea }, { 3539, 0x561e }, { 3547, 0xc649 }, { 3554, 0x1190 },
2261 { 3558, 0x5324 }, { 3564, 0xe83a }, { 3572, 0xcfdb }, { 3584, 0x8124 },
2262 { 3588, 0x18f1 }, { 3595, 0x6342 }, { 3601, 0x5853 }, { 3608, 0x1a8a },
2263 /* 0x6e00 */
2264 { 3614, 0x7420 }, { 3619, 0x24d3 }, { 3626, 0xaa3b }, { 3635, 0x0514 },
2265 { 3639, 0x6018 }, { 3643, 0x8958 }, { 3649, 0x4800 }, { 3651, 0xc000 },
2266 { 3653, 0x8268 }, { 3658, 0x9101 }, { 3662, 0x84a4 }, { 3667, 0x2cd6 },
2267 { 3675, 0x8886 }, { 3680, 0xc4ba }, { 3688, 0x0377 }, { 3696, 0x0210 },
2268 /* 0x6f00 */
2269 { 3698, 0x8244 }, { 3702, 0x0038 }, { 3705, 0xae11 }, { 3712, 0x404a },
```

```

2270 { 3716, 0x28c0 }, { 3720, 0x5100 }, { 3723, 0x6044 }, { 3727, 0x1514 },
2271 { 3732, 0x7310 }, { 3738, 0x1000 }, { 3739, 0x0082 }, { 3741, 0x0248 },
2272 { 3744, 0x0205 }, { 3747, 0x4006 }, { 3750, 0xc003 }, { 3754, 0x0000 },
2273 /* 0x7000 */
2274 { 3754, 0x0000 }, { 3754, 0x0c02 }, { 3757, 0x0008 }, { 3758, 0x0220 },
2275 { 3760, 0x9000 }, { 3762, 0x4000 }, { 3763, 0xb800 }, { 3767, 0xd161 },
2276 { 3774, 0x4621 }, { 3779, 0x3274 }, { 3786, 0xf800 }, { 3791, 0x3b8a },
2277 { 3799, 0x050f }, { 3805, 0x8b00 }, { 3809, 0xbb00 }, { 3818, 0x2280 },
2278 /* 0x7100 */
2279 { 3821, 0x0600 }, { 3823, 0x0769 }, { 3830, 0x8040 }, { 3832, 0x0043 },
2280 { 3835, 0x5420 }, { 3839, 0x5000 }, { 3841, 0x41d0 }, { 3846, 0x250c },
2281 { 3851, 0x8410 }, { 3854, 0x8310 }, { 3858, 0x1101 }, { 3861, 0x0228 },
2282 { 3864, 0x4008 }, { 3866, 0x0030 }, { 3868, 0x40a1 }, { 3872, 0x0200 },
2283 /* 0x7200 */
2284 { 3873, 0x0040 }, { 3874, 0x2000 }, { 3875, 0x1500 }, { 3878, 0xab3e },
2285 { 3888, 0x3180 }, { 3892, 0xaa44 }, { 3898, 0xc2c6 }, { 3905, 0xc624 },
2286 { 3911, 0xac13 }, { 3918, 0x8004 }, { 3920, 0xb000 }, { 3923, 0x03d1 },
2287 { 3929, 0x611e }, { 3936, 0x4285 }, { 3941, 0xf303 }, { 3949, 0x1d9f },
2288 /* 0x7300 */
2289 { 3959, 0x440a }, { 3963, 0x78e8 }, { 3971, 0x5e26 }, { 3979, 0xc392 },
2290 { 3986, 0x2000 }, { 3987, 0x0085 }, { 3990, 0xb001 }, { 3994, 0x4000 },
2291 { 3995, 0x4a90 }, { 4000, 0x8842 }, { 4004, 0xca04 }, { 4009, 0x0c8d },
2292 { 4015, 0xa705 }, { 4022, 0x4203 }, { 4026, 0x22a1 }, { 4031, 0x0004 },
2293 /* 0x7400 */
2294 { 4032, 0x8668 }, { 4038, 0x0c01 }, { 4041, 0x5564 }, { 4048, 0x1079 },
2295 { 4054, 0x0002 }, { 4055, 0xdea0 }, { 4063, 0x2000 }, { 4064, 0x40c1 },
2296 { 4068, 0x488b }, { 4074, 0x5001 }, { 4077, 0x0380 }, { 4080, 0x0400 },
2297 { 4081, 0x0000 }, { 4081, 0x5004 }, { 4084, 0xc05d }, { 4091, 0x80d0 },
2298 /* 0x7500 */
2299 { 4095, 0xa010 }, { 4098, 0x970a }, { 4105, 0xbb20 }, { 4112, 0x4daf },
2300 { 4122, 0xd921 }, { 4129, 0x1e10 }, { 4134, 0x0460 }, { 4137, 0x8314 },
2301 { 4142, 0x8848 }, { 4146, 0xa6d6 }, { 4155, 0xd83b }, { 4164, 0x733f },
2302 { 4175, 0x27bc }, { 4184, 0x4974 }, { 4191, 0x0ddc }, { 4199, 0x9213 },
2303 /* 0x7600 */
2304 { 4205, 0x142b }, { 4211, 0x8ba1 }, { 4218, 0x2e75 }, { 4227, 0xd139 },
2305 { 4235, 0x3009 }, { 4239, 0x5050 }, { 4243, 0x8808 }, { 4246, 0x6900 },
2306 { 4250, 0x49d4 }, { 4257, 0x024a }, { 4261, 0x4010 }, { 4263, 0x8016 },
2307 { 4267, 0xe564 }, { 4275, 0x89d7 }, { 4284, 0xc020 }, { 4287, 0x5316 },
2308 /* 0x7700 */
2309 { 4294, 0x2b92 }, { 4301, 0x8600 }, { 4304, 0xa345 }, { 4311, 0x15e0 },
2310 { 4317, 0x008b }, { 4321, 0x0c03 }, { 4325, 0x196e }, { 4333, 0xe200 },
2311 { 4337, 0x7031 }, { 4343, 0x8006 }, { 4346, 0x16a5 }, { 4353, 0xa829 },
2312 { 4359, 0x2000 }, { 4360, 0x1880 }, { 4363, 0x7aac }, { 4372, 0xe148 },
2313 /* 0x7800 */
2314 { 4378, 0x3207 }, { 4384, 0xb5d6 }, { 4394, 0x32e8 }, { 4401, 0x5f91 },
2315 { 4410, 0x50a1 }, { 4415, 0x20e5 }, { 4421, 0x7c00 }, { 4426, 0x1080 },
2316 { 4428, 0x7280 }, { 4433, 0x9d8a }, { 4441, 0x00aa }, { 4445, 0x421f },
2317 { 4452, 0x0e22 }, { 4457, 0x0231 }, { 4461, 0x1100 }, { 4463, 0x0494 },
2318 /* 0x7900 */
2319 { 4467, 0x0022 }, { 4469, 0x4008 }, { 4471, 0x0010 }, { 4472, 0x5c10 },
2320 { 4477, 0x0343 }, { 4482, 0xfcc8 }, { 4491, 0xa1a5 }, { 4498, 0x0580 },
2321 { 4501, 0x8433 }, { 4507, 0x0400 }, { 4508, 0x0080 }, { 4509, 0x6e08 },
2322 { 4515, 0x2a4b }, { 4522, 0x8126 }, { 4527, 0xaaad }, { 4535, 0x2901 },
2323 /* 0x7a00 */
2324 { 4539, 0x684d }, { 4546, 0x4490 }, { 4550, 0x0009 }, { 4552, 0xba88 },
2325 { 4559, 0x0040 }, { 4560, 0x0082 }, { 4562, 0x0000 }, { 4562, 0x87d1 },
2326 { 4570, 0x215b }, { 4577, 0xb1e6 }, { 4586, 0x3161 }, { 4592, 0x8008 },
2327 { 4594, 0x0800 }, { 4595, 0xc240 }, { 4599, 0xa069 }, { 4605, 0xa600 },
2328 /* 0x7b00 */
2329 { 4609, 0x8d58 }, { 4616, 0x4a32 }, { 4622, 0x5d71 }, { 4631, 0x550a },
2330 { 4637, 0x9aa0 }, { 4643, 0x2d57 }, { 4652, 0x4005 }, { 4655, 0x4aa6 },
2331 { 4662, 0x2021 }, { 4665, 0x30b1 }, { 4671, 0x3fc6 }, { 4681, 0x0112 },
2332 { 4684, 0x10c2 }, { 4688, 0x260a }, { 4693, 0x4462 }, { 4698, 0x5082 },
2333 /* 0x7c00 */
2334 { 4702, 0x9880 }, { 4706, 0x8040 }, { 4708, 0x04c0 }, { 4711, 0x8100 },
2335 { 4713, 0x2003 }, { 4716, 0x0000 }, { 4716, 0x0000 }, { 4716, 0x3818 },
2336 { 4721, 0x0200 }, { 4722, 0xf1a6 }, { 4731, 0x4434 }, { 4736, 0x720e },
2337 { 4743, 0x35a2 }, { 4750, 0x92e0 }, { 4756, 0x8101 }, { 4759, 0x0900 },
2338 /* 0x7d00 */
2339 { 4761, 0x0400 }, { 4762, 0x0000 }, { 4762, 0x8885 }, { 4767, 0x0000 },
2340 { 4767, 0x0000 }, { 4767, 0x0000 }, { 4767, 0x4000 }, { 4768, 0x0080 },
2341 { 4769, 0x0000 }, { 4769, 0x0000 }, { 4769, 0x4040 }, { 4771, 0x0000 },
2342 { 4771, 0x0000 }, { 4771, 0x0000 }, { 4771, 0x0000 }, { 4771, 0x0000 },
2343 /* 0x7e00 */
2344 { 4771, 0x0000 }, { 4771, 0x0000 }, { 4771, 0x0000 }, { 4771, 0x0800 },
2345 { 4772, 0x0082 }, { 4774, 0x0000 }, { 4774, 0x0000 }, { 4774, 0x0000 },
2346 { 4774, 0x0004 }, { 4775, 0x8800 }, { 4777, 0xbfff }, { 4792, 0xe7ef },
2347 { 4805, 0xffff }, { 4821, 0xffbf }, { 4836, 0xfefef }, { 4850, 0xfdff },
2348 /* 0x7f00 */
2349 { 4865, 0xfbff }, { 4880, 0xbffe }, { 4894, 0xffff }, { 4910, 0x057f },
2350 { 4919, 0x0034 }, { 4922, 0x85b3 }, { 4930, 0x4706 }, { 4936, 0x4216 },
2351 { 4941, 0x5402 }, { 4945, 0xe410 }, { 4950, 0x8092 }, { 4954, 0xb305 },
2352 { 4961, 0x5422 }, { 4966, 0x8130 }, { 4970, 0x4263 }, { 4976, 0x180b },
2353 /* 0x8000 */
2354 { 4981, 0x387b }, { 4990, 0x13f5 }, { 4999, 0x07e5 }, { 5007, 0xa9ea },
2355 { 5016, 0x3c4c }, { 5023, 0x0514 }, { 5027, 0x0600 }, { 5029, 0x8002 },
2356 { 5031, 0x1ad9 }, { 5039, 0xbd48 }, { 5047, 0xee37 }, { 5058, 0xf496 },

```

```

2357 { 5067, 0x705f }, { 5076, 0x7ec0 }, { 5084, 0xbfb2 }, { 5095, 0x355f },
2358 /* 0x8100 */
2359 { 5105, 0xe644 }, { 5112, 0x455f }, { 5121, 0x9000 }, { 5123, 0x4146 },
2360 { 5128, 0x1d40 }, { 5133, 0x063b }, { 5140, 0x62a1 }, { 5146, 0xfe13 },
2361 { 5156, 0x8505 }, { 5161, 0x3902 }, { 5166, 0x0548 }, { 5170, 0x0c08 },
2362 { 5173, 0x144f }, { 5180, 0x0000 }, { 5180, 0x3488 }, { 5185, 0x5818 },
2363 /* 0x8200 */
2364 { 5190, 0x3077 }, { 5198, 0xd815 }, { 5205, 0xbd0e }, { 5214, 0x4bfb },
2365 { 5225, 0x8a90 }, { 5230, 0x8500 }, { 5233, 0xc100 }, { 5236, 0xe61d },
2366 { 5245, 0xed14 }, { 5253, 0xb386 }, { 5261, 0xff72 }, { 5273, 0x639b },
2367 { 5282, 0xfd92 }, { 5292, 0xd9be }, { 5303, 0x887b }, { 5311, 0x0a92 },
2368 /* 0x8300 */
2369 { 5316, 0xd3fe }, { 5328, 0x1cb2 }, { 5335, 0xb980 }, { 5341, 0x177a },
2370 { 5350, 0x82c9 }, { 5356, 0xdc17 }, { 5365, 0xffffb }, { 5380, 0x3980 },
2371 { 5385, 0x4260 }, { 5389, 0x590c }, { 5395, 0x0f01 }, { 5400, 0x37df },
2372 { 5412, 0x94a3 }, { 5419, 0xb150 }, { 5425, 0x0623 }, { 5430, 0x2307 },
2373 /* 0x8400 */
2374 { 5436, 0xf85a }, { 5445, 0x3102 }, { 5449, 0x01f0 }, { 5454, 0x3102 },
2375 { 5458, 0x0040 }, { 5459, 0x1e82 }, { 5465, 0x3a0a }, { 5471, 0x056a },
2376 { 5477, 0x5b84 }, { 5484, 0x1280 }, { 5487, 0x8002 }, { 5489, 0xa714 },
2377 { 5496, 0x2612 }, { 5501, 0xa04b }, { 5507, 0x1069 }, { 5512, 0x9001 },
2378 /* 0x8500 */
2379 { 5515, 0x1000 }, { 5516, 0x848a }, { 5521, 0x1802 }, { 5524, 0x3f80 },
2380 { 5531, 0x0708 }, { 5535, 0x4240 }, { 5538, 0x0110 }, { 5540, 0x4e14 },
2381 { 5546, 0x80b0 }, { 5550, 0x1800 }, { 5552, 0xc510 }, { 5557, 0x0281 },
2382 { 5560, 0x8202 }, { 5563, 0x1029 }, { 5567, 0x0210 }, { 5569, 0x8800 },
2383 /* 0x8600 */
2384 { 5571, 0x0020 }, { 5572, 0x0042 }, { 5574, 0x0280 }, { 5576, 0x1100 },
2385 { 5578, 0xe000 }, { 5581, 0x4413 }, { 5586, 0x5804 }, { 5590, 0xfe02 },
2386 { 5598, 0x3c07 }, { 5605, 0x3028 }, { 5609, 0x9798 }, { 5617, 0x0473 },
2387 { 5623, 0xcd1 }, { 5632, 0xcb13 }, { 5640, 0x6210 }, { 5644, 0x431f },
2388 /* 0x8700 */
2389 { 5652, 0x278d }, { 5660, 0x55ac }, { 5668, 0x422e }, { 5674, 0xc892 },
2390 { 5680, 0x5380 }, { 5685, 0x0288 }, { 5688, 0x4039 }, { 5693, 0x7851 },
2391 { 5700, 0x292c }, { 5706, 0x8088 }, { 5709, 0xb900 }, { 5714, 0x2428 },
2392 { 5718, 0x0c41 }, { 5722, 0x080e }, { 5726, 0x4421 }, { 5730, 0x4200 },
2393 /* 0x8800 */
2394 { 5732, 0x0408 }, { 5734, 0x0868 }, { 5738, 0x0006 }, { 5740, 0x1204 },
2395 { 5743, 0x3031 }, { 5748, 0x0290 }, { 5751, 0x5b3e }, { 5761, 0xe085 },
2396 { 5767, 0x2936 }, { 5774, 0x1044 }, { 5777, 0x2814 }, { 5781, 0x1082 },
2397 { 5784, 0x4266 }, { 5790, 0x8334 }, { 5796, 0x013c }, { 5801, 0x531b },
2398 /* 0x8900 */
2399 { 5809, 0x0404 }, { 5811, 0x0e0d }, { 5817, 0x0c22 }, { 5821, 0x0051 },
2400 { 5824, 0x0012 }, { 5826, 0xc000 }, { 5828, 0x0040 }, { 5829, 0x8800 },
2401 { 5831, 0x004a }, { 5834, 0x0000 }, { 5834, 0x0000 }, { 5834, 0x0000 },
2402 { 5834, 0xdf6 }, { 5847, 0x5447 }, { 5854, 0x8868 }, { 5859, 0x0008 },
2403 /* 0x8a00 */
2404 { 5860, 0x0081 }, { 5862, 0x0000 }, { 5862, 0x0000 }, { 5862, 0x4000 },
2405 { 5863, 0x0100 }, { 5864, 0x0000 }, { 5864, 0x0000 }, { 5864, 0x0200 },
2406 { 5865, 0x0600 }, { 5867, 0x0008 }, { 5868, 0x0000 }, { 5868, 0x0000 },
2407 { 5868, 0x0000 }, { 5868, 0x0000 }, { 5868, 0x0000 }, { 5868, 0x0000 },
2408 /* 0x8b00 */
2409 { 5868, 0x0080 }, { 5869, 0x0000 }, { 5869, 0x0040 }, { 5870, 0x0000 },
2410 { 5870, 0x0000 }, { 5870, 0x0000 }, { 5870, 0x1040 }, { 5872, 0x0000 },
2411 { 5872, 0x0000 }, { 5872, 0x0000 }, { 5872, 0xffff }, { 5887, 0xf7fd },
2412 { 5901, 0xff7f }, { 5916, 0xffffe }, { 5931, 0xfbff }, { 5946, 0xfffff },
2413 /* 0x8c00 */
2414 { 5962, 0xfdf }, { 5977, 0xbfff }, { 5992, 0xfffff }, { 6008, 0x00ff },
2415 { 6016, 0x12c2 }, { 6021, 0x0420 }, { 6023, 0x0c06 }, { 6027, 0x0708 },
2416 { 6031, 0x1624 }, { 6036, 0x0110 }, { 6038, 0x0000 }, { 6038, 0x0000 },
2417 { 6038, 0x0000 }, { 6038, 0x0000 }, { 6038, 0x0000 }, { 6038, 0x0000 },
2418 /* 0x8d00 */
2419 { 6038, 0x0000 }, { 6038, 0xe000 }, { 6041, 0xffffe }, { 6056, 0xfffff },
2420 { 6072, 0xfffff }, { 6088, 0x7f79 }, { 6100, 0x28df }, { 6109, 0x00f9 },
2421 { 6115, 0x0c32 }, { 6120, 0x8012 }, { 6123, 0x0008 }, { 6124, 0xd53a },
2422 { 6133, 0xd858 }, { 6140, 0xecc2 }, { 6148, 0x9d18 }, { 6155, 0x2fa8 },
2423 /* 0x8e00 */
2424 { 6163, 0x9620 }, { 6168, 0xe010 }, { 6172, 0xd60c }, { 6179, 0x2622 },
2425 { 6184, 0x0f97 }, { 6193, 0x0206 }, { 6196, 0xb240 }, { 6201, 0x9055 },
2426 { 6207, 0x80a2 }, { 6211, 0x5011 }, { 6215, 0x9800 }, { 6218, 0x0404 },
2427 { 6220, 0x4000 }, { 6221, 0x0000 }, { 6221, 0x0000 }, { 6221, 0x0000 },
2428 /* 0x8f00 */
2429 { 6221, 0x0000 }, { 6221, 0x0000 }, { 6221, 0x0000 }, { 6221, 0x0000 },
2430 { 6221, 0x0000 }, { 6221, 0x0000 }, { 6221, 0xfbc0 }, { 6230, 0xfffff },
2431 { 6246, 0xeffe }, { 6260, 0xdffb }, { 6274, 0x0b08 }, { 6278, 0x6243 },
2432 { 6284, 0x41b6 }, { 6291, 0xfb3b }, { 6303, 0x6f74 }, { 6313, 0x2389 },
2433 /* 0x9000 */
2434 { 6319, 0xae7f }, { 6331, 0xecd7 }, { 6342, 0xe047 }, { 6349, 0x5960 },
2435 { 6355, 0xa096 }, { 6361, 0x098f }, { 6368, 0x612c }, { 6374, 0xa030 },
2436 { 6378, 0x090d }, { 6383, 0x2aaa }, { 6390, 0xd44e }, { 6398, 0x4f7b },
2437 { 6409, 0xc4b2 }, { 6416, 0x388b }, { 6423, 0xa9c6 }, { 6431, 0x6110 },
2438 /* 0x9100 */
2439 { 6435, 0x0014 }, { 6437, 0x4200 }, { 6439, 0x800c }, { 6442, 0x0202 },
2440 { 6444, 0xfe48 }, { 6453, 0x6485 }, { 6459, 0xd63e }, { 6469, 0xe3f7 },
2441 { 6481, 0x3aa0 }, { 6487, 0x0c07 }, { 6492, 0xe40c }, { 6498, 0x0430 },
2442 { 6501, 0xf680 }, { 6508, 0x1002 }, { 6510, 0x0000 }, { 6510, 0x0000 },
2443 /* 0x9200 */

```

```

2444 { 6510, 0x0000 }, { 6510, 0x0000 }, { 6510, 0x0000 }, { 6510, 0x0000 },
2445 { 6510, 0x0000 }, { 6510, 0x0000 }, { 6510, 0x0000 }, { 6510, 0x0010 },
2446 { 6511, 0x4000 }, { 6512, 0x0000 }, { 6512, 0x4000 }, { 6513, 0x0000 },
2447 { 6513, 0x0100 }, { 6514, 0x0000 }, { 6514, 0x0000 }, { 6514, 0x0000 },
2448 /* 0x9300 */
2449 { 6514, 0x0000 }, { 6514, 0x0000 }, { 6514, 0x0000 }, { 6514, 0x4000 },
2450 { 6515, 0x0000 }, { 6515, 0x0000 }, { 6515, 0x0400 }, { 6516, 0x0000 },
2451 { 6516, 0x8000 }, { 6517, 0x0000 }, { 6517, 0x0000 }, { 6517, 0x0000 },
2452 { 6517, 0x0400 }, { 6518, 0x0040 }, { 6519, 0x0000 }, { 6519, 0x0000 },
2453 /* 0x9400 */
2454 { 6519, 0x0000 }, { 6519, 0x0000 }, { 6519, 0x0000 }, { 6519, 0x4000 },
2455 { 6520, 0x0000 }, { 6520, 0x0000 }, { 6520, 0x0800 }, { 6521, 0x0000 },
2456 { 6521, 0xffe0 }, { 6532, 0xfebd }, { 6545, 0xffff }, { 6561, 0xffff },
2457 { 6577, 0x7f7f }, { 6591, 0xfbe7 }, { 6604, 0xfbf7 }, { 6619, 0xf7ff },
2458 /* 0x9500 */
2459 { 6634, 0xffff }, { 6650, 0xffff }, { 6665, 0xff7e }, { 6679, 0xdff7 },
2460 { 6693, 0xf6f7 }, { 6706, 0xfbdf }, { 6720, 0xbffe }, { 6734, 0x804f },
2461 { 6740, 0x0000 }, { 6740, 0x0000 }, { 6740, 0x0000 }, { 6740, 0x0000 },
2462 { 6740, 0x0000 }, { 6740, 0x0000 }, { 6740, 0xef00 }, { 6747, 0x7fff },
2463 /* 0x9600 */
2464 { 6762, 0xff7f }, { 6777, 0xb6f7 }, { 6789, 0x4406 }, { 6793, 0xb87e },
2465 { 6803, 0x3bf5 }, { 6814, 0x8831 }, { 6819, 0x1796 }, { 6827, 0x00f4 },
2466 { 6832, 0xa960 }, { 6838, 0x1391 }, { 6844, 0x0080 }, { 6845, 0x7249 },
2467 { 6852, 0xf2f3 }, { 6863, 0x0024 }, { 6865, 0x8701 }, { 6870, 0x42c8 },
2468 /* 0x9700 */
2469 { 6875, 0xe3d3 }, { 6885, 0x5048 }, { 6889, 0x2400 }, { 6891, 0x4305 },
2470 { 6896, 0x0000 }, { 6896, 0x4a4c }, { 6902, 0x0227 }, { 6907, 0x1058 },
2471 { 6911, 0x2820 }, { 6914, 0x0116 }, { 6918, 0xa809 }, { 6923, 0x0014 },
2472 { 6925, 0x0000 }, { 6925, 0x0000 }, { 6925, 0x3ec0 }, { 6932, 0x0068 },
2473 /* 0x9800 */
2474 { 6935, 0x0000 }, { 6935, 0x0000 }, { 6935, 0x0000 }, { 6935, 0x0000 },
2475 { 6935, 0x0000 }, { 6935, 0x0000 }, { 6935, 0x0000 }, { 6935, 0xffe0 },
2476 { 6946, 0xb7ff }, { 6960, 0xfddb }, { 6973, 0x00f7 }, { 6980, 0x0000 },
2477 { 6980, 0x4000 }, { 6981, 0xc72e }, { 6990, 0x0180 }, { 6992, 0x0000 },
2478 /* 0x9900 */
2479 { 6992, 0x2000 }, { 6993, 0x0001 }, { 6994, 0x4000 }, { 6995, 0x0000 },
2480 { 6995, 0x0000 }, { 6995, 0x0030 }, { 6997, 0xffa8 }, { 7008, 0xb4f7 },
2481 { 7019, 0xadf3 }, { 7030, 0x03ff }, { 7040, 0x0120 }, { 7042, 0x0000 },
2482 { 7042, 0x0000 }, { 7042, 0x0000 }, { 7042, 0x0000 }, { 7042, 0x0000 },
2483 /* 0x9a00 */
2484 { 7042, 0x0000 }, { 7042, 0x0000 }, { 7042, 0x0000 }, { 7042, 0x0000 },
2485 { 7042, 0x0000 }, { 7042, 0x0000 }, { 7042, 0xf000 }, { 7046, 0xfffb },
2486 { 7061, 0x9df7 }, { 7073, 0xfdcf }, { 7086, 0x01bf }, { 7094, 0x15c3 },
2487 { 7101, 0x1827 }, { 7107, 0x810a }, { 7111, 0xa842 }, { 7116, 0x0a00 },
2488 /* 0x9b00 */
2489 { 7118, 0x8108 }, { 7121, 0x8008 }, { 7123, 0x8008 }, { 7125, 0x1804 },
2490 { 7128, 0xa3be }, { 7138, 0x0012 }, { 7140, 0x0000 }, { 7140, 0x0000 },
2491 { 7140, 0x0000 }, { 7140, 0x0000 }, { 7140, 0x0000 }, { 7140, 0x0000 },
2492 { 7140, 0x0000 }, { 7140, 0x0000 }, { 7140, 0x0000 }, { 7140, 0x0000 },
2493 /* 0x9c00 */
2494 { 7140, 0x0000 }, { 7140, 0x0000 }, { 7140, 0x0000 }, { 7140, 0x0000 },
2495 { 7140, 0x0000 }, { 7140, 0x0000 }, { 7140, 0x0000 }, { 7140, 0x9000 },
2496 { 7142, 0x69e6 }, { 7151, 0xdc37 }, { 7161, 0x6bff }, { 7174, 0x3dff },
2497 { 7187, 0xfcf8 }, { 7198, 0xf3f9 }, { 7210, 0x0004 },
2498 };
2499 static const Summary16 gb2312_uni2indx_page9e[27] = {
2500 /* 0x9e00 */
2501 { 7211, 0x0000 }, { 7211, 0x8000 }, { 7212, 0xbf6f }, { 7225, 0xe7ee },
2502 { 7237, 0xdffe }, { 7251, 0x5da2 }, { 7259, 0x3fd8 }, { 7269, 0xc00b },
2503 { 7274, 0x0984 }, { 7278, 0xa00c }, { 7282, 0x0040 }, { 7283, 0x6910 },
2504 { 7288, 0xe210 }, { 7293, 0xb912 }, { 7300, 0x86a5 }, { 7307, 0x5a00 },
2505 /* 0x9f00 */
2506 { 7311, 0x6800 }, { 7314, 0x0289 }, { 7318, 0x9005 }, { 7322, 0x6a80 },
2507 { 7327, 0x0010 }, { 7328, 0x0003 }, { 7330, 0x0000 }, { 7330, 0x8000 },
2508 { 7331, 0x1ff9 }, { 7342, 0x8e00 }, { 7346, 0x0001 },
2509 };
2510 static const Summary16 gb2312_uni2indx_pageff[15] = {
2511 /* 0xff00 */
2512 { 7347, 0xffff }, { 7362, 0xffff }, { 7378, 0xffff }, { 7394, 0xffff },
2513 { 7410, 0xffff }, { 7426, 0x7fff }, { 7441, 0x0000 }, { 7441, 0x0000 },
2514 { 7441, 0x0000 }, { 7441, 0x0000 }, { 7441, 0x0000 }, { 7441, 0x0000 },
2515 { 7441, 0x0000 }, { 7441, 0x0000 }, { 7441, 0x002b },
2516 };
2517
2518 static int
2519 gb2312_wctomb (conv_t conv, unsigned char *r, ucs4_t wc, int n)
2520 {
2521     if (n >= 2) {
2522         const Summary16 *summary = NULL;
2523         if (wc < 0x0460)
2524             summary = &gb2312_uni2indx_page00[(wc>4)];
2525         else if (wc >= 0x2000 && wc < 0x2650)
2526             summary = &gb2312_uni2indx_page20[(wc>4)-0x200];
2527         else if (wc >= 0x3000 && wc < 0x3230)
2528             summary = &gb2312_uni2indx_page30[(wc>4)-0x300];
2529         else if (wc >= 0x4e00 && wc < 0x9cf0)
2530             summary = &gb2312_uni2indx_page4e[(wc>4)-0x4e0];

```

```

2531     else if (wc >= 0x9e00 && wc < 0x9fb0)
2532         summary = &gb2312_uni2indx_page9e[(wc>4)-0x9e0];
2533     else if (wc >= 0xff00 && wc < 0xfff0)
2534         summary = &gb2312_uni2indx_pageff[(wc>4)-0xff0];
2535     if (summary) {
2536         unsigned short used = summary->used;
2537         unsigned int i = wc & 0x0f;
2538         if (used & ((unsigned short) 1 << i)) {
2539             unsigned short c;
2540             /* Keep in 'used' only the bits 0..i-1. */
2541             used &= ((unsigned short) 1 << i) - 1;
2542             /* Add 'summary->indx' and the number of bits set in 'used'. */
2543             used = (used & 0x5555) + ((used & 0xaaaa) >> 1);
2544             used = (used & 0x3333) + ((used & 0xcccc) >> 2);
2545             used = (used & 0x0f0f) + ((used & 0xf0f0) >> 4);
2546             used = (used & 0x00ff) + (used >> 8);
2547             c = gb2312_2charset[summary->indx + used];
2548             r[0] = (c >> 8); r[1] = (c & 0xff);
2549             return 2;
2550         }
2551     }
2552     return RET_ILSEQ;
2553 }
2554 return RET_TOOSMALL;
2555 }
2556 #endif /* NEED_TOMB */

```

32.209 georgian_academy.h

```

1 /* $XFree86: xc/lib/X11/lcUniConv/georgian_academy.h,v 1.3 2000/11/29 17:40:29 dawes Exp $ */
2
3 /*
4 * GEORGIAN-ACADEMY
5 */
6
7 static const unsigned short georgian_academy_2uni[32] = {
8     /* 0x80 */
9     0x0080, 0x0081, 0x201a, 0x0192, 0x201e, 0x2026, 0x2020, 0x2021,
10    0x02c6, 0x2030, 0x0160, 0x2039, 0x0152, 0x008d, 0x008e, 0x008f,
11    /* 0x90 */
12    0x0090, 0x2018, 0x2019, 0x201c, 0x201d, 0x2022, 0x2013, 0x2014,
13    0x02dc, 0x2122, 0x0161, 0x203a, 0x0153, 0x009d, 0x009e, 0x0178,
14 };
15
16 static int
17 georgian_academy_mbtowc (conv_t conv, ucs4_t *pwc, const unsigned char *s, int n)
18 {
19     unsigned char c = *s;
20     if (c >= 0x80 && c < 0xa0)
21         *pwc = (ucs4_t) georgian_academy_2uni[c-0x80];
22     else if (c >= 0xc0 && c < 0xe7)
23         *pwc = (ucs4_t) c + 0x1010;
24     else
25         *pwc = (ucs4_t) c;
26     return 1;
27 }
28
29 static const unsigned char georgian_academy_page00[32] = {
30     0x80, 0x81, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x80-0x87 */
31     0x00, 0x00, 0x00, 0x00, 0x00, 0x8d, 0x8e, 0x8f, /* 0x88-0x8f */
32     0x90, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x90-0x97 */
33     0x00, 0x00, 0x00, 0x00, 0x00, 0x9d, 0x9e, 0x00, /* 0x98-0x9f */
34 };
35 static const unsigned char georgian_academy_page01[72] = {
36     0x00, 0x00, 0x8c, 0x9c, 0x00, 0x00, 0x00, 0x00, /* 0x50-0x57 */
37     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x58-0x5f */
38     0x8a, 0x9a, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x60-0x67 */
39     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x68-0x6f */
40     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x70-0x77 */
41     0x9f, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x78-0x7f */
42     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x80-0x87 */
43     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x88-0x8f */
44     0x00, 0x00, 0x83, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x90-0x97 */
45 };
46 static const unsigned char georgian_academy_page02[32] = {
47     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x88, 0x00, /* 0xc0-0xc7 */
48     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xc8-0xcf */
49     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xd0-0xd7 */
50     0x00, 0x00, 0x00, 0x00, 0x98, 0x00, 0x00, 0x00, /* 0xd8-0xdf */
51 };
52 static const unsigned char georgian_academy_page20[48] = {
53     0x00, 0x00, 0x00, 0x96, 0x97, 0x00, 0x00, 0x00, /* 0x10-0x17 */
54     0x91, 0x92, 0x82, 0x00, 0x93, 0x94, 0x84, 0x00, /* 0x18-0x1f */
55     0x86, 0x87, 0x95, 0x00, 0x00, 0x00, 0x85, 0x00, /* 0x20-0x27 */
56     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x28-0x2f */

```

```

57 0x89, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x30-0x37 */
58 0x00, 0x8b, 0x9b, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x38-0x3f */
59 };
60
61 static int
62 georgian_academy_wctomb (conv_t conv, unsigned char *r, ucs4_t wc, int n)
63 {
64     unsigned char c = 0;
65     if (wc < 0x0080) {
66         *r = wc;
67         return 1;
68     }
69     else if (wc >= 0x0080 && wc < 0x00a0)
70         c = georgian_academy_page00[wc-0x0080];
71     else if ((wc >= 0x00a0 && wc < 0x00c0) || (wc >= 0x00e7 && wc < 0x0100))
72         c = wc;
73     else if (wc >= 0x0150 && wc < 0x0198)
74         c = georgian_academy_page01[wc-0x0150];
75     else if (wc >= 0x02c0 && wc < 0x02e0)
76         c = georgian_academy_page02[wc-0x02c0];
77     else if (wc >= 0x10d0 && wc < 0x10f7)
78         c = wc-0x1010;
79     else if (wc >= 0x2010 && wc < 0x2040)
80         c = georgian_academy_page20[wc-0x2010];
81     else if (wc == 0x2122)
82         c = 0x99;
83     if (c != 0) {
84         *r = c;
85         return 1;
86     }
87     return RET_ILSEQ;
88 }

```

32.210 georgian_ps.h

```

1 /* $XFree86: xc/lib/X11/lcUniConv/georgian_ps.h,v 1.3 2000/11/29 17:40:29 dawes Exp $ */
2
3 /*
4 * GEORGIAN-PS
5 */
6
7 static const unsigned short georgian_ps_2uni_1[32] = {
8     /* 0x80 */
9     0x0080, 0x0081, 0x201a, 0x0192, 0x201e, 0x2026, 0x2020, 0x2021,
10    0x02c6, 0x2030, 0x0160, 0x2039, 0x0152, 0x008d, 0x008e, 0x008f,
11    /* 0x90 */
12    0x0090, 0x2018, 0x2019, 0x201c, 0x201d, 0x2022, 0x2013, 0x2014,
13    0x02dc, 0x2122, 0x0161, 0x203a, 0x0153, 0x009d, 0x009e, 0x0178,
14 };
15 static const unsigned short georgian_ps_2uni_2[39] = {
16     /* 0xc0 */
17     0x10d0, 0x10d1, 0x10d2, 0x10d3, 0x10d4, 0x10d5, 0x10d6, 0x10f1,
18     0x10d7, 0x10d8, 0x10d9, 0x10da, 0x10db, 0x10dc, 0x10f2, 0x10dd,
19     /* 0xd0 */
20     0x10de, 0x10df, 0x10e0, 0x10e1, 0x10e2, 0x10f3, 0x10e3, 0x10e4,
21     0x10e5, 0x10e6, 0x10e7, 0x10e8, 0x10e9, 0x10ea, 0x10eb, 0x10ec,
22     /* 0xe0 */
23     0x10ed, 0x10ee, 0x10f4, 0x10ef, 0x10f0, 0x10f5,
24 };
25
26 static int
27 georgian_ps_mbtowc (conv_t conv, ucs4_t *pwc, const unsigned char *s, int n)
28 {
29     unsigned char c = *s;
30     if (c >= 0x80 && c < 0xa0)
31         *pwc = (ucs4_t) georgian_ps_2uni_1[c-0x80];
32     else if (c >= 0xc0 && c < 0xe6)
33         *pwc = (ucs4_t) georgian_ps_2uni_2[c-0xc0];
34     else
35         *pwc = (ucs4_t) c;
36     return 1;
37 }
38
39 static const unsigned char georgian_ps_page00[32] = {
40     0x80, 0x81, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x80-0x87 */
41     0x00, 0x00, 0x00, 0x00, 0x00, 0x8d, 0x8e, 0x8f, /* 0x88-0x8f */
42     0x90, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x90-0x97 */
43     0x00, 0x00, 0x00, 0x00, 0x00, 0x9d, 0x9e, 0x00, /* 0x98-0x9f */
44 };
45 static const unsigned char georgian_ps_page01[72] = {
46     0x00, 0x00, 0x8c, 0x9c, 0x00, 0x00, 0x00, 0x00, /* 0x50-0x57 */
47     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x58-0x5f */
48     0x8a, 0x9a, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x60-0x67 */
49     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x68-0x6f */
50     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x70-0x77 */

```



```

51 0x9f, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x78-0x7f */
52 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x80-0x87 */
53 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x88-0x8f */
54 0x00, 0x00, 0x83, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x90-0x97 */
55 };
56 static const unsigned char georgian_ps_page02[32] = {
57 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x88, 0x00, /* 0xc0-0xc7 */
58 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xc8-0xcf */
59 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xd0-0xd7 */
60 0x00, 0x00, 0x00, 0x00, 0x98, 0x00, 0x00, 0x00, /* 0xd8-0xdf */
61 };
62 static const unsigned char georgian_ps_page10[40] = {
63 0xc0, 0xc1, 0xc2, 0xc3, 0xc4, 0xc5, 0xc6, 0xc8, /* 0xd0-0xd7 */
64 0xc9, 0xca, 0xcb, 0xcc, 0xcd, 0xcf, 0xd0, 0xd1, /* 0xd8-0xdf */
65 0xd2, 0xd3, 0xd4, 0xd6, 0xd7, 0xd8, 0xd9, 0xda, /* 0xe0-0xe7 */
66 0xdb, 0xdc, 0xdd, 0xde, 0xdf, 0xe0, 0xe1, 0xe3, /* 0xe8-0xef */
67 0xe4, 0xc7, 0xce, 0xd5, 0xe2, 0xe5, 0x00, 0x00, /* 0xf0-0xf7 */
68 };
69 static const unsigned char georgian_ps_page20[48] = {
70 0x00, 0x00, 0x00, 0x96, 0x97, 0x00, 0x00, 0x00, /* 0x10-0x17 */
71 0x91, 0x92, 0x82, 0x00, 0x93, 0x94, 0x84, 0x00, /* 0x18-0x1f */
72 0x86, 0x87, 0x95, 0x00, 0x00, 0x00, 0x85, 0x00, /* 0x20-0x27 */
73 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x28-0x2f */
74 0x89, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x30-0x37 */
75 0x00, 0x8b, 0x9b, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x38-0x3f */
76 };
77
78 static int
79 georgian_ps_wctomb (conv_t conv, unsigned char *r, ucs4_t wc, int n)
80 {
81     unsigned char c = 0;
82     if (wc < 0x0080) {
83         *r = wc;
84         return 1;
85     }
86     else if (wc >= 0x0080 && wc < 0x00a0)
87         c = georgian_ps_page00[wc-0x0080];
88     else if ((wc >= 0x00a0 && wc < 0x00c0) || (wc >= 0x00e6 && wc < 0x0100))
89         c = wc;
90     else if (wc >= 0x0150 && wc < 0x0198)
91         c = georgian_ps_page01[wc-0x0150];
92     else if (wc >= 0x02c0 && wc < 0x02e0)
93         c = georgian_ps_page02[wc-0x02c0];
94     else if (wc >= 0x10d0 && wc < 0x10f8)
95         c = georgian_ps_page10[wc-0x10d0];
96     else if (wc >= 0x2010 && wc < 0x2040)
97         c = georgian_ps_page20[wc-0x2010];
98     else if (wc == 0x2122)
99         c = 0x99;
100    if (c != 0) {
101        *r = c;
102        return 1;
103    }
104    return RET_ILSEQ;
105 }

```

32.211 iso8859_1.h

```

1 /* $XFree86: xc/lib/X11/lcUniConv/iso8859_1.h,v 1.3 2000/11/29 17:40:30 dawes Exp $ */
2
3 /*
4 * ISO-8859-1
5 */
6
7 static int
8 iso8859_1_mbtowc (conv_t conv, ucs4_t *pwc, const unsigned char *s, int n)
9 {
10     unsigned char c = *s;
11     *pwc = (ucs4_t) c;
12     return 1;
13 }
14
15 static int
16 iso8859_1_wctomb (conv_t conv, unsigned char *r, ucs4_t wc, int n)
17 {
18     if (wc < 0x0100) {
19         *r = wc;
20         return 1;
21     }
22     return RET_ILSEQ;
23 }

```

32.212 iso8859_10.h

```

1 /* $XFree86: xc/lib/X11/lcUniConv/iso8859_10.h,v 1.3 2000/11/29 17:40:30 dawes Exp $ */
2
3 /*
4 * ISO-8859-10
5 */
6
7 #ifdef NEED_TOWC
8 static const unsigned short iso8859_10_2uni[96] = {
9 /* 0xa0 */
10 0x00a0, 0x0104, 0x0112, 0x0122, 0x012a, 0x0128, 0x0136, 0x00a7,
11 0x013b, 0x0110, 0x0160, 0x0166, 0x017d, 0x00ad, 0x016a, 0x014a,
12 /* 0xb0 */
13 0x00b0, 0x0105, 0x0113, 0x0123, 0x012b, 0x0129, 0x0137, 0x00b7,
14 0x013c, 0x0111, 0x0161, 0x0167, 0x017e, 0x2015, 0x016b, 0x014b,
15 /* 0xc0 */
16 0x0100, 0x00c1, 0x00c2, 0x00c3, 0x00c4, 0x00c5, 0x00c6, 0x012e,
17 0x010c, 0x00c9, 0x0118, 0x00cb, 0x0116, 0x00cd, 0x00ce, 0x00cf,
18 /* 0xd0 */
19 0x00d0, 0x0145, 0x014c, 0x00d3, 0x00d4, 0x00d5, 0x00d6, 0x0168,
20 0x00d8, 0x00172, 0x00da, 0x00db, 0x00dc, 0x00dd, 0x00de, 0x00df,
21 /* 0xe0 */
22 0x0101, 0x00e1, 0x00e2, 0x00e3, 0x00e4, 0x00e5, 0x00e6, 0x012f,
23 0x010d, 0x00e9, 0x0119, 0x00eb, 0x0117, 0x00ed, 0x00ee, 0x00ef,
24 /* 0xf0 */
25 0x00f0, 0x0146, 0x014d, 0x00f3, 0x00f4, 0x00f5, 0x00f6, 0x0169,
26 0x00f8, 0x0173, 0x00fa, 0x00fb, 0x00fc, 0x00fd, 0x00fe, 0x0138,
27 };
28
29 static int
30 iso8859_10_mbtowc (conv_t conv, ucs4_t *pwc, const unsigned char *s, int n)
31 {
32     unsigned char c = *s;
33     if (c < 0xa0)
34         *pwc = (ucs4_t) c;
35     else
36         *pwc = (ucs4_t) iso8859_10_2uni[c-0xa0];
37     return 1;
38 }
39 #endif /* NEED_TOWC */
40
41 #ifdef NEED_TOMB
42 static const unsigned char iso8859_10_page00[224] = {
43 0xa0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xa7, /* 0xa0-0xa7 */
44 0x00, 0x00, 0x00, 0x00, 0x00, 0xad, 0x00, 0x00, /* 0xa8-0xaf */
45 0xb0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xb7, /* 0xb0-0xb7 */
46 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xb8-0xbf */
47 0x00, 0xc1, 0xc2, 0xc3, 0xc4, 0xc5, 0xc6, 0x00, /* 0xc0-0xc7 */
48 0x00, 0xc9, 0x00, 0xcb, 0x00, 0xcd, 0xce, 0xcf, /* 0xc8-0xcf */
49 0xd0, 0x00, 0x00, 0xd3, 0xd4, 0xd5, 0xd6, 0x00, /* 0xd0-0xd7 */
50 0xd8, 0x00, 0xda, 0xdb, 0xdc, 0xdd, 0xde, 0xdf, /* 0xd8-0xdf */
51 0x00, 0xe1, 0xe2, 0xe3, 0xe4, 0xe5, 0xe6, 0x00, /* 0xe0-0xe7 */
52 0x00, 0xe9, 0x00, 0xeb, 0x00, 0xed, 0xee, 0xef, /* 0xe8-0xef */
53 0xf0, 0x00, 0x00, 0xf3, 0xf4, 0xf5, 0xf6, 0x00, /* 0xf0-0xf7 */
54 0xf8, 0x00, 0xfa, 0xfb, 0xfc, 0xfd, 0xfe, 0x00, /* 0xf8-0xff */
55 /* 0x0100 */
56 0xc0, 0xe0, 0x00, 0x00, 0xa1, 0xb1, 0x00, 0x00, /* 0x00-0x07 */
57 0x00, 0x00, 0x00, 0x00, 0xc8, 0xe8, 0x00, 0x00, /* 0x08-0x0f */
58 0xa9, 0xb9, 0xa2, 0xb2, 0x00, 0x00, 0xcc, 0xec, /* 0x10-0x17 */
59 0xca, 0xea, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x18-0x1f */
60 0x00, 0x00, 0xa3, 0xb3, 0x00, 0x00, 0x00, 0x00, /* 0x20-0x27 */
61 0xa5, 0xb5, 0xa4, 0xb4, 0x00, 0x00, 0xc7, 0xe7, /* 0x28-0x2f */
62 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xa6, 0xb6, /* 0x30-0x37 */
63 0xff, 0x00, 0x00, 0xa8, 0xb8, 0x00, 0x00, 0x00, /* 0x38-0x3f */
64 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xd1, 0xf1, 0x00, /* 0x40-0x47 */
65 0x00, 0x00, 0xaf, 0xbf, 0xd2, 0xf2, 0x00, 0x00, /* 0x48-0x4f */
66 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x50-0x57 */
67 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x58-0x5f */
68 0xaa, 0xba, 0x00, 0x00, 0x00, 0x00, 0xab, 0xbb, /* 0x60-0x67 */
69 0xd7, 0xf7, 0xae, 0xbe, 0x00, 0x00, 0x00, 0x00, /* 0x68-0x6f */
70 0x00, 0x00, 0xd9, 0xf9, 0x00, 0x00, 0x00, 0x00, /* 0x70-0x77 */
71 0x00, 0x00, 0x00, 0x00, 0x00, 0xac, 0xbc, 0x00, /* 0x78-0x7f */
72 };
73
74 static int
75 iso8859_10_wctomb (conv_t conv, unsigned char *r, ucs4_t wc, int n)
76 {
77     unsigned char c = 0;
78     if (wc < 0x00a0) {
79         *r = wc;
80         return 1;
81     }
82     else if (wc >= 0x00a0 && wc < 0x0180)
83         c = iso8859_10_page00[wc-0x00a0];
84     else if (wc == 0x2015)
85         c = 0xbd;

```

```

86  if (c != 0) {
87      *r = c;
88      return 1;
89  }
90  return RET_ILSEQ;
91 }
92 #endif /* NEED_TOMB */

```

32.213 iso8859_11.h

```

1  /* $XFree86: xc/lib/X11/lcUniConv/iso8859_11.h,v 1.2 2002/10/09 16:38:19 tsi Exp $ */
2
3  /*
4  * ISO8859-11
5  */
6
7  #ifdef NEED_TOWC
8  static const unsigned short iso8859_11_2uni[96] = {
9      /* 0xa0 */
10     0x00a0, 0x0e01, 0x0e02, 0x0e03, 0x0e04, 0x0e05, 0x0e06, 0x0e07,
11     0x0e08, 0x0e09, 0x0e0a, 0x0e0b, 0x0e0c, 0x0e0d, 0x0e0e, 0x0e0f,
12     /* 0xb0 */
13     0x0e10, 0x0e11, 0x0e12, 0x0e13, 0x0e14, 0x0e15, 0x0e16, 0x0e17,
14     0x0e18, 0x0e19, 0x0e1a, 0x0e1b, 0x0e1c, 0x0e1d, 0x0e1e, 0x0e1f,
15     /* 0xc0 */
16     0x0e20, 0x0e21, 0x0e22, 0x0e23, 0x0e24, 0x0e25, 0x0e26, 0x0e27,
17     0x0e28, 0x0e29, 0x0e2a, 0x0e2b, 0x0e2c, 0x0e2d, 0x0e2e, 0x0e2f,
18     /* 0xd0 */
19     0x0e30, 0x0e31, 0x0e32, 0x0e33, 0x0e34, 0x0e35, 0x0e36, 0x0e37,
20     0x0e38, 0x0e39, 0x0e3a, 0xffff, 0xffff, 0xffff, 0xffff, 0x0e3f,
21     /* 0xe0 */
22     0x0e40, 0x0e41, 0x0e42, 0x0e43, 0x0e44, 0x0e45, 0x0e46, 0x0e47,
23     0x0e48, 0x0e49, 0x0e4a, 0x0e4b, 0x0e4c, 0x0e4d, 0x0e4e, 0x0e4f,
24     /* 0xf0 */
25     0x0e50, 0x0e51, 0x0e52, 0x0e53, 0x0e54, 0x0e55, 0x0e56, 0x0e57,
26     0x0e58, 0x0e59, 0x0e5a, 0x0e5b, 0xffff, 0xffff, 0xffff, 0xffff,
27 };
28
29 static int
30 iso8859_11_mbtowc (conv_t conv, ucs4_t *pwc, const unsigned char *s, int n)
31 {
32     unsigned char c = *s;
33     if (c < 0x80) {
34         *pwc = (ucs4_t) c;
35         return 1;
36     }
37     else if (c < 0xa0) {
38     }
39     else {
40         unsigned short wc = iso8859_11_2uni[c-0xa0];
41         if (wc != 0xffff) {
42             *pwc = (ucs4_t) wc;
43             return 1;
44         }
45     }
46     return RET_ILSEQ;
47 }
48 #endif /* NEED_TOWC */
49
50 #ifdef NEED_TOMB
51 static const unsigned char iso8859_11_page0e[96] = {
52     0x00, 0xa1, 0xa2, 0xa3, 0xa4, 0xa5, 0xa6, 0xa7, /* 0x00-0x07 */
53     0xa8, 0xa9, 0xaa, 0xab, 0xac, 0xad, 0xae, 0xaf, /* 0x08-0x0f */
54     0xb0, 0xb1, 0xb2, 0xb3, 0xb4, 0xb5, 0xb6, 0xb7, /* 0x10-0x17 */
55     0xb8, 0xb9, 0xba, 0xbb, 0xbc, 0xbd, 0xbe, 0xbf, /* 0x18-0x1f */
56     0xc0, 0xc1, 0xc2, 0xc3, 0xc4, 0xc5, 0xc6, 0xc7, /* 0x20-0x27 */
57     0xc8, 0xc9, 0xca, 0xcb, 0xcc, 0xcd, 0xce, 0xcf, /* 0x28-0x2f */
58     0xd0, 0xd1, 0xd2, 0xd3, 0xd4, 0xd5, 0xd6, 0xd7, /* 0x30-0x37 */
59     0xd8, 0xd9, 0xda, 0x00, 0x00, 0x00, 0x00, 0xdf, /* 0x38-0x3f */
60     0xe0, 0xe1, 0xe2, 0xe3, 0xe4, 0xe5, 0xe6, 0xe7, /* 0x40-0x47 */
61     0xe8, 0xe9, 0xea, 0xeb, 0xec, 0xed, 0xee, 0xef, /* 0x48-0x4f */
62     0xf0, 0xf1, 0xf2, 0xf3, 0xf4, 0xf5, 0xf6, 0xf7, /* 0x50-0x57 */
63     0xf8, 0xf9, 0xfa, 0xfb, 0x00, 0x00, 0x00, 0x00, /* 0x58-0x5f */
64 };
65
66 static int
67 iso8859_11_wctomb (conv_t conv, unsigned char *r, ucs4_t wc, int n)
68 {
69     unsigned char c = 0;
70     if (wc < 0x0080 || wc == 0x00a0) {
71         *r = wc;
72         return 1;
73     }
74     else if (wc >= 0x0e00 && wc < 0x0e60)
75         c = iso8859_11_page0e[wc-0x0e00];

```

```

76  if (c != 0) {
77      *r = c;
78      return 1;
79  }
80  return RET_ILSEQ;
81 }
82 #endif /* NEED_TOMB */

```

32.214 iso8859_13.h

```

1 /* $XFree86: xc/lib/X11/lcUniConv/iso8859_13.h,v 1.2 2000/11/28 16:10:26 dawes Exp $ */
2
3 /*
4 * ISO-8859-13
5 */
6
7 #ifdef NEED_TOWC
8 static const unsigned short iso8859_13_2uni[96] = {
9     /* 0xa0 */
10    0x00a0, 0x201d, 0x00a2, 0x00a3, 0x00a4, 0x201e, 0x00a6, 0x00a7,
11    0x00d8, 0x00a9, 0x0156, 0x00ab, 0x00ac, 0x00ad, 0x00ae, 0x00c6,
12    /* 0xb0 */
13    0x00b0, 0x00b1, 0x00b2, 0x00b3, 0x201c, 0x00b5, 0x00b6, 0x00b7,
14    0x00f8, 0x00b9, 0x0157, 0x00bb, 0x00bc, 0x00bd, 0x00be, 0x00e6,
15    /* 0xc0 */
16    0x0104, 0x012e, 0x0100, 0x0106, 0x00c4, 0x00c5, 0x0118, 0x0112,
17    0x010c, 0x00c9, 0x0179, 0x0116, 0x0122, 0x0136, 0x012a, 0x013b,
18    /* 0xd0 */
19    0x0160, 0x0143, 0x0145, 0x00d3, 0x014c, 0x00d5, 0x00d6, 0x00d7,
20    0x0172, 0x0141, 0x015a, 0x016a, 0x00dc, 0x017b, 0x017d, 0x00df,
21    /* 0xe0 */
22    0x0105, 0x012f, 0x0101, 0x0107, 0x00e4, 0x00e5, 0x0119, 0x0113,
23    0x010d, 0x00e9, 0x017a, 0x0117, 0x0123, 0x0137, 0x012b, 0x013c,
24    /* 0xf0 */
25    0x0161, 0x0144, 0x0146, 0x00f3, 0x014d, 0x00f5, 0x00f6, 0x00f7,
26    0x0173, 0x0142, 0x015b, 0x016b, 0x00fc, 0x017c, 0x017e, 0x2019,
27 };
28
29 static int
30 iso8859_13_mbtowc (conv_t conv, ucs4_t *pwc, const unsigned char *s, int n)
31 {
32     unsigned char c = *s;
33     if (c < 0xa0)
34         *pwc = (ucs4_t) c;
35     else
36         *pwc = (ucs4_t) iso8859_13_2uni[c-0xa0];
37     return 1;
38 }
39 #endif /* NEED_TOWC */
40
41 #ifdef NEED_TOMB
42 static const unsigned char iso8859_13_page00[224] = {
43     0xa0, 0x00, 0xa2, 0xa3, 0xa4, 0x00, 0xa6, 0xa7, /* 0xa0-0xa7 */
44     0x00, 0xa9, 0x00, 0xab, 0xac, 0xad, 0xae, 0x00, /* 0xa8-0xaf */
45     0xb0, 0xb1, 0xb2, 0xb3, 0x00, 0xb5, 0xb6, 0xb7, /* 0xb0-0xb7 */
46     0x00, 0xb9, 0x00, 0xbb, 0xbc, 0xbd, 0xbe, 0x00, /* 0xb8-0xbf */
47     0x00, 0x00, 0x00, 0x00, 0xc4, 0xc5, 0xaf, 0x00, /* 0xc0-0xc7 */
48     0x00, 0xc9, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xc8-0xcf */
49     0x00, 0x00, 0x00, 0xd3, 0x00, 0xd5, 0xd6, 0xd7, /* 0xd0-0xd7 */
50     0xa8, 0x00, 0x00, 0x00, 0xdc, 0x00, 0x00, 0xdf, /* 0xd8-0xdf */
51     0x00, 0x00, 0x00, 0x00, 0xe4, 0xe5, 0xbf, 0x00, /* 0xe0-0xe7 */
52     0x00, 0xe9, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xe8-0xef */
53     0x00, 0x00, 0x00, 0xf3, 0x00, 0xf5, 0xf6, 0xf7, /* 0xf0-0xf7 */
54     0xb8, 0x00, 0x00, 0x00, 0xfc, 0x00, 0x00, 0x00, /* 0xf8-0xff */
55     /* 0x0100 */
56     0xc2, 0xe2, 0x00, 0x00, 0xc0, 0xe0, 0xc3, 0xe3, /* 0x00-0x07 */
57     0x00, 0x00, 0x00, 0xc8, 0xe8, 0x00, 0x00, /* 0x08-0x0f */
58     0x00, 0x00, 0xc7, 0xe7, 0x00, 0x00, 0xcb, 0xeb, /* 0x10-0x17 */
59     0xc6, 0xe6, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x18-0x1f */
60     0x00, 0x00, 0xcc, 0xec, 0x00, 0x00, 0x00, 0x00, /* 0x20-0x27 */
61     0x00, 0x00, 0xce, 0xee, 0x00, 0x00, 0xc1, 0xe1, /* 0x28-0x2f */
62     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xcd, 0xed, /* 0x30-0x37 */
63     0x00, 0x00, 0xcf, 0xef, 0x00, 0x00, 0x00, 0x00, /* 0x38-0x3f */
64     0x00, 0xd9, 0xf9, 0xd1, 0xf1, 0xd2, 0xf2, 0x00, /* 0x40-0x47 */
65     0x00, 0x00, 0x00, 0x00, 0xd4, 0xf4, 0x00, 0x00, /* 0x48-0x4f */
66     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xaa, 0xba, /* 0x50-0x57 */
67     0x00, 0x00, 0xda, 0xfa, 0x00, 0x00, 0x00, 0x00, /* 0x58-0x5f */
68     0xd0, 0xf0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x60-0x67 */
69     0x00, 0x00, 0xdb, 0xfb, 0x00, 0x00, 0x00, 0x00, /* 0x68-0x6f */
70     0x00, 0x00, 0xd8, 0xf8, 0x00, 0x00, 0x00, 0x00, /* 0x70-0x77 */
71     0x00, 0xca, 0xea, 0xdd, 0xfd, 0xde, 0xfe, 0x00, /* 0x78-0x7f */
72 };
73 static const unsigned char iso8859_13_page20[8] = {
74     0x00, 0xff, 0x00, 0x00, 0xb4, 0xa1, 0xa5, 0x00, /* 0x18-0x1f */
75 };

```

```

76
77 static int
78 iso8859_13_wctomb (conv_t conv, unsigned char *, ucs4_t wc, int n)
79 {
80     unsigned char c = 0;
81     if (wc < 0x00a0) {
82         *r = wc;
83         return 1;
84     }
85     else if (wc >= 0x00a0 && wc < 0x0180)
86         c = iso8859_13_page00[wc-0x00a0];
87     else if (wc >= 0x2018 && wc < 0x2020)
88         c = iso8859_13_page20[wc-0x2018];
89     if (c != 0) {
90         *r = c;
91         return 1;
92     }
93     return RET_ILSEQ;
94 }
95 #endif /* NEED_TOWC */

```

32.215 iso8859_14.h

```

1 /* $XFree86: xc/lib/X11/lcUniConv/iso8859_14.h,v 1.3 2000/11/29 17:40:30 dawes Exp $ */
2
3 /*
4  * ISO-8859-14
5  */
6
7 #ifdef NEED_TOWC
8 static const unsigned short iso8859_14_2uni[96] = {
9     /* 0xa0 */
10    0x00a0, 0x1e02, 0x1e03, 0x00a3, 0x010a, 0x010b, 0x1e0a, 0x00a7,
11    0x1e80, 0x00a9, 0x1e82, 0x1e0b, 0x1ef2, 0x00ad, 0x00ae, 0x0178,
12    /* 0xb0 */
13    0x1e1e, 0x1ef, 0x0120, 0x0121, 0x1e40, 0x1e41, 0x00b6, 0x1e56,
14    0x1e81, 0x1e57, 0x1e83, 0x1e60, 0x1ef3, 0x1e84, 0x1e85, 0x1e61,
15    /* 0xc0 */
16    0x00c0, 0x00c1, 0x00c2, 0x00c3, 0x00c4, 0x00c5, 0x00c6, 0x00c7,
17    0x00c8, 0x00c9, 0x00ca, 0x00cb, 0x00cc, 0x00cd, 0x00ce, 0x00cf,
18    /* 0xd0 */
19    0x0174, 0x00d1, 0x00d2, 0x00d3, 0x00d4, 0x00d5, 0x00d6, 0x1e6a,
20    0x00d8, 0x00d9, 0x00da, 0x00db, 0x00dc, 0x00dd, 0x0176, 0x00df,
21    /* 0xe0 */
22    0x00e0, 0x00e1, 0x00e2, 0x00e3, 0x00e4, 0x00e5, 0x00e6, 0x00e7,
23    0x00e8, 0x00e9, 0x00ea, 0x00eb, 0x00ec, 0x00ed, 0x00ee, 0x00ef,
24    /* 0xf0 */
25    0x0175, 0x00f1, 0x00f2, 0x00f3, 0x00f4, 0x00f5, 0x00f6, 0x1e6b,
26    0x00f8, 0x00f9, 0x00fa, 0x00fb, 0x00fc, 0x00fd, 0x0177, 0x00ff,
27 };
28
29 static int
30 iso8859_14_mbtowc (conv_t conv, ucs4_t *pwc, const unsigned char *s, int n)
31 {
32     unsigned char c = *s;
33     if (c >= 0xa0)
34         *pwc = (ucs4_t) iso8859_14_2uni[c-0xa0];
35     else
36         *pwc = (ucs4_t) c;
37     return 1;
38 }
39 #endif /* NEED_TOWC */
40
41 #ifdef NEED_TOMB
42 static const unsigned char iso8859_14_page00[96] = {
43    0xa0, 0x00, 0x00, 0xa3, 0x00, 0x00, 0x00, 0xa7, /* 0xa0-0xa7 */
44    0x00, 0xa9, 0x00, 0x00, 0x00, 0xad, 0xae, 0x00, /* 0xa8-0xaf */
45    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xb6, 0x00, /* 0xb0-0xb7 */
46    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xb8-0xbf */
47    0xc0, 0xc1, 0xc2, 0xc3, 0xc4, 0xc5, 0xc6, 0xc7, /* 0xc0-0xc7 */
48    0xc8, 0xc9, 0xca, 0xcb, 0xcc, 0xcd, 0xce, 0xcf, /* 0xc8-0xcf */
49    0x00, 0xd1, 0xd2, 0xd3, 0xd4, 0xd5, 0xd6, 0x00, /* 0xd0-0xd7 */
50    0xd8, 0xd9, 0xda, 0xdb, 0xdc, 0xdd, 0x00, 0xdf, /* 0xd8-0xdf */
51    0xe0, 0xe1, 0xe2, 0xe3, 0xe4, 0xe5, 0xe6, 0xe7, /* 0xe0-0xe7 */
52    0xe8, 0xe9, 0xea, 0xeb, 0xec, 0xed, 0xee, 0xef, /* 0xe8-0xef */
53    0x00, 0xf1, 0xf2, 0xf3, 0xf4, 0xf5, 0xf6, 0x00, /* 0xf0-0xf7 */
54    0xf8, 0xf9, 0xfa, 0xfb, 0xfc, 0xfd, 0x00, 0xff, /* 0xf8-0xff */
55 };
56 static const unsigned char iso8859_14_page01_0[32] = {
57    0x00, 0x00, 0xa4, 0xa5, 0x00, 0x00, 0x00, 0x00, /* 0x08-0x0f */
58    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x10-0x17 */
59    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x18-0x1f */
60    0xb2, 0xb3, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x20-0x27 */
61 };
62 static const unsigned char iso8859_14_page01_1[16] = {

```

```

63 0x00, 0x00, 0x00, 0x00, 0xd0, 0xf0, 0xde, 0xfe, /* 0x70-0x77 */
64 0xaf, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x78-0x7f */
65 };
66 static const unsigned char iso8859_14_pagele_0[136] = {
67 0x00, 0x00, 0xa1, 0xa2, 0x00, 0x00, 0x00, 0x00, /* 0x00-0x07 */
68 0x00, 0x00, 0xa6, 0xab, 0x00, 0x00, 0x00, 0x00, /* 0x08-0x0f */
69 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x10-0x17 */
70 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xb0, 0xb1, /* 0x18-0x1f */
71 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x20-0x27 */
72 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x28-0x2f */
73 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x30-0x37 */
74 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x38-0x3f */
75 0xb4, 0xb5, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x40-0x47 */
76 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x48-0x4f */
77 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xb7, 0xb9, /* 0x50-0x57 */
78 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x58-0x5f */
79 0xbb, 0xbf, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x60-0x67 */
80 0x00, 0x00, 0xd7, 0xf7, 0x00, 0x00, 0x00, 0x00, /* 0x68-0x6f */
81 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x70-0x77 */
82 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x78-0x7f */
83 0xa8, 0xb8, 0xaa, 0xba, 0xbd, 0xbe, 0x00, 0x00, /* 0x80-0x87 */
84 };
85 static const unsigned char iso8859_14_pagele_1[8] = {
86 0x00, 0x00, 0xac, 0xbc, 0x00, 0x00, 0x00, 0x00, /* 0xf0-0xf7 */
87 };
88
89 static int
90 iso8859_14_wctomb (conv_t conv, unsigned char *, ucs4_t wc, int n)
91 {
92     unsigned char c = 0;
93     if (wc < 0x00a0) {
94         *r = wc;
95         return 1;
96     }
97     else if (wc >= 0x00a0 && wc < 0x0100)
98         c = iso8859_14_page00[wc-0x00a0];
99     else if (wc >= 0x0108 && wc < 0x0128)
100        c = iso8859_14_page01_0[wc-0x0108];
101     else if (wc >= 0x0170 && wc < 0x0180)
102        c = iso8859_14_page01_1[wc-0x0170];
103     else if (wc >= 0x1e00 && wc < 0x1e88)
104        c = iso8859_14_pagele_0[wc-0x1e00];
105     else if (wc >= 0x1ef0 && wc < 0x1ef8)
106        c = iso8859_14_pagele_1[wc-0x1ef0];
107     if (c != 0) {
108         *r = c;
109         return 1;
110     }
111     return RET_ILSEQ;
112 }
113 #endif /* NEED_TOMB */

```

32.216 iso8859_15.h

```

1 /* $XFree86: xc/lib/X11/lcUniConv/iso8859_15.h,v 1.3 2000/11/29 17:40:31 dawes Exp $ */
2
3 /*
4 * ISO-8859-15
5 */
6
7 #ifndef NEED_TOWC
8 static const unsigned short iso8859_15_2uni[32] = {
9     /* 0xa0 */
10    0x00a0, 0x00a1, 0x00a2, 0x00a3, 0x20ac, 0x00a5, 0x0160, 0x00a7,
11    0x0161, 0x00a9, 0x00aa, 0x00ab, 0x00ac, 0x00ad, 0x00ae, 0x00af,
12    /* 0xb0 */
13    0x00b0, 0x00b1, 0x00b2, 0x00b3, 0x017d, 0x00b5, 0x00b6, 0x00b7,
14    0x017e, 0x00b9, 0x00ba, 0x00bb, 0x0152, 0x0153, 0x0178, 0x00bf,
15 };
16
17 static int
18 iso8859_15_mbtowc (conv_t conv, ucs4_t *pwc, const unsigned char *s, int n)
19 {
20     unsigned char c = *s;
21     if (c >= 0xa0 && c < 0xc0)
22         *pwc = (ucs4_t) iso8859_15_2uni[c-0xa0];
23     else
24         *pwc = (ucs4_t) c;
25     return 1;
26 }
27 #endif /* NEED_TOWC */
28
29 #ifndef NEED_TOMB
30 static const unsigned char iso8859_15_page00[32] = {
31     0xa0, 0xa1, 0xa2, 0xa3, 0x00, 0xa5, 0x00, 0xa7, /* 0xa0-0xa7 */

```

```

32 0x00, 0xa9, 0xaa, 0xab, 0xac, 0xad, 0xae, 0xaf, /* 0xa8-0xaf */
33 0xb0, 0xb1, 0xb2, 0xb3, 0x00, 0xb5, 0xb6, 0xb7, /* 0xb0-0xb7 */
34 0x00, 0xb9, 0xba, 0xbb, 0x00, 0x00, 0x00, 0xbf, /* 0xb8-0xbf */
35 };
36 static const unsigned char iso8859_15_page01[48] = {
37 0x00, 0x00, 0xbc, 0xbd, 0x00, 0x00, 0x00, 0x00, /* 0x50-0x57 */
38 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x58-0x5f */
39 0xa6, 0xa8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x60-0x67 */
40 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x68-0x6f */
41 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x70-0x77 */
42 0xbe, 0x00, 0x00, 0x00, 0x00, 0xb4, 0xb8, 0x00, /* 0x78-0x7f */
43 };
44
45 static int
46 iso8859_15_wctomb (conv_t conv, unsigned char *r, ucs4_t wc, int n)
47 {
48     unsigned char c = 0;
49     if (wc < 0x00a0) {
50         *r = wc;
51         return 1;
52     }
53     else if (wc >= 0x00a0 && wc < 0x00c0)
54         c = iso8859_15_page00[wc-0x00a0];
55     else if (wc >= 0x00c0 && wc < 0x0100)
56         c = wc;
57     else if (wc >= 0x0150 && wc < 0x0180)
58         c = iso8859_15_page01[wc-0x0150];
59     else if (wc == 0x20ac)
60         c = 0xa4;
61     if (c != 0) {
62         *r = c;
63         return 1;
64     }
65     return RET_ILSEQ;
66 }
67 #endif /* NEED_TOMB */

```

32.217 iso8859_16.h

```

1 /* $XFree86: xc/lib/X11/lcUniConv/iso8859_16.h,v 1.4 2003/07/15 17:33:45 pascal Exp $ */
2
3 /*
4  * ISO-8859-16
5  */
6
7 static const unsigned short iso8859_16_2uni[96] = {
8     /* 0xa0 */
9     0x00a0, 0x0104, 0x0105, 0x0141, 0x20ac, 0x201e, 0x0160, 0x00a7,
10    0x0161, 0x00a9, 0x0218, 0x00ab, 0x0179, 0x00ad, 0x017a, 0x017b,
11    /* 0xb0 */
12    0x00b0, 0x00b1, 0x010c, 0x0142, 0x017d, 0x201d, 0x00b6, 0x00b7,
13    0x017e, 0x010d, 0x0219, 0x00bb, 0x0152, 0x0153, 0x0178, 0x017c,
14    /* 0xc0 */
15    0x00c0, 0x00c1, 0x00c2, 0x0102, 0x00c4, 0x0106, 0x00c6, 0x00c7,
16    0x00c8, 0x00c9, 0x00ca, 0x00cb, 0x00cc, 0x00cd, 0x00ce, 0x00cf,
17    /* 0xd0 */
18    0x0110, 0x0143, 0x00d2, 0x00d3, 0x00d4, 0x0150, 0x00d6, 0x015a,
19    0x0170, 0x00d9, 0x00da, 0x00db, 0x00dc, 0x0118, 0x021a, 0x00df,
20    /* 0xe0 */
21    0x00e0, 0x00e1, 0x00e2, 0x0103, 0x00e4, 0x0107, 0x00e6, 0x00e7,
22    0x00e8, 0x00e9, 0x00ea, 0x00eb, 0x00ec, 0x00ed, 0x00ee, 0x00ef,
23    /* 0xf0 */
24    0x0111, 0x0144, 0x00f2, 0x00f3, 0x00f4, 0x0151, 0x00f6, 0x015b,
25    0x0171, 0x00f9, 0x00fa, 0x00fb, 0x00fc, 0x0119, 0x021b, 0x00ff,
26 };
27
28 static int
29 iso8859_16_mbtowc (conv_t conv, ucs4_t *pwc, const unsigned char *s, int n)
30 {
31     unsigned char c = *s;
32     if (c < 0xa0)
33         *pwc = (ucs4_t) c;
34     else
35         *pwc = (ucs4_t) iso8859_16_2uni[c-0xa0];
36     return 1;
37 }
38
39 static const unsigned char iso8859_16_page00[224] = {
40 0xa0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xa7, /* 0xa0-0xa7 */
41 0x00, 0xa9, 0x00, 0xab, 0x00, 0xad, 0x00, 0x00, /* 0xa8-0xaf */
42 0xb0, 0xb1, 0x00, 0x00, 0x00, 0x00, 0xb6, 0xb7, /* 0xb0-0xb7 */
43 0x00, 0x00, 0x00, 0xbb, 0x00, 0x00, 0x00, 0x00, /* 0xb8-0xbf */
44 0xc0, 0xc1, 0xc2, 0x00, 0xc4, 0x00, 0xc6, 0xc7, /* 0xc0-0xc7 */
45 0xc8, 0xc9, 0xca, 0xcb, 0xcc, 0xcd, 0xce, 0xcf, /* 0xc8-0xcf */
46 0x00, 0x00, 0xd2, 0xd3, 0xd4, 0x00, 0xd6, 0x00, /* 0xd0-0xd7 */

```

```

47 0x00, 0xd9, 0xda, 0xdb, 0xdc, 0x00, 0x00, 0xdf, /* 0xd8-0xdf */
48 0xe0, 0xe1, 0xe2, 0x00, 0xe4, 0x00, 0xe6, 0xe7, /* 0xe0-0xe7 */
49 0xe8, 0xe9, 0xea, 0xeb, 0xec, 0xed, 0xee, 0xef, /* 0xe8-0xef */
50 0x00, 0x00, 0xf2, 0xf3, 0xf4, 0x00, 0xf6, 0x00, /* 0xf0-0xf7 */
51 0x00, 0xf9, 0xfa, 0xfb, 0xfc, 0x00, 0x00, 0xff, /* 0xf8-0xff */
52 /* 0x0100 */
53 0x00, 0x00, 0xc3, 0xe3, 0xa1, 0xa2, 0xc5, 0xe5, /* 0x00-0x07 */
54 0x00, 0x00, 0x00, 0x00, 0xb2, 0xb9, 0x00, 0x00, /* 0x08-0x0f */
55 0xd0, 0xf0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x10-0x17 */
56 0xdd, 0xfd, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x18-0x1f */
57 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x20-0x27 */
58 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x28-0x2f */
59 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x30-0x37 */
60 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x38-0x3f */
61 0x00, 0xa3, 0xb3, 0xd1, 0xf1, 0x00, 0x00, 0x00, /* 0x40-0x47 */
62 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x48-0x4f */
63 0xd5, 0xf5, 0xbc, 0xbd, 0x00, 0x00, 0x00, 0x00, /* 0x50-0x57 */
64 0x00, 0x00, 0x00, 0xd7, 0xf7, 0x00, 0x00, 0x00, /* 0x58-0x5f */
65 0xa6, 0xa8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x60-0x67 */
66 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x68-0x6f */
67 0xd8, 0xf8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x70-0x77 */
68 0xbe, 0xac, 0xae, 0xaf, 0xbf, 0xb4, 0xb8, 0x00, /* 0x78-0x7f */
69 };
70 static const unsigned char iso8859_16_page02[8] = {
71 0xaa, 0xba, 0xde, 0xfe, 0x00, 0x00, 0x00, 0x00, /* 0x18-0x1f */
72 };
73 static const unsigned char iso8859_16_page20[8] = {
74 0x00, 0x00, 0x00, 0x00, 0x00, 0xb5, 0xa5, 0x00, /* 0x18-0x1f */
75 };
76
77 static int
78 iso8859_16_wctomb (conv_t conv, unsigned char *r, ucs4_t wc, int n)
79 {
80 unsigned char c = 0;
81 if (wc < 0x00a0) {
82 *r = wc;
83 return 1;
84 }
85 else if (wc >= 0x00a0 && wc < 0x0180)
86 c = iso8859_16_page00[wc-0x00a0];
87 else if (wc >= 0x0218 && wc < 0x0220)
88 c = iso8859_16_page02[wc-0x0218];
89 else if (wc >= 0x2018 && wc < 0x2020)
90 c = iso8859_16_page20[wc-0x2018];
91 else if (wc == 0x20ac)
92 c = 0xa4;
93 if (c != 0) {
94 *r = c;
95 return 1;
96 }
97 return RET_ILSEQ;
98 }

```

32.218 iso8859_2.h

```

1 /* $XFree86: xc/lib/X11/lcUniConv/iso8859_2.h,v 1.3 2000/11/29 17:40:31 dawes Exp $ */
2
3 /*
4 * ISO-8859-2
5 */
6
7 #ifdef NEED_TOWC
8 static const unsigned short iso8859_2_uni[96] = {
9 /* 0xa0 */
10 0x00a0, 0x0104, 0x02d8, 0x0141, 0x00a4, 0x013d, 0x015a, 0x00a7,
11 0x00a8, 0x0160, 0x015e, 0x0164, 0x0179, 0x00ad, 0x017d, 0x017b,
12 /* 0xb0 */
13 0x00b0, 0x0105, 0x02db, 0x0142, 0x00b4, 0x013e, 0x015b, 0x02c7,
14 0x00b8, 0x0161, 0x015f, 0x0165, 0x017a, 0x02dd, 0x017e, 0x017c,
15 /* 0xc0 */
16 0x0154, 0x00c1, 0x00c2, 0x0102, 0x00c4, 0x0139, 0x0106, 0x00c7,
17 0x010c, 0x00c9, 0x0118, 0x00cb, 0x011a, 0x00cd, 0x00ce, 0x010e,
18 /* 0xd0 */
19 0x0110, 0x0143, 0x0147, 0x00d3, 0x00d4, 0x0150, 0x00d6, 0x00d7,
20 0x0158, 0x016e, 0x00da, 0x0170, 0x00dc, 0x00dd, 0x0162, 0x00df,
21 /* 0xe0 */
22 0x0155, 0x00e1, 0x00e2, 0x0103, 0x00e4, 0x013a, 0x0107, 0x00e7,
23 0x010d, 0x00e9, 0x0119, 0x00eb, 0x011b, 0x00ed, 0x00ee, 0x010f,
24 /* 0xf0 */
25 0x0111, 0x0144, 0x0148, 0x00f3, 0x00f4, 0x0151, 0x00f6, 0x00f7,
26 0x0159, 0x016f, 0x00fa, 0x0171, 0x00fc, 0x00fd, 0x0163, 0x02d9,
27 };
28
29 static int
30 iso8859_2_mbtowc (conv_t conv, ucs4_t *pwc, const unsigned char *s, int n)

```



```

31 {
32   unsigned char c = *s;
33   if (c < 0xa0)
34     *pwc = (ucs4_t) c;
35   else
36     *pwc = (ucs4_t) iso8859_2_uni[c-0xa0];
37   return 1;
38 }
39 #endif /* NEED_TOWC */
40
41 #ifndef NEED_TOMB
42 static const unsigned char iso8859_2_page00[224] = {
43   0xa0, 0x00, 0x00, 0x00, 0xa4, 0x00, 0x00, 0xa7, /* 0xa0-0xa7 */
44   0xa8, 0x00, 0x00, 0x00, 0x00, 0xad, 0x00, 0x00, /* 0xa8-0xaf */
45   0xb0, 0x00, 0x00, 0x00, 0xb4, 0x00, 0x00, 0x00, /* 0xb0-0xb7 */
46   0xb8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xb8-0xbf */
47   0x00, 0xc1, 0xc2, 0x00, 0xc4, 0x00, 0x00, 0xc7, /* 0xc0-0xc7 */
48   0x00, 0xc9, 0x00, 0xcb, 0x00, 0xcd, 0xce, 0x00, /* 0xc8-0xcf */
49   0x00, 0x00, 0x00, 0xd3, 0xd4, 0x00, 0xd6, 0xd7, /* 0xd0-0xd7 */
50   0x00, 0x00, 0xda, 0x00, 0xdc, 0xdd, 0x00, 0xdf, /* 0xd8-0xdf */
51   0x00, 0xe1, 0xe2, 0x00, 0xe4, 0x00, 0x00, 0xe7, /* 0xe0-0xe7 */
52   0x00, 0xe9, 0x00, 0xeb, 0x00, 0xed, 0xee, 0x00, /* 0xe8-0xef */
53   0x00, 0x00, 0x00, 0xf3, 0xf4, 0x00, 0xf6, 0xf7, /* 0xf0-0xf7 */
54   0x00, 0x00, 0xfa, 0x00, 0xfc, 0xfd, 0x00, 0x00, /* 0xf8-0xff */
55   /* 0x0100 */
56   0x00, 0x00, 0xc3, 0xe3, 0xa1, 0xb1, 0xc6, 0xe6, /* 0x00-0x07 */
57   0x00, 0x00, 0x00, 0x00, 0xc8, 0xe8, 0xcf, 0xef, /* 0x08-0x0f */
58   0xd0, 0xf0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x10-0x17 */
59   0xca, 0xea, 0xcc, 0xec, 0x00, 0x00, 0x00, 0x00, /* 0x18-0x1f */
60   0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x20-0x27 */
61   0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x28-0x2f */
62   0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x30-0x37 */
63   0x00, 0xc5, 0xe5, 0x00, 0x00, 0xa5, 0xb5, 0x00, /* 0x38-0x3f */
64   0x00, 0xa3, 0xb3, 0xd1, 0xf1, 0x00, 0x00, 0xd2, /* 0x40-0x47 */
65   0xf2, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x48-0x4f */
66   0xd5, 0xf5, 0x00, 0x00, 0xc0, 0xe0, 0x00, 0x00, /* 0x50-0x57 */
67   0xd8, 0xf8, 0xa6, 0xb6, 0x00, 0x00, 0xaa, 0xba, /* 0x58-0x5f */
68   0xa9, 0xb9, 0xde, 0xfe, 0xab, 0xbb, 0x00, 0x00, /* 0x60-0x67 */
69   0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xd9, 0xf9, /* 0x68-0x6f */
70   0xdb, 0xfb, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x70-0x77 */
71   0x00, 0xac, 0xbc, 0xaf, 0xbf, 0xae, 0xbe, 0x00, /* 0x78-0x7f */
72 };
73 static const unsigned char iso8859_2_page02[32] = {
74   0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xb7, /* 0xc0-0xc7 */
75   0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xc8-0xcf */
76   0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xd0-0xd7 */
77   0xa2, 0xff, 0x00, 0xb2, 0x00, 0xbd, 0x00, 0x00, /* 0xd8-0xdf */
78 };
79
80 /*
81 static int
82 iso8859_2_wctomb (conv_t conv, unsigned char *r, ucs4_t wc, int n)
83 {
84   unsigned char c = 0;
85   if (wc < 0x00a0) {
86     *r = wc;
87     return 1;
88   }
89   else if (wc >= 0x00a0 && wc < 0x0180)
90     c = iso8859_2_page00[wc-0x00a0];
91   else if (wc >= 0x02c0 && wc < 0x02e0)
92     c = iso8859_2_page02[wc-0x02c0];
93   if (c != 0) {
94     *r = c;
95     return 1;
96   }
97   return RET_ILSEQ;
98 }
99 */
100 #endif /* NEED_TOMB */

```

32.219 iso8859_3.h

```

1 /* $XFree86: xc/lib/X11/lcUniConv/iso8859_3.h,v 1.3 2000/11/29 17:40:31 dawes Exp $ */
2
3 /*
4 * ISO-8859-3
5 */
6
7 #ifndef NEED_TOWC
8 static const unsigned short iso8859_3_uni[96] = {
9   /* 0xa0 */
10  0x00a0, 0x0126, 0x02d8, 0x00a3, 0x00a4, 0xffff, 0x0124, 0x00a7,
11  0x00a8, 0x0130, 0x015e, 0x011e, 0x0134, 0x00ad, 0xffff, 0x017b,
12  /* 0xb0 */

```

```

13 0x00b0, 0x0127, 0x00b2, 0x00b3, 0x00b4, 0x00b5, 0x0125, 0x00b7,
14 0x00b8, 0x0131, 0x015f, 0x011f, 0x0135, 0x00bd, 0xffffd, 0x017c,
15 /* 0xc0 */
16 0x00c0, 0x00c1, 0x00c2, 0xffffd, 0x00c4, 0x010a, 0x0108, 0x00c7,
17 0x00c8, 0x00c9, 0x00ca, 0x00cb, 0x00cc, 0x00cd, 0x00ce, 0x00cf,
18 /* 0xd0 */
19 0xffffd, 0x00d1, 0x00d2, 0x00d3, 0x00d4, 0x0120, 0x00d6, 0x00d7,
20 0x011c, 0x00d9, 0x00da, 0x00db, 0x00dc, 0x016c, 0x015c, 0x00df,
21 /* 0xe0 */
22 0x00e0, 0x00e1, 0x00e2, 0xffffd, 0x00e4, 0x010b, 0x0109, 0x00e7,
23 0x00e8, 0x00e9, 0x00ea, 0x00eb, 0x00ec, 0x00ed, 0x00ee, 0x00ef,
24 /* 0xf0 */
25 0xffffd, 0x00f1, 0x00f2, 0x00f3, 0x00f4, 0x0121, 0x00f6, 0x00f7,
26 0x011d, 0x00f9, 0x00fa, 0x00fb, 0x00fc, 0x016d, 0x015d, 0x02d9,
27 };
28
29 static int
30 iso8859_3_mbtowc (conv_t conv, ucs4_t *pwc, const unsigned char *s, int n)
31 {
32  unsigned char c = *s;
33  if (c < 0xa0) {
34      *pwc = (ucs4_t) c;
35      return 1;
36  }
37  else {
38      unsigned short wc = iso8859_3_2uni[c-0xa0];
39      if (wc != 0xffffd) {
40          *pwc = (ucs4_t) wc;
41          return 1;
42      }
43  }
44  return RET_ILSEQ;
45 }
46 #endif /* NEED_TOWC */
47
48 #ifdef NEED_TOMB
49 static const unsigned char iso8859_3_page00[96] = {
50  0xa0, 0x00, 0x00, 0xa3, 0xa4, 0x00, 0x00, 0xa7, /* 0xa0-0xa7 */
51  0xa8, 0x00, 0x00, 0x00, 0x00, 0xad, 0x00, 0x00, /* 0xa8-0xaf */
52  0xb0, 0x00, 0xb2, 0xb3, 0xb4, 0xb5, 0x00, 0xb7, /* 0xb0-0xb7 */
53  0xb8, 0x00, 0x00, 0x00, 0x00, 0xbd, 0x00, 0x00, /* 0xb8-0xbf */
54  0xc0, 0xc1, 0xc2, 0x00, 0xc4, 0x00, 0x00, 0xc7, /* 0xc0-0xc7 */
55  0xc8, 0xc9, 0xca, 0xcb, 0xcc, 0xcd, 0xce, 0xcf, /* 0xc8-0xcf */
56  0x00, 0xd1, 0xd2, 0xd3, 0xd4, 0x00, 0xd6, 0xd7, /* 0xd0-0xd7 */
57  0x00, 0xd9, 0xda, 0xdb, 0xdc, 0x00, 0x00, 0xdf, /* 0xd8-0xdf */
58  0xe0, 0xe1, 0xe2, 0x00, 0xe4, 0x00, 0x00, 0xe7, /* 0xe0-0xe7 */
59  0xe8, 0xe9, 0xea, 0xeb, 0xec, 0xed, 0xee, 0xef, /* 0xe8-0xef */
60  0x00, 0xf1, 0xf2, 0xf3, 0xf4, 0x00, 0xf6, 0xf7, /* 0xf0-0xf7 */
61  0x00, 0xf9, 0xfa, 0xfb, 0xfc, 0x00, 0x00, 0x00, /* 0xf8-0xff */
62 };
63 static const unsigned char iso8859_3_page01[120] = {
64  0xc6, 0xe6, 0xc5, 0xe5, 0x00, 0x00, 0x00, 0x00, /* 0x08-0x0f */
65  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x10-0x17 */
66  0x00, 0x00, 0x00, 0x00, 0xd8, 0xf8, 0xab, 0xbb, /* 0x18-0x1f */
67  0xd5, 0xf5, 0x00, 0x00, 0xa6, 0xb6, 0xa1, 0xb1, /* 0x20-0x27 */
68  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x28-0x2f */
69  0xa9, 0xb9, 0x00, 0x00, 0xac, 0xbc, 0x00, 0x00, /* 0x30-0x37 */
70  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x38-0x3f */
71  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x40-0x47 */
72  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x48-0x4f */
73  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x50-0x57 */
74  0x00, 0x00, 0x00, 0x00, 0xde, 0xfe, 0xaa, 0xba, /* 0x58-0x5f */
75  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x60-0x67 */
76  0x00, 0x00, 0x00, 0x00, 0xdd, 0xfd, 0x00, 0x00, /* 0x68-0x6f */
77  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x70-0x77 */
78  0x00, 0x00, 0x00, 0xaf, 0xbf, 0x00, 0x00, 0x00, /* 0x78-0x7f */
79 };
80 static const unsigned char iso8859_3_page02[8] = {
81  0xa2, 0xff, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xd8-0xdf */
82 };
83
84 static int
85 iso8859_3_wctomb (conv_t conv, unsigned char *r, ucs4_t wc, int n)
86 {
87  unsigned char c = 0;
88  if (wc < 0x00a0) {
89      *r = wc;
90      return 1;
91  }
92  else if (wc >= 0x00a0 && wc < 0x0100)
93      c = iso8859_3_page00[wc-0x00a0];
94  else if (wc >= 0x0108 && wc < 0x0180)
95      c = iso8859_3_page01[wc-0x0108];
96  else if (wc >= 0x02d8 && wc < 0x02e0)
97      c = iso8859_3_page02[wc-0x02d8];
98  if (c != 0) {
99      *r = c;

```

```
100     return 1;
101 }
102 return RET_ILSEQ;
103 }
104 #endif /* NEED_TOMB */
```

32.220 iso8859_4.h

```
1 /* $XFree86: xc/lib/X11/lcUniConv/iso8859_4.h,v 1.3 2000/11/29 17:40:31 dawes Exp $ */
2
3 /*
4 * ISO-8859-4
5 */
6
7 #ifdef NEED_TOWC
8 static const unsigned short iso8859_4_2uni[96] = {
9     /* 0xa0 */
10    0x00a0, 0x0104, 0x0138, 0x0156, 0x00a4, 0x0128, 0x013b, 0x00a7,
11    0x00a8, 0x0160, 0x0112, 0x0122, 0x0166, 0x00ad, 0x017d, 0x00af,
12    /* 0xb0 */
13    0x00b0, 0x0105, 0x02db, 0x0157, 0x00b4, 0x0129, 0x013c, 0x02c7,
14    0x00b8, 0x0161, 0x0113, 0x0123, 0x0167, 0x014a, 0x017e, 0x014b,
15    /* 0xc0 */
16    0x0100, 0x00c1, 0x00c2, 0x00c3, 0x00c4, 0x00c5, 0x00c6, 0x012e,
17    0x010c, 0x00c9, 0x0118, 0x00cb, 0x0116, 0x00cd, 0x00ce, 0x012a,
18    /* 0xd0 */
19    0x0110, 0x0145, 0x014c, 0x0136, 0x00d4, 0x00d5, 0x00d6, 0x00d7,
20    0x00d8, 0x0172, 0x00da, 0x00db, 0x00dc, 0x0168, 0x016a, 0x00df,
21    /* 0xe0 */
22    0x0101, 0x00e1, 0x00e2, 0x00e3, 0x00e4, 0x00e5, 0x00e6, 0x012f,
23    0x010d, 0x00e9, 0x0119, 0x00eb, 0x0117, 0x00ed, 0x00ee, 0x012b,
24    /* 0xf0 */
25    0x0111, 0x0146, 0x014d, 0x0137, 0x00f4, 0x00f5, 0x00f6, 0x00f7,
26    0x00f8, 0x0173, 0x00fa, 0x00fb, 0x00fc, 0x0169, 0x016b, 0x02d9,
27 };
28
29 static int
30 iso8859_4_mbtowc (conv_t conv, ucs4_t *pwc, const unsigned char *s, int n)
31 {
32     unsigned char c = *s;
33     if (c < 0xa0)
34         *pwc = (ucs4_t) c;
35     else
36         *pwc = (ucs4_t) iso8859_4_2uni[c-0xa0];
37     return 1;
38 }
39 #endif /* NEED_TOWC */
40
41 #ifdef NEED_TOMB
42 static const unsigned char iso8859_4_page00[224] = {
43     0xa0, 0x00, 0x00, 0x00, 0xa4, 0x00, 0x00, 0xa7, /* 0xa0-0xa7 */
44     0xa8, 0x00, 0x00, 0x00, 0xad, 0x00, 0xaf, /* 0xa8-0xaf */
45     0xb0, 0x00, 0x00, 0xb4, 0x00, 0x00, 0x00, /* 0xb0-0xb7 */
46     0xb8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xb8-0xbf */
47     0x00, 0xc1, 0xc2, 0xc3, 0xc4, 0xc5, 0xc6, 0x00, /* 0xc0-0xc7 */
48     0x00, 0xc9, 0x00, 0xcb, 0x00, 0xcd, 0xce, 0x00, /* 0xc8-0xcf */
49     0x00, 0x00, 0x00, 0x00, 0xd4, 0xd5, 0xd6, 0xd7, /* 0xd0-0xd7 */
50     0xd8, 0x00, 0xda, 0xdb, 0xdc, 0x00, 0x00, 0xdf, /* 0xd8-0xdf */
51     0x00, 0xe1, 0xe2, 0xe3, 0xe4, 0xe5, 0xe6, 0x00, /* 0xe0-0xe7 */
52     0x00, 0xe9, 0x00, 0xeb, 0x00, 0xed, 0xee, 0x00, /* 0xe8-0xef */
53     0x00, 0x00, 0xf0, 0xf4, 0xf5, 0xf6, 0xf7, /* 0xf0-0xf7 */
54     0xf8, 0x00, 0xfa, 0xfb, 0xfc, 0x00, 0x00, 0x00, /* 0xf8-0xff */
55     /* 0x0100 */
56     0xc0, 0xe0, 0x00, 0x00, 0xa1, 0xb1, 0x00, 0x00, /* 0x00-0x07 */
57     0x00, 0x00, 0x00, 0x00, 0xc8, 0xe8, 0x00, 0x00, /* 0x08-0x0f */
58     0xd0, 0xf0, 0xaa, 0xba, 0x00, 0x00, 0xcc, 0xec, /* 0x10-0x17 */
59     0xca, 0xea, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x18-0x1f */
60     0x00, 0x00, 0xab, 0xbb, 0x00, 0x00, 0x00, 0x00, /* 0x20-0x27 */
61     0xa5, 0xb5, 0xcf, 0xef, 0x00, 0x00, 0xc7, 0xe7, /* 0x28-0x2f */
62     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xd3, 0xf3, /* 0x30-0x37 */
63     0xa2, 0x00, 0x00, 0xa6, 0xb6, 0x00, 0x00, 0x00, /* 0x38-0x3f */
64     0x00, 0x00, 0x00, 0x00, 0x00, 0xd1, 0xf1, 0x00, /* 0x40-0x47 */
65     0x00, 0x00, 0xbd, 0xbf, 0xd2, 0xf2, 0x00, 0x00, /* 0x48-0x4f */
66     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xa3, 0xb3, /* 0x50-0x57 */
67     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x58-0x5f */
68     0xa9, 0xb9, 0x00, 0x00, 0x00, 0x00, 0xac, 0xbc, /* 0x60-0x67 */
69     0xdd, 0xfd, 0xde, 0xfe, 0x00, 0x00, 0x00, 0x00, /* 0x68-0x6f */
70     0x00, 0x00, 0xd9, 0xf9, 0x00, 0x00, 0x00, 0x00, /* 0x70-0x77 */
71     0x00, 0x00, 0x00, 0x00, 0x00, 0xae, 0xbe, 0x00, /* 0x78-0x7f */
72 };
73 static const unsigned char iso8859_4_page02[32] = {
74     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xb7, /* 0xc0-0xc7 */
75     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xc8-0xcf */
76     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xd0-0xd7 */
77     0x00, 0xff, 0x00, 0xb2, 0x00, 0x00, 0x00, 0x00, /* 0xd8-0xdf */

```

```
78 };
79
80 static int
81 iso8859_4_wctomb (conv_t conv, unsigned char *r, ucs4_t wc, int n)
82 {
83     unsigned char c = 0;
84     if (wc < 0x00a0) {
85         *r = wc;
86         return 1;
87     }
88     else if (wc >= 0x00a0 && wc < 0x0180)
89         c = iso8859_4_page00[wc-0x00a0];
90     else if (wc >= 0x02c0 && wc < 0x02e0)
91         c = iso8859_4_page02[wc-0x02c0];
92     if (c != 0) {
93         *r = c;
94         return 1;
95     }
96     return RET_ILSEQ;
97 }
98 #endif /* NEED_TOMB */
```

32.221 iso8859_5.h

```
1 /* $XFree86: xc/lib/X11/lcUniConv/iso8859_5.h,v 1.3 2000/11/29 17:40:32 dawes Exp $ */
2
3 /*
4 * ISO-8859-5
5 */
6
7 #ifndef NEED_TOWC
8 static const unsigned short iso8859_5_uni[96] = {
9     /* 0xa0 */
10     0x00a0, 0x0401, 0x0402, 0x0403, 0x0404, 0x0405, 0x0406, 0x0407,
11     0x0408, 0x0409, 0x040a, 0x040b, 0x040c, 0x00ad, 0x040e, 0x040f,
12     /* 0xb0 */
13     0x0410, 0x0411, 0x0412, 0x0413, 0x0414, 0x0415, 0x0416, 0x0417,
14     0x0418, 0x0419, 0x041a, 0x041b, 0x041c, 0x041d, 0x041e, 0x041f,
15     /* 0xc0 */
16     0x0420, 0x0421, 0x0422, 0x0423, 0x0424, 0x0425, 0x0426, 0x0427,
17     0x0428, 0x0429, 0x042a, 0x042b, 0x042c, 0x042d, 0x042e, 0x042f,
18     /* 0xd0 */
19     0x0430, 0x0431, 0x0432, 0x0433, 0x0434, 0x0435, 0x0436, 0x0437,
20     0x0438, 0x0439, 0x043a, 0x043b, 0x043c, 0x043d, 0x043e, 0x043f,
21     /* 0xe0 */
22     0x0440, 0x0441, 0x0442, 0x0443, 0x0444, 0x0445, 0x0446, 0x0447,
23     0x0448, 0x0449, 0x044a, 0x044b, 0x044c, 0x044d, 0x044e, 0x044f,
24     /* 0xf0 */
25     0x2116, 0x0451, 0x0452, 0x0453, 0x0454, 0x0455, 0x0456, 0x0457,
26     0x0458, 0x0459, 0x045a, 0x045b, 0x045c, 0x00a7, 0x045e, 0x045f,
27 };
28
29 static int
30 iso8859_5_mbtowc (conv_t conv, ucs4_t *pwc, const unsigned char *s, int n)
31 {
32     unsigned char c = *s;
33     if (c < 0xa0)
34         *pwc = (ucs4_t) c;
35     else
36         *pwc = (ucs4_t) iso8859_5_uni[c-0xa0];
37     return 1;
38 }
39 #endif /* NEED_TOWC */
40
41 #ifndef NEED_TOMB
42 static const unsigned char iso8859_5_page00[16] = {
43     0xa0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xfd, /* 0xa0-0xa7 */
44     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xad, 0x00, 0x00, /* 0xa8-0xaf */
45 };
46 static const unsigned char iso8859_5_page04[96] = {
47     0x00, 0xa1, 0xa2, 0xa3, 0xa4, 0xa5, 0xa6, 0xa7, /* 0x00-0x07 */
48     0xa8, 0xa9, 0xaa, 0xab, 0xac, 0x00, 0xae, 0xaf, /* 0x08-0x0f */
49     0xb0, 0xb1, 0xb2, 0xb3, 0xb4, 0xb5, 0xb6, 0xb7, /* 0x10-0x17 */
50     0xb8, 0xb9, 0xba, 0xbb, 0xbc, 0xbd, 0xbe, 0xbf, /* 0x18-0x1f */
51     0xc0, 0xc1, 0xc2, 0xc3, 0xc4, 0xc5, 0xc6, 0xc7, /* 0x20-0x27 */
52     0xc8, 0xc9, 0xca, 0xcb, 0xcc, 0xcd, 0xce, 0xcf, /* 0x28-0x2f */
53     0xd0, 0xd1, 0xd2, 0xd3, 0xd4, 0xd5, 0xd6, 0xd7, /* 0x30-0x37 */
54     0xd8, 0xd9, 0xda, 0xdb, 0xdc, 0xdd, 0xde, 0xdf, /* 0x38-0x3f */
55     0xe0, 0xe1, 0xe2, 0xe3, 0xe4, 0xe5, 0xe6, 0xe7, /* 0x40-0x47 */
56     0xe8, 0xe9, 0xea, 0xeb, 0xec, 0xed, 0xee, 0xef, /* 0x48-0x4f */
57     0x00, 0xf1, 0xf2, 0xf3, 0xf4, 0xf5, 0xf6, 0xf7, /* 0x50-0x57 */
58     0xf8, 0xf9, 0xfa, 0xfb, 0xfc, 0x00, 0xfe, 0xff, /* 0x58-0x5f */
59 };
60
61 static int
```

```
62 iso8859_5_wctomb (conv_t conv, unsigned char *r, ucs4_t wc, int n)
63 {
64     unsigned char c = 0;
65     if (wc < 0x00a0) {
66         *r = wc;
67         return 1;
68     }
69     else if (wc >= 0x00a0 && wc < 0x00b0)
70         c = iso8859_5_page00[wc-0x00a0];
71     else if (wc >= 0x0400 && wc < 0x0460)
72         c = iso8859_5_page04[wc-0x0400];
73     else if (wc == 0x2116)
74         c = 0xf0;
75     if (c != 0) {
76         *r = c;
77         return 1;
78     }
79     return RET_ILSEQ;
80 }
81 #endif /* NEED_TOMB */
```

32.222 iso8859_6.h

```
1 /* $XFree86: xc/lib/X11/lcUniConv/iso8859_6.h,v 1.3 2000/11/29 17:40:32 dawes Exp $ */
2
3 /*
4  * ISO-8859-6
5  */
6
7 #ifdef NEED_TOWC
8 static const unsigned short iso8859_6_2uni[96] = {
9     /* 0xa0 */
10    0x00a0, 0xffff, 0xffff, 0xffff, 0x00a4, 0xffff, 0xffff, 0xffff,
11    0xffff, 0xffff, 0xffff, 0xffff, 0x060c, 0x00ad, 0xffff, 0xffff,
12    /* 0xb0 */
13    0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
14    0xffff, 0xffff, 0xffff, 0x061b, 0xffff, 0xffff, 0xffff, 0x061f,
15    /* 0xc0 */
16    0xffff, 0x0621, 0x0622, 0x0623, 0x0624, 0x0625, 0x0626, 0x0627,
17    0x0628, 0x0629, 0x062a, 0x062b, 0x062c, 0x062d, 0x062e, 0x062f,
18    /* 0xd0 */
19    0x0630, 0x0631, 0x0632, 0x0633, 0x0634, 0x0635, 0x0636, 0x0637,
20    0x0638, 0x0639, 0x063a, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
21    /* 0xe0 */
22    0x0640, 0x0641, 0x0642, 0x0643, 0x0644, 0x0645, 0x0646, 0x0647,
23    0x0648, 0x0649, 0x064a, 0x064b, 0x064c, 0x064d, 0x064e, 0x064f,
24    /* 0xf0 */
25    0x0650, 0x0651, 0x0652, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
26    0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
27 };
28
29 static int
30 iso8859_6_mbtowc (conv_t conv, ucs4_t *pwc, const unsigned char *s, int n)
31 {
32     unsigned char c = *s;
33     if (c < 0xa0) {
34         *pwc = (ucs4_t) c;
35         return 1;
36     }
37     else {
38         unsigned short wc = iso8859_6_2uni[c-0xa0];
39         if (wc != 0xffff) {
40             *pwc = (ucs4_t) wc;
41             return 1;
42         }
43     }
44     return RET_ILSEQ;
45 }
46 #endif /* NEED_TOWC */
47
48 #ifdef NEED_TOMB
49 static const unsigned char iso8859_6_page00[16] = {
50     0xa0, 0x00, 0x00, 0x00, 0xa4, 0x00, 0x00, 0x00, /* 0xa0-0xa7 */
51     0x00, 0x00, 0x00, 0x00, 0x00, 0xad, 0x00, 0x00, /* 0xa8-0xaf */
52 };
53 static const unsigned char iso8859_6_page06[80] = {
54     0x00, 0x00, 0x00, 0x00, 0xac, 0x00, 0x00, 0x00, /* 0x08-0x0f */
55     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x10-0x17 */
56     0x00, 0x00, 0x00, 0xbb, 0x00, 0x00, 0x00, 0xbf, /* 0x18-0x1f */
57     0x00, 0xc1, 0xc2, 0xc3, 0xc4, 0xc5, 0xc6, 0xc7, /* 0x20-0x27 */
58     0xc8, 0xc9, 0xca, 0xcb, 0xcc, 0xcd, 0xce, 0xcf, /* 0x28-0x2f */
59     0xd0, 0xd1, 0xd2, 0xd3, 0xd4, 0xd5, 0xd6, 0xd7, /* 0x30-0x37 */
60     0xd8, 0xd9, 0xda, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x38-0x3f */
61     0xe0, 0xe1, 0xe2, 0xe3, 0xe4, 0xe5, 0xe6, 0xe7, /* 0x40-0x47 */
62     0xe8, 0xe9, 0xea, 0xeb, 0xec, 0xed, 0xee, 0xef, /* 0x48-0x4f */

```

```

63 0xf0, 0xf1, 0xf2, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x50-0x57 */
64 };
65
66 static int
67 iso8859_6_wctomb (conv_t conv, unsigned char *r, ucs4_t wc, int n)
68 {
69     unsigned char c = 0;
70     if (wc < 0x00a0) {
71         *r = wc;
72         return 1;
73     }
74     else if (wc >= 0x00a0 && wc < 0x00b0)
75         c = iso8859_6_page00[wc-0x00a0];
76     else if (wc >= 0x0608 && wc < 0x0658)
77         c = iso8859_6_page06[wc-0x0608];
78     if (c != 0) {
79         *r = c;
80         return 1;
81     }
82     return RET_ILSEQ;
83 }
84 #endif /* NEED_TOMB */

```

32.223 iso8859_7.h

```

1 /* $XFree86: xc/lib/X11/lcUniConv/iso8859_7.h,v 1.3 2000/11/29 17:40:32 dawes Exp $ */
2
3 /*
4  * ISO-8859-7
5  */
6
7 #ifndef NEED_TOWC
8 static const unsigned short iso8859_7_2uni[96] = {
9     /* 0xa0 */
10    0x00a0, 0x2018, 0x2019, 0x00a3, 0xffff, 0xffff, 0x00a6, 0x00a7,
11    0x00a8, 0x00a9, 0xffff, 0x00ab, 0x00ac, 0x00ad, 0xffff, 0x2015,
12    /* 0xb0 */
13    0x00b0, 0x00b1, 0x00b2, 0x00b3, 0x0384, 0x0385, 0x0386, 0x00b7,
14    0x0388, 0x0389, 0x038a, 0x00bb, 0x038c, 0x00bd, 0x038e, 0x038f,
15    /* 0xc0 */
16    0x0390, 0x0391, 0x0392, 0x0393, 0x0394, 0x0395, 0x0396, 0x0397,
17    0x0398, 0x0399, 0x039a, 0x039b, 0x039c, 0x039d, 0x039e, 0x039f,
18    /* 0xd0 */
19    0x03a0, 0x03a1, 0xffff, 0x03a3, 0x03a4, 0x03a5, 0x03a6, 0x03a7,
20    0x03a8, 0x03a9, 0x03aa, 0x03ab, 0x03ac, 0x03ad, 0x03ae, 0x03af,
21    /* 0xe0 */
22    0x03b0, 0x03b1, 0x03b2, 0x03b3, 0x03b4, 0x03b5, 0x03b6, 0x03b7,
23    0x03b8, 0x03b9, 0x03ba, 0x03bb, 0x03bc, 0x03bd, 0x03be, 0x03bf,
24    /* 0xf0 */
25    0x03c0, 0x03c1, 0x03c2, 0x03c3, 0x03c4, 0x03c5, 0x03c6, 0x03c7,
26    0x03c8, 0x03c9, 0x03ca, 0x03cb, 0x03cc, 0x03cd, 0x03ce, 0xffff,
27 };
28
29 static int
30 iso8859_7_mbtowc (conv_t conv, ucs4_t *pwc, const unsigned char *s, int n)
31 {
32     unsigned char c = *s;
33     if (c < 0xa0) {
34         *pwc = (ucs4_t) c;
35         return 1;
36     }
37     else {
38         unsigned short wc = iso8859_7_2uni[c-0xa0];
39         if (wc != 0xffff) {
40             *pwc = (ucs4_t) wc;
41             return 1;
42         }
43     }
44     return RET_ILSEQ;
45 }
46 #endif /* NEED_TOWC */
47
48 #ifndef NEED_TOMB
49 static const unsigned char iso8859_7_page00[32] = {
50     0xa0, 0x00, 0x00, 0xa3, 0x00, 0x00, 0xa6, 0xa7, /* 0xa0-0xa7 */
51     0xa8, 0xa9, 0x00, 0xab, 0xac, 0xad, 0x00, 0x00, /* 0xa8-0xaf */
52     0xb0, 0xb1, 0xb2, 0xb3, 0x00, 0x00, 0x00, 0xb7, /* 0xb0-0xb7 */
53     0x00, 0x00, 0x00, 0xbb, 0x00, 0xbd, 0x00, 0x00, /* 0xb8-0xbf */
54 };
55 static const unsigned char iso8859_7_page03[80] = {
56     0x00, 0x00, 0x00, 0x00, 0xb4, 0xb5, 0xb6, 0x00, /* 0x80-0x87 */
57     0xb8, 0xb9, 0xba, 0x00, 0xbc, 0x00, 0xbe, 0xbf, /* 0x88-0x8f */
58     0xc0, 0xc1, 0xc2, 0xc3, 0xc4, 0xc5, 0xc6, 0xc7, /* 0x90-0x97 */
59     0xc8, 0xc9, 0xca, 0xcb, 0xcc, 0xcd, 0xce, 0xcf, /* 0x98-0x9f */
60     0xd0, 0xd1, 0x00, 0xd3, 0xd4, 0xd5, 0xd6, 0xd7, /* 0xa0-0xa7 */

```

```

61 0xd8, 0xd9, 0xda, 0xdb, 0xdc, 0xdd, 0xde, 0xdf, /* 0xa8-0xaf */
62 0xe0, 0xe1, 0xe2, 0xe3, 0xe4, 0xe5, 0xe6, 0xe7, /* 0xb0-0xb7 */
63 0xe8, 0xe9, 0xea, 0xeb, 0xec, 0xed, 0xee, 0xef, /* 0xb8-0xbf */
64 0xf0, 0xf1, 0xf2, 0xf3, 0xf4, 0xf5, 0xf6, 0xf7, /* 0xc0-0xc7 */
65 0xf8, 0xf9, 0xfa, 0xfb, 0xfc, 0xfd, 0xfe, 0x00, /* 0xc8-0xcf */
66 };
67 static const unsigned char iso8859_7_page20[16] = {
68 0x00, 0x00, 0x00, 0x00, 0x00, 0xaf, 0x00, 0x00, /* 0x10-0x17 */
69 0xa1, 0xa2, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x18-0x1f */
70 };
71
72 static int
73 iso8859_7_wctomb (conv_t conv, unsigned char *r, ucs4_t wc, int n)
74 {
75     unsigned char c = 0;
76     if (wc < 0x00a0) {
77         *r = wc;
78         return 1;
79     }
80     else if (wc >= 0x00a0 && wc < 0x00c0)
81         c = iso8859_7_page00[wc-0x00a0];
82     else if (wc >= 0x0380 && wc < 0x03d0)
83         c = iso8859_7_page03[wc-0x0380];
84     else if (wc >= 0x2010 && wc < 0x2020)
85         c = iso8859_7_page20[wc-0x2010];
86     if (c != 0) {
87         *r = c;
88         return 1;
89     }
90     return RET_ILSEQ;
91 }
92 #endif /* NEED_TOMB */

```

32.224 iso8859_8.h

```

1 /* $XFree86: xc/lib/X11/lcUniConv/iso8859_8.h,v 1.3 2000/11/29 17:40:32 dawes Exp $ */
2
3 /*
4 * ISO-8859-8
5 */
6
7 #ifdef NEED_TOWC
8 static const unsigned short iso8859_8_2uni[96] = {
9     /* 0xa0 */
10     0x00a0, 0xffff, 0x00a2, 0x00a3, 0x00a4, 0x00a5, 0x00a6, 0x00a7,
11     0x00a8, 0x00a9, 0x00d7, 0x00ab, 0x00ac, 0x00ad, 0x00ae, 0x00af,
12     /* 0xb0 */
13     0x00b0, 0x00b1, 0x00b2, 0x00b3, 0x00b4, 0x00b5, 0x00b6, 0x00b7,
14     0x00b8, 0x00b9, 0x00f7, 0x00bb, 0x00bc, 0x00bd, 0x00be, 0xffff,
15     /* 0xc0 */
16     0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
17     0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
18     /* 0xd0 */
19     0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
20     0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x2017,
21     /* 0xe0 */
22     0x05d0, 0x05d1, 0x05d2, 0x05d3, 0x05d4, 0x05d5, 0x05d6, 0x05d7,
23     0x05d8, 0x05d9, 0x05da, 0x05db, 0x05dc, 0x05dd, 0x05de, 0x05df,
24     /* 0xf0 */
25     0x05e0, 0x05e1, 0x05e2, 0x05e3, 0x05e4, 0x05e5, 0x05e6, 0x05e7,
26     0x05e8, 0x05e9, 0x05ea, 0xffff, 0xffff, 0x200e, 0x200f, 0xffff,
27 };
28
29 static int
30 iso8859_8_mbtowc (conv_t conv, ucs4_t *pwc, const unsigned char *s, int n)
31 {
32     unsigned char c = *s;
33     if (c >= 0xa0) {
34         unsigned short wc = iso8859_8_2uni[c-0xa0];
35         if (wc != 0xffff) {
36             *pwc = (ucs4_t) wc;
37             return 1;
38         }
39     }
40     else {
41         *pwc = (ucs4_t) c;
42         return 1;
43     }
44     return RET_ILSEQ;
45 }
46 #endif /* NEED_TOWC */
47
48 #ifdef NEED_TOMB
49 static const unsigned char iso8859_8_page00[88] = {
50     0xa0, 0x00, 0xa2, 0xa3, 0xa4, 0xa5, 0xa6, 0xa7, /* 0xa0-0xa7 */

```

```

51 0xa8, 0xa9, 0x00, 0xab, 0xac, 0xad, 0xae, 0xaf, /* 0xa8-0xaf */
52 0xb0, 0xb1, 0xb2, 0xb3, 0xb4, 0xb5, 0xb6, 0xb7, /* 0xb0-0xb7 */
53 0xb8, 0xb9, 0x00, 0xbb, 0xbc, 0xbd, 0xbe, 0x00, /* 0xb8-0xbf */
54 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xc0-0xc7 */
55 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xc8-0xcf */
56 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xaa, /* 0xd0-0xd7 */
57 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xd8-0xdf */
58 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xe0-0xe7 */
59 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xe8-0xef */
60 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xba, /* 0xf0-0xf7 */
61 };
62 static const unsigned char iso8859_8_page05[32] = {
63 0xe0, 0xe1, 0xe2, 0xe3, 0xe4, 0xe5, 0xe6, 0xe7, /* 0xd0-0xd7 */
64 0xe8, 0xe9, 0xea, 0xeb, 0xec, 0xed, 0xee, 0xef, /* 0xd8-0xdf */
65 0xf0, 0xf1, 0xf2, 0xf3, 0xf4, 0xf5, 0xf6, 0xf7, /* 0xe0-0xe7 */
66 0xf8, 0xf9, 0xfa, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xe8-0xef */
67 };
68 static const unsigned char iso8859_8_page20[16] = {
69 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xfd, 0xfe, /* 0x08-0x0f */
70 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xdf, /* 0x10-0x17 */
71 };
72
73 static int
74 iso8859_8_wctomb (conv_t conv, unsigned char *r, ucs4_t wc, int n)
75 {
76     unsigned char c = 0;
77     if (wc < 0x00a0) {
78         *r = wc;
79         return 1;
80     }
81     else if (wc >= 0x00a0 && wc < 0x00f8)
82         c = iso8859_8_page00[wc-0x00a0];
83     else if (wc >= 0x00d0 && wc < 0x05f0)
84         c = iso8859_8_page05[wc-0x00d0];
85     else if (wc >= 0x2008 && wc < 0x2018)
86         c = iso8859_8_page20[wc-0x2008];
87     if (c != 0) {
88         *r = c;
89         return 1;
90     }
91     return RET_ILSEQ;
92 }
93 #endif /* NEED_TOMB */

```

32.225 iso8859_9.h

```

1 /* $XFree86: xc/lib/X11/lcUniConv/iso8859_9.h,v 1.3 2000/11/29 17:40:32 dawes Exp $ */
2
3 /*
4 * ISO-8859-9
5 */
6
7 #ifdef NEED_TOWC
8 static const unsigned short iso8859_9_2uni[48] = {
9     /* 0xd0 */
10 0x011e, 0x00d1, 0x00d2, 0x00d3, 0x00d4, 0x00d5, 0x00d6, 0x00d7,
11 0x00d8, 0x00d9, 0x00da, 0x00db, 0x00dc, 0x0130, 0x015e, 0x00df,
12     /* 0xe0 */
13 0x00e0, 0x00e1, 0x00e2, 0x00e3, 0x00e4, 0x00e5, 0x00e6, 0x00e7,
14 0x00e8, 0x00e9, 0x00ea, 0x00eb, 0x00ec, 0x00ed, 0x00ee, 0x00ef,
15     /* 0xf0 */
16 0x011f, 0x00f1, 0x00f2, 0x00f3, 0x00f4, 0x00f5, 0x00f6, 0x00f7,
17 0x00f8, 0x00f9, 0x00fa, 0x00fb, 0x00fc, 0x0131, 0x015f, 0x00ff,
18 };
19
20 static int
21 iso8859_9_mbtowc (conv_t conv, ucs4_t *pwc, const unsigned char *s, int n)
22 {
23     unsigned char c = *s;
24     if (c >= 0xd0)
25         *pwc = (ucs4_t) iso8859_9_2uni[c-0xd0];
26     else
27         *pwc = (ucs4_t) c;
28     return 1;
29 }
30 #endif /* NEED_TOWC */
31
32 #ifdef NEED_TOMB
33 static const unsigned char iso8859_9_page00[48] = {
34 0x00, 0xd1, 0xd2, 0xd3, 0xd4, 0xd5, 0xd6, 0xd7, /* 0xd0-0xd7 */
35 0xd8, 0xd9, 0xda, 0xdb, 0xdc, 0x00, 0x00, 0xdf, /* 0xd8-0xdf */
36 0xe0, 0xe1, 0xe2, 0xe3, 0xe4, 0xe5, 0xe6, 0xe7, /* 0xe0-0xe7 */
37 0xe8, 0xe9, 0xea, 0xeb, 0xec, 0xed, 0xee, 0xef, /* 0xe8-0xef */
38 0x00, 0xf1, 0xf2, 0xf3, 0xf4, 0xf5, 0xf6, 0xf7, /* 0xf0-0xf7 */
39 0xf8, 0xf9, 0xfa, 0xfb, 0xfc, 0x00, 0x00, 0xff, /* 0xf8-0xff */

```



```

40 };
41 static const unsigned char iso8859_9_page01[72] = {
42 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xd0, 0xf0, /* 0x18-0x1f */
43 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x20-0x27 */
44 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x28-0x2f */
45 0xdd, 0xfd, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x30-0x37 */
46 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x38-0x3f */
47 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x40-0x47 */
48 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x48-0x4f */
49 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x50-0x57 */
50 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xde, 0xfe, /* 0x58-0x5f */
51 };
52
53 static int
54 iso8859_9_wctomb (conv_t conv, unsigned char *r, ucs4_t wc, int n)
55 {
56     unsigned char c = 0;
57     if (wc < 0x00d0) {
58         *r = wc;
59         return 1;
60     }
61     else if (wc >= 0x00d0 && wc < 0x0100)
62         c = iso8859_9_page00[wc-0x00d0];
63     else if (wc >= 0x0118 && wc < 0x0160)
64         c = iso8859_9_page01[wc-0x0118];
65     if (c != 0) {
66         *r = c;
67         return 1;
68     }
69     return RET_ILSEQ;
70 }
71 #endif /* NEED_TOMB */

```

32.226 iso8859_9e.h

```

1 /* $XFree86: xc/lib/X11/lcUniConv/iso8859_9e.h,v 1.3 2000/11/28 16:10:28 dawes Exp $ */
2
3 /*
4 * ISO-8859-9E
5 */
6
7 static const unsigned short iso8859_9e_2uni[96] = {
8     /* 0xa0 */
9     0x00a0, 0x017d, 0x00a2, 0x00a3, 0x20ac, 0x00a5, 0x012c, 0x00a7,
10    0x016c, 0x00a9, 0x01e6, 0x00ab, 0x014a, 0x00ad, 0x00ae, 0x01d1,
11    /* 0xb0 */
12    0x00b0, 0x017e, 0x00b2, 0x00b3, 0x00b4, 0x00b5, 0x012d, 0x00b7,
13    0x016d, 0x00b9, 0x01e7, 0x00bb, 0x014b, 0x00bd, 0x0178, 0x01d2,
14    /* 0xc0 */
15    0x00c0, 0x00c1, 0x00c2, 0x00c3, 0x00c4, 0x00c5, 0x018f, 0x00c7,
16    0x00c8, 0x00c9, 0x00ca, 0x00cb, 0x00cc, 0x00cd, 0x00ce, 0x00cf,
17    /* 0xd0 */
18    0x011e, 0x00d1, 0x00d2, 0x00d3, 0x00d4, 0x00d5, 0x00d6, 0x00dd,
19    0x019f, 0x00d9, 0x00da, 0x00db, 0x00dc, 0x0130, 0x015e, 0x00df,
20    /* 0xe0 */
21    0x00e0, 0x00e1, 0x00e2, 0x00e3, 0x00e4, 0x00e5, 0x0259, 0x00e7,
22    0x00e8, 0x00e9, 0x00ea, 0x00eb, 0x00ec, 0x00ed, 0x00ee, 0x00ef,
23    /* 0xf0 */
24    0x011f, 0x00f1, 0x00f2, 0x00f3, 0x00f4, 0x00f5, 0x00f6, 0x00fd,
25    0x0275, 0x00f9, 0x00fa, 0x00fb, 0x00fc, 0x0131, 0x015f, 0x00ff,
26 };
27
28 static int
29 iso8859_9e_mbtowc (conv_t conv, ucs4_t *pwc, const unsigned char *s, int n)
30 {
31     unsigned char c = *s;
32     if (c >= 0xa0)
33         *pwc = (ucs4_t) iso8859_9e_2uni[c-0xa0];
34     else
35         *pwc = (ucs4_t) c;
36     return 1;
37 }
38
39 static const unsigned char iso8859_9e_page00[96] = {
40 0xa0, 0x00, 0xa2, 0xa3, 0x00, 0xa5, 0x00, 0xa7, /* 0xa0-0xa7 */
41 0x00, 0xa9, 0x00, 0xab, 0x00, 0xad, 0xae, 0x00, /* 0xa8-0xaf */
42 0xb0, 0x00, 0xb2, 0xb3, 0xb4, 0xb5, 0x00, 0xb7, /* 0xb0-0xb7 */
43 0x00, 0xb9, 0x00, 0xbb, 0x00, 0xbd, 0x00, 0x00, /* 0xb8-0xbf */
44 0xc0, 0xc1, 0xc2, 0xc3, 0xc4, 0xc5, 0x00, 0xc7, /* 0xc0-0xc7 */
45 0xc8, 0xc9, 0xca, 0xcb, 0xcc, 0xcd, 0xce, 0xcf, /* 0xc8-0xcf */
46 0x00, 0xd1, 0xd2, 0xd3, 0xd4, 0xd5, 0xd6, 0x00, /* 0xd0-0xd7 */
47 0x00, 0xd9, 0xda, 0xdb, 0xdc, 0xdd, 0x00, 0xdf, /* 0xd8-0xdf */
48 0xe0, 0xe1, 0xe2, 0xe3, 0xe4, 0xe5, 0x00, 0xe7, /* 0xe0-0xe7 */
49 0xe8, 0xe9, 0xea, 0xeb, 0xec, 0xed, 0xee, 0xef, /* 0xe8-0xef */
50 0x00, 0xf1, 0xf2, 0xf3, 0xf4, 0xf5, 0xf6, 0x00, /* 0xf0-0xf7 */

```

```

51 0x00, 0xf9, 0xfa, 0xfb, 0xfc, 0xf7, 0x00, 0xff, /* 0xf8-0xff */
52 };
53 static const unsigned char iso8859_9e_page01[136] = {
54 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xd0, 0xf0, /* 0x18-0x1f */
55 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x20-0x27 */
56 0x00, 0x00, 0x00, 0x00, 0xa6, 0xb6, 0x00, 0x00, /* 0x28-0x2f */
57 0xdd, 0xfd, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x30-0x37 */
58 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x38-0x3f */
59 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x40-0x47 */
60 0x00, 0x00, 0xac, 0xbc, 0x00, 0x00, 0x00, 0x00, /* 0x48-0x4f */
61 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x50-0x57 */
62 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xde, 0xfe, /* 0x58-0x5f */
63 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x60-0x67 */
64 0x00, 0x00, 0x00, 0x00, 0xa8, 0xb8, 0x00, 0x00, /* 0x68-0x6f */
65 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x70-0x77 */
66 0xbe, 0x00, 0x00, 0x00, 0x00, 0xa1, 0xb1, 0x00, /* 0x78-0x7f */
67 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x80-0x87 */
68 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xc6, /* 0x88-0x8f */
69 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x90-0x97 */
70 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xd8, /* 0x98-0x9f */
71 };
72 static const unsigned char iso8859_9e_page01_d[24] = {
73 0x00, 0xaf, 0xbf, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xd0-0xd7 */
74 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xd8-0xdf */
75 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xaa, 0xba, /* 0xe0-0xe7 */
76 };
77
78 static int
79 iso8859_9e_wctomb (conv_t conv, unsigned char *r, ucs4_t wc, int n)
80 {
81     unsigned char c = 0;
82     if (wc < 0x00a0) {
83         *r = wc;
84         return 1;
85     }
86     else if (wc >= 0x00a0 && wc < 0x0100)
87         c = iso8859_9e_page00[wc-0x00a0];
88     else if (wc >= 0x0118 && wc < 0x01a0)
89         c = iso8859_9e_page01[wc-0x0118];
90     else if (wc >= 0x01d0 && wc < 0x01e8)
91         c = iso8859_9e_page01_d[wc-0x01d0];
92     else if (wc == 0x0259)
93         c = 0xe6;
94     else if (wc == 0x0275)
95         c = 0xf8;
96     else if (wc == 0x20ac)
97         c = 0xa4;
98     if (c != 0) {
99         *r = c;
100        return 1;
101    }
102    return RET_ILSEQ;
103 }

```

32.227 jisx0201.h

```

1 /* $XFree86: xc/lib/X11/lcUniConv/jisx0201.h,v 1.3 2000/11/29 17:40:33 dawes Exp $ */
2
3 /*
4  * JISX0201.1976-0
5  */
6 #ifdef NEED_TOWC
7
8 static int
9 jisx0201_mbtowc (conv_t conv, ucs4_t *pwc, const unsigned char *s, int n)
10 {
11     unsigned char c = *s;
12     if (c < 0x80) {
13         if (c == 0x5c)
14             *pwc = (ucs4_t) 0x00a5;
15         else if (c == 0x7e)
16             *pwc = (ucs4_t) 0x203e;
17         else
18             *pwc = (ucs4_t) c;
19         return 1;
20     } else {
21         if (c >= 0xa1 && c < 0xe0) {
22             *pwc = (ucs4_t) c + 0xfec0;
23             return 1;
24         }
25     }
26     return RET_ILSEQ;
27 }
28 #endif /* NEED_TOWC */
29

```

```
30 #ifdef NEED_TOMB
31
32 static int
33 jisx0201_wctomb (conv_t conv, unsigned char *r, ucs4_t wc, int n)
34 {
35     if (wc < 0x0080 && !(wc == 0x005c || wc == 0x007e)) {
36         *r = wc;
37         return 1;
38     }
39     if (wc == 0x00a5) {
40         *r = 0x5c;
41         return 1;
42     }
43     if (wc == 0x203e) {
44         *r = 0x7e;
45         return 1;
46     }
47     if (wc >= 0xff61 && wc < 0xffa0) {
48         *r = wc - 0xfec0;
49         return 1;
50     }
51     return RET_ILSEQ;
52 }
53 #endif /* NEED_TOMB */
```

32.228 jisx0208.h

```
1 /* $XFree86: xc/lib/X11/lcUniConv/jisx0208.h,v 1.6 2003/05/27 22:26:31 tsi Exp $ */
2
3 /*
4 * JISX0208.1990-0
5 */
6 #ifdef NEED_TOWC
7
8 static const unsigned short jisx0208_2uni_page21[690] = {
9     /* 0x21 */
10    0x3000, 0x3001, 0x3002, 0xff0c, 0xff0e, 0x30fb, 0xff1a, 0xff1b,
11    0xff1f, 0xff01, 0x309b, 0x309c, 0x00b4, 0xff40, 0x00a8, 0xff3e,
12    0xffe3, 0xff3f, 0x30fd, 0x30fe, 0x309d, 0x309e, 0x3003, 0x4edd,
13    0x3005, 0x3006, 0x3007, 0x30fc, 0x2015, 0x2010, 0xff0f, 0xff3c,
14    0x301c, 0x2016, 0xff5c, 0x2026, 0x2025, 0x2018, 0x2019, 0x201c,
15    0x201d, 0xff08, 0xff09, 0x3014, 0x3015, 0xff3b, 0xff3d, 0xff5b,
16    0xff5d, 0x3008, 0x3009, 0x300a, 0x300b, 0x300c, 0x300d, 0x300e,
17    0x300f, 0x3010, 0x3011, 0xff0b, 0x2212, 0x00b1, 0x00d7, 0x00f7,
18    0xff1d, 0x2260, 0xff1c, 0xff1e, 0x2266, 0x2267, 0x221e, 0x2234,
19    0x2642, 0x2640, 0x00b0, 0x2032, 0x2033, 0x2103, 0xffe5, 0xff04,
20    0x00a2, 0x00a3, 0xff05, 0xff03, 0xff06, 0xff0a, 0xff20, 0x00a7,
21    0x2606, 0x2605, 0x25cb, 0x25cf, 0x25ce, 0x25c7,
22    /* 0x22 */
23    0x25c6, 0x25a1, 0x25a0, 0x25b3, 0x25b2, 0x25bd, 0x25bc, 0x203b,
24    0x3012, 0x2192, 0x2190, 0x2191, 0x2193, 0x3013, 0xffff, 0xffff,
25    0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
26    0xffff, 0x2208, 0x220b, 0x2286, 0x2287, 0x2282, 0x2283, 0x222a,
27    0x2229, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
28    0xffff, 0x2227, 0x2228, 0x00ac, 0x21d2, 0x21d4, 0x2200, 0x2203,
29    0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
30    0xffff, 0xffff, 0xffff, 0x2220, 0x22a5, 0x2312, 0x2202, 0x2207,
31    0x2261, 0x2252, 0x226a, 0x226b, 0x221a, 0x223d, 0x221d, 0x2235,
32    0x222b, 0x222c, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
33    0xffff, 0x212b, 0x2030, 0x266f, 0x266d, 0x266a, 0x2020, 0x2021,
34    0x00b6, 0xffff, 0xffff, 0xffff, 0xffff, 0x25ef,
35    /* 0x23 */
36    0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
37    0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xff10,
38    0xff11, 0xff12, 0xff13, 0xff14, 0xff15, 0xff16, 0xff17, 0xff18,
39    0xff19, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
40    0xff21, 0xff22, 0xff23, 0xff24, 0xff25, 0xff26, 0xff27, 0xff28,
41    0xff29, 0xff2a, 0xff2b, 0xff2c, 0xff2d, 0xff2e, 0xff2f, 0xffff,
42    0xff31, 0xff32, 0xff33, 0xff34, 0xff35, 0xff36, 0xff37, 0xff38,
43    0xff39, 0xff3a, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
44    0xff41, 0xff42, 0xff43, 0xff44, 0xff45, 0xff46, 0xff47, 0xff48,
45    0xff49, 0xff4a, 0xff4b, 0xff4c, 0xff4d, 0xff4e, 0xff4f, 0xffff,
46    0xff51, 0xff52, 0xff53, 0xff54, 0xff55, 0xff56, 0xff57, 0xff58,
47    0xff59, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
48    /* 0x24 */
49    0x3041, 0x3042, 0x3043, 0x3044, 0x3045, 0x3046, 0x3047, 0x3048,
50    0x3049, 0x304a, 0x304b, 0x304c, 0x304d, 0x304e, 0x304f, 0x3050,
51    0x3051, 0x3052, 0x3053, 0x3054, 0x3055, 0x3056, 0x3057, 0x3058,
52    0x3059, 0x305a, 0x305b, 0x305c, 0x305d, 0x305e, 0x305f, 0x3060,
53    0x3061, 0x3062, 0x3063, 0x3064, 0x3065, 0x3066, 0x3067, 0x3068,
54    0x3069, 0x306a, 0x306b, 0x306c, 0x306d, 0x306e, 0x306f, 0x3070,
55    0x3071, 0x3072, 0x3073, 0x3074, 0x3075, 0x3076, 0x3077, 0x3078,
56    0x3079, 0x307a, 0x307b, 0x307c, 0x307d, 0x307e, 0x307f, 0x3080,
57    0x3081, 0x3082, 0x3083, 0x3084, 0x3085, 0x3086, 0x3087, 0x3088,
58    0x3089, 0x308a, 0x308b, 0x308c, 0x308d, 0x308e, 0x308f, 0x3090,
```

```

59 0x3091, 0x3092, 0x3093, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
60 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
61 /* 0x25 */
62 0x30a1, 0x30a2, 0x30a3, 0x30a4, 0x30a5, 0x30a6, 0x30a7, 0x30a8,
63 0x30a9, 0x30aa, 0x30ab, 0x30ac, 0x30ad, 0x30ae, 0x30af, 0x30b0,
64 0x30b1, 0x30b2, 0x30b3, 0x30b4, 0x30b5, 0x30b6, 0x30b7, 0x30b8,
65 0x30b9, 0x30ba, 0x30bb, 0x30bc, 0x30bd, 0x30be, 0x30bf, 0x30c0,
66 0x30c1, 0x30c2, 0x30c3, 0x30c4, 0x30c5, 0x30c6, 0x30c7, 0x30c8,
67 0x30c9, 0x30ca, 0x30cb, 0x30cc, 0x30cd, 0x30ce, 0x30cf, 0x30d0,
68 0x30d1, 0x30d2, 0x30d3, 0x30d4, 0x30d5, 0x30d6, 0x30d7, 0x30d8,
69 0x30d9, 0x30da, 0x30db, 0x30dc, 0x30dd, 0x30de, 0x30df, 0x30e0,
70 0x30e1, 0x30e2, 0x30e3, 0x30e4, 0x30e5, 0x30e6, 0x30e7, 0x30e8,
71 0x30e9, 0x30ea, 0x30eb, 0x30ec, 0x30ed, 0x30ee, 0x30ef, 0x30f0,
72 0x30f1, 0x30f2, 0x30f3, 0x30f4, 0x30f5, 0x30f6, 0xffff, 0xffff,
73 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
74 /* 0x26 */
75 0x0391, 0x0392, 0x0393, 0x0394, 0x0395, 0x0396, 0x0397, 0x0398,
76 0x0399, 0x039a, 0x039b, 0x039c, 0x039d, 0x039e, 0x039f, 0x03a0,
77 0x03a1, 0x03a3, 0x03a4, 0x03a5, 0x03a6, 0x03a7, 0x03a8, 0x03a9,
78 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
79 0x03b1, 0x03b2, 0x03b3, 0x03b4, 0x03b5, 0x03b6, 0x03b7, 0x03b8,
80 0x03b9, 0x03ba, 0x03bb, 0x03bc, 0x03bd, 0x03be, 0x03bf, 0x03c0,
81 0x03c1, 0x03c3, 0x03c4, 0x03c5, 0x03c6, 0x03c7, 0x03c8, 0x03c9,
82 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
83 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
84 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
85 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
86 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
87 /* 0x27 */
88 0x0410, 0x0411, 0x0412, 0x0413, 0x0414, 0x0415, 0x0416, 0x0417,
89 0x0418, 0x0419, 0x041a, 0x041b, 0x041c, 0x041d, 0x041e,
90 0x041f, 0x0420, 0x0421, 0x0422, 0x0423, 0x0424, 0x0425, 0x0426,
91 0x0427, 0x0428, 0x0429, 0x042a, 0x042b, 0x042c, 0x042d, 0x042e,
92 0x042f, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
93 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
94 0x0430, 0x0431, 0x0432, 0x0433, 0x0434, 0x0435, 0x0436, 0x0437,
95 0x0438, 0x0439, 0x043a, 0x043b, 0x043c, 0x043d, 0x043e,
96 0x043f, 0x0440, 0x0441, 0x0442, 0x0443, 0x0444, 0x0445, 0x0446,
97 0x0447, 0x0448, 0x0449, 0x044a, 0x044b, 0x044c, 0x044d, 0x044e,
98 0x044f, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
99 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
100 /* 0x28 */
101 0x2500, 0x2502, 0x250c, 0x2510, 0x2518, 0x2514, 0x251c, 0x252c,
102 0x2524, 0x2534, 0x253c, 0x2501, 0x2503, 0x250f, 0x2513, 0x251b,
103 0x2517, 0x2523, 0x2533, 0x252b, 0x253b, 0x254b, 0x2520, 0x252f,
104 0x2528, 0x2537, 0x253f, 0x251d, 0x2530, 0x2525, 0x2538, 0x2542,
105 };
106 static const unsigned short jisx0208_2uni_page30[6398] = {
107 /* 0x30 */
108 0x4e9c, 0x5516, 0x5a03, 0x963f, 0x54c0, 0x611b, 0x6328, 0x59f6,
109 0x9022, 0x8475, 0x831c, 0x7a50, 0x60aa, 0x63e1, 0x6e25, 0x65ed,
110 0x8466, 0x82a6, 0x9b5f, 0x6893, 0x5727, 0x65a1, 0x6271, 0x5b9b,
111 0x59d0, 0x867b, 0x98f4, 0x7d62, 0x7dbe, 0x9b8e, 0x6216, 0x7c9f,
112 0x88b7, 0x5b89, 0x5eb5, 0x6309, 0x6697, 0x6848, 0x95c7, 0x978d,
113 0x674f, 0x4ee5, 0x4f0a, 0x4f4d, 0x4f9d, 0x5049, 0x56f2, 0x5937,
114 0x59d4, 0x5a01, 0x5c09, 0x60df, 0x610f, 0x6170, 0x6613, 0x6905,
115 0x70ba, 0x754f, 0x7570, 0x79fb, 0x7dad, 0x7def, 0x80c3, 0x840e,
116 0x8863, 0x8b02, 0x9055, 0x907a, 0x533b, 0x4e95, 0x4ea5, 0x57df,
117 0x80b2, 0x90c1, 0x78ef, 0x4e00, 0x58f1, 0x6ea2, 0x9038, 0x7a32,
118 0x8328, 0x828b, 0x9c2f, 0x5141, 0x5370, 0x54bd, 0x54e1, 0x56e0,
119 0x59fb, 0x5f15, 0x98f2, 0x6deb, 0x80e4, 0x852d,
120 /* 0x31 */
121 0x9662, 0x9670, 0x96a0, 0x97fb, 0x540b, 0x53f3, 0x5b87, 0x70cf,
122 0x7fbd, 0x8fc2, 0x96e8, 0x536f, 0x9d5c, 0x7aba, 0x4e11, 0x7893,
123 0x81fc, 0x6e26, 0x5618, 0x5504, 0x6b1d, 0x851a, 0x9c3b, 0x59e5,
124 0x53a9, 0x6d66, 0x74dc, 0x958f, 0x5642, 0x4e91, 0x904b, 0x96f2,
125 0x834f, 0x990c, 0x53e1, 0x55b6, 0x5b30, 0x5f71, 0x6620, 0x66f3,
126 0x6804, 0x6c38, 0x6cf3, 0x6d29, 0x745b, 0x76c8, 0x7a4e, 0x9834,
127 0x82f1, 0x885b, 0x8a60, 0x92ed, 0x6db2, 0x75ab, 0x76ca, 0x99c5,
128 0x60a6, 0x8b01, 0x8d8a, 0x95b2, 0x698e, 0x53ad, 0x5186, 0x5712,
129 0x5830, 0x5944, 0x5bb4, 0x5ef6, 0x6028, 0x63a9, 0x63f4, 0x6cbf,
130 0x6f14, 0x708e, 0x7114, 0x7159, 0x71d5, 0x733f, 0x7e01, 0x8276,
131 0x82d1, 0x8597, 0x9060, 0x925b, 0x9d1b, 0x5869, 0x65bc, 0x6c5a,
132 0x7525, 0x51f9, 0x592e, 0x5965, 0x5f80, 0x5fdc,
133 /* 0x32 */
134 0x62bc, 0x65fa, 0x6a2a, 0x6b27, 0x6bb4, 0x738b, 0x7fc1, 0x8956,
135 0x9d2c, 0x9d0e, 0x9ec4, 0x5ca1, 0x6c96, 0x837b, 0x5104, 0x5c4b,
136 0x61b6, 0x81c6, 0x6876, 0x7261, 0x4e59, 0x4ffa, 0x5378, 0x6069,
137 0x6e29, 0x7a4f, 0x97f3, 0x4e0b, 0x5316, 0x4eee, 0x4f55, 0x4f3d,
138 0x4fa1, 0x4f73, 0x52a0, 0x53ef, 0x5609, 0x590f, 0x5ac1, 0x5bb6,
139 0x5be1, 0x79d1, 0x6687, 0x679c, 0x67b6, 0x6b4c, 0x6cb3, 0x706b,
140 0x73c2, 0x798d, 0x79be, 0x7a3c, 0x7b87, 0x82b1, 0x82db, 0x8304,
141 0x8377, 0x83ef, 0x83d3, 0x8766, 0x8ab2, 0x5629, 0x8ca8, 0x8fe6,
142 0x904e, 0x971e, 0x868a, 0x4fc4, 0x5ce8, 0x6211, 0x7259, 0x753b,
143 0x81e5, 0x82bd, 0x86fe, 0x8cc0, 0x96c5, 0x9913, 0x99d5, 0x4ecb,
144 0x4f1a, 0x89e3, 0x56de, 0x584a, 0x58ca, 0x5efb, 0x5feb, 0x602a,
145 0x6094, 0x6062, 0x61d0, 0x6212, 0x62d0, 0x6539,

```



```
494 0x83b1, 0x983c, 0x96f7, 0x6dlb, 0x7d61, 0x843d, 0x916a, 0x4e71,
495 0x5375, 0x5d50, 0x6b04, 0x6feb, 0x85cd, 0x862d, 0x89a7, 0x5229,
496 0x540f, 0x5c65, 0x674e, 0x68a8, 0x7406, 0x7483,
497 /* 0x4e */
498 0x75e2, 0x88cf, 0x88e1, 0x91cc, 0x96e2, 0x9678, 0x5f8b, 0x7387,
499 0x7acb, 0x844e, 0x63a0, 0x7565, 0x5289, 0x6d41, 0x6e9c, 0x7409,
500 0x7559, 0x786b, 0x7c92, 0x9686, 0x7adc, 0x9f8d, 0x4fb6, 0x616e,
501 0x65c5, 0x865c, 0x4e86, 0x4eae, 0x50da, 0x4e21, 0x51cc, 0x5bee,
502 0x6599, 0x6881, 0x6dbc, 0x731f, 0x7642, 0x77ad, 0x7alc, 0x7ce7,
503 0x826f, 0x8ad2, 0x907c, 0x91cf, 0x9675, 0x9818, 0x529b, 0x7dd1,
504 0x502b, 0x5398, 0x6797, 0x6dcb, 0x71d0, 0x7433, 0x81e8, 0x8f2a,
505 0x96a3, 0x9c57, 0x9e9f, 0x7460, 0x5841, 0x6d99, 0x7d2f, 0x985e,
506 0x4ee4, 0x4f36, 0x4f8b, 0x51b7, 0x52b1, 0x5dba, 0x601c, 0x73b2,
507 0x793c, 0x82d3, 0x9234, 0x96b7, 0x96f6, 0x970a, 0x9e97, 0x9f62,
508 0x66a6, 0x6b74, 0x5217, 0x52a3, 0x70c8, 0x88c2, 0x5ec9, 0x604b,
509 0x6190, 0x6f23, 0x7149, 0x7c3e, 0x7df4, 0x806f,
510 /* 0x4f */
511 0x84ee, 0x9023, 0x932c, 0x5442, 0x9b6f, 0x6ad3, 0x7089, 0x8cc2,
512 0x8def, 0x9732, 0x52b4, 0x5a41, 0x5eca, 0x5f04, 0x6717, 0x697c,
513 0x6994, 0x6d6a, 0x6f0f, 0x7262, 0x72fc, 0x7bed, 0x8001, 0x807e,
514 0x874b, 0x90ce, 0x916d, 0x9e93, 0x7984, 0x808b, 0x9332, 0x8ad6,
515 0x502d, 0x548c, 0x8a71, 0x6b6a, 0x8cc4, 0x8107, 0x60d1, 0x67a0,
516 0x9df2, 0x4e99, 0x4e98, 0x9c10, 0x8a6b, 0x85c1, 0x8568, 0x6900,
517 0x6e7e, 0x7897, 0x8155, 0xffff, 0xffff, 0xffff, 0xffff,
518 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
519 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
520 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
521 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
522 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
523 /* 0x50 */
524 0x5f0c, 0x4e10, 0x4e15, 0x4e2a, 0x4e31, 0x4e36, 0x4e3c, 0x4e3f,
525 0x4e42, 0x4e56, 0x4e58, 0x4e82, 0x4e85, 0x8c6b, 0x4e8a, 0x8212,
526 0x5f0d, 0x4e8e, 0x4e9e, 0x4e9f, 0x4ea0, 0x4ea2, 0x4eb0, 0x4eb3,
527 0x4eb6, 0x4ece, 0x4ecd, 0x4ec4, 0x4ec6, 0x4ec2, 0x4ed7, 0x4ede,
528 0x4eed, 0x4edf, 0x4ef7, 0x4f09, 0x4f5a, 0x4f30, 0x4f5b, 0x4f5d,
529 0x4f57, 0x4f47, 0x4f76, 0x4f88, 0x4f8f, 0x4f98, 0x4f7b, 0x4f69,
530 0x4f70, 0x4f91, 0x4f6f, 0x4f86, 0x4f96, 0x5118, 0x4fd4, 0x4fdf,
531 0x4fce, 0x4fd8, 0x4fdb, 0x4fd1, 0x4fda, 0x4fd0, 0x4fe4, 0x4fe5,
532 0x501a, 0x5028, 0x5014, 0x502a, 0x5025, 0x5005, 0x4ffc, 0x4ff6,
533 0x5021, 0x5029, 0x502c, 0x4ffe, 0x4fef, 0x5011, 0x5006, 0x5043,
534 0x5047, 0x6703, 0x5055, 0x5050, 0x5048, 0x505a, 0x5056, 0x506c,
535 0x5078, 0x5080, 0x509a, 0x5085, 0x50b4, 0x50b2,
536 /* 0x51 */
537 0x50c9, 0x50ca, 0x50b3, 0x50c2, 0x50d6, 0x50de, 0x50e5, 0x50ed,
538 0x50e3, 0x50ee, 0x50f9, 0x50f5, 0x5109, 0x5101, 0x5102, 0x5116,
539 0x5115, 0x5114, 0x511a, 0x5121, 0x513a, 0x5137, 0x513c, 0x513b,
540 0x513f, 0x5140, 0x5152, 0x514c, 0x5154, 0x5162, 0x7af8, 0x5169,
541 0x516a, 0x516e, 0x5180, 0x5182, 0x56d8, 0x518c, 0x5189, 0x518f,
542 0x5191, 0x5193, 0x5195, 0x5196, 0x51a4, 0x51a6, 0x51a2, 0x51a9,
543 0x51aa, 0x51ab, 0x51b3, 0x51b1, 0x51b2, 0x51b0, 0x51b5, 0x51bd,
544 0x51c5, 0x51c9, 0x51db, 0x51e0, 0x8655, 0x51e9, 0x51ed, 0x51f0,
545 0x51f5, 0x51fe, 0x5204, 0x520b, 0x5214, 0x520e, 0x5227, 0x522a,
546 0x522e, 0x5233, 0x5239, 0x524f, 0x5244, 0x524b, 0x524c, 0x525e,
547 0x5254, 0x526a, 0x5274, 0x5269, 0x5273, 0x527f, 0x527d, 0x528d,
548 0x5294, 0x5292, 0x5271, 0x5288, 0x5291, 0x8fa8,
549 /* 0x52 */
550 0x8fa7, 0x52ac, 0x52ad, 0x52bc, 0x52b5, 0x52c1, 0x52cd, 0x52d7,
551 0x52de, 0x52e3, 0x52e6, 0x98ed, 0x52e0, 0x52f3, 0x52f5, 0x52f8,
552 0x52f9, 0x5306, 0x5308, 0x7538, 0x530d, 0x5310, 0x530f, 0x5315,
553 0x531a, 0x5323, 0x532f, 0x5331, 0x5333, 0x5338, 0x5340, 0x5346,
554 0x5345, 0x4e17, 0x5349, 0x534d, 0x51d6, 0x535e, 0x5369, 0x536e,
555 0x5918, 0x537b, 0x5377, 0x5382, 0x5396, 0x53a0, 0x53a6, 0x53a5,
556 0x53ae, 0x53b0, 0x53b6, 0x53c3, 0x7c12, 0x96d9, 0x53df, 0x66fc,
557 0x71ee, 0x53ee, 0x53e8, 0x53ed, 0x53fa, 0x5401, 0x543d, 0x5440,
558 0x542c, 0x542d, 0x543c, 0x542e, 0x5436, 0x5429, 0x541d, 0x544e,
559 0x548f, 0x5475, 0x548e, 0x545f, 0x5471, 0x5477, 0x5470, 0x5492,
560 0x547b, 0x5480, 0x5476, 0x5484, 0x5490, 0x5486, 0x54c7, 0x54a2,
561 0x54b8, 0x54a5, 0x54ac, 0x54c4, 0x54c8, 0x54a8,
562 /* 0x53 */
563 0x54ab, 0x54c2, 0x54a4, 0x54be, 0x54bc, 0x54d8, 0x54e5, 0x54e6,
564 0x550f, 0x5514, 0x54fd, 0x54ee, 0x54ed, 0x54fa, 0x54e2, 0x5539,
565 0x5540, 0x5563, 0x554c, 0x552e, 0x555c, 0x5545, 0x5556, 0x5557,
566 0x5538, 0x5533, 0x555d, 0x5599, 0x5580, 0x54af, 0x558a, 0x559f,
567 0x557b, 0x557e, 0x5598, 0x559e, 0x55ae, 0x557c, 0x5583, 0x55a9,
568 0x5587, 0x55a8, 0x55da, 0x55c5, 0x55df, 0x55c4, 0x55dc, 0x55e4,
569 0x55d4, 0x5614, 0x55f7, 0x5616, 0x55fe, 0x55fd, 0x561b, 0x55f9,
570 0x564e, 0x5650, 0x71df, 0x5634, 0x5636, 0x5632, 0x5638, 0x566b,
571 0x5664, 0x562f, 0x566c, 0x566a, 0x5686, 0x5680, 0x568a, 0x56a0,
572 0x5694, 0x568f, 0x56a5, 0x56ae, 0x56b6, 0x56b4, 0x56c2, 0x56bc,
573 0x56c1, 0x56c3, 0x56c0, 0x56c8, 0x56ce, 0x56d1, 0x56d3, 0x56d7,
574 0x56ee, 0x56f9, 0x5700, 0x56ff, 0x5704, 0x5709,
575 /* 0x54 */
576 0x5708, 0x570b, 0x570d, 0x5713, 0x5718, 0x5716, 0x55c7, 0x571c,
577 0x5726, 0x5737, 0x5738, 0x574e, 0x573b, 0x5740, 0x574f, 0x5769,
578 0x57c0, 0x5788, 0x5761, 0x577f, 0x5789, 0x5793, 0x57a0, 0x57b3,
579 0x57a4, 0x57aa, 0x57b0, 0x57c3, 0x57c6, 0x57d4, 0x57d2, 0x57d3,
580 0x580a, 0x57d6, 0x57e3, 0x580b, 0x5819, 0x581d, 0x5872, 0x5821,
```

```
581 0x5862, 0x584b, 0x5870, 0x6bc0, 0x5852, 0x583d, 0x5879, 0x5885,  
582 0x58b9, 0x589f, 0x58ab, 0x58ba, 0x58de, 0x58bb, 0x58b8, 0x58ae,  
583 0x58c5, 0x58d3, 0x58d3, 0x58d1, 0x58d7, 0x58d9, 0x58d8, 0x58e5, 0x58dc,  
584 0x58e4, 0x58df, 0x58ef, 0x58fa, 0x58f9, 0x58fb, 0x58fc, 0x58fd,  
585 0x5902, 0x590a, 0x5910, 0x591b, 0x68a6, 0x5925, 0x592c, 0x592d,  
586 0x5932, 0x5938, 0x593e, 0x7ad2, 0x5955, 0x5950, 0x594e, 0x595a,  
587 0x5958, 0x5962, 0x5960, 0x5967, 0x596c, 0x5969,  
588 /* 0x55 */  
589 0x5978, 0x5981, 0x599d, 0x4f5e, 0x4fab, 0x59a3, 0x59b2, 0x59c6,  
590 0x59e8, 0x59dc, 0x598d, 0x59d9, 0x59da, 0x5a25, 0x5a1f, 0x5a11,  
591 0x5a1c, 0x5a09, 0x5a1a, 0x5a40, 0x5a6c, 0x5a49, 0x5a35, 0x5a36,  
592 0x5a62, 0x5a6a, 0x5a9a, 0x5abc, 0x5abe, 0x5ach, 0x5ac2, 0x5abd,  
593 0x5ae3, 0x5ad7, 0x5ae6, 0x5ae9, 0x5ad6, 0x5afa, 0x5afb, 0x5b0c,  
594 0x5b0b, 0x5b16, 0x5b32, 0x5ad0, 0x5b2a, 0x5b36, 0x5b3e, 0x5b43,  
595 0x5b45, 0x5b40, 0x5b51, 0x5b55, 0x5b5a, 0x5b5b, 0x5b65, 0x5b69,  
596 0x5b70, 0x5b73, 0x5b75, 0x5b78, 0x6588, 0x5b7a, 0x5b80, 0x5b83,  
597 0x5ba6, 0x5bb8, 0x5bc3, 0x5bc7, 0x5bc9, 0x5bd4, 0x5bd0, 0x5be4,  
598 0x5be6, 0x5be2, 0x5bde, 0x5be5, 0x5beb, 0x5bf0, 0x5bf6, 0x5bf3,  
599 0x5c05, 0x5c07, 0x5c08, 0x5c0d, 0x5c13, 0x5c20, 0x5c22, 0x5c28,  
600 0x5c38, 0x5c39, 0x5c41, 0x5c46, 0x5c4e, 0x5c53,  
601 /* 0x56 */  
602 0x5c50, 0x5c4f, 0x5b71, 0x5c6c, 0x5c6e, 0x4e62, 0x5c76, 0x5c79,  
603 0x5c8c, 0x5c91, 0x5c94, 0x599b, 0x5cab, 0x5cbb, 0x5cb6, 0x5cbc,  
604 0x5cb7, 0x5cc5, 0x5ccb, 0x5cbe, 0x5cc7, 0x5cd9, 0x5ce9, 0x5cfd, 0x5cfa,  
605 0x5ced, 0x5d8c, 0x5cea, 0x5d0b, 0x5d15, 0x5d17, 0x5d5c, 0x5d1f,  
606 0x5d1b, 0x5d11, 0x5d14, 0x5d22, 0x5d1a, 0x5d19, 0x5d18, 0x5d4c,  
607 0x5d52, 0x5d4e, 0x5d4b, 0x5d4b, 0x5d6c, 0x5d73, 0x5d76, 0x5d87, 0x5d84,  
608 0x5d82, 0x5da2, 0x5d9d, 0x5dac, 0x5dae, 0x5dbd, 0x5d90, 0x5db7,  
609 0x5dbc, 0x5dc9, 0x5dcd, 0x5dd3, 0x5dd2, 0x5dd6, 0x5ddb, 0x5deb,  
610 0x5df2, 0x5df5, 0x5e0b, 0x5e1a, 0x5e19, 0x5e11, 0x5e1b, 0x5e36,  
611 0x5e37, 0x5e44, 0x5e43, 0x5e40, 0x5e4e, 0x5e57, 0x5e54, 0x5e5f,  
612 0x5e62, 0x5e64, 0x5e47, 0x5e75, 0x5e76, 0x5e7a, 0x9ebc, 0x5e7f,  
613 0x5ea0, 0x5ec1, 0x5ec2, 0x5ec8, 0x5ed0, 0x5ecf,  
614 /* 0x57 */  
615 0x5ed6, 0x5ee3, 0x5edd, 0x5eda, 0x5edb, 0x5ee2, 0x5ee1, 0x5ee8,  
616 0x5ee9, 0x5eec, 0x5ef1, 0x5ef3, 0x5ef0, 0x5ef4, 0x5ef8, 0x5efe,  
617 0x5f03, 0x5f09, 0x5f5d, 0x5f5c, 0x5f0b, 0x5f11, 0x5f16, 0x5f29,  
618 0x5f2a, 0x5f38, 0x5f41, 0x5f48, 0x5f4c, 0x5f4e, 0x5f2f, 0x5f51,  
619 0x5f56, 0x5f57, 0x5f59, 0x5f61, 0x5f6d, 0x5f73, 0x5f77, 0x5f83,  
620 0x5f82, 0x5f7f, 0x5f8a, 0x5f88, 0x5f91, 0x5f87, 0x5f9e, 0x5f99,  
621 0x5f98, 0x5fa0, 0x5fa8, 0x5fad, 0x5fbc, 0x5fd6, 0x5ffb, 0x5fe4,  
622 0x5ff8, 0x5ff1, 0x5ffd, 0x60b3, 0x5fff, 0x6021, 0x6060, 0x6019,  
623 0x6010, 0x6029, 0x600e, 0x6031, 0x601b, 0x6015, 0x602b, 0x6026,  
624 0x600f, 0x603a, 0x605a, 0x6041, 0x606a, 0x6077, 0x605f, 0x604a,  
625 0x6046, 0x604d, 0x6063, 0x6043, 0x6064, 0x6042, 0x606c, 0x606b,  
626 0x6059, 0x6081, 0x608d, 0x60e7, 0x6083, 0x609a,  
627 /* 0x58 */  
628 0x6084, 0x609b, 0x6096, 0x6097, 0x6092, 0x60a7, 0x608b, 0x60e1,  
629 0x60b8, 0x60e0, 0x60d3, 0x60b4, 0x5ff0, 0x60bd, 0x60c6, 0x60b5,  
630 0x60d8, 0x614d, 0x6115, 0x6106, 0x60f6, 0x60f7, 0x6100, 0x60f4,  
631 0x60fa, 0x6103, 0x6121, 0x60fb, 0x60f1, 0x610d, 0x610e, 0x6147,  
632 0x613e, 0x6128, 0x6127, 0x614a, 0x613f, 0x613c, 0x612c, 0x6134,  
633 0x613d, 0x6142, 0x6144, 0x6173, 0x6177, 0x6158, 0x6159, 0x615a,  
634 0x616b, 0x6174, 0x616f, 0x6165, 0x6171, 0x615f, 0x615d, 0x6153,  
635 0x6175, 0x6199, 0x6196, 0x6187, 0x61ac, 0x6194, 0x619a, 0x618a,  
636 0x6191, 0x61ab, 0x61ae, 0x61cc, 0x61ca, 0x61c9, 0x61f7, 0x61c8,  
637 0x61c3, 0x61c6, 0x61ba, 0x61cb, 0x7f79, 0x61cd, 0x61e6, 0x61e3,  
638 0x61f6, 0x61fa, 0x61f4, 0x61ff, 0x61fd, 0x61fc, 0x61fe, 0x6200,  
639 0x6208, 0x6209, 0x620d, 0x620c, 0x6214, 0x621b,  
640 /* 0x59 */  
641 0x621e, 0x6221, 0x622a, 0x622e, 0x6230, 0x6232, 0x6233, 0x6241,  
642 0x624e, 0x625e, 0x6263, 0x625b, 0x6260, 0x6268, 0x627c, 0x6282,  
643 0x6289, 0x627e, 0x6292, 0x6293, 0x6296, 0x62d4, 0x6283, 0x6294,  
644 0x62d7, 0x62d1, 0x62bb, 0x62cf, 0x62ff, 0x62c6, 0x64d4, 0x62c8,  
645 0x62dc, 0x62cc, 0x62ca, 0x62c2, 0x62c7, 0x629b, 0x62c9, 0x630c,  
646 0x62ee, 0x62f1, 0x6327, 0x6302, 0x6308, 0x62ef, 0x62f5, 0x6350,  
647 0x633e, 0x634d, 0x641c, 0x634f, 0x6396, 0x638e, 0x6380, 0x63ab,  
648 0x6376, 0x63a3, 0x638f, 0x6389, 0x639f, 0x63b5, 0x636b, 0x6369,  
649 0x63be, 0x63e9, 0x63c0, 0x63c6, 0x63e3, 0x63c9, 0x63d2, 0x63f6,  
650 0x63c4, 0x6416, 0x6434, 0x6406, 0x6413, 0x6426, 0x6436, 0x651d,  
651 0x6417, 0x6428, 0x640f, 0x6467, 0x646f, 0x6476, 0x644e, 0x652a,  
652 0x6495, 0x6493, 0x64a5, 0x64a9, 0x6488, 0x64bc,  
653 /* 0x5a */  
654 0x64da, 0x64d2, 0x64c5, 0x64c7, 0x64bb, 0x64d8, 0x64c2, 0x64f1,  
655 0x64e7, 0x8209, 0x64e0, 0x64e1, 0x62ac, 0x64e3, 0x64ef, 0x652c,  
656 0x64f6, 0x64f4, 0x64f2, 0x64fa, 0x6500, 0x64fd, 0x6518, 0x651c,  
657 0x6505, 0x6524, 0x6523, 0x652b, 0x6534, 0x6535, 0x6537, 0x6536,  
658 0x6538, 0x754b, 0x6548, 0x6556, 0x6555, 0x654d, 0x6558, 0x655e,  
659 0x655d, 0x6572, 0x6578, 0x6582, 0x6583, 0x8b8a, 0x659b, 0x659f,  
660 0x65ab, 0x65b7, 0x65c3, 0x65c6, 0x65c1, 0x65c4, 0x65cc, 0x65d2,  
661 0x65db, 0x65d9, 0x65e0, 0x65e1, 0x65f1, 0x6772, 0x660a, 0x6603,  
662 0x65fb, 0x6773, 0x6635, 0x6636, 0x6634, 0x661c, 0x664f, 0x6644,  
663 0x6649, 0x6641, 0x665e, 0x665d, 0x6664, 0x6667, 0x6668, 0x666f,  
664 0x6662, 0x6670, 0x6683, 0x6688, 0x668e, 0x6689, 0x6684, 0x6698,  
665 0x669d, 0x66c1, 0x66b9, 0x66c9, 0x66be, 0x66bc,  
666 /* 0x5b */  
667 0x66c4, 0x66b8, 0x66d6, 0x66da, 0x66e0, 0x663f, 0x66e6, 0x66e9,
```

```

668 0x66f0, 0x66f5, 0x66f7, 0x670f, 0x6716, 0x671e, 0x6726, 0x6727,
669 0x9738, 0x672e, 0x673f, 0x6736, 0x6741, 0x6738, 0x6737, 0x6746,
670 0x675e, 0x6760, 0x6760, 0x6759, 0x6763, 0x6764, 0x6789, 0x6770, 0x67a9,
671 0x677c, 0x676a, 0x678c, 0x678b, 0x67a6, 0x67a1, 0x6785, 0x67b7,
672 0x67ef, 0x67b4, 0x67ec, 0x67b3, 0x67e9, 0x67b8, 0x67e4, 0x67de,
673 0x67dd, 0x67e2, 0x67ee, 0x67b9, 0x67ce, 0x67c6, 0x67e7, 0x6a9c,
674 0x681e, 0x6846, 0x6829, 0x6840, 0x684d, 0x6832, 0x684e, 0x68b3,
675 0x682b, 0x6859, 0x6863, 0x6877, 0x687f, 0x689f, 0x688f, 0x68ad,
676 0x6894, 0x689d, 0x689b, 0x689b, 0x6883, 0x6aae, 0x68b9, 0x6874, 0x68b5,
677 0x68a0, 0x68ba, 0x690f, 0x688d, 0x687e, 0x6901, 0x68ca, 0x6908,
678 0x68d8, 0x6922, 0x6926, 0x68e1, 0x690c, 0x68cd,
679 /* 0x5c */
680 0x68d4, 0x68e7, 0x68d5, 0x6936, 0x6912, 0x6904, 0x68d7, 0x68e3,
681 0x6925, 0x68f9, 0x68e0, 0x68ef, 0x6928, 0x692a, 0x691a, 0x6923,
682 0x6921, 0x68c6, 0x6979, 0x6977, 0x695c, 0x6978, 0x696b, 0x6954,
683 0x697e, 0x696e, 0x6939, 0x6974, 0x693d, 0x6959, 0x6930, 0x6961,
684 0x695e, 0x695d, 0x6981, 0x696a, 0x69b2, 0x69ae, 0x69d0, 0x69bf,
685 0x69c1, 0x69d3, 0x69be, 0x69ce, 0x5be8, 0x69ca, 0x69dd, 0x69bb,
686 0x69c3, 0x69a7, 0x6a2e, 0x6991, 0x69a0, 0x699c, 0x6995, 0x69b4,
687 0x69de, 0x69e8, 0x6a02, 0x6a1b, 0x69ff, 0x6b0a, 0x69f9, 0x69f2,
688 0x69e7, 0x6a05, 0x69b1, 0x6a1e, 0x69ed, 0x6a14, 0x69eb, 0x6a0a,
689 0x6a12, 0x6ac1, 0x6a23, 0x6a13, 0x6a44, 0x6a0c, 0x6a72, 0x6a36,
690 0x6a78, 0x6a47, 0x6a62, 0x6a59, 0x6a66, 0x6a48, 0x6a38, 0x6a22,
691 0x6a90, 0x6a8d, 0x6aa0, 0x6a84, 0x6aa2, 0x6aa3,
692 /* 0x5d */
693 0x6a97, 0x8617, 0x6abb, 0x6ac3, 0x6ac2, 0x6ab8, 0x6ab3, 0x6aac,
694 0x6ade, 0x6ad1, 0x6adf, 0x6aaa, 0x6ada, 0x6aea, 0x6afb, 0x6b05,
695 0x8616, 0x6afa, 0x6b12, 0x6b16, 0x9b31, 0x6b1f, 0x6b38, 0x6b37,
696 0x76dc, 0x6b39, 0x98ee, 0x6b47, 0x6b43, 0x6b49, 0x6b50, 0x6b59,
697 0x6b54, 0x6b5b, 0x6b5f, 0x6b61, 0x6b78, 0x6b79, 0x6b7f, 0x6b80,
698 0x6b84, 0x6b83, 0x6b8d, 0x6b98, 0x6b95, 0x6b9e, 0x6ba4, 0x6baa,
699 0x6bab, 0x6baf, 0x6bb2, 0x6bb1, 0x6bb3, 0x6bb7, 0x6bbc, 0x6bc6,
700 0x6bcb, 0x6bd3, 0x6bdf, 0x6bec, 0x6beb, 0x6bf3, 0x6bef, 0x9ebe,
701 0x6c08, 0x6c13, 0x6c14, 0x6c1b, 0x6c24, 0x6c23, 0x6c5e, 0x6c55,
702 0x6c62, 0x6c6a, 0x6c82, 0x6c8d, 0x6c9a, 0x6c81, 0x6c9b, 0x6c7e,
703 0x6c68, 0x6c73, 0x6c92, 0x6c90, 0x6cc4, 0x6cfl, 0x6cd3, 0x6cbd,
704 0x6cd7, 0x6cc5, 0x6cdd, 0x6cae, 0x6cb1, 0x6cbe,
705 /* 0x5e */
706 0x6cba, 0x6cdb, 0x6cef, 0x6cd9, 0x6cea, 0x6d1f, 0x884d, 0x6d36,
707 0x6d2b, 0x6d3d, 0x6d38, 0x6d19, 0x6d35, 0x6d33, 0x6d12, 0x6d0c,
708 0x6d63, 0x6d93, 0x6d64, 0x6d5a, 0x6d79, 0x6d59, 0x6d8e, 0x6d95,
709 0x6fe4, 0x6d85, 0x6df9, 0x6e15, 0x6e0a, 0x6db5, 0x6dc7, 0x6de6,
710 0x6db8, 0x6dc6, 0x6dec, 0x6dde, 0x6dcc, 0x6de8, 0x6dd2, 0x6dc5,
711 0x6dfa, 0x6dd9, 0x6de4, 0x6dd5, 0x6dea, 0x6dee, 0x6e2d, 0x6e6e,
712 0x6e2e, 0x6e19, 0x6e72, 0x6e5f, 0x6e3e, 0x6e23, 0x6e6b, 0x6e2b,
713 0x6e76, 0x6e4d, 0x6e1f, 0x6e43, 0x6e3a, 0x6e4e, 0x6e24, 0x6eff,
714 0x6e1d, 0x6e38, 0x6e82, 0x6eaa, 0x6e98, 0x6ec9, 0x6eb7, 0x6ed3,
715 0x6ebd, 0x6eaf, 0x6ec4, 0x6eb2, 0x6ed4, 0x6ed5, 0x6e8f, 0x6ea5,
716 0x6ec2, 0x6e9f, 0x6f41, 0x6f11, 0x704c, 0x6eec, 0x6ef8, 0x6efe,
717 0x6f3f, 0x6ef2, 0x6f31, 0x6eef, 0x6f32, 0x6ecc,
718 /* 0x5f */
719 0x6f3e, 0x6f13, 0x6ef7, 0x6f86, 0x6f7a, 0x6f78, 0x6f81, 0x6f80,
720 0x6f6f, 0x6f5b, 0x6ff3, 0x6f6d, 0x6f82, 0x6f7c, 0x6f58, 0x6f8e,
721 0x6f91, 0x6fc2, 0x6ff6, 0x6fb3, 0x6fa3, 0x6fa1, 0x6fa4, 0x6fb9,
722 0x6fc6, 0x6faa, 0x6fdf, 0x6fd5, 0x6fec, 0x6fd4, 0x6fd8, 0x6ff1,
723 0x6fee, 0x6fdb, 0x7009, 0x700b, 0x6ffa, 0x7011, 0x7001, 0x700f,
724 0x6ffe, 0x701b, 0x701a, 0x6ff7, 0x701d, 0x7018, 0x701f, 0x7030,
725 0x703e, 0x7032, 0x7051, 0x7063, 0x7099, 0x7092, 0x70af, 0x70f1,
726 0x70ac, 0x70b8, 0x70b3, 0x70ae, 0x70df, 0x70cb, 0x70dd, 0x70d9,
727 0x7109, 0x70fd, 0x711c, 0x7119, 0x7165, 0x7155, 0x7188, 0x7166,
728 0x7162, 0x714c, 0x7156, 0x716c, 0x718f, 0x71fb, 0x7184, 0x7195,
729 0x71a8, 0x71ac, 0x71d7, 0x71b9, 0x71be, 0x71d2, 0x71c9, 0x71d4,
730 0x71ce, 0x71e0, 0x71ec, 0x71e7, 0x71f5, 0x71fc,
731 /* 0x60 */
732 0x71f9, 0x71ff, 0x720d, 0x7210, 0x721b, 0x7228, 0x722d, 0x722c,
733 0x7230, 0x7232, 0x723b, 0x723c, 0x723f, 0x7240, 0x7246, 0x724b,
734 0x7258, 0x7274, 0x727e, 0x7282, 0x7281, 0x7287, 0x7292, 0x7296,
735 0x72a2, 0x72a7, 0x72b9, 0x72b2, 0x72c3, 0x72c6, 0x72c4, 0x72ce,
736 0x72d2, 0x72e2, 0x72e0, 0x72e1, 0x72f9, 0x72f7, 0x500f, 0x7317,
737 0x730a, 0x731c, 0x7316, 0x731d, 0x7334, 0x732f, 0x7329, 0x7325,
738 0x733e, 0x734e, 0x734f, 0x9ed8, 0x7357, 0x736a, 0x7368, 0x7370,
739 0x7378, 0x7375, 0x737b, 0x737a, 0x73c8, 0x73b3, 0x73ce, 0x73bb,
740 0x73c0, 0x73e5, 0x73ee, 0x73de, 0x74a2, 0x7405, 0x746f, 0x7425,
741 0x73f8, 0x7432, 0x743a, 0x7455, 0x743f, 0x745f, 0x7459, 0x7441,
742 0x745c, 0x7469, 0x7470, 0x7463, 0x746a, 0x7476, 0x747e, 0x748b,
743 0x749e, 0x74a7, 0x74ca, 0x74cf, 0x74d4, 0x73f1,
744 /* 0x61 */
745 0x74e0, 0x74e3, 0x74e7, 0x74e9, 0x74ee, 0x74f2, 0x74f0, 0x74f1,
746 0x74f8, 0x74f7, 0x7504, 0x7503, 0x7505, 0x750c, 0x750e, 0x750d,
747 0x7515, 0x7513, 0x751e, 0x7526, 0x752c, 0x753c, 0x7544, 0x754d,
748 0x754a, 0x7549, 0x755b, 0x7546, 0x755a, 0x7569, 0x7564, 0x7567,
749 0x756b, 0x756d, 0x7578, 0x7576, 0x7586, 0x7587, 0x7574, 0x758a,
750 0x7589, 0x7582, 0x7594, 0x759d, 0x759d, 0x75a5, 0x75a3, 0x75c2,
751 0x75b3, 0x75c3, 0x75b5, 0x75bd, 0x75b8, 0x75bc, 0x75b1, 0x75cd,
752 0x75ca, 0x75d2, 0x75d9, 0x75e3, 0x75de, 0x75fe, 0x75ff, 0x75fc,
753 0x7601, 0x75f0, 0x75fa, 0x75f2, 0x75f3, 0x760b, 0x760d, 0x7609,
754 0x761f, 0x7627, 0x7620, 0x7621, 0x7622, 0x7624, 0x7634, 0x7630,

```



```
929 0x93b9, 0x93d6, 0x93d7, 0x93e8, 0x93e5, 0x93d8, 0x93c3, 0x93dd,
930 0x93d0, 0x93c8, 0x93e4, 0x941a, 0x9414, 0x9413, 0x9403, 0x9407,
931 0x9410, 0x9436, 0x942b, 0x942b, 0x9435, 0x9421, 0x943a, 0x9441, 0x9452,
932 0x9444, 0x945b, 0x9460, 0x9462, 0x945e, 0x946a, 0x9229, 0x9470,
933 0x9475, 0x9477, 0x947d, 0x945a, 0x947c, 0x947e, 0x9481, 0x947f,
934 0x9582, 0x9587, 0x958a, 0x958a, 0x9594, 0x9596, 0x9598, 0x9599, 0x95a0,
935 0x95a8, 0x95a7, 0x95ad, 0x95bc, 0x95bb, 0x95b9, 0x95be, 0x95ca,
936 0x6ff6, 0x95c3, 0x95cd, 0x95cc, 0x95d5, 0x95d4, 0x95d6, 0x95dc,
937 0x95e1, 0x95e5, 0x95e2, 0x9621, 0x9628, 0x962e, 0x962f, 0x9642,
938 0x964c, 0x964f, 0x964b, 0x9677, 0x965c, 0x965e,
939 /* 0x70 */
940 0x965d, 0x965f, 0x9666, 0x9672, 0x966c, 0x968d, 0x9698, 0x9695,
941 0x9697, 0x96aa, 0x96a7, 0x96b1, 0x96b2, 0x96b0, 0x96b4, 0x96b6,
942 0x96b8, 0x96b9, 0x96ce, 0x96cb, 0x96c9, 0x96cd, 0x894d, 0x96dc,
943 0x970d, 0x96d5, 0x96f9, 0x9704, 0x9706, 0x9708, 0x9713, 0x970e,
944 0x9711, 0x970f, 0x9716, 0x9719, 0x9724, 0x972a, 0x9730, 0x9739,
945 0x973d, 0x973e, 0x9744, 0x9746, 0x9748, 0x9742, 0x9749, 0x975c,
946 0x9760, 0x9764, 0x9766, 0x9768, 0x52d2, 0x976b, 0x9771, 0x9779,
947 0x9785, 0x977c, 0x9781, 0x977a, 0x9786, 0x978b, 0x978f, 0x9790,
948 0x979c, 0x97a8, 0x97a6, 0x97a3, 0x97b3, 0x97b4, 0x97c3, 0x97c6,
949 0x97c8, 0x97cb, 0x97dc, 0x97ed, 0x97f2, 0x97ad, 0x97f6,
950 0x97f5, 0x980f, 0x980c, 0x9838, 0x9824, 0x9821, 0x9837, 0x983d,
951 0x9846, 0x984f, 0x984b, 0x986b, 0x986f, 0x9870,
952 /* 0x71 */
953 0x9871, 0x9874, 0x9873, 0x98aa, 0x98af, 0x98b1, 0x98b6, 0x98c4,
954 0x98c3, 0x98c6, 0x98e9, 0x98eb, 0x9903, 0x9909, 0x9912, 0x9914,
955 0x9918, 0x9921, 0x991d, 0x991e, 0x9924, 0x9920, 0x992c, 0x992e,
956 0x993d, 0x993e, 0x9942, 0x9949, 0x9945, 0x9950, 0x994b, 0x9951,
957 0x9952, 0x994c, 0x9955, 0x9997, 0x9998, 0x99a5, 0x99ad, 0x99ae,
958 0x99bc, 0x99df, 0x99db, 0x99dd, 0x99d8, 0x99d1, 0x99de, 0x99ee,
959 0x99f1, 0x99f2, 0x99fb, 0x99f8, 0x9a01, 0x9a0f, 0x9a05, 0x99e2,
960 0x9a19, 0x9a2b, 0x9a37, 0x9a45, 0x9a42, 0x9a40, 0x9a43, 0x9a3e,
961 0x9a55, 0x9a4d, 0x9a5b, 0x9a57, 0x9a5f, 0x9a62, 0x9a65, 0x9a64,
962 0x9a69, 0x9a6b, 0x9a6a, 0x9aad, 0x9ab0, 0x9abc, 0x9ac0, 0x9acf,
963 0x9ad1, 0x9ad3, 0x9ad4, 0x9ade, 0x9adf, 0x9ae2, 0x9ae3, 0x9ae6,
964 0x9aef, 0x9aeb, 0x9aee, 0x9af4, 0x9af1, 0x9af7,
965 /* 0x72 */
966 0x9afb, 0x9b06, 0x9b18, 0x9b1a, 0x9b1f, 0x9b22, 0x9b23, 0x9b25,
967 0x9b27, 0x9b28, 0x9b29, 0x9b2a, 0x9b2e, 0x9b2f, 0x9b32, 0x9b44,
968 0x9b43, 0x9b4f, 0x9b4d, 0x9b4e, 0x9b51, 0x9b58, 0x9b74, 0x9b93,
969 0x9b83, 0x9b91, 0x9b96, 0x9b97, 0x9b9f, 0x9ba0, 0x9ba8, 0x9bb4,
970 0x9bc0, 0x9bca, 0x9bb9, 0x9bc6, 0x9bcf, 0x9bd1, 0x9bd2, 0x9be3,
971 0x9be2, 0x9be4, 0x9bd4, 0x9bel, 0x9c3a, 0x9bf2, 0x9bf1, 0x9bf0,
972 0x9c15, 0x9c14, 0x9c09, 0x9c13, 0x9c0c, 0x9c06, 0x9c08, 0x9c12,
973 0x9c0a, 0x9c04, 0x9c2e, 0x9c1b, 0x9c25, 0x9c24, 0x9c21, 0x9c30,
974 0x9c47, 0x9c32, 0x9c46, 0x9c3e, 0x9c5a, 0x9c60, 0x9c67, 0x9c76,
975 0x9c78, 0x9ce7, 0x9cec, 0x9cf0, 0x9d09, 0x9d08, 0x9ceb, 0x9d03,
976 0x9d06, 0x9d2a, 0x9d26, 0x9daf, 0x9d23, 0x9d1f, 0x9d44, 0x9d15,
977 0x9d12, 0x9d41, 0x9d3f, 0x9d3e, 0x9d46, 0x9d48,
978 /* 0x73 */
979 0x9d5d, 0x9d5e, 0x9d64, 0x9d51, 0x9d50, 0x9d59, 0x9d72, 0x9d89,
980 0x9d87, 0x9dab, 0x9d6f, 0x9d7a, 0x9d9a, 0x9da4, 0x9da9, 0x9db2,
981 0x9dc4, 0x9dc1, 0x9dbb, 0x9db8, 0x9dba, 0x9dc6, 0x9dcf, 0x9dc2,
982 0x9dd9, 0x9dd3, 0x9df8, 0x9de6, 0x9ded, 0x9def, 0x9dfd, 0x9e1a,
983 0x9e1b, 0x9e1e, 0x9e75, 0x9e79, 0x9e7d, 0x9e81, 0x9e88, 0x9e8b,
984 0x9e8c, 0x9e92, 0x9e95, 0x9e91, 0x9e9d, 0x9ea5, 0x9ea9, 0x9eb8,
985 0x9eaa, 0x9ead, 0x9761, 0x9ecc, 0x9ece, 0x9ecf, 0x9ed0, 0x9ed4,
986 0x9edc, 0x9ede, 0x9edd, 0x9ee0, 0x9ee5, 0x9ee8, 0x9eef, 0x9ef4,
987 0x9ef6, 0x9ef7, 0x9ef9, 0x9efb, 0x9efc, 0x9efd, 0x9f07, 0x9f08,
988 0x76b7, 0x9f15, 0x9f21, 0x9f2c, 0x9f3e, 0x9f4a, 0x9f52, 0x9f54,
989 0x9f63, 0x9f5f, 0x9f60, 0x9f61, 0x9f66, 0x9f67, 0x9f6c, 0x9f6a,
990 0x9f77, 0x9f72, 0x9f76, 0x9f95, 0x9f9c, 0x9fa0,
991 /* 0x74 */
992 0x582f, 0x69c7, 0x9059, 0x7464, 0x51dc, 0x7199,
993 };
994
995 static int
996 jisx0208_mbtowc (conv_t conv, ucs4_t *pwc, const unsigned char *s, int n)
997 {
998     unsigned char c1 = (s[0] & 0x7F);
999     if ((c1 >= 0x21 && c1 <= 0x28) || (c1 >= 0x30 && c1 <= 0x74)) {
1000         if (n >= 2) {
1001             unsigned char c2 = (s[1] & 0x7F);
1002             if (c2 >= 0x21 && c2 < 0x7f) {
1003                 unsigned int i = 94 * (c1 - 0x21) + (c2 - 0x21);
1004                 unsigned short wc = 0xffffd;
1005                 if (i < 1410) {
1006                     if (i < 690)
1007                         wc = jisx0208_2uni_page21[i];
1008                     } else {
1009                         if (i < 7808)
1010                             wc = jisx0208_2uni_page30[i-1410];
1011                     }
1012                 } if (wc != 0xffffd) {
1013                     *pwc = (ucs4_t) wc;
1014                     return 2;
1015                 }
            }
        }
```

```
1016     }
1017     return RET_ILSEQ;
1018     }
1019     return RET_TOOFEW(0);
1020 }
1021 return RET_ILSEQ;
1022 }
1023 #endif /* NEED_TOWC */
1024
1025 #ifdef NEED_TOMB
1026 static const unsigned short jsx0208_2charset[6879] = {
1027 0x2140, 0x2171, 0x2172, 0x2178, 0x212f, 0x224c, 0x216b, 0x215e,
1028 0x212d, 0x2279, 0x215f, 0x2160, 0x2621, 0x2622, 0x2623, 0x2624,
1029 0x2625, 0x2626, 0x2627, 0x2628, 0x2629, 0x262a, 0x262b, 0x262c,
1030 0x262d, 0x262e, 0x262f, 0x2630, 0x2631, 0x2632, 0x2633, 0x2634,
1031 0x2635, 0x2636, 0x2637, 0x2638, 0x2641, 0x2642, 0x2643, 0x2644,
1032 0x2645, 0x2646, 0x2647, 0x2648, 0x2649, 0x264a, 0x264b, 0x264c,
1033 0x264d, 0x264e, 0x264f, 0x2650, 0x2651, 0x2652, 0x2653, 0x2654,
1034 0x2655, 0x2656, 0x2657, 0x2658, 0x2727, 0x2721, 0x2722, 0x2723,
1035 0x2724, 0x2725, 0x2726, 0x2728, 0x2729, 0x272a, 0x272b, 0x272c,
1036 0x272d, 0x272e, 0x272f, 0x2730, 0x2731, 0x2732, 0x2733, 0x2734,
1037 0x2735, 0x2736, 0x2737, 0x2738, 0x2739, 0x273a, 0x273b, 0x273c,
1038 0x273d, 0x273e, 0x273f, 0x2740, 0x2741, 0x2751, 0x2752, 0x2753,
1039 0x2754, 0x2755, 0x2756, 0x2757, 0x2758, 0x2759, 0x275a, 0x275b,
1040 0x275d, 0x275e, 0x275f, 0x2760, 0x2761, 0x2762, 0x2763, 0x2764,
1041 0x2765, 0x2766, 0x2767, 0x2768, 0x2769, 0x276a, 0x276b, 0x276c,
1042 0x276d, 0x276e, 0x276f, 0x2770, 0x2771, 0x2775, 0x213e, 0x213d,
1043 0x2142, 0x2146, 0x2147, 0x2148, 0x2149, 0x2277, 0x2278, 0x2145,
1044 0x2144, 0x2273, 0x216c, 0x216d, 0x2228, 0x216e, 0x2272, 0x222b,
1045 0x222c, 0x222a, 0x222d, 0x222d, 0x224d, 0x224e, 0x224f, 0x2250,
1046 0x2260, 0x223a, 0x223b, 0x215d, 0x2265, 0x2267, 0x2167, 0x225c,
1047 0x224a, 0x224b, 0x2241, 0x2240, 0x2269, 0x226a, 0x2168, 0x2268,
1048 0x2266, 0x2262, 0x2162, 0x2261, 0x2165, 0x2166, 0x2263, 0x2264,
1049 0x223e, 0x223f, 0x223c, 0x223d, 0x225d, 0x225e, 0x2821, 0x282c,
1050 0x2822, 0x282d, 0x2823, 0x282e, 0x2824, 0x282f, 0x2826, 0x2831,
1051 0x2825, 0x2830, 0x2827, 0x283c, 0x2837, 0x2832, 0x2829, 0x283e,
1052 0x2839, 0x2834, 0x2828, 0x2838, 0x283d, 0x2833, 0x282a, 0x283a,
1053 0x283f, 0x2835, 0x282b, 0x283b, 0x2840, 0x2836, 0x2223, 0x2222,
1054 0x2225, 0x2224, 0x2227, 0x2226, 0x2221, 0x217e, 0x217b, 0x217d,
1055 0x217c, 0x227e, 0x217a, 0x2179, 0x216a, 0x2169, 0x2276, 0x2275,
1056 0x2274, 0x2121, 0x2122, 0x2123, 0x2137, 0x2139, 0x213a, 0x213b,
1057 0x2152, 0x2153, 0x2154, 0x2155, 0x2156, 0x2157, 0x2158, 0x2159,
1058 0x215a, 0x215b, 0x2229, 0x222e, 0x214c, 0x214d, 0x2141, 0x2421,
1059 0x2422, 0x2423, 0x2424, 0x2425, 0x2426, 0x2427, 0x2428, 0x2429,
1060 0x242a, 0x242b, 0x242c, 0x242d, 0x242e, 0x242f, 0x2430, 0x2431,
1061 0x2432, 0x2433, 0x2434, 0x2435, 0x2436, 0x2437, 0x2438, 0x2439,
1062 0x243a, 0x243b, 0x243c, 0x243d, 0x243e, 0x243f, 0x2440, 0x2441,
1063 0x2442, 0x2443, 0x2444, 0x2445, 0x2446, 0x2447, 0x2448, 0x2449,
1064 0x244a, 0x244b, 0x244c, 0x244d, 0x244e, 0x244f, 0x2450, 0x2451,
1065 0x2452, 0x2453, 0x2454, 0x2455, 0x2456, 0x2457, 0x2458, 0x2459,
1066 0x245a, 0x245b, 0x245c, 0x245d, 0x245e, 0x245f, 0x2460, 0x2461,
1067 0x2462, 0x2463, 0x2464, 0x2465, 0x2466, 0x2467, 0x2468, 0x2469,
1068 0x246a, 0x246b, 0x246c, 0x246d, 0x246e, 0x246f, 0x2470, 0x2471,
1069 0x2472, 0x2473, 0x212b, 0x212c, 0x2135, 0x2136, 0x2521, 0x2522,
1070 0x2523, 0x2524, 0x2525, 0x2526, 0x2527, 0x2528, 0x2529, 0x252a,
1071 0x252b, 0x252c, 0x252d, 0x252e, 0x252f, 0x2530, 0x2531, 0x2532,
1072 0x2533, 0x2534, 0x2535, 0x2536, 0x2537, 0x2538, 0x2539, 0x253a,
1073 0x253b, 0x253c, 0x253d, 0x253e, 0x253f, 0x2540, 0x2541, 0x2542,
1074 0x2543, 0x2544, 0x2545, 0x2546, 0x2547, 0x2548, 0x2549, 0x254a,
1075 0x254b, 0x254c, 0x254d, 0x254e, 0x254f, 0x2550, 0x2551, 0x2552,
1076 0x2553, 0x2554, 0x2555, 0x2556, 0x2557, 0x2558, 0x2559, 0x255a,
1077 0x255b, 0x255c, 0x255d, 0x255e, 0x255f, 0x2560, 0x2561, 0x2562,
1078 0x2563, 0x2564, 0x2565, 0x2566, 0x2567, 0x2568, 0x2569, 0x256a,
1079 0x256b, 0x256c, 0x256d, 0x256e, 0x256f, 0x2570, 0x2571, 0x2572,
1080 0x2573, 0x2574, 0x2575, 0x2576, 0x2126, 0x213c, 0x2133, 0x2134,
1081 0x306c, 0x437a, 0x3c37, 0x4b7c, 0x3e66, 0x3b30, 0x3e65, 0x323c,
1082 0x4954, 0x4d3f, 0x5022, 0x312f, 0x336e, 0x5023, 0x4024, 0x5242,
1083 0x3556, 0x4a3a, 0x3e67, 0x4e3e, 0x4a42, 0x5024, 0x4366, 0x5025,
1084 0x367a, 0x5026, 0x345d, 0x4330, 0x3c67, 0x5027, 0x5028, 0x5029,
1085 0x4735, 0x3557, 0x4737, 0x4663, 0x3843, 0x4b33, 0x6949, 0x502a,
1086 0x3e68, 0x502b, 0x3235, 0x3665, 0x3870, 0x4c69, 0x5626, 0x4d70,
1087 0x467d, 0x3425, 0x3535, 0x502c, 0x502d, 0x4e3b, 0x4d3d, 0x4168,
1088 0x502f, 0x3b76, 0x4673, 0x5032, 0x313e, 0x385f, 0x385e, 0x3066,
1089 0x4f4b, 0x4f4a, 0x3a33, 0x3021, 0x5033, 0x5034, 0x5035, 0x4b34,
1090 0x5036, 0x3872, 0x3067, 0x4b72, 0x357c, 0x357d, 0x357e, 0x4462,
1091 0x4e3c, 0x5037, 0x5038, 0x5039, 0x3f4d, 0x3d3a, 0x3f4e, 0x503e,
1092 0x503c, 0x503d, 0x3558, 0x3a23, 0x3270, 0x503b, 0x503a, 0x4a29,
1093 0x3b46, 0x3b45, 0x423e, 0x503f, 0x4955, 0x4067, 0x2138, 0x5040,
1094 0x5042, 0x4265, 0x4e61, 0x304a, 0x5041, 0x323e, 0x3644, 0x4367,
1095 0x376f, 0x5043, 0x4724, 0x346b, 0x5044, 0x304b, 0x3860, 0x346c,
1096 0x497a, 0x4832, 0x3559, 0x3271, 0x5067, 0x4541, 0x476c, 0x5046,
1097 0x483c, 0x4e62, 0x3f2d, 0x3b47, 0x3b77, 0x3240, 0x4451, 0x4322,
1098 0x504a, 0x304c, 0x4463, 0x3d3b, 0x3a34, 0x4d24, 0x424e, 0x323f,
1099 0x5049, 0x4d3e, 0x5045, 0x5047, 0x3a6e, 0x5048, 0x5524, 0x5050,
1100 0x5053, 0x5051, 0x3242, 0x4a3b, 0x504b, 0x504f, 0x3873, 0x3b48,
1101 0x3426, 0x5054, 0x504c, 0x4e63, 0x3b78, 0x504d, 0x5052, 0x5055,
1102 0x504e, 0x3621, 0x304d, 0x3622, 0x3241, 0x5525, 0x4b79, 0x496e,
```



```
1190 0x5432, 0x5435, 0x373f, 0x5436, 0x5437, 0x3924, 0x3340, 0x5439,
1191 0x543a, 0x543b, 0x5438, 0x5431, 0x543c, 0x543d, 0x4b64, 0x3e6b,
1192 0x543f, 0x5440, 0x5441, 0x543e, 0x5442, 0x4738, 0x3068, 0x4956, 0x5443,
1193 0x3e7d, 0x3c39, 0x475d, 0x3470, 0x3a6b, 0x4b59, 0x4632, 0x3778,
1194 0x424f, 0x5441, 0x5444, 0x4244, 0x5445, 0x5446, 0x5448, 0x4469,
1195 0x342e, 0x7421, 0x3161, 0x4a73, 0x3e6c, 0x4548, 0x3a66, 0x544e,
1196 0x4a3d, 0x4e5d, 0x3274, 0x544a, 0x413a, 0x544d, 0x4563, 0x4549,
1197 0x4564, 0x4839, 0x444d, 0x3a49, 0x5449, 0x3176, 0x4536, 0x544b,
1198 0x5447, 0x3f50, 0x544f, 0x544f, 0x3d4e, 0x362d, 0x5450, 0x4a68, 0x417d,
1199 0x4446, 0x5452, 0x4b4f, 0x5453, 0x5458, 0x4a2f, 0x5457, 0x5451,
1200 0x5454, 0x5456, 0x3a26, 0x4a49, 0x5459, 0x4345, 0x3275, 0x3e6d,
1201 0x545b, 0x545a, 0x3968, 0x545c, 0x545e, 0x545d, 0x5460, 0x5455,
1202 0x5462, 0x5461, 0x545f, 0x3b4e, 0x3f51, 0x4154, 0x5463, 0x403c,
1203 0x306d, 0x4764, 0x445b, 0x5465, 0x5464, 0x5466, 0x5467, 0x5468,
1204 0x5469, 0x4a51, 0x546a, 0x3246, 0x546b, 0x4d3c, 0x3330, 0x5249,
1205 0x3d48, 0x423f, 0x546c, 0x4c6b, 0x4c34, 0x546e, 0x4267, 0x4537,
1206 0x4240, 0x4957, 0x546f, 0x5470, 0x317b, 0x3c3a, 0x5471, 0x3050,
1207 0x5472, 0x5473, 0x3162, 0x3471, 0x4660, 0x4a74, 0x5477, 0x4155,
1208 0x5476, 0x3740, 0x4b5b, 0x5475, 0x4565, 0x5479, 0x5478, 0x547b,
1209 0x547a, 0x317c, 0x547c, 0x3e29, 0x547e, 0x4325, 0x547d, 0x4a33,
1210 0x3d77, 0x455b, 0x5521, 0x3925, 0x5522, 0x4721, 0x485e, 0x4c51,
1211 0x4725, 0x552b, 0x3538, 0x4d45, 0x4c2f, 0x562c, 0x5523, 0x5526,
1212 0x4245, 0x4b38, 0x454a, 0x5527, 0x4b65, 0x3a4a, 0x3e2a, 0x5528,
1213 0x3b50, 0x3b4f, 0x3039, 0x3848, 0x402b, 0x3051, 0x552c, 0x552d,
1214 0x552a, 0x3138, 0x342f, 0x5529, 0x4c45, 0x4931, 0x3028, 0x3079,
1215 0x3b51, 0x3052, 0x3023, 0x5532, 0x5530, 0x4c3c, 0x5533, 0x5531,
1216 0x552f, 0x3f31, 0x552e, 0x4a5a, 0x3864, 0x5537, 0x5538, 0x3e2b,
1217 0x5534, 0x4f2c, 0x474c, 0x5536, 0x3a27, 0x5539, 0x4958, 0x553a,
1218 0x5535, 0x4c3b, 0x475e, 0x553b, 0x4932, 0x553c, 0x5540, 0x553d,
1219 0x3247, 0x553f, 0x3c3b, 0x553e, 0x3779, 0x554c, 0x5545, 0x5542,
1220 0x4364, 0x5541, 0x5543, 0x5544, 0x5546, 0x5547, 0x3472, 0x5549,
1221 0x5548, 0x554a, 0x3e6e, 0x554d, 0x445c, 0x3145, 0x554b, 0x554e,
1222 0x554f, 0x5552, 0x5550, 0x5551, 0x3b52, 0x5553, 0x3926, 0x5554,
1223 0x3b7a, 0x4238, 0x5555, 0x5556, 0x3b5a, 0x3927, 0x4c52, 0x3528,
1224 0x3849, 0x5557, 0x3358, 0x5558, 0x4239, 0x5559, 0x5623, 0x555a,
1225 0x555b, 0x555c, 0x555e, 0x555f, 0x5560, 0x4270, 0x3127, 0x3c69,
1226 0x3042, 0x4157, 0x3430, 0x3c35, 0x3928, 0x4566, 0x3d21, 0x3431,
1227 0x4368, 0x446a, 0x3038, 0x3539, 0x4a75, 0x3c42, 0x3552, 0x406b,
1228 0x3c3c, 0x4d28, 0x5561, 0x355c, 0x3a4b, 0x3332, 0x3163, 0x3e2c,
1229 0x3248, 0x5562, 0x4d46, 0x3d49, 0x3c64, 0x5563, 0x3473, 0x4652,
1230 0x4c29, 0x5564, 0x5565, 0x4959, 0x5567, 0x3428, 0x3677, 0x5566,
1231 0x3432, 0x3f32, 0x556b, 0x3b21, 0x3249, 0x556a, 0x5568, 0x556c,
1232 0x5569, 0x472b, 0x5c4d, 0x3f33, 0x556d, 0x4e40, 0x556e, 0x5570,
1233 0x437e, 0x556f, 0x4023, 0x3b7b, 0x4250, 0x3c77, 0x4975, 0x406c,
1234 0x3c4d, 0x5571, 0x3e2d, 0x5572, 0x5573, 0x3053, 0x423a, 0x3f52,
1235 0x5574, 0x4633, 0x3e2e, 0x3e2f, 0x5575, 0x406d, 0x3e30, 0x5576,
1236 0x5577, 0x4c60, 0x5578, 0x3646, 0x3d22, 0x5579, 0x557a, 0x3c5c,
1237 0x3f2c, 0x4674, 0x3f54, 0x4878, 0x4722, 0x3649, 0x557b, 0x356f,
1238 0x557c, 0x367e, 0x464f, 0x3230, 0x3b53, 0x557d, 0x5622, 0x5621,
1239 0x367d, 0x557e, 0x4538, 0x4230, 0x454b, 0x3c48, 0x4158, 0x4d7a,
1240 0x5624, 0x5625, 0x4656, 0x3b33, 0x5627, 0x5628, 0x5629, 0x3474,
1241 0x562a, 0x562b, 0x322c, 0x413b, 0x3464, 0x562d, 0x4c28, 0x4252,
1242 0x3359, 0x562f, 0x5631, 0x345f, 0x562e, 0x5630, 0x5633, 0x5632,
1243 0x5634, 0x5635, 0x463d, 0x362e, 0x362f, 0x5636, 0x563b, 0x5639,
1244 0x4a77, 0x4a76, 0x4567, 0x5638, 0x3d54, 0x5637, 0x3f72, 0x563c,
1245 0x3a6a, 0x5642, 0x5643, 0x563d, 0x3333, 0x563e, 0x5647, 0x5646,
1246 0x5645, 0x5641, 0x5640, 0x5644, 0x4a78, 0x564b, 0x5648, 0x564a,
1247 0x4d72, 0x5649, 0x563f, 0x3f73, 0x564c, 0x3a37, 0x564d, 0x564e,
1248 0x5651, 0x5650, 0x564f, 0x4568, 0x563a, 0x5657, 0x5653, 0x5652,
1249 0x5654, 0x5655, 0x5658, 0x4e66, 0x5659, 0x5656, 0x565a, 0x3460,
1250 0x565b, 0x565d, 0x565c, 0x565e, 0x565f, 0x406e, 0x3d23, 0x3d64,
1251 0x4163, 0x3929, 0x3a38, 0x392a, 0x3570, 0x5660, 0x3a39, 0x384a,
1252 0x5661, 0x4c26, 0x4743, 0x5662, 0x392b, 0x342c, 0x4327, 0x3652,
1253 0x3b54, 0x495b, 0x4841, 0x5663, 0x3475, 0x5666, 0x4421, 0x5665,
1254 0x5664, 0x5667, 0x446b, 0x3f63, 0x3b55, 0x404a, 0x4253, 0x3522,
1255 0x4422, 0x5668, 0x5669, 0x3e6f, 0x4b39, 0x566c, 0x566b, 0x566a,
1256 0x497d, 0x5673, 0x4b5a, 0x566d, 0x566f, 0x4b6b, 0x566e, 0x5670,
1257 0x4828, 0x5671, 0x4a3e, 0x5672, 0x3433, 0x4a3f, 0x472f, 0x5674,
1258 0x5675, 0x392c, 0x3434, 0x5676, 0x3838, 0x4d44, 0x4d29, 0x3476,
1259 0x5678, 0x4423, 0x392d, 0x3e31, 0x485f, 0x3e32, 0x3d78, 0x446c,
1260 0x4a79, 0x4539, 0x392e, 0x495c, 0x5679, 0x4559, 0x3a42, 0x384b,
1261 0x446d, 0x3043, 0x3d6e, 0x392f, 0x4d47, 0x567a, 0x567b, 0x4751,
1262 0x567c, 0x4e77, 0x4f2d, 0x567e, 0x567d, 0x3347, 0x5721, 0x5724,
1263 0x5725, 0x5723, 0x4940, 0x3e33, 0x5727, 0x5726, 0x5722, 0x5728,
1264 0x5729, 0x572a, 0x572d, 0x572b, 0x572c, 0x572e, 0x3164, 0x446e,
1265 0x572f, 0x377a, 0x3276, 0x4736, 0x5730, 0x467b, 0x4a5b, 0x5731,
1266 0x4f2e, 0x5732, 0x4a40, 0x5735, 0x5021, 0x5031, 0x3c30, 0x4675,
1267 0x5736, 0x355d, 0x4424, 0x307a, 0x5737, 0x4a26, 0x3930, 0x4350,
1268 0x446f, 0x4c6f, 0x3839, 0x384c, 0x5738, 0x5739, 0x573f, 0x3c65,
1269 0x4425, 0x362f, 0x573a, 0x492b, 0x4346, 0x573b, 0x573c, 0x3630,
1270 0x573d, 0x573e, 0x5740, 0x4576, 0x5741, 0x5742, 0x5743, 0x5734,
1271 0x5733, 0x5744, 0x3741, 0x4927, 0x3a4c, 0x4937, 0x4426, 0x494b,
1272 0x5745, 0x3e34, 0x3146, 0x5746, 0x5747, 0x4c72, 0x4860, 0x574a,
1273 0x317d, 0x402c, 0x5749, 0x5748, 0x3742, 0x4254, 0x574e, 0x574c,
1274 0x574b, 0x4e27, 0x3865, 0x3d79, 0x574d, 0x454c, 0x3d3e, 0x4640,
1275 0x5751, 0x5750, 0x574f, 0x5752, 0x3866, 0x5753, 0x497c, 0x3d5b,
1276 0x5754, 0x4879, 0x4641, 0x4427, 0x4530, 0x5755, 0x352b, 0x3f34,
```

```

1277 0x492c, 0x3477, 0x4726, 0x5756, 0x3b56, 0x4b3a, 0x4b3b, 0x317e,
1278 0x575b, 0x4369, 0x5758, 0x3277, 0x582d, 0x575a, 0x4730, 0x5759,
1279 0x5757, 0x397a, 0x575d, 0x575d, 0x5763, 0x5769, 0x5761, 0x455c, 0x5766,
1280 0x495d, 0x5760, 0x5765, 0x4e67, 0x3b57, 0x4255, 0x575e, 0x355e,
1281 0x5768, 0x402d, 0x3165, 0x5762, 0x3278, 0x5767, 0x3631, 0x5764,
1282 0x576a, 0x576c, 0x5776, 0x5776, 0x5774, 0x5771, 0x5770, 0x4e78, 0x5772,
1283 0x3632, 0x3931, 0x3d7a, 0x5779, 0x576b, 0x576f, 0x575f, 0x327a,
1284 0x5773, 0x5775, 0x4351, 0x3a28, 0x3238, 0x576d, 0x5778, 0x5777,
1285 0x3633, 0x4229, 0x3366, 0x3743, 0x576e, 0x577a, 0x577d, 0x5821,
1286 0x3c3d, 0x5827, 0x4470, 0x577b, 0x5825, 0x3279, 0x5823, 0x5824,
1287 0x577e, 0x5822, 0x3867, 0x4d2a, 0x3435, 0x3159, 0x5826, 0x473a,
1288 0x302d, 0x4861, 0x575c, 0x582c, 0x5830, 0x4c65, 0x5829, 0x4569,
1289 0x582e, 0x3e70, 0x582f, 0x4657, 0x4f47, 0x582b, 0x5831, 0x397b,
1290 0x404b, 0x3054, 0x582a, 0x5828, 0x415a, 0x577c, 0x3b34, 0x4246,
1291 0x583d, 0x415b, 0x5838, 0x5835, 0x5836, 0x3c66, 0x5839, 0x583c,
1292 0x5837, 0x3d25, 0x583a, 0x5834, 0x4c7c, 0x4c7b, 0x583e, 0x583f,
1293 0x3055, 0x5833, 0x3672, 0x3026, 0x3436, 0x583b, 0x5843, 0x5842,
1294 0x5847, 0x5848, 0x5846, 0x5849, 0x5841, 0x5845, 0x584a, 0x584b,
1295 0x5840, 0x3b7c, 0x5844, 0x4256, 0x3932, 0x5832, 0x3f35, 0x5858,
1296 0x4a69, 0x584e, 0x584f, 0x5850, 0x5857, 0x5856, 0x4b7d, 0x3437,
1297 0x5854, 0x3745, 0x3334, 0x5851, 0x4e38, 0x5853, 0x3056, 0x5855,
1298 0x584c, 0x5852, 0x5859, 0x3744, 0x584d, 0x4d5d, 0x4d2b, 0x585c,
1299 0x5860, 0x417e, 0x4e79, 0x5861, 0x585e, 0x585b, 0x585a, 0x585f,
1300 0x4a30, 0x4634, 0x3746, 0x5862, 0x585d, 0x5863, 0x377b, 0x3231,
1301 0x586b, 0x3438, 0x5869, 0x586a, 0x3a29, 0x5868, 0x5866, 0x5865,
1302 0x586c, 0x5864, 0x586e, 0x327b, 0x5870, 0x586f, 0x4428, 0x5873,
1303 0x5871, 0x5867, 0x377c, 0x5872, 0x5876, 0x5875, 0x5877, 0x5874,
1304 0x5878, 0x5879, 0x587a, 0x4a6a, 0x587c, 0x587b, 0x3d3f, 0x402e,
1305 0x3266, 0x327c, 0x587d, 0x303f, 0x404c, 0x587e, 0x6c43, 0x5921,
1306 0x3761, 0x5922, 0x406f, 0x5923, 0x5924, 0x353a, 0x5925, 0x5926,
1307 0x5927, 0x4257, 0x384d, 0x4c61, 0x4b3c, 0x3d6a, 0x5928, 0x4070,
1308 0x6e3d, 0x4862, 0x3c6a, 0x3a4d, 0x5929, 0x4247, 0x4a27, 0x4271,
1309 0x592c, 0x592a, 0x592d, 0x592b, 0x592e, 0x4a31, 0x3037, 0x495e,
1310 0x4863, 0x592f, 0x5932, 0x3e35, 0x353b, 0x5930, 0x5937, 0x3e36,
1311 0x5931, 0x4744, 0x4d5e, 0x5933, 0x5934, 0x5938, 0x456a, 0x5935,
1312 0x3933, 0x405e, 0x5946, 0x4834, 0x4272, 0x4864, 0x5a2d, 0x4a7a,
1313 0x4471, 0x4b75, 0x593b, 0x3221, 0x436a, 0x5944, 0x4334, 0x593e,
1314 0x5945, 0x5940, 0x5947, 0x5943, 0x5942, 0x476f, 0x593c, 0x327d,
1315 0x593a, 0x3571, 0x4273, 0x5936, 0x5939, 0x3934, 0x405b, 0x3e37,
1316 0x5941, 0x4752, 0x3572, 0x3348, 0x3367, 0x3f21, 0x5949, 0x594e,
1317 0x594a, 0x377d, 0x594f, 0x3b22, 0x3969, 0x3d26, 0x593d, 0x3b7d,
1318 0x594c, 0x3b58, 0x594d, 0x3044, 0x5948, 0x4429, 0x3573, 0x3634,
1319 0x594b, 0x3027, 0x3a43, 0x3f36, 0x4472, 0x4854, 0x5951, 0x415e,
1320 0x422a, 0x3b2b, 0x5952, 0x5954, 0x5950, 0x4a61, 0x443d, 0x415c,
1321 0x4a7b, 0x3c4e, 0x5960, 0x595f, 0x3f78, 0x377e, 0x5959, 0x3e39,
1322 0x4668, 0x4731, 0x5957, 0x415d, 0x3c78, 0x595c, 0x3e38, 0x5956,
1323 0x595b, 0x4753, 0x5955, 0x3721, 0x335d, 0x595d, 0x4e2b, 0x3a4e,
1324 0x4335, 0x595a, 0x405c, 0x3935, 0x3f64, 0x3166, 0x413c, 0x5958,
1325 0x3545, 0x3747, 0x444f, 0x595e, 0x415f, 0x5961, 0x5963, 0x4237,
1326 0x5969, 0x5964, 0x5966, 0x4941, 0x4473, 0x5967, 0x4d2c, 0x4d48,
1327 0x3439, 0x302e, 0x5965, 0x5962, 0x3478, 0x3167, 0x5968, 0x4d49,
1328 0x596c, 0x423b, 0x5973, 0x596d, 0x596a, 0x5971, 0x5953, 0x596e,
1329 0x5972, 0x4842, 0x456b, 0x596b, 0x596f, 0x3748, 0x3a71, 0x405d,
1330 0x5977, 0x4526, 0x5974, 0x4b60, 0x5975, 0x5976, 0x4c4e, 0x4022,
1331 0x3762, 0x597d, 0x3b35, 0x597a, 0x5979, 0x4732, 0x4635, 0x4531,
1332 0x597b, 0x597c, 0x496f, 0x4745, 0x3b23, 0x4071, 0x4b50, 0x3349,
1333 0x5a25, 0x597e, 0x4d4a, 0x5a27, 0x5a23, 0x5a24, 0x4160, 0x5a22,
1334 0x593f, 0x5a26, 0x5a21, 0x5a2b, 0x5a2c, 0x4527, 0x5a2e, 0x3b24,
1335 0x5a29, 0x353c, 0x5a2f, 0x5a28, 0x5a33, 0x5a32, 0x5a31, 0x5a34,
1336 0x5a36, 0x3e71, 0x5a35, 0x5a39, 0x5a37, 0x5a38, 0x5970, 0x5a3b,
1337 0x5a3a, 0x5978, 0x5a3c, 0x5a30, 0x3b59, 0x5a3d, 0x5a3e, 0x5a40,
1338 0x5a3f, 0x5a41, 0x327e, 0x3936, 0x4a7c, 0x402f, 0x384e, 0x5a43,
1339 0x5a46, 0x4952, 0x355f, 0x5a45, 0x5a44, 0x4754, 0x5a47, 0x3635,
1340 0x5a49, 0x5a48, 0x343a, 0x3b36, 0x4658, 0x3749, 0x3f74, 0x5a4a,
1341 0x4030, 0x4528, 0x495f, 0x5a4b, 0x5a4c, 0x5a4d, 0x4a38, 0x555d,
1342 0x4046, 0x494c, 0x3a58, 0x4865, 0x4843, 0x454d, 0x4e41, 0x5a4f,
1343 0x3c50, 0x5a50, 0x3036, 0x3654, 0x404d, 0x4960, 0x5a51, 0x3b42,
1344 0x4347, 0x3b5b, 0x3f37, 0x5a52, 0x4a7d, 0x3177, 0x3b5c, 0x5a55,
1345 0x5a53, 0x5a56, 0x4e39, 0x5a54, 0x407b, 0x5a57, 0x4232, 0x5a58,
1346 0x347a, 0x5a5a, 0x5a59, 0x5a5b, 0x5a5c, 0x347b, 0x467c, 0x4336,
1347 0x356c, 0x3b5d, 0x4161, 0x3d5c, 0x3030, 0x5a5d, 0x3222, 0x5a61,
1348 0x3937, 0x5a60, 0x3a2b, 0x3e3a, 0x5a5f, 0x3e3b, 0x4c40, 0x3a2a,
1349 0x3057, 0x404e, 0x5a66, 0x4031, 0x3147, 0x3d55, 0x4b66, 0x3a72,
1350 0x3e3c, 0x4027, 0x5a65, 0x5a63, 0x5a64, 0x436b, 0x5b26, 0x5a6a,
1351 0x3b7e, 0x3938, 0x5a68, 0x5a69, 0x3f38, 0x5a67, 0x3b2f, 0x5a6c,
1352 0x5a6b, 0x5a70, 0x5a71, 0x5a6d, 0x3322, 0x5a6e, 0x5a6f, 0x4855,
1353 0x4961, 0x374a, 0x5a72, 0x4032, 0x3e3d, 0x4352, 0x3647, 0x5a73,
1354 0x5a77, 0x324b, 0x5a74, 0x5a76, 0x5a75, 0x3d6b, 0x4348, 0x3045,
1355 0x5a78, 0x5a79, 0x442a, 0x4e71, 0x3b43, 0x4a6b, 0x4b3d, 0x5b22,
1356 0x5a7b, 0x5a7e, 0x5a7d, 0x5a7a, 0x5b21, 0x465e, 0x5a7c, 0x5b23,
1357 0x3d6c, 0x5b24, 0x4d4b, 0x4778, 0x5b25, 0x5b27, 0x5b28, 0x5b29,
1358 0x364a, 0x3148, 0x3939, 0x5b2a, 0x5b2b, 0x3d71, 0x4162, 0x5258,
1359 0x413e, 0x413d, 0x4258, 0x3a47, 0x5072, 0x376e, 0x4d2d, 0x4a7e,
1360 0x497e, 0x5b2c, 0x3a73, 0x443f, 0x5b2d, 0x4f2f, 0x4b3e, 0x442b,
1361 0x5b2e, 0x347c, 0x5b2f, 0x5b30, 0x4c5a, 0x4c24, 0x4b76, 0x4b5c,
1362 0x3b25, 0x5b32, 0x3c6b, 0x4b51, 0x5b34, 0x5b37, 0x5b36, 0x3479,
1363 0x3560, 0x5b33, 0x5b35, 0x5b38, 0x3f79, 0x4d7b, 0x3049, 0x3a60,

```

```
1364 0x423c, 0x3c5d, 0x3e73, 0x5b3b, 0x454e, 0x5b39, 0x422b, 0x5b3a,  
1365 0x3e72, 0x4c5d, 0x5b3c, 0x5b3d, 0x4d68, 0x5b42, 0x393a, 0x4755,  
1366 0x5b3f, 0x456c, 0x456c, 0x5a5e, 0x5a62, 0x354f, 0x4747, 0x5b41, 0x3e3e,  
1367 0x4844, 0x5b47, 0x487a, 0x5b3e, 0x5b44, 0x5b43, 0x404f, 0x4b6d,  
1368 0x4e53, 0x4b67, 0x324c, 0x3b5e, 0x4f48, 0x5b46, 0x3f75, 0x5b45,  
1369 0x5b40, 0x384f, 0x384f, 0x5b4c, 0x5b4a, 0x324d, 0x5b48, 0x5b4e, 0x5b54,  
1370 0x4248, 0x4a41, 0x5b56, 0x4922, 0x5b55, 0x4770, 0x4b3f, 0x343b,  
1371 0x4077, 0x3d40, 0x4453, 0x4d2e, 0x5b51, 0x5b50, 0x5b52, 0x5b4f,  
1372 0x5b57, 0x5b4d, 0x5b4b, 0x5b53, 0x5b49, 0x436c, 0x4c78, 0x3c46,  
1373 0x3a74, 0x3a3a, 0x4b6f, 0x3341, 0x444e, 0x464a, 0x3149, 0x4072,  
1374 0x4034, 0x372a, 0x5b59, 0x393b, 0x337c, 0x5b5b, 0x3374, 0x5b61,  
1375 0x5b5e, 0x4073, 0x334b, 0x3a2c, 0x334a, 0x3a4f, 0x5b5c, 0x3765,  
1376 0x374b, 0x456d, 0x5b5a, 0x3046, 0x5b5d, 0x5b5f, 0x364d, 0x372c,  
1377 0x343c, 0x354b, 0x5b62, 0x3a79, 0x4b71, 0x3b37, 0x5b63, 0x4930,  
1378 0x5b6f, 0x3233, 0x5b64, 0x5b75, 0x5b65, 0x4e42, 0x5b6c, 0x475f,  
1379 0x5b74, 0x5b67, 0x3034, 0x5b69, 0x393c, 0x5b6b, 0x5b6a, 0x5b66,  
1380 0x5b71, 0x3e3f, 0x546d, 0x3868, 0x4d7c, 0x5b68, 0x4474, 0x3323,  
1381 0x3a2d, 0x5b60, 0x5b70, 0x3361, 0x5b6e, 0x5b72, 0x456e, 0x347e,  
1382 0x5c32, 0x4c49, 0x5b77, 0x347d, 0x5b7e, 0x4b40, 0x5c21, 0x5c23,  
1383 0x5c27, 0x5b79, 0x432a, 0x456f, 0x5c2b, 0x5b7c, 0x5c28, 0x5c22,  
1384 0x3f39, 0x5c2c, 0x4033, 0x4033, 0x5c2a, 0x343d, 0x4f50, 0x5b76, 0x5c26,  
1385 0x3058, 0x5b78, 0x4c3a, 0x5b7d, 0x3f22, 0x4447, 0x5b73, 0x5c25,  
1386 0x3f7a, 0x5c2f, 0x3371, 0x3821, 0x5c31, 0x5b7a, 0x5c30, 0x5c29,  
1387 0x5b7b, 0x5c2d, 0x5c2e, 0x5c2f, 0x464e, 0x5c24, 0x5c3b, 0x5c3d,  
1388 0x4458, 0x4d4c, 0x4976, 0x5c38, 0x424a, 0x5c3e, 0x413f, 0x5c35,  
1389 0x5c42, 0x5c41, 0x466f, 0x5c40, 0x466a, 0x5c44, 0x5c37, 0x3648,  
1390 0x5c3a, 0x3d50, 0x4760, 0x5c3c, 0x364b, 0x5c34, 0x5c36, 0x5c33,  
1391 0x4f30, 0x335a, 0x5c39, 0x5c43, 0x3335, 0x3a67, 0x315d, 0x5c54,  
1392 0x4f31, 0x5c57, 0x3f3a, 0x5c56, 0x5c55, 0x5c52, 0x5c46, 0x5c63,  
1393 0x5c45, 0x5c58, 0x5c50, 0x5c4b, 0x5c48, 0x5c49, 0x5c51, 0x7422,  
1394 0x5c4e, 0x393d, 0x4448, 0x4164, 0x5c4c, 0x5c47, 0x5c4a, 0x4d4d,  
1395 0x4b6a, 0x5c4f, 0x5c59, 0x5c61, 0x5c5a, 0x5c67, 0x5c65, 0x5c60,  
1396 0x5c5f, 0x4450, 0x4165, 0x5c5d, 0x5c5b, 0x5c62, 0x5c68, 0x4875,  
1397 0x5c6e, 0x5c69, 0x5c6c, 0x5c66, 0x4374, 0x4938, 0x5c5c, 0x5c64,  
1398 0x3e40, 0x4c4f, 0x5c78, 0x5c6b, 0x3822, 0x3223, 0x335f, 0x5c53,  
1399 0x3e41, 0x5c70, 0x5c77, 0x3c79, 0x3372, 0x432e, 0x5c6d, 0x5c72,  
1400 0x5c76, 0x3636, 0x354c, 0x5c74, 0x3521, 0x464b, 0x5c73, 0x5c75,  
1401 0x5c6f, 0x5c71, 0x3360, 0x4349, 0x5c7c, 0x5c7a, 0x3869, 0x5c79,  
1402 0x5d21, 0x5b58, 0x5c7b, 0x5c7d, 0x5c7e, 0x5d2c, 0x5d28, 0x5b6d,  
1403 0x5d27, 0x5d26, 0x5d23, 0x5c6a, 0x5d25, 0x5d24, 0x5d2a, 0x4f26,  
1404 0x5d2d, 0x367b, 0x5d29, 0x5d2b, 0x4827, 0x5d2e, 0x5d32, 0x5d2f,  
1405 0x4d73, 0x5d30, 0x5c5e, 0x5d33, 0x5d34, 0x3135, 0x5d36, 0x3767,  
1406 0x3c21, 0x3655, 0x3224, 0x4d5f, 0x5d38, 0x5d37, 0x5d3a, 0x353d,  
1407 0x3656, 0x343e, 0x5d3d, 0x5d3c, 0x5d3e, 0x324e, 0x4337, 0x5d3f,  
1408 0x343f, 0x5d41, 0x5d40, 0x5d42, 0x5d43, 0x5d44, 0x3b5f, 0x4035,  
1409 0x3a21, 0x4970, 0x4a62, 0x4f44, 0x3b75, 0x3a50, 0x4e72, 0x5d45,  
1410 0x5d46, 0x3b60, 0x5d47, 0x5d48, 0x5d4a, 0x5d49, 0x4b58, 0x3d5e,  
1411 0x3c6c, 0x3b44, 0x5d4b, 0x5d4d, 0x3f23, 0x5d4c, 0x5d4e, 0x5d4f,  
1412 0x5d50, 0x5d51, 0x5d52, 0x5d54, 0x5d53, 0x5d55, 0x3225, 0x434a,  
1413 0x5d56, 0x3b26, 0x334c, 0x5d57, 0x4542, 0x544c, 0x3523, 0x5d58,  
1414 0x5d59, 0x4a6c, 0x4b68, 0x4647, 0x5d5a, 0x4866, 0x487b, 0x4c53,  
1415 0x5d5b, 0x5d5d, 0x5d5c, 0x5d5f, 0x5d5e, 0x5d61, 0x3b61, 0x4c31,  
1416 0x5d62, 0x5d63, 0x3524, 0x5d64, 0x5d66, 0x5d65, 0x3f65, 0x4939,  
1417 0x314a, 0x4845, 0x4475, 0x3d41, 0x3561, 0x4846, 0x3c2e, 0x5d68,  
1418 0x3440, 0x3178, 0x4672, 0x5d67, 0x393e, 0x4353, 0x5d69, 0x5d71,  
1419 0x5d6a, 0x4241, 0x3562, 0x5d72, 0x3768, 0x3525, 0x5d70, 0x5d6e,  
1420 0x5d6b, 0x4d60, 0x4440, 0x4659, 0x5d6c, 0x5d74, 0x5d73, 0x3723,  
1421 0x322d, 0x3a3b, 0x5d6d, 0x5d6f, 0x4b57, 0x4274, 0x4b77, 0x5d7c,  
1422 0x5d7d, 0x324f, 0x4a28, 0x4c7d, 0x5e21, 0x3c23, 0x3e42, 0x5d78,  
1423 0x5d7e, 0x3168, 0x3637, 0x5d75, 0x5d7a, 0x4074, 0x4771, 0x4867,  
1424 0x5d77, 0x4b21, 0x5d79, 0x5e24, 0x5e22, 0x5d7b, 0x4b22, 0x4748,  
1425 0x3563, 0x4525, 0x436d, 0x5e25, 0x5e23, 0x4259, 0x5d76, 0x314b,  
1426 0x4d4e, 0x5e30, 0x5e2f, 0x4076, 0x5e2c, 0x4d6c, 0x4636, 0x5e26,  
1427 0x4445, 0x314c, 0x393f, 0x5e29, 0x3d27, 0x5e2e, 0x5e2d, 0x5e28,  
1428 0x5e2b, 0x3368, 0x5e2a, 0x4749, 0x4e2e, 0x3e74, 0x4075, 0x5e36,  
1429 0x5e34, 0x494d, 0x5e31, 0x5e33, 0x313a, 0x3940, 0x4f32, 0x333d,  
1430 0x4962, 0x4d61, 0x3324, 0x3f3b, 0x5e35, 0x5e3a, 0x3e43, 0x4d30,  
1431 0x5e37, 0x5e32, 0x5e38, 0x4e5e, 0x4573, 0x4642, 0x3336, 0x3155,  
1432 0x5e3e, 0x5e41, 0x4e43, 0x4d64, 0x5e48, 0x5e42, 0x5e3f, 0x4e54,  
1433 0x5e45, 0x3d4a, 0x5e47, 0x5e4c, 0x4571, 0x5e4a, 0x5e44, 0x4338,  
1434 0x5e4b, 0x5e40, 0x5e46, 0x5e4d, 0x307c, 0x5e43, 0x5e4e, 0x3f3c,  
1435 0x3d5f, 0x4a25, 0x3a2e, 0x5e3b, 0x5e49, 0x453a, 0x4036, 0x3369,  
1436 0x3a51, 0x3e44, 0x5e3d, 0x3d42, 0x374c, 0x5e3c, 0x5e52, 0x3d6d,  
1437 0x383a, 0x5e61, 0x5e5b, 0x3574, 0x454f, 0x5e56, 0x5e5f, 0x302f,  
1438 0x3132, 0x3239, 0x5e58, 0x422c, 0x5e4f, 0x5e51, 0x3941, 0x5e62,  
1439 0x5e5d, 0x5e55, 0x5e5c, 0x4c2b, 0x5e5a, 0x5e5e, 0x3850, 0x3e45,  
1440 0x4339, 0x5e54, 0x4d2f, 0x5e57, 0x5e50, 0x4572, 0x5e53, 0x5e59,  
1441 0x4f51, 0x3c3e, 0x4b7e, 0x5e63, 0x482e, 0x5e6f, 0x383b, 0x3d60,  
1442 0x5e65, 0x4e2f, 0x3942, 0x5e72, 0x306e, 0x5e70, 0x5e64, 0x5e6a,  
1443 0x5e6c, 0x4d4f, 0x5e67, 0x452e, 0x5e69, 0x5e71, 0x5e6b, 0x4c47,  
1444 0x5e66, 0x3c22, 0x5e7e, 0x336a, 0x5e68, 0x5e6d, 0x5e6e, 0x426c,  
1445 0x425a, 0x5e76, 0x5e7c, 0x5e7a, 0x4529, 0x5f23, 0x5e77, 0x5e78,  
1446 0x5e60, 0x3579, 0x493a, 0x3c3f, 0x3977, 0x4f33, 0x5e74, 0x5f22,  
1447 0x3169, 0x4166, 0x4779, 0x3441, 0x4e7a, 0x4c21, 0x4452, 0x5e7b,  
1448 0x5e7d, 0x4132, 0x5f21, 0x5e79, 0x5e73, 0x3443, 0x3769, 0x5f2f,  
1449 0x5f2a, 0x4078, 0x3363, 0x3d61, 0x5f33, 0x5f2c, 0x442c, 0x5f29,  
1450 0x4459, 0x5f4c, 0x5f26, 0x5f25, 0x5f2e, 0x5f28, 0x5f27, 0x5f2d,
```

1451 0x4021, 0x5f24, 0x5f30, 0x5f31, 0x3442, 0x5f36, 0x5f35, 0x5f37,
1452 0x5f3a, 0x4543, 0x5f34, 0x5f38, 0x3763, 0x4279, 0x5f32, 0x473b,
1453 0x5f39, 0x5f3e, 0x5f3c, 0x5f3d, 0x5f3f, 0x5f42, 0x5f3b, 0x396a, 0x4728,
1454 0x5e39, 0x4d74, 0x5f3d, 0x5f41, 0x4275, 0x5f40, 0x5f2b, 0x6f69,
1455 0x5f45, 0x5f49, 0x5f47, 0x5f43, 0x5f44, 0x5f48, 0x5f46, 0x494e,
1456 0x5f4e, 0x5f4b, 0x5f4a, 0x5f4a, 0x5f4d, 0x4654, 0x5f4f, 0x4375, 0x426d,
1457 0x4025, 0x5f50, 0x5f52, 0x5f51, 0x5e75, 0x5f53, 0x4667, 0x5f54,
1458 0x3250, 0x4574, 0x3325, 0x3564, 0x3c5e, 0x3a52, 0x4f27, 0x3f66,
1459 0x316a, 0x5f56, 0x5f55, 0x5f59, 0x433a, 0x5f5c, 0x5f57, 0x5f5b,
1460 0x5f5a, 0x4540, 0x3059, 0x4e75, 0x5f5e, 0x3128, 0x5f60, 0x5f5f,
1461 0x5f5d, 0x5f58, 0x4b23, 0x5f62, 0x5f61, 0x316b, 0x5f64, 0x4a32,
1462 0x5f63, 0x4c35, 0x3e47, 0x4133, 0x3e46, 0x4e7b, 0x5f6a, 0x4079,
1463 0x5f66, 0x5f6b, 0x316c, 0x5f69, 0x4761, 0x5f65, 0x5f68, 0x3e48,
1464 0x4851, 0x5f6c, 0x3c51, 0x407a, 0x5f6f, 0x5f67, 0x3727, 0x5f6d,
1465 0x4d50, 0x5f70, 0x7426, 0x4133, 0x3d4f, 0x5f71, 0x5f72, 0x472e, 0x5f74,
1466 0x5f75, 0x4733, 0x4575, 0x5f77, 0x5f79, 0x4e55, 0x5f76, 0x5f78,
1467 0x316d, 0x5f73, 0x535b, 0x5f7a, 0x4167, 0x3b38, 0x5f7c, 0x5f7b,
1468 0x3f24, 0x5259, 0x5f7d, 0x6021, 0x5f6e, 0x5f7e, 0x6022, 0x477a,
1469 0x6023, 0x6024, 0x6025, 0x6026, 0x445e, 0x6028, 0x6027, 0x6029,
1470 0x602a, 0x3c5f, 0x4963, 0x4c6c, 0x602b, 0x602c, 0x4156, 0x3c24,
1471 0x602d, 0x602e, 0x602f, 0x602f, 0x4a52, 0x4847, 0x6030, 0x4757, 0x442d,
1472 0x6031, 0x3267, 0x356d, 0x4c46, 0x4c36, 0x3234, 0x4f34, 0x4b52,
1473 0x4a2a, 0x4037, 0x6032, 0x4643, 0x3823, 0x6033, 0x3a54, 0x6035,
1474 0x6034, 0x6036, 0x6037, 0x6038, 0x353e, 0x6039, 0x603a, 0x3824,
1475 0x4848, 0x603c, 0x3e75, 0x603b, 0x3638, 0x603d, 0x603f, 0x603e,
1476 0x6040, 0x3851, 0x6041, 0x3669, 0x4140, 0x397d, 0x6043, 0x6044,
1477 0x6042, 0x3c6d, 0x4648, 0x3639, 0x6046, 0x432c, 0x6045, 0x4f35,
1478 0x4762, 0x6049, 0x604b, 0x6048, 0x4c54, 0x604a, 0x604c, 0x4e44,
1479 0x6050, 0x604f, 0x4376, 0x472d, 0x3825, 0x604e, 0x604d, 0x4d31,
1480 0x4d32, 0x6051, 0x316e, 0x3976, 0x3b62, 0x6052, 0x6053, 0x6055,
1481 0x3d43, 0x6057, 0x6056, 0x6058, 0x334d, 0x605a, 0x6059, 0x605c,
1482 0x605b, 0x383c, 0x4e28, 0x364c, 0x3226, 0x366a, 0x3461, 0x4e68,
1483 0x605e, 0x6060, 0x6061, 0x3251, 0x605d, 0x3b39, 0x4441, 0x605f,
1484 0x6064, 0x3c6e, 0x6062, 0x373e, 0x4849, 0x6063, 0x607e, 0x6069,
1485 0x383d, 0x3565, 0x6066, 0x4d7d, 0x4e30, 0x4276, 0x6068, 0x606a,
1486 0x4e56, 0x3657, 0x487c, 0x474a, 0x606b, 0x606d, 0x6070, 0x606c,
1487 0x606f, 0x386a, 0x314d, 0x6071, 0x3f70, 0x606e, 0x4e5c, 0x6074,
1488 0x7424, 0x6072, 0x6075, 0x6067, 0x6073, 0x3a3c, 0x6076, 0x6077,
1489 0x4d7e, 0x6078, 0x6079, 0x6065, 0x607a, 0x3444, 0x3c25, 0x607b,
1490 0x607c, 0x607d, 0x313b, 0x6121, 0x493b, 0x6122, 0x3424, 0x6123,
1491 0x6124, 0x6125, 0x6127, 0x6128, 0x6126, 0x4953, 0x612a, 0x6129,
1492 0x612c, 0x612b, 0x612d, 0x612e, 0x6130, 0x612f, 0x3979, 0x6132,
1493 0x6131, 0x3445, 0x3f53, 0x453c, 0x6133, 0x4038, 0x3b3a, 0x3179,
1494 0x6134, 0x4d51, 0x4a63, 0x6135, 0x4544, 0x4d33, 0x3943, 0x3f3d,
1495 0x434b, 0x5234, 0x442e, 0x3268, 0x6136, 0x6137, 0x613c, 0x613a,
1496 0x6139, 0x5a42, 0x3326, 0x6138, 0x305a, 0x482a, 0x484a, 0x4e31,
1497 0x613d, 0x613b, 0x435c, 0x4026, 0x482b, 0x492d, 0x613f, 0x4e2c,
1498 0x374d, 0x6140, 0x613e, 0x4856, 0x6141, 0x6142, 0x305b, 0x3e76,
1499 0x6147, 0x6144, 0x466d, 0x6143, 0x3526, 0x614a, 0x6145, 0x6146,
1500 0x6149, 0x6148, 0x4925, 0x4142, 0x4141, 0x353f, 0x614b, 0x614c,
1501 0x614d, 0x614f, 0x614e, 0x3156, 0x6157, 0x4868, 0x6151, 0x6153,
1502 0x6155, 0x3f3e, 0x6156, 0x6154, 0x3c40, 0x6150, 0x6152, 0x4942,
1503 0x3e49, 0x6159, 0x6158, 0x615a, 0x3c26, 0x3a2f, 0x4577, 0x615b,
1504 0x444b, 0x615d, 0x4e21, 0x615c, 0x4169, 0x6162, 0x6164, 0x6165,
1505 0x4354, 0x6163, 0x6160, 0x615e, 0x615f, 0x6161, 0x6168, 0x6166,
1506 0x6167, 0x6169, 0x616b, 0x616c, 0x616d, 0x616e, 0x616a, 0x6170,
1507 0x616f, 0x6171, 0x4e45, 0x6174, 0x6172, 0x6173, 0x3462, 0x4c7e,
1508 0x4a4a, 0x6176, 0x6175, 0x6177, 0x6178, 0x617c, 0x6179, 0x617a,
1509 0x617b, 0x617d, 0x617e, 0x6221, 0x6222, 0x6223, 0x482f, 0x4550,
1510 0x6224, 0x4772, 0x4934, 0x6225, 0x6226, 0x452a, 0x3327, 0x3944,
1511 0x6227, 0x6228, 0x6229, 0x3b29, 0x622b, 0x622a, 0x622c, 0x622d,
1512 0x4869, 0x622e, 0x622f, 0x7369, 0x6230, 0x6231, 0x6232, 0x3b2e,
1513 0x6233, 0x4756, 0x4b5f, 0x314e, 0x3157, 0x6234, 0x6236, 0x6235,
1514 0x4570, 0x4039, 0x5d39, 0x6237, 0x4c41, 0x6238, 0x3446, 0x4857,
1515 0x6239, 0x623a, 0x623b, 0x4c5c, 0x4c55, 0x443e, 0x416a, 0x623d,
1516 0x3d62, 0x3e4a, 0x6240, 0x623f, 0x623e, 0x487d, 0x3447, 0x3829,
1517 0x6246, 0x6243, 0x3f3f, 0x4c32, 0x6242, 0x6244, 0x6245, 0x6241,
1518 0x6247, 0x6248, 0x442f, 0x3463, 0x4365, 0x6249, 0x624a, 0x624d,
1519 0x3f67, 0x4644, 0x624e, 0x4b53, 0x624b, 0x624c, 0x6251, 0x6250,
1520 0x624f, 0x6253, 0x6252, 0x6254, 0x6256, 0x6255, 0x4a4d, 0x3d56,
1521 0x4e46, 0x6257, 0x4637, 0x6258, 0x6259, 0x625d, 0x625b, 0x625c,
1522 0x625a, 0x625e, 0x625f, 0x6260, 0x6261, 0x4c37, 0x6262, 0x4c70,
1523 0x6263, 0x434e, 0x476a, 0x366b, 0x433b, 0x6264, 0x363a, 0x4050,
1524 0x6265, 0x3a3d, 0x6266, 0x6267, 0x3826, 0x3a55, 0x6269, 0x4556,
1525 0x3a56, 0x354e, 0x4b24, 0x474b, 0x4557, 0x395c, 0x626b, 0x3e4b,
1526 0x4e32, 0x3945, 0x3827, 0x4823, 0x626d, 0x626f, 0x386b, 0x626e,
1527 0x4476, 0x6271, 0x3337, 0x626c, 0x486a, 0x3130, 0x3a6c, 0x4f52,
1528 0x6270, 0x6272, 0x4a4b, 0x4059, 0x6274, 0x6275, 0x6273, 0x334e,
1529 0x627b, 0x627a, 0x3c27, 0x627c, 0x6277, 0x627d, 0x6278, 0x4858,
1530 0x6276, 0x6279, 0x6322, 0x6321, 0x4b61, 0x627e, 0x306b, 0x6324,
1531 0x6323, 0x3e4c, 0x6325, 0x4143, 0x6327, 0x6326, 0x6328, 0x6268,
1532 0x626a, 0x632a, 0x6329, 0x3c28, 0x4e69, 0x3c52, 0x632b, 0x3737,
1533 0x3540, 0x3527, 0x3b63, 0x4d34, 0x6331, 0x6330, 0x4144, 0x632d,
1534 0x632f, 0x3d4b, 0x3f40, 0x632e, 0x632c, 0x472a, 0x3e4d, 0x493c,
1535 0x3a57, 0x4578, 0x6332, 0x6333, 0x6349, 0x3658, 0x4f3d, 0x4135,
1536 0x6334, 0x3252, 0x4477, 0x4a21, 0x6335, 0x357a, 0x6336, 0x6338,
1537 0x6339, 0x4729, 0x633a, 0x633b, 0x633c, 0x3659, 0x3253, 0x4645,

1538 0x3d28, 0x3b64, 0x633d, 0x3d29, 0x324a, 0x4943, 0x633e, 0x486b,
1539 0x4145, 0x6341, 0x6342, 0x4769, 0x3f41, 0x633f, 0x4361, 0x6340,
1540 0x3e4e, 0x305c, 0x3529, 0x6343, 0x4478, 0x6344, 0x4047, 0x4c2d,
1541 0x4923, 0x6345, 0x6346, 0x4355, 0x4e47, 0x6348, 0x6347, 0x3c6f,
1542 0x634a, 0x3070, 0x634d, 0x634b, 0x3254, 0x374e, 0x634c, 0x394e,
1543 0x3972, 0x4a66, 0x634e, 0x4b54, 0x6350, 0x4051, 0x314f, 0x323a,
1544 0x302c, 0x634f, 0x6351, 0x6352, 0x3e77, 0x6353, 0x334f, 0x6355,
1545 0x376a, 0x3566, 0x6356, 0x3675, 0x6357, 0x407c, 0x464d, 0x4060,
1546 0x3a75, 0x6358, 0x4362, 0x416b, 0x635a, 0x635c, 0x6359, 0x635b,
1547 0x3722, 0x635d, 0x3726, 0x3567, 0x4d52, 0x635f, 0x6360, 0x312e,
1548 0x6363, 0x3376, 0x6362, 0x6361, 0x6365, 0x635e, 0x6366, 0x4e29,
1549 0x6367, 0x6368, 0x5474, 0x636a, 0x6369, 0x636b, 0x636c, 0x4e35,
1550 0x636d, 0x706f, 0x3e4f, 0x636e, 0x636f, 0x3d57, 0x4638, 0x6370,
1551 0x4328, 0x6371, 0x433c, 0x6372, 0x3625, 0x513f, 0x435d, 0x3c33,
1552 0x3448, 0x6373, 0x6422, 0x6376, 0x3568, 0x6375, 0x6424, 0x6374,
1553 0x3e50, 0x6378, 0x6379, 0x452b, 0x637a, 0x335e, 0x3f5a, 0x4964,
1554 0x637c, 0x4268, 0x6377, 0x637b, 0x637d, 0x3a7b, 0x6426, 0x492e,
1555 0x4826, 0x4579, 0x365a, 0x6425, 0x6423, 0x4835, 0x637e, 0x435e,
1556 0x457b, 0x457a, 0x3a76, 0x6438, 0x6428, 0x642a, 0x642d, 0x642e,
1557 0x642b, 0x642c, 0x6429, 0x6427, 0x6421, 0x4a4f, 0x3255, 0x6435,
1558 0x6432, 0x6437, 0x6436, 0x643e, 0x4773, 0x4c27, 0x3b3b, 0x6430, 0x6439,
1559 0x6434, 0x6433, 0x642f, 0x6431, 0x3449, 0x433d, 0x407d, 0x4822,
1560 0x643e, 0x4824, 0x4061, 0x643b, 0x484f, 0x643f, 0x4a53, 0x435b,
1561 0x643a, 0x643c, 0x643d, 0x6440, 0x3c44, 0x4646, 0x6445, 0x6444,
1562 0x6441, 0x4f36, 0x644a, 0x644e, 0x644b, 0x6447, 0x6448, 0x644d,
1563 0x6442, 0x5255, 0x6449, 0x6443, 0x644c, 0x6452, 0x344a, 0x644f,
1564 0x6450, 0x6451, 0x6454, 0x6453, 0x4876, 0x6455, 0x4e7c, 0x4a6d,
1565 0x645a, 0x6457, 0x6456, 0x4052, 0x6459, 0x645b, 0x6458, 0x645f,
1566 0x645c, 0x645d, 0x6446, 0x645e, 0x6460, 0x6461, 0x4a46, 0x6462,
1567 0x4c62, 0x364e, 0x3729, 0x6463, 0x4a34, 0x3f68, 0x4c30, 0x6464,
1568 0x4e33, 0x4774, 0x4146, 0x4734, 0x3d4d, 0x3040, 0x6469, 0x6467,
1569 0x6465, 0x3421, 0x3e51, 0x646a, 0x6468, 0x6466, 0x646e, 0x646d,
1570 0x646c, 0x646b, 0x646f, 0x6470, 0x403a, 0x6471, 0x6473, 0x6472,
1571 0x3852, 0x4138, 0x6475, 0x457c, 0x6474, 0x6476, 0x4a35, 0x416c,
1572 0x3947, 0x6477, 0x4e48, 0x6479, 0x647a, 0x647b, 0x647c, 0x3b65,
1573 0x647d, 0x374f, 0x356a, 0x352a, 0x6521, 0x4c73, 0x3948, 0x647e,
1574 0x6524, 0x4c66, 0x473c, 0x4933, 0x3d63, 0x6523, 0x3c53, 0x3949,
1575 0x3b66, 0x3569, 0x4a36, 0x6522, 0x4147, 0x4b42, 0x3a77, 0x3b67,
1576 0x445d, 0x6527, 0x4e5f, 0x3a59, 0x6528, 0x3f42, 0x652a, 0x3e52,
1577 0x3a30, 0x6529, 0x3d2a, 0x383e, 0x4148, 0x6525, 0x652b, 0x6526,
1578 0x3750, 0x652e, 0x6532, 0x376b, 0x652d, 0x6536, 0x394a, 0x4d6d,
1579 0x303c, 0x6533, 0x356b, 0x6530, 0x6531, 0x457d, 0x652f, 0x652c,
1580 0x3328, 0x4064, 0x3828, 0x6538, 0x6535, 0x6537, 0x6534, 0x3751,
1581 0x4233, 0x6539, 0x416e, 0x6546, 0x6542, 0x653c, 0x6540, 0x3c7a,
1582 0x305d, 0x653b, 0x6543, 0x6547, 0x394b, 0x4c56, 0x4456, 0x653d,
1583 0x6545, 0x653a, 0x433e, 0x653f, 0x303d, 0x4c4a, 0x653e, 0x365b,
1584 0x486c, 0x416d, 0x4e50, 0x3d6f, 0x656e, 0x6548, 0x407e, 0x6544,
1585 0x6549, 0x654b, 0x4479, 0x654e, 0x654a, 0x4a54, 0x344b, 0x4c4b,
1586 0x305e, 0x654d, 0x4e7d, 0x654c, 0x316f, 0x466c, 0x654f, 0x6556,
1587 0x6550, 0x6557, 0x6553, 0x477b, 0x3c4a, 0x6555, 0x6552, 0x6558,
1588 0x6551, 0x3d44, 0x4b25, 0x3d4c, 0x6554, 0x6560, 0x655c, 0x655f,
1589 0x655d, 0x6561, 0x655b, 0x6541, 0x4053, 0x484b, 0x655e, 0x6559,
1590 0x4121, 0x3752, 0x3d2b, 0x3f25, 0x4136, 0x6564, 0x6566, 0x6567,
1591 0x6563, 0x6565, 0x655a, 0x6562, 0x656a, 0x6569, 0x4b7a, 0x372b,
1592 0x6568, 0x656c, 0x656b, 0x656f, 0x6571, 0x3b3c, 0x656d, 0x6572,
1593 0x6573, 0x6574, 0x657a, 0x453b, 0x6576, 0x6575, 0x6577, 0x6578,
1594 0x6579, 0x657b, 0x657c, 0x344c, 0x657d, 0x657e, 0x6621, 0x6622,
1595 0x6623, 0x6624, 0x6625, 0x6626, 0x6628, 0x6627, 0x6629, 0x662a,
1596 0x662b, 0x662e, 0x662c, 0x662d, 0x3a61, 0x3753, 0x4356, 0x4833,
1597 0x3d70, 0x474d, 0x486d, 0x662f, 0x586d, 0x6630, 0x6632, 0x4d65,
1598 0x6631, 0x6634, 0x6633, 0x4d53, 0x6635, 0x487e, 0x6636, 0x6639,
1599 0x6638, 0x6637, 0x663a, 0x3732, 0x4122, 0x3541, 0x663e, 0x663b,
1600 0x663c, 0x663f, 0x6640, 0x663d, 0x3129, 0x3227, 0x6642, 0x6643,
1601 0x6644, 0x4d62, 0x3d2c, 0x6646, 0x6645, 0x3f69, 0x6647, 0x6648,
1602 0x6649, 0x3465, 0x344d, 0x664a, 0x664b, 0x4b5d, 0x4d63, 0x4d54,
1603 0x4f37, 0x394d, 0x664e, 0x3c54, 0x664d, 0x664f, 0x3c29, 0x4251,
1604 0x6650, 0x394c, 0x4c57, 0x6651, 0x6652, 0x6653, 0x6654, 0x6655,
1605 0x3c2a, 0x4c6d, 0x6657, 0x433f, 0x6656, 0x6659, 0x6658, 0x665a,
1606 0x403b, 0x665b, 0x665c, 0x4a39, 0x665d, 0x416f, 0x665e, 0x665f,
1607 0x4e7e, 0x6662, 0x6661, 0x6660, 0x4430, 0x6663, 0x3f26, 0x6664,
1608 0x6665, 0x4f38, 0x6666, 0x6667, 0x6669, 0x6668, 0x4825, 0x4679,
1609 0x4f3e, 0x4829, 0x666b, 0x3e53, 0x492a, 0x666c, 0x666a, 0x344e,
1610 0x3854, 0x3b68, 0x486e, 0x382a, 0x4b43, 0x666f, 0x666d, 0x394e,
1611 0x394f, 0x3069, 0x3a68, 0x4759, 0x305f, 0x6674, 0x4340, 0x4758,
1612 0x425b, 0x6676, 0x6672, 0x6675, 0x6670, 0x6673, 0x4b26, 0x3855,
1613 0x307d, 0x6671, 0x6678, 0x6679, 0x4639, 0x363b, 0x6726, 0x473d,
1614 0x3b69, 0x363c, 0x4048, 0x4f46, 0x4c2e, 0x6677, 0x4054, 0x3553,
1615 0x667a, 0x667c, 0x667b, 0x667d, 0x4326, 0x473e, 0x4431, 0x6723,
1616 0x6722, 0x667e, 0x3f55, 0x4965, 0x6725, 0x6724, 0x3950, 0x4f53,
1617 0x6735, 0x6729, 0x672a, 0x3c70, 0x6728, 0x3978, 0x6727, 0x672b,
1618 0x4432, 0x4a22, 0x4123, 0x425c, 0x672f, 0x6730, 0x672c, 0x672d,
1619 0x672e, 0x3951, 0x6736, 0x6732, 0x4966, 0x4b6c, 0x4928, 0x6731,
1620 0x6734, 0x6733, 0x4b44, 0x6737, 0x6738, 0x4137, 0x6739, 0x673b,
1621 0x673f, 0x673c, 0x673a, 0x473f, 0x673d, 0x673e, 0x3232, 0x6745,
1622 0x6740, 0x6741, 0x6742, 0x4221, 0x6744, 0x6743, 0x6746, 0x6747,
1623 0x6748, 0x3f43, 0x3269, 0x6749, 0x4e57, 0x3c2b, 0x3d2d, 0x3b6a,
1624 0x4357, 0x674a, 0x674b, 0x3131, 0x674c, 0x674d, 0x674e, 0x674f,

```

1625 0x6750, 0x363d, 0x5a2a, 0x6751, 0x4065, 0x6752, 0x3c4b, 0x6753,
1626 0x5030, 0x6754, 0x4a5e, 0x345c, 0x4124, 0x3d58, 0x4971, 0x3d2e,
1627 0x6755, 0x3952, 0x6756, 0x484c, 0x6764, 0x6758, 0x4249, 0x4775,
1628 0x383f, 0x6757, 0x4125, 0x6759, 0x447a, 0x675b, 0x675a, 0x675d,
1629 0x675c, 0x675e, 0x6760, 0x675f, 0x344f, 0x6761, 0x6762, 0x6763,
1630 0x3a31, 0x4e49, 0x6765, 0x6765, 0x3f27, 0x3170, 0x6766, 0x6767, 0x6768,
1631 0x3072, 0x6769, 0x676a, 0x4967, 0x3c47, 0x676c, 0x3329, 0x3032,
1632 0x676b, 0x676e, 0x474e, 0x3f44, 0x3256, 0x4b27, 0x375d, 0x365c,
1633 0x676d, 0x326a, 0x3423, 0x3171, 0x6772, 0x4e6a, 0x425d, 0x4944,
1634 0x677e, 0x3257, 0x677c, 0x677a, 0x6771, 0x676f, 0x6770, 0x3c63,
1635 0x366c, 0x4377, 0x4651, 0x3151, 0x6774, 0x6773, 0x6779, 0x6775,
1636 0x6778, 0x4c50, 0x6777, 0x3258, 0x337d, 0x677b, 0x677d, 0x3754,
1637 0x6823, 0x682c, 0x682d, 0x302b, 0x6834, 0x3071, 0x682b, 0x682a,
1638 0x6825, 0x6824, 0x6822, 0x6821, 0x4363, 0x427b, 0x6827, 0x6826,
1639 0x6829, 0x4170, 0x3755, 0x3141, 0x6828, 0x3953, 0x4171, 0x683a,
1640 0x683b, 0x3259, 0x322e, 0x6838, 0x682e, 0x6836, 0x683d, 0x6837,
1641 0x6835, 0x6776, 0x6833, 0x682f, 0x3450, 0x6831, 0x683c, 0x6832,
1642 0x683e, 0x6830, 0x477c, 0x4d69, 0x6839, 0x684f, 0x6847, 0x3f7b,
1643 0x3546, 0x365d, 0x6842, 0x325b, 0x3e54, 0x6845, 0x3a5a, 0x4551,
1644 0x684a, 0x4a6e, 0x6841, 0x325a, 0x3856, 0x4929, 0x684b, 0x683f,
1645 0x6848, 0x6852, 0x6843, 0x6844, 0x463a, 0x6849, 0x6846, 0x4b28,
1646 0x684c, 0x3060, 0x6840, 0x684e, 0x684d, 0x476b, 0x6854, 0x685f,
1647 0x337e, 0x6862, 0x6850, 0x6855, 0x4d6e, 0x685e, 0x4d55, 0x4e2a,
1648 0x4378, 0x336b, 0x4972, 0x6864, 0x4621, 0x3031, 0x685d, 0x6859,
1649 0x4172, 0x6853, 0x685b, 0x6860, 0x472c, 0x302a, 0x6858, 0x6861,
1650 0x4978, 0x685c, 0x6857, 0x3e55, 0x3d2f, 0x3c2c, 0x4c58, 0x4947,
1651 0x6867, 0x6870, 0x685a, 0x3377, 0x3e78, 0x6865, 0x686a, 0x4173,
1652 0x6866, 0x686d, 0x435f, 0x686e, 0x4d56, 0x6863, 0x3338, 0x6869,
1653 0x686c, 0x4c2c, 0x686f, 0x6868, 0x686b, 0x4b29, 0x4f21, 0x6873,
1654 0x687a, 0x6872, 0x3c43, 0x6851, 0x4a4e, 0x4c22, 0x6879, 0x6878,
1655 0x6874, 0x6875, 0x3136, 0x6877, 0x6871, 0x4455, 0x6876, 0x307e,
1656 0x4222, 0x4a43, 0x687b, 0x6921, 0x4859, 0x687e, 0x3e56, 0x3c49,
1657 0x6923, 0x363e, 0x6924, 0x4979, 0x687d, 0x6856, 0x687c, 0x4f4f,
1658 0x4622, 0x4973, 0x692b, 0x6931, 0x6932, 0x6925, 0x4776, 0x692f,
1659 0x6927, 0x6929, 0x6933, 0x6928, 0x692c, 0x3172, 0x4665, 0x692d,
1660 0x6930, 0x6926, 0x4126, 0x692a, 0x3b27, 0x3f45, 0x3730, 0x4c74,
1661 0x4c79, 0x3d72, 0x6937, 0x6935, 0x4f4e, 0x6934, 0x4d75, 0x6936,
1662 0x6938, 0x6939, 0x693c, 0x693a, 0x4623, 0x693b, 0x484d, 0x692e,
1663 0x3d73, 0x693d, 0x6942, 0x4174, 0x6941, 0x6922, 0x6943, 0x4149,
1664 0x693e, 0x6940, 0x693f, 0x5d31, 0x5d22, 0x6945, 0x6944, 0x4d76,
1665 0x623c, 0x6946, 0x6947, 0x6948, 0x3857, 0x3554, 0x694a, 0x515d,
1666 0x3575, 0x4e3a, 0x3673, 0x694b, 0x694c, 0x436e, 0x694d, 0x467a,
1667 0x303a, 0x3263, 0x6952, 0x6953, 0x694e, 0x3b3d, 0x694f, 0x4742,
1668 0x6950, 0x6951, 0x695b, 0x6955, 0x6958, 0x6954, 0x6956, 0x6957,
1669 0x3c58, 0x6959, 0x4341, 0x3756, 0x3342, 0x695c, 0x333f, 0x6961,
1670 0x695d, 0x6960, 0x483a, 0x695e, 0x695f, 0x4948, 0x485a, 0x6962,
1671 0x427d, 0x696c, 0x6968, 0x326b, 0x6966, 0x4b2a, 0x6967, 0x6964,
1672 0x6965, 0x696a, 0x696d, 0x696b, 0x6969, 0x6963, 0x4358, 0x6974,
1673 0x4c2a, 0x6972, 0x6973, 0x696e, 0x6970, 0x6971, 0x696f, 0x4066,
1674 0x4f39, 0x6978, 0x6979, 0x6a21, 0x3f2a, 0x697b, 0x697e, 0x6976,
1675 0x6975, 0x6a22, 0x325c, 0x697c, 0x6a23, 0x697d, 0x697a, 0x4433,
1676 0x6977, 0x4768, 0x6a27, 0x4d3b, 0x6a26, 0x6a25, 0x6a2e, 0x6a28,
1677 0x6a30, 0x4d66, 0x6a33, 0x6a2a, 0x6a2b, 0x6a2f, 0x6a32, 0x6a31,
1678 0x6a29, 0x6a2c, 0x6a3d, 0x6a36, 0x6a34, 0x6a35, 0x6a3a, 0x6a3b,
1679 0x332a, 0x3542, 0x6a39, 0x6a24, 0x6a38, 0x6a3c, 0x6a37, 0x6a3e,
1680 0x6a40, 0x6a3f, 0x6a42, 0x6a41, 0x695a, 0x6a46, 0x6a43, 0x6a44,
1681 0x6a45, 0x6a47, 0x376c, 0x6a49, 0x6a48, 0x3d30, 0x3954, 0x5e27,
1682 0x6a4a, 0x3d51, 0x3339, 0x6a4b, 0x3152, 0x3e57, 0x6a4c, 0x3955,
1683 0x6a4d, 0x3061, 0x493d, 0x6a4e, 0x3f6a, 0x6a55, 0x6a52, 0x436f,
1684 0x6a53, 0x6a50, 0x365e, 0x6a4f, 0x6a56, 0x3736, 0x425e, 0x6a5c,
1685 0x6a58, 0x4235, 0x6a57, 0x6a5a, 0x6a51, 0x6a5b, 0x6a5d, 0x486f,
1686 0x6a59, 0x6a5e, 0x6a60, 0x3853, 0x6a54, 0x3041, 0x6a5f, 0x3a5b,
1687 0x4e76, 0x6a61, 0x6a62, 0x4175, 0x4e22, 0x6a63, 0x4d35, 0x6a64,
1688 0x6a65, 0x4a64, 0x6a66, 0x3a40, 0x4e23, 0x6a6b, 0x6a6c, 0x3e58,
1689 0x6a6a, 0x4d67, 0x6a67, 0x6a69, 0x403d, 0x3f7e, 0x6a68, 0x6a6d,
1690 0x4a23, 0x6a6f, 0x6a6e, 0x336c, 0x4b2b, 0x6a70, 0x6a7c, 0x6a72,
1691 0x6a73, 0x6a74, 0x6a75, 0x6a79, 0x6a7a, 0x6a78, 0x6a76, 0x6a71,
1692 0x6a77, 0x6a7b, 0x7037, 0x3228, 0x6a7e, 0x365f, 0x6a7d, 0x6b22,
1693 0x6b21, 0x6b24, 0x6b23, 0x6b25, 0x3d31, 0x6b26, 0x6b27, 0x6b28,
1694 0x403e, 0x4d57, 0x6b29, 0x4a24, 0x4746, 0x6b2a, 0x6b2b, 0x382b,
1695 0x352c, 0x6b2c, 0x3b6b, 0x4741, 0x6b2d, 0x3350, 0x6b2e, 0x6b30,
1696 0x4d77, 0x6b2f, 0x3f46, 0x6b31, 0x6b32, 0x6b33, 0x3451, 0x6b34,
1697 0x6b35, 0x6b36, 0x6b37, 0x3351, 0x6b38, 0x6b39, 0x6b3a, 0x3272,
1698 0x3f28, 0x6b3b, 0x6b3c, 0x6b3d, 0x3840, 0x447b, 0x6b3e, 0x3757,
1699 0x3f56, 0x6b41, 0x4624, 0x6b40, 0x3731, 0x6b3f, 0x4277, 0x352d,
1700 0x6b42, 0x6b43, 0x3e59, 0x376d, 0x6b44, 0x4b2c, 0x405f, 0x3576,
1701 0x4c75, 0x414a, 0x6b45, 0x3f47, 0x4370, 0x3e5a, 0x6b46, 0x6b49,
1702 0x6b4a, 0x3a3e, 0x4242, 0x6b48, 0x3e5b, 0x493e, 0x6b47, 0x3b6c,
1703 0x3153, 0x6b4e, 0x3758, 0x3b6e, 0x3b6d, 0x4f4d, 0x6b4d, 0x6b4c,
1704 0x4127, 0x354d, 0x4f43, 0x333a, 0x3e5c, 0x6b4b, 0x6b50, 0x6b51,
1705 0x6b4f, 0x3858, 0x4d40, 0x3b6f, 0x4727, 0x6b54, 0x4040, 0x4342,
1706 0x4d36, 0x6b57, 0x386c, 0x403f, 0x6b53, 0x6b58, 0x386d, 0x6b55,
1707 0x6b56, 0x6b52, 0x4062, 0x4649, 0x432f, 0x325d, 0x4870, 0x3543,
1708 0x4434, 0x6b5b, 0x6b59, 0x434c, 0x4041, 0x3452, 0x6b5a, 0x3f5b,
1709 0x4e4a, 0x4f40, 0x6b5c, 0x6b67, 0x4435, 0x6b66, 0x6b63, 0x6b6b,
1710 0x6b64, 0x6b60, 0x447c, 0x6b5f, 0x6b5d, 0x4d21, 0x3b70, 0x6b61,
1711 0x6b5e, 0x6b65, 0x3d74, 0x3841, 0x427a, 0x4b45, 0x315a, 0x3062,

```

```

1712 0x4625, 0x6b69, 0x6b68, 0x4666, 0x6b6d, 0x6b62, 0x6b6c, 0x6b6e,
1713 0x382c, 0x6b6a, 0x3956, 0x3c55, 0x6b6f, 0x4d58, 0x6b72, 0x6b75,
1714 0x6b73, 0x4935, 0x6b70, 0x3660, 0x6b74, 0x6b76, 0x6b7a, 0x6b77,
1715 0x6b79, 0x6b78, 0x6b7b, 0x3c31, 0x6b7d, 0x6b7c, 0x4968, 0x6c21,
1716 0x3759, 0x6b7e, 0x6c22, 0x6c23, 0x3544, 0x6641, 0x3e79, 0x6c24,
1717 0x386e, 0x6c25, 0x6c26, 0x3b3e, 0x5a4e, 0x6c27, 0x6c28, 0x3d32,
1718 0x6c29, 0x6c2a, 0x6c2b, 0x6c2c, 0x6c2d, 0x432b, 0x6c2e, 0x6c30,
1719 0x6c2f, 0x4626, 0x6c31, 0x4b2d, 0x6c32, 0x6c33, 0x6c34, 0x6c35,
1720 0x465a, 0x3e5d, 0x6c36, 0x396b, 0x502e, 0x6c37, 0x6c38, 0x493f,
1721 0x6c39, 0x6c41, 0x6c3a, 0x6c3c, 0x6c3b, 0x6c3d, 0x4b46, 0x6c3e,
1722 0x6c3f, 0x6c40, 0x6c42, 0x332d, 0x4467, 0x4969, 0x3a62, 0x3957,
1723 0x494f, 0x325f, 0x484e, 0x6c45, 0x3453, 0x4055, 0x6c44, 0x6c49,
1724 0x4379, 0x4c63, 0x6c47, 0x6c48, 0x352e, 0x6c4a, 0x4763, 0x425f,
1725 0x4871, 0x453d, 0x6c46, 0x4b47, 0x326c, 0x6c4c, 0x4f28, 0x4442,
1726 0x4f45, 0x3b71, 0x6c4b, 0x4231, 0x6c5c, 0x4128, 0x4678, 0x4950,
1727 0x6c4f, 0x3b3f, 0x3b72, 0x3e5e, 0x4765, 0x382d, 0x6c4e, 0x6c4d,
1728 0x496a, 0x3c41, 0x4552, 0x6c51, 0x6c52, 0x3958, 0x6c50, 0x6c53,
1729 0x6c54, 0x6c56, 0x4223, 0x6c55, 0x3466, 0x6c58, 0x6c57, 0x6c59,
1730 0x6c5b, 0x6c5d, 0x6c5e, 0x4056, 0x3c4f, 0x6c5f, 0x3352, 0x6c60,
1731 0x4176, 0x6c61, 0x6c62, 0x496b, 0x352f, 0x6c63, 0x4436, 0x315b,
1732 0x6c64, 0x3c71, 0x3f76, 0x422d, 0x6c67, 0x6c66, 0x6c65, 0x6c6d,
1733 0x6c6b, 0x6c68, 0x6c6a, 0x6c69, 0x6c6c, 0x3577, 0x6c70, 0x4057,
1734 0x6c71, 0x3859, 0x6c6e, 0x6c6f, 0x4f29, 0x4437, 0x4129, 0x6c72,
1735 0x6c75, 0x6c73, 0x6c74, 0x4d59, 0x4627, 0x6c78, 0x6c76, 0x6c77,
1736 0x6c79, 0x6d29, 0x6c7c, 0x6c7d, 0x6c7b, 0x6c7a, 0x447d, 0x6d21,
1737 0x6d25, 0x6d22, 0x6c7e, 0x6d23, 0x6d24, 0x6d2b, 0x6d26, 0x4058,
1738 0x6d28, 0x6d2a, 0x6d27, 0x6d2d, 0x3d33, 0x6d2c, 0x6d2e, 0x6d2f,
1739 0x6d32, 0x6d31, 0x6d30, 0x6d34, 0x6d33, 0x4c76, 0x6d36, 0x6d35,
1740 0x6d37, 0x6d38, 0x6d3a, 0x6d39, 0x3f48, 0x6d3b, 0x366d, 0x6d3c,
1741 0x6d3e, 0x6d3f, 0x6d40, 0x6d3d, 0x6d41, 0x3c56, 0x6d42, 0x3530,
1742 0x3733, 0x382e, 0x6d43, 0x4670, 0x453e, 0x6d44, 0x6d47, 0x3c34,
1743 0x6d46, 0x6d45, 0x375a, 0x6d48, 0x3353, 0x6d4a, 0x3a5c, 0x6d49,
1744 0x6d52, 0x6d4c, 0x6d4e, 0x4a65, 0x6d4b, 0x6d4d, 0x6d51, 0x6d4f,
1745 0x3531, 0x6d50, 0x6d53, 0x475a, 0x4e58, 0x3d34, 0x6d54, 0x4d22,
1746 0x6d56, 0x6d55, 0x6d59, 0x4d41, 0x6d58, 0x336d, 0x6d57, 0x6d5c,
1747 0x6d5b, 0x6d5a, 0x4532, 0x6d5d, 0x6d5e, 0x6d5f, 0x396c, 0x3725,
1748 0x6d60, 0x6d61, 0x6d62, 0x3f49, 0x6d63, 0x3c2d, 0x6d64, 0x6d65,
1749 0x5221, 0x517e, 0x6d66, 0x6570, 0x6d67, 0x4324, 0x3f2b, 0x4740,
1750 0x6d68, 0x4a55, 0x4454, 0x397e, 0x4329, 0x312a, 0x4b78, 0x3f57,
1751 0x375e, 0x3661, 0x4a56, 0x6d69, 0x6d6b, 0x6d6a, 0x3260, 0x4676,
1752 0x6d6c, 0x4777, 0x4533, 0x6d6d, 0x3d52, 0x6d6f, 0x4c42, 0x6d7e,
1753 0x6d71, 0x6d72, 0x4449, 0x4260, 0x4177, 0x4628, 0x6d70, 0x3555,
1754 0x6d79, 0x6d76, 0x6e25, 0x4629, 0x4360, 0x6d73, 0x447e, 0x4553,
1755 0x6d74, 0x6d78, 0x3f60, 0x4767, 0x444c, 0x4042, 0x6d77, 0x422e,
1756 0x4224, 0x6d75, 0x3029, 0x4f22, 0x6d7a, 0x4261, 0x3d35, 0x3f4a,
1757 0x6d7c, 0x6d7b, 0x306f, 0x6d7d, 0x492f, 0x6e27, 0x465b, 0x3f6b,
1758 0x4359, 0x3678, 0x6e26, 0x4d37, 0x313f, 0x4a57, 0x3261, 0x6e21,
1759 0x6e22, 0x6e23, 0x6e24, 0x463b, 0x4323, 0x3063, 0x6e28, 0x6e29,
1760 0x7423, 0x423d, 0x6e2a, 0x3173, 0x414c, 0x382f, 0x4d5a, 0x6e2b,
1761 0x452c, 0x4178, 0x3c57, 0x6e2c, 0x6e2f, 0x3d65, 0x6e2d, 0x412b,
1762 0x412a, 0x3064, 0x4e4b, 0x6e31, 0x4872, 0x6e33, 0x6e32, 0x6e30,
1763 0x6364, 0x3454, 0x6d6e, 0x6e35, 0x6e34, 0x6e36, 0x4d38, 0x4661,
1764 0x4b2e, 0x6e37, 0x3c59, 0x6e38, 0x6e39, 0x6e3a, 0x4521, 0x306a,
1765 0x3959, 0x4f3a, 0x6e3e, 0x3734, 0x6e3b, 0x6e3c, 0x4974, 0x3354,
1766 0x4d39, 0x363f, 0x4554, 0x6e3f, 0x6e40, 0x6e41, 0x4522, 0x6e43,
1767 0x6e42, 0x4653, 0x6e44, 0x3d36, 0x3c60, 0x475b, 0x4371, 0x3c72,
1768 0x3f6c, 0x6e45, 0x6e46, 0x3f5d, 0x6e47, 0x6e48, 0x6e49, 0x4d6f,
1769 0x3d37, 0x6e4b, 0x6e4a, 0x395a, 0x3973, 0x3b40, 0x6e4e, 0x3d66,
1770 0x6e4d, 0x6e4c, 0x4269, 0x386f, 0x4043, 0x4830, 0x3d39, 0x6e4f,
1771 0x3e5f, 0x6e52, 0x6e50, 0x6e51, 0x6e54, 0x6e53, 0x3e7a, 0x6e55,
1772 0x6e56, 0x6e57, 0x4850, 0x3a53, 0x3c61, 0x6e58, 0x6e59, 0x4e24,
1773 0x3d45, 0x4c6e, 0x4e4c, 0x6e5a, 0x3662, 0x6e5b, 0x4523, 0x6e5e,
1774 0x3378, 0x3f4b, 0x6e5c, 0x6e5d, 0x4460, 0x4b55, 0x367c, 0x6e60,
1775 0x6e61, 0x6e5f, 0x6e63, 0x465f, 0x3343, 0x6e67, 0x6e64, 0x6e66,
1776 0x6e62, 0x6f4f, 0x6e65, 0x4e6b, 0x385a, 0x6e6f, 0x4534, 0x6e6a,
1777 0x6e6d, 0x6e6b, 0x6e70, 0x6e71, 0x6e69, 0x6e76, 0x3174, 0x6e68,
1778 0x482d, 0x6e6c, 0x3e60, 0x395b, 0x4b48, 0x3664, 0x3d46, 0x463c,
1779 0x412d, 0x6e74, 0x6e6e, 0x6e73, 0x4c43, 0x4438, 0x6e75, 0x6e72,
1780 0x412c, 0x6e79, 0x6e78, 0x6e77, 0x4b2f, 0x3d7b, 0x6e7a, 0x4a5f,
1781 0x3154, 0x4946, 0x4372, 0x3578, 0x6e7c, 0x395d, 0x3b2c, 0x6e7b,
1782 0x3f6d, 0x3f6e, 0x6f21, 0x6f23, 0x3e7b, 0x6f22, 0x6f24, 0x3653,
1783 0x4945, 0x3c62, 0x4f23, 0x6e7e, 0x3a78, 0x4f3f, 0x6f26, 0x6f25,
1784 0x6f27, 0x6e7d, 0x4669, 0x4555, 0x4457, 0x6f2c, 0x4343, 0x6f28,
1785 0x6f29, 0x372d, 0x6f2b, 0x3830, 0x6f2a, 0x3e61, 0x3379, 0x6f30,
1786 0x3a3f, 0x4179, 0x444a, 0x333b, 0x6f2e, 0x6f2f, 0x4443, 0x6f2d,
1787 0x6f31, 0x6f37, 0x6f3a, 0x6f39, 0x452d, 0x6f32, 0x6f33, 0x6f36,
1788 0x6f38, 0x3640, 0x6f3b, 0x6f35, 0x6f34, 0x6f3f, 0x6f40, 0x6f41,
1789 0x6f3e, 0x6f3d, 0x3e62, 0x462a, 0x6f3c, 0x6f45, 0x6f43, 0x6f44,
1790 0x6f42, 0x4278, 0x6f46, 0x6f47, 0x6f49, 0x3455, 0x6f48, 0x4c7a,
1791 0x6f54, 0x6f4a, 0x6f4d, 0x6f4b, 0x6f4c, 0x6f4e, 0x6f50, 0x6f51,
1792 0x6f52, 0x6f55, 0x6f53, 0x6f56, 0x6f58, 0x6f57, 0x4439, 0x4c67,
1793 0x6f59, 0x412e, 0x6f5a, 0x4a44, 0x6f5b, 0x332b, 0x313c, 0x3457,
1794 0x3456, 0x6f5c, 0x6f5d, 0x6f5e, 0x6f5f, 0x6f60, 0x3458, 0x3355,
1795 0x395e, 0x4836, 0x6f62, 0x6f61, 0x6f63, 0x315c, 0x6f66, 0x6f65,
1796 0x6f64, 0x6f67, 0x6f6a, 0x3047, 0x6f68, 0x6f6c, 0x6f6b, 0x6f6e,
1797 0x6f6d, 0x6f6f, 0x462e, 0x6f70, 0x6f71, 0x6f73, 0x6f72, 0x496c,
1798 0x6f74, 0x6f75, 0x3a65, 0x6f76, 0x6f77, 0x4b49, 0x414b, 0x3024,

```

1799 0x424b, 0x6f78, 0x496d, 0x6f7b, 0x6f79, 0x395f, 0x6f7a, 0x3842,
1800 0x4a45, 0x6f7d, 0x7021, 0x6f7e, 0x7022, 0x3121, 0x3f58, 0x3d7c,
1801 0x3459, 0x7023, 0x4766, 0x7025, 0x3122, 0x7024, 0x4444, 0x4e4d,
1802 0x462b, 0x6f7c, 0x4e26, 0x3831, 0x4d5b, 0x3679, 0x4e34, 0x3728,
1803 0x4262, 0x6721, 0x7026, 0x332c, 0x3f6f, 0x3356, 0x7028, 0x7029,
1804 0x7027, 0x3764, 0x3a5d, 0x3e63, 0x3123, 0x4e59, 0x702b, 0x6e2e,
1805 0x702a, 0x702e, 0x702c, 0x702d, 0x702f, 0x7030, 0x4e6c, 0x7031,
1806 0x7032, 0x4049, 0x483b, 0x3f7d, 0x3467, 0x4d3a, 0x326d, 0x3d38,
1807 0x385b, 0x7035, 0x7034, 0x3b73, 0x7036, 0x7033, 0x3b28, 0x703a,
1808 0x6a2d, 0x5256, 0x3f77, 0x7038, 0x4e25, 0x4671, 0x312b, 0x4063,
1809 0x3c36, 0x4a37, 0x3140, 0x4e6d, 0x4d6b, 0x703b, 0x4545, 0x3c7b,
1810 0x703c, 0x703d, 0x3f4c, 0x703e, 0x4e6e, 0x7039, 0x7040, 0x7042,
1811 0x7041, 0x703f, 0x7043, 0x7044, 0x417a, 0x3262, 0x7045, 0x4c38,
1812 0x7046, 0x7047, 0x4f2a, 0x5b31, 0x7048, 0x7049, 0x704a, 0x704e,
1813 0x704b, 0x704c, 0x704d, 0x704e, 0x704f, 0x4044, 0x4c77, 0x4045, 0x7050,
1814 0x4873, 0x7051, 0x7353, 0x4c4c, 0x7052, 0x7053, 0x7054, 0x3357,
1815 0x7056, 0x3f59, 0x7057, 0x3724, 0x7058, 0x705c, 0x705a, 0x705b,
1816 0x3373, 0x7059, 0x705d, 0x705e, 0x3048, 0x705f, 0x7060, 0x3e64,
1817 0x7061, 0x3547, 0x7064, 0x7063, 0x7062, 0x6b71, 0x4a5c, 0x7065,
1818 0x7066, 0x7067, 0x7068, 0x7069, 0x706a, 0x345a, 0x706b, 0x706c,
1819 0x4723, 0x706e, 0x323b, 0x7071, 0x7070, 0x3124, 0x3641, 0x4a47,
1820 0x443a, 0x3a22, 0x3960, 0x3d67, 0x3f5c, 0x7073, 0x7072, 0x4d42,
1821 0x3468, 0x4852, 0x465c, 0x3f7c, 0x4e4e, 0x375b, 0x7076, 0x7075,
1822 0x4b4b, 0x462c, 0x462c, 0x3150, 0x7077, 0x7074, 0x4951, 0x4d6a, 0x7078,
1823 0x7079, 0x707b, 0x426a, 0x335b, 0x335c, 0x707a, 0x3469, 0x3832,
1824 0x346a, 0x453f, 0x4e60, 0x385c, 0x707c, 0x707d, 0x707e, 0x7121,
1825 0x7123, 0x7122, 0x4977, 0x7124, 0x7125, 0x7126, 0x7127, 0x7129,
1826 0x7128, 0x712a, 0x4874, 0x664c, 0x3f29, 0x3532, 0x712b, 0x712c,
1827 0x522c, 0x5d3b, 0x4853, 0x307b, 0x303b, 0x3b74, 0x4b30, 0x3e7e,
1828 0x712d, 0x4c5f, 0x712e, 0x4d5c, 0x3142, 0x3b41, 0x712f, 0x326e,
1829 0x7130, 0x7131, 0x7133, 0x7134, 0x7136, 0x7132, 0x7135, 0x345b,
1830 0x7137, 0x7138, 0x7139, 0x713a, 0x713b, 0x713d, 0x713c, 0x713f,
1831 0x7142, 0x713e, 0x7140, 0x7141, 0x7143, 0x3642, 0x3c73, 0x7144,
1832 0x7145, 0x3961, 0x7146, 0x333e, 0x474f, 0x7147, 0x7148, 0x435a,
1833 0x466b, 0x7149, 0x477d, 0x424c, 0x3158, 0x366e, 0x366f, 0x4373,
1834 0x714e, 0x3670, 0x326f, 0x714d, 0x714b, 0x714c, 0x714a, 0x7158,
1835 0x714f, 0x7150, 0x7151, 0x7152, 0x7154, 0x7153, 0x3d59, 0x7155,
1836 0x7157, 0x3533, 0x7156, 0x417b, 0x3833, 0x7159, 0x424d, 0x715a,
1837 0x462d, 0x715b, 0x7160, 0x715e, 0x715d, 0x715f, 0x715c, 0x7162,
1838 0x7161, 0x7164, 0x3643, 0x7163, 0x7165, 0x7166, 0x7168, 0x7167,
1839 0x7169, 0x716b, 0x716a, 0x397c, 0x716c, 0x716d, 0x333c, 0x716e,
1840 0x716f, 0x3f71, 0x7170, 0x7171, 0x7172, 0x7173, 0x3962, 0x7174,
1841 0x7175, 0x7176, 0x7177, 0x7178, 0x4831, 0x717a, 0x4926, 0x717b,
1842 0x7179, 0x717d, 0x717c, 0x717e, 0x7221, 0x7222, 0x7223, 0x7224,
1843 0x7225, 0x7226, 0x7227, 0x7228, 0x7229, 0x722a, 0x722b, 0x722c,
1844 0x722d, 0x722e, 0x5d35, 0x722f, 0x6478, 0x3534, 0x3321, 0x3a32,
1845 0x7231, 0x7230, 0x4c25, 0x7233, 0x7234, 0x7232, 0x7235, 0x4b62,
1846 0x7236, 0x357b, 0x4f25, 0x7237, 0x7239, 0x303e, 0x723a, 0x4a2b,
1847 0x7238, 0x723b, 0x723c, 0x723d, 0x723e, 0x723f, 0x4b6e, 0x3b2d,
1848 0x3a7a, 0x412f, 0x7240, 0x7243, 0x7241, 0x7244, 0x3871, 0x7242,
1849 0x7245, 0x7246, 0x7247, 0x724b, 0x3b2a, 0x4264, 0x724c, 0x7249,
1850 0x7248, 0x724a, 0x375f, 0x7250, 0x724f, 0x724e, 0x3033, 0x725a,
1851 0x7256, 0x7257, 0x7253, 0x7259, 0x7255, 0x3362, 0x4f4c, 0x7258,
1852 0x7254, 0x7252, 0x7251, 0x725c, 0x725e, 0x725d, 0x4949,
1853 0x725b, 0x3073, 0x7260, 0x7262, 0x336f, 0x724d, 0x3137, 0x7264,
1854 0x7263, 0x7261, 0x432d, 0x4b70, 0x4e5a, 0x7265, 0x7266, 0x7267,
1855 0x7268, 0x7269, 0x443b, 0x726a, 0x4837, 0x726f, 0x726b, 0x726c,
1856 0x4b31, 0x4c44, 0x4650, 0x7270, 0x7271, 0x463e, 0x726e, 0x726d,
1857 0x322a, 0x7279, 0x7278, 0x3175, 0x7276, 0x7275, 0x7273, 0x337b,
1858 0x7272, 0x3c32, 0x3229, 0x3963, 0x727c, 0x727b, 0x727a, 0x7277,
1859 0x727d, 0x727e, 0x7325, 0x7324, 0x7326, 0x312d, 0x7321, 0x7322,
1860 0x3974, 0x4c39, 0x7323, 0x4b32, 0x732b, 0x7327, 0x732c, 0x7329,
1861 0x7328, 0x375c, 0x732d, 0x732e, 0x732f, 0x732a, 0x7274, 0x7330,
1862 0x4461, 0x7334, 0x7335, 0x7333, 0x7332, 0x7338, 0x7331, 0x7336,
1863 0x7337, 0x733a, 0x7339, 0x733c, 0x733d, 0x733e, 0x4f49, 0x733b,
1864 0x426b, 0x3a6d, 0x733f, 0x7340, 0x7341, 0x7342, 0x7343, 0x3834,
1865 0x7344, 0x7345, 0x3c2f, 0x7346, 0x7347, 0x7348, 0x7349, 0x734c,
1866 0x734a, 0x4f3c, 0x734b, 0x4e6f, 0x734d, 0x4e5b, 0x734e, 0x477e,
1867 0x734f, 0x7351, 0x7352, 0x7350, 0x396d, 0x4c4d, 0x4b63, 0x5677,
1868 0x5d60, 0x4b7b, 0x322b, 0x7354, 0x3550, 0x7355, 0x7356, 0x7357,
1869 0x3975, 0x7358, 0x6054, 0x4c5b, 0x4263, 0x7359, 0x735b, 0x735a,
1870 0x735c, 0x735d, 0x735e, 0x735f, 0x7360, 0x7361, 0x7362, 0x7363,
1871 0x7364, 0x7365, 0x7366, 0x7367, 0x7368, 0x4524, 0x385d, 0x736a,
1872 0x414d, 0x736b, 0x736c, 0x4921, 0x736d, 0x736e, 0x6337, 0x6c5a,
1873 0x706d, 0x736f, 0x7370, 0x7372, 0x7373, 0x7374, 0x4e70, 0x7371,
1874 0x7375, 0x7376, 0x7378, 0x7377, 0x737a, 0x737b, 0x7379, 0x4e36,
1875 0x737c, 0x737d, 0x6354, 0x737e, 0x212a, 0x2174, 0x2170, 0x2173,
1876 0x2175, 0x214a, 0x214b, 0x2176, 0x215c, 0x2124, 0x2125, 0x213f,
1877 0x2330, 0x2331, 0x2332, 0x2333, 0x2334, 0x2335, 0x2336, 0x2337,
1878 0x2338, 0x2339, 0x2127, 0x2128, 0x2163, 0x2161, 0x2164, 0x2129,
1879 0x2177, 0x2341, 0x2342, 0x2343, 0x2344, 0x2345, 0x2346, 0x2347,
1880 0x2348, 0x2349, 0x234a, 0x234b, 0x234c, 0x234d, 0x234e, 0x234f,
1881 0x2350, 0x2351, 0x2352, 0x2353, 0x2354, 0x2355, 0x2356, 0x2357,
1882 0x2358, 0x2359, 0x235a, 0x214e, 0x214f, 0x2130, 0x2132, 0x212e,
1883 0x2361, 0x2362, 0x2363, 0x2364, 0x2365, 0x2366, 0x2367, 0x2368,
1884 0x2369, 0x236a, 0x236b, 0x236c, 0x236d, 0x236e, 0x236f, 0x2370,
1885 0x2371, 0x2372, 0x2373, 0x2374, 0x2375, 0x2376, 0x2377, 0x2378,


```
1886   0x2379, 0x237a, 0x2150, 0x2143, 0x2151, 0x2131, 0x216f,
1887 };
1888
1889 static const Summary16 jisx0208_uni2indx_page00[16] = {
1890     /* 0x0000 */
1891     { 0, 0x0000 }, { 0, 0x0000 }, { 0, 0x0000 }, { 0, 0x0000 },
1892     { 0, 0x0000 }, { 0, 0x1000 }, { 1, 0x0000 }, { 1, 0x0000 },
1893     { 1, 0x0000 }, { 1, 0x0000 }, { 1, 0x118c }, { 6, 0x0053 },
1894     { 10, 0x0000 }, { 10, 0x0080 }, { 11, 0x0000 }, { 11, 0x0080 },
1895 };
1896 static const Summary16 jisx0208_uni2indx_page03[22] = {
1897     /* 0x0300 */
1898     { 12, 0x0000 }, { 12, 0x0000 }, { 12, 0x0000 }, { 12, 0x0000 },
1899     { 12, 0x0000 }, { 12, 0x0000 }, { 12, 0x0000 }, { 12, 0x0000 },
1900     { 12, 0x0000 }, { 12, 0xffff }, { 27, 0x03fb }, { 36, 0xffff },
1901     { 51, 0x03fb }, { 60, 0x0000 }, { 60, 0x0000 }, { 60, 0x0000 },
1902     /* 0x0400 */
1903     { 60, 0x0002 }, { 61, 0xffff }, { 77, 0xffff }, { 93, 0xffff },
1904     { 109, 0xffff }, { 125, 0x0002 },
1905 };
1906 static const Summary16 jisx0208_uni2indx_page20[50] = {
1907     /* 0x2000 */
1908     { 126, 0x0000 }, { 126, 0x3361 }, { 133, 0x0063 }, { 137, 0x080d },
1909     { 141, 0x0000 }, { 141, 0x0000 }, { 141, 0x0000 }, { 141, 0x0000 },
1910     { 141, 0x0000 }, { 141, 0x0000 }, { 141, 0x0000 }, { 141, 0x0000 },
1911     { 141, 0x0000 }, { 141, 0x0000 }, { 141, 0x0000 }, { 141, 0x0000 },
1912     /* 0x2100 */
1913     { 141, 0x0008 }, { 142, 0x0000 }, { 142, 0x0800 }, { 143, 0x0000 },
1914     { 143, 0x0000 }, { 143, 0x0000 }, { 143, 0x0000 }, { 143, 0x0000 },
1915     { 143, 0x0000 }, { 143, 0x000f }, { 147, 0x0000 }, { 147, 0x0000 },
1916     { 147, 0x0000 }, { 147, 0x0014 }, { 149, 0x0000 }, { 149, 0x0000 },
1917     /* 0x2200 */
1918     { 149, 0x098d }, { 155, 0x6404 }, { 159, 0x1f81 }, { 166, 0x2030 },
1919     { 169, 0x0000 }, { 169, 0x0004 }, { 170, 0x0cc3 }, { 176, 0x0000 },
1920     { 176, 0x00cc }, { 180, 0x0000 }, { 180, 0x0020 }, { 181, 0x0000 },
1921     { 181, 0x0000 }, { 181, 0x0000 }, { 181, 0x0000 }, { 181, 0x0000 },
1922     /* 0x2300 */
1923     { 181, 0x0000 }, { 181, 0x0004 },
1924 };
1925 static const Summary16 jisx0208_uni2indx_page25[23] = {
1926     /* 0x2500 */
1927     { 182, 0x900f }, { 188, 0x3999 }, { 196, 0x9939 }, { 204, 0x9999 },
1928     { 212, 0x0804 }, { 214, 0x0000 }, { 214, 0x0000 }, { 214, 0x0000 },
1929     { 214, 0x0000 }, { 214, 0x0000 }, { 214, 0x0003 }, { 216, 0x300c },
1930     { 220, 0xc8c0 }, { 225, 0x0000 }, { 225, 0x8000 }, { 226, 0x0000 },
1931     /* 0x2600 */
1932     { 226, 0x0060 }, { 228, 0x0000 }, { 228, 0x0000 }, { 228, 0x0000 },
1933     { 228, 0x0005 }, { 230, 0x0000 }, { 230, 0xa400 },
1934 };
1935 static const Summary16 jisx0208_uni2indx_page30[16] = {
1936     /* 0x3000 */
1937     { 233, 0xffff }, { 248, 0x103f }, { 255, 0x0000 }, { 255, 0x0000 },
1938     { 255, 0xffff }, { 270, 0xffff }, { 286, 0xffff }, { 302, 0xffff },
1939     { 318, 0xffff }, { 334, 0x780f }, { 342, 0xffff }, { 357, 0xffff },
1940     { 373, 0xffff }, { 389, 0xffff }, { 405, 0xffff }, { 421, 0x787f },
1941 };
1942 static const Summary16 jisx0208_uni2indx_page4e[1307] = {
1943     /* 0x4e00 */
1944     { 432, 0x6f8b }, { 442, 0x43f3 }, { 451, 0x2442 }, { 455, 0x9b46 },
1945     { 463, 0xe82c }, { 470, 0xe3e0 }, { 478, 0x0004 }, { 479, 0x400a },
1946     { 482, 0x5f65 }, { 492, 0xdb36 }, { 502, 0x7977 }, { 513, 0x0449 },
1947     { 517, 0xecd7 }, { 528, 0xe3f0 }, { 537, 0x6038 }, { 542, 0x08c5 },
1948     /* 0x4f00 */
1949     { 547, 0xe602 }, { 553, 0x3403 }, { 558, 0x8000 }, { 559, 0x3551 },
1950     { 566, 0xe0c8 }, { 572, 0x7eab }, { 583, 0x8200 }, { 585, 0x9869 },
1951     { 592, 0xa948 }, { 598, 0x2942 }, { 603, 0xe803 }, { 609, 0x8060 },
1952     { 612, 0x441c }, { 617, 0xad93 }, { 626, 0xc03a }, { 632, 0x4568 },
1953     /* 0x5000 */
1954     { 638, 0xaa60 }, { 644, 0x8656 }, { 651, 0x3f7a }, { 662, 0x0240 },
1955     { 664, 0x8388 }, { 669, 0x1461 }, { 674, 0x1020 }, { 676, 0x2174 },
1956     { 682, 0x2021 }, { 685, 0x0702 }, { 689, 0x3000 }, { 691, 0x40bc },
1957     { 697, 0xa624 }, { 703, 0x4462 }, { 708, 0x60a8 }, { 713, 0x0a20 },
1958     /* 0x5100 */
1959     { 716, 0x0217 }, { 721, 0x8574 }, { 728, 0x0402 }, { 730, 0x9c84 },
1960     { 736, 0x7bfb }, { 749, 0x1415 }, { 754, 0x7f24 }, { 763, 0x11e2 },
1961     { 769, 0xb665 }, { 778, 0x02ef }, { 786, 0x1f75 }, { 796, 0x20ff },
1962     { 805, 0x3a70 }, { 812, 0x3840 }, { 816, 0x26c3 }, { 823, 0x6763 },
1963     /* 0x5200 */
1964     { 832, 0x4dd9 }, { 841, 0x2092 }, { 845, 0x46b0 }, { 851, 0x0fc9 },
1965     { 859, 0xbc98 }, { 867, 0x4850 }, { 871, 0x8638 }, { 877, 0xa03f },
1966     { 885, 0x2388 }, { 890, 0x8816 }, { 895, 0x3e09 }, { 902, 0x5232 },
1967     { 908, 0x22aa }, { 914, 0xe3a4 }, { 922, 0x00dd }, { 928, 0xc72c },
1968     /* 0x5300 */
1969     { 936, 0xa166 }, { 943, 0x26e1 }, { 950, 0x840b }, { 955, 0x8f0a },
1970     { 962, 0x27eb }, { 972, 0x559e }, { 981, 0xc241 }, { 986, 0x89bb },
1971     { 995, 0x0014 }, { 997, 0x8540 }, { 1001, 0x6361 }, { 1008, 0x0849 },
1972     { 1012, 0x7f0c }, { 1021, 0x8ad0 }, { 1027, 0xff3e }, { 1040, 0x05cf },
```



```
2060 { 2494, 0xa120 }, { 2498, 0x63e2 }, { 2506, 0x104c }, { 2510, 0x01b5 },
2061 { 2516, 0x538c }, { 2523, 0x9a83 }, { 2530, 0xb8b2 }, { 2538, 0x3281 },
2062 { 2543, 0x987a }, { 2551, 0x0a84 }, { 2555, 0x33e7 }, { 2565, 0x0c02 },
2063 /* 0x6600 */
2064 { 2568, 0xd4cc }, { 2576, 0x9018 }, { 2580, 0xa1a1 }, { 2586, 0x9070 },
2065 { 2591, 0x8a1e }, { 2598, 0xe004 }, { 2602, 0xc3d4 }, { 2610, 0x0451 },
2066 { 2614, 0x439a }, { 2621, 0x21c2 }, { 2626, 0x4844 }, { 2630, 0x5310 },
2067 { 2635, 0x0292 }, { 2639, 0x3640 }, { 2644, 0x0241 }, { 2647, 0xf3bd },
2068 /* 0x6700 */
2069 { 2659, 0xab09 }, { 2666, 0xe8f0 }, { 2674, 0x7dc0 }, { 2682, 0xa5d2 },
2070 { 2690, 0xc242 }, { 2695, 0xd24b }, { 2703, 0xa43f }, { 2712, 0xd0af },
2071 { 2721, 0x1aa0 }, { 2726, 0x34a1 }, { 2732, 0x8247 }, { 2738, 0x03d8 },
2072 { 2744, 0xc452 }, { 2750, 0x651b }, { 2758, 0xd294 }, { 2765, 0xc83a },
2073 /* 0x6800 */
2074 { 2772, 0x001c }, { 2775, 0x40c8 }, { 2779, 0x0e06 }, { 2784, 0x3314 },
2075 { 2790, 0x614f }, { 2798, 0xb21b }, { 2806, 0x0088 }, { 2808, 0xc0d0 },
2076 { 2813, 0xa02a }, { 2818, 0xa898 }, { 2824, 0xa1c5 }, { 2831, 0x166b },
2077 { 2839, 0x2e50 }, { 2845, 0x85b4 }, { 2852, 0xc08b }, { 2858, 0x0604 },
2078 /* 0x6900 */
2079 { 2861, 0xf933 }, { 2871, 0x1e04 }, { 2876, 0x056e }, { 2883, 0xa251 },
2080 { 2889, 0x0400 }, { 2890, 0x7638 }, { 2898, 0xec07 }, { 2906, 0x73b8 },
2081 { 2915, 0x4406 }, { 2919, 0x1832 }, { 2924, 0x4081 }, { 2927, 0xc816 },
2082 { 2933, 0x7c8a }, { 2941, 0x6309 }, { 2947, 0x2980 }, { 2951, 0xaa04 },
2083 /* 0x6a00 */
2084 { 2956, 0x1c24 }, { 2961, 0xca9c }, { 2969, 0x4e0e }, { 2976, 0x2760 },
2085 { 2982, 0x0990 }, { 2986, 0x8300 }, { 2989, 0x0046 }, { 2992, 0x8104 },
2086 { 2995, 0x6011 }, { 2999, 0x1081 }, { 3002, 0x54d0 }, { 3008, 0x0908 },
2087 { 3011, 0x000e }, { 3014, 0xcc0a }, { 3020, 0x0500 }, { 3022, 0x0c00 },
2088 /* 0x6b00 */
2089 { 3024, 0x0430 }, { 3027, 0xa044 }, { 3031, 0x008b }, { 3035, 0x6784 },
2090 { 3042, 0x5288 }, { 3047, 0x8a19 }, { 3053, 0x865e }, { 3061, 0x8b18 },
2091 { 3067, 0x2e59 }, { 3075, 0x4160 }, { 3079, 0x8c10 }, { 3083, 0x9cbe },
2092 { 3093, 0x6861 }, { 3099, 0x891c }, { 3105, 0x9800 }, { 3108, 0x0008 },
2093 /* 0x6c00 */
2094 { 3109, 0x8100 }, { 3111, 0x089a }, { 3116, 0x0018 }, { 3118, 0x4190 },
2095 { 3122, 0x4007 }, { 3126, 0xe4a1 }, { 3133, 0x0505 }, { 3137, 0x640d },
2096 { 3143, 0x310e }, { 3149, 0x0e4d }, { 3156, 0x4806 }, { 3160, 0xff0a },
2097 { 3170, 0x1632 }, { 3176, 0x2aa8 }, { 3182, 0x852e }, { 3189, 0x000b },
2098 /* 0x6d00 */
2099 { 3192, 0x1800 }, { 3194, 0xca84 }, { 3200, 0x0e20 }, { 3204, 0x696c },
2100 { 3212, 0x0032 }, { 3215, 0x1600 }, { 3218, 0x5658 }, { 3225, 0x0390 },
2101 { 3229, 0x5120 }, { 3233, 0x1a28 }, { 3238, 0x8000 }, { 3239, 0x1124 },
2102 { 3243, 0x18e1 }, { 3249, 0x4326 }, { 3255, 0x5d52 }, { 3263, 0x0eaa },
2103 /* 0x6e00 */
2104 { 3270, 0x0fa0 }, { 3276, 0xae28 }, { 3283, 0xfa7b }, { 3295, 0x4500 },
2105 { 3298, 0x6408 }, { 3302, 0x8940 }, { 3306, 0xc880 }, { 3310, 0xc044 },
2106 { 3314, 0x9005 }, { 3318, 0xb141 }, { 3324, 0x8424 }, { 3328, 0x24c4 },
2107 { 3333, 0x1a34 }, { 3339, 0x603a }, { 3345, 0x9000 }, { 3347, 0xc194 },
2108 /* 0x6f00 */
2109 { 3353, 0x8246 }, { 3358, 0x003a }, { 3362, 0x180d }, { 3367, 0xc106 },
2110 { 3372, 0x0022 }, { 3374, 0x9910 }, { 3379, 0xe050 }, { 3384, 0x1511 },
2111 { 3389, 0x4057 }, { 3395, 0x0082 }, { 3397, 0x041a }, { 3401, 0x020a },
2112 { 3404, 0x004f }, { 3409, 0x8930 }, { 3414, 0xd813 }, { 3421, 0x444a },
2113 /* 0x7000 */
2114 { 3426, 0x8a02 }, { 3430, 0xed22 }, { 3438, 0x10c0 }, { 3441, 0x4005 },
2115 { 3444, 0x1000 }, { 3445, 0x0102 }, { 3447, 0x8808 }, { 3450, 0x3101 },
2116 { 3454, 0x4600 }, { 3457, 0x0204 }, { 3459, 0xf000 }, { 3463, 0x0708 },
2117 { 3467, 0x8900 }, { 3470, 0xa200 }, { 3473, 0x0000 }, { 3473, 0x2202 },
2118 /* 0x7100 */
2119 { 3476, 0x0200 }, { 3477, 0x1610 }, { 3481, 0x0042 }, { 3483, 0x1040 },
2120 { 3485, 0x5200 }, { 3488, 0x0260 }, { 3491, 0x52f4 }, { 3499, 0x2000 },
2121 { 3500, 0x8510 }, { 3504, 0x8230 }, { 3508, 0x1100 }, { 3510, 0x4202 },
2122 { 3513, 0x4308 }, { 3517, 0x80b5 }, { 3523, 0x70e1 }, { 3530, 0x9a20 },
2123 /* 0x7200 */
2124 { 3535, 0x2040 }, { 3537, 0x0801 }, { 3539, 0x3500 }, { 3543, 0xfc65 },
2125 { 3553, 0x19c1 }, { 3559, 0xab04 }, { 3565, 0x0286 }, { 3569, 0x6214 },
2126 { 3574, 0x0087 }, { 3578, 0x0044 }, { 3580, 0x9085 }, { 3585, 0x0244 },
2127 { 3588, 0x405c }, { 3593, 0x0a85 }, { 3598, 0x3207 }, { 3604, 0x3380 },
2128 /* 0x7300 */
2129 { 3609, 0x0400 }, { 3610, 0xb8c0 }, { 3616, 0xce20 }, { 3622, 0xc0d0 },
2130 { 3627, 0xc030 }, { 3631, 0x0080 }, { 3632, 0x0508 }, { 3635, 0x0d25 },
2131 { 3641, 0x0a90 }, { 3645, 0x0040 }, { 3646, 0x0200 }, { 3647, 0x080c },
2132 { 3650, 0x6505 }, { 3656, 0x4000 }, { 3657, 0x6421 }, { 3662, 0x4102 },
2133 /* 0x7400 */
2134 { 3665, 0x0268 }, { 3669, 0x0000 }, { 3669, 0x0024 }, { 3671, 0x847c },
2135 { 3678, 0x0002 }, { 3679, 0xde20 }, { 3686, 0x8619 }, { 3692, 0x4049 },
2136 { 3696, 0x0808 }, { 3698, 0x4000 }, { 3699, 0x0084 }, { 3701, 0x2001 },
2137 { 3703, 0x8400 }, { 3705, 0x1010 }, { 3707, 0x42cd }, { 3714, 0x01c7 },
2138 /* 0x7500 */
2139 { 3720, 0x7038 }, { 3726, 0xd52a }, { 3734, 0x1968 }, { 3740, 0x1d8f },
2140 { 3749, 0xbe50 }, { 3757, 0x3e12 }, { 3764, 0x2ef5 }, { 3774, 0x81d9 },
2141 { 3781, 0xcec4 }, { 3789, 0x2412 }, { 3793, 0x0828 }, { 3796, 0x732e },
2142 { 3805, 0x24ac }, { 3811, 0x4b34 }, { 3818, 0x020c }, { 3821, 0xd41d },
2143 /* 0x7600 */
2144 { 3829, 0x2a02 }, { 3833, 0x8000 }, { 3834, 0x0097 }, { 3839, 0x0811 },
2145 { 3842, 0x11c4 }, { 3847, 0x1144 }, { 3851, 0x178e }, { 3858, 0x7d45 },
2146 { 3867, 0x49d9 }, { 3875, 0x0649 }, { 3880, 0x4000 }, { 3881, 0x8791 },
```

```
2147 { 3888, 0x254c }, { 3894, 0xd8c4 }, { 3901, 0x44ba }, { 3908, 0x4914 },
2148 /* 0x7700 */
2149 { 3913, 0x1b92 }, { 3920, 0xc800 }, { 3923, 0x0271 }, { 3928, 0x1580 },
2150 { 3932, 0x0081 }, { 3934, 0x0c00 }, { 3936, 0x096a }, { 3942, 0xc200 },
2151 { 3945, 0x4800 }, { 3947, 0x4002 }, { 3949, 0x3021 }, { 3953, 0xba49 },
2152 { 3961, 0x2080 }, { 3963, 0x1c80 }, { 3967, 0xe2ac }, { 3975, 0x1008 },
2153 /* 0x7800 */
2154 { 3977, 0x1004 }, { 3979, 0x0034 }, { 3982, 0x00e1 }, { 3986, 0x8414 },
2155 { 3990, 0x0020 }, { 3991, 0x2000 }, { 3992, 0x9800 }, { 3995, 0x1014 },
2156 { 3998, 0x70c2 }, { 4004, 0x04aa }, { 4009, 0x8688 }, { 4014, 0x5420 },
2157 { 4018, 0x0c62 }, { 4023, 0x0413 }, { 4027, 0x9180 }, { 4031, 0x2010 },
2158 /* 0x7900 */
2159 { 4033, 0x4082 }, { 4036, 0x0206 }, { 4039, 0x1c40 }, { 4043, 0x5400 },
2160 { 4046, 0x0383 }, { 4051, 0xe4e9 }, { 4060, 0x2125 }, { 4065, 0x8480 },
2161 { 4068, 0xe433 }, { 4076, 0x2000 }, { 4077, 0x44c0 }, { 4081, 0xe609 },
2162 { 4088, 0x0a03 }, { 4092, 0x8126 }, { 4097, 0x12da }, { 4104, 0x0801 },
2163 /* 0x7a00 */
2164 { 4106, 0x6901 }, { 4111, 0x9790 }, { 4118, 0x4001 }, { 4120, 0xf886 },
2165 { 4128, 0xe24d }, { 4136, 0x0081 }, { 4138, 0x0a0e }, { 4143, 0xa651 },
2166 { 4150, 0x011a }, { 4154, 0x81ec }, { 4161, 0xc600 }, { 4165, 0x8441 },
2167 { 4169, 0xad8 }, { 4178, 0xb62c }, { 4186, 0xa46f }, { 4195, 0x8741 },
2168 /* 0x7b00 */
2169 { 4201, 0x8d54 }, { 4208, 0x4b02 }, { 4213, 0x1161 }, { 4218, 0x0268 },
2170 { 4222, 0xbb60 }, { 4230, 0x2057 }, { 4236, 0x50a0 }, { 4240, 0x0433 },
2171 { 4245, 0xa8c0 }, { 4250, 0xb7b4 }, { 4260, 0x2402 }, { 4263, 0x0112 },
2172 { 4266, 0x9ad3 }, { 4275, 0x2000 }, { 4276, 0x2271 }, { 4282, 0x00c8 },
2173 /* 0x7c00 */
2174 { 4285, 0x2081 }, { 4288, 0x809e }, { 4294, 0x0c8a }, { 4299, 0xe180 },
2175 { 4304, 0xb009 }, { 4309, 0x8151 }, { 4314, 0x1031 }, { 4318, 0x4028 },
2176 { 4321, 0x2a0e }, { 4327, 0x89a5 }, { 4334, 0x69b6 }, { 4343, 0x620e },
2177 { 4349, 0x4425 }, { 4354, 0xd144 }, { 4360, 0x8085 }, { 4364, 0x4d54 },
2178 /* 0x7d00 */
2179 { 4371, 0x2c75 }, { 4379, 0x1fb1 }, { 4388, 0xd807 }, { 4395, 0x862d },
2180 { 4402, 0xd87c }, { 4411, 0x4841 }, { 4415, 0x414e }, { 4421, 0x226e },
2181 { 4428, 0x8200 }, { 4430, 0x9e08 }, { 4436, 0xf80c }, { 4443, 0xed37 },
2182 { 4454, 0x8c80 }, { 4458, 0x7526 }, { 4466, 0x9313 }, { 4473, 0x0814 },
2183 /* 0x7e00 */
2184 { 4476, 0x0e32 }, { 4482, 0xc804 }, { 4486, 0x484e }, { 4492, 0x6ea6 },
2185 { 4501, 0x2c4a }, { 4507, 0x6670 }, { 4514, 0x26c0 }, { 4519, 0xba01 },
2186 { 4525, 0xd30c }, { 4532, 0x185d }, { 4539, 0x0000 }, { 4539, 0x0000 },
2187 { 4539, 0x0000 }, { 4539, 0x0000 }, { 4539, 0x0000 }, { 4539, 0x0000 },
2188 /* 0x7f00 */
2189 { 4539, 0x0000 }, { 4539, 0x0000 }, { 4539, 0x0000 }, { 4539, 0x0540 },
2190 { 4542, 0x7020 }, { 4546, 0x8133 }, { 4552, 0x4f81 }, { 4559, 0x03a5 },
2191 { 4565, 0x55ec }, { 4574, 0x6410 }, { 4578, 0xc318 }, { 4584, 0x2344 },
2192 { 4589, 0x1462 }, { 4594, 0x0034 }, { 4597, 0x0a43 }, { 4602, 0x1a09 },
2193 /* 0x8000 */
2194 { 4607, 0x187b }, { 4615, 0x13a5 }, { 4622, 0x0102 }, { 4624, 0xa848 },
2195 { 4629, 0x0440 }, { 4631, 0xc544 }, { 4637, 0x8106 }, { 4641, 0xe2dd },
2196 { 4651, 0x1af0 }, { 4658, 0x2d48 }, { 4664, 0xb626 }, { 4672, 0x0416 },
2197 { 4676, 0x5058 }, { 4681, 0x6e40 }, { 4687, 0x8032 }, { 4691, 0x3112 },
2198 /* 0x8100 */
2199 { 4696, 0x07e4 }, { 4703, 0x0c00 }, { 4705, 0x8208 }, { 4708, 0x420a },
2200 { 4712, 0x4840 }, { 4715, 0x803b }, { 4721, 0x4860 }, { 4725, 0x8713 },
2201 { 4732, 0x850d }, { 4738, 0x3428 }, { 4743, 0x0319 }, { 4748, 0xe529 },
2202 { 4756, 0x2345 }, { 4762, 0x870a }, { 4768, 0x25a9 }, { 4775, 0x5c18 },
2203 /* 0x8200 */
2204 { 4781, 0x77a6 }, { 4791, 0xd9c5 }, { 4800, 0x5e00 }, { 4805, 0x03e8 },
2205 { 4811, 0x0081 }, { 4813, 0xa700 }, { 4818, 0xcd54 }, { 4826, 0x41c6 },
2206 { 4832, 0x2800 }, { 4834, 0xa204 }, { 4838, 0xb860 }, { 4844, 0x2b0a },
2207 { 4850, 0x0020 }, { 4851, 0xda9e }, { 4861, 0x08ea }, { 4867, 0x0e1a },
2208 /* 0x8300 */
2209 { 4873, 0x427c }, { 4880, 0x11c0 }, { 4884, 0x8908 }, { 4888, 0x0376 },
2210 { 4895, 0x8621 }, { 4900, 0x0105 }, { 4903, 0x0000 }, { 4903, 0x18a8 },
2211 { 4908, 0x46a0 }, { 4913, 0xc448 }, { 4918, 0xd005 }, { 4923, 0x2022 },
2212 { 4926, 0x5422 }, { 4931, 0x9148 }, { 4936, 0x8a01 }, { 4940, 0x2897 },
2213 /* 0x8400 */
2214 { 4947, 0x7898 }, { 4954, 0x0008 }, { 4955, 0x1605 }, { 4960, 0x3122 },
2215 { 4965, 0x4240 }, { 4968, 0x0880 }, { 4970, 0xfa4e }, { 4980, 0x06a2 },
2216 { 4985, 0x0814 }, { 4988, 0x9211 }, { 4993, 0x2002 }, { 4995, 0x9b04 },
2217 { 5001, 0x2e52 }, { 5008, 0x0643 }, { 5013, 0x5000 }, { 5015, 0x9010 },
2218 /* 0x8500 */
2219 { 5018, 0x0041 }, { 5020, 0x85ba }, { 5028, 0x3042 }, { 5032, 0x2020 },
2220 { 5034, 0x4f0b }, { 5042, 0x05a0 }, { 5046, 0x2708 }, { 5051, 0x4080 },
2221 { 5053, 0x0591 }, { 5058, 0x1a93 }, { 5065, 0xdf50 }, { 5074, 0x0600 },
2222 { 5076, 0xa202 }, { 5080, 0x3021 }, { 5084, 0x0630 }, { 5088, 0x4e80 },
2223 /* 0x8600 */
2224 { 5093, 0x0cc4 }, { 5098, 0x04c8 }, { 5102, 0xa004 }, { 5105, 0x8001 },
2225 { 5107, 0x6000 }, { 5109, 0xd431 }, { 5116, 0x0880 }, { 5118, 0x0a02 },
2226 { 5121, 0x1c00 }, { 5124, 0x0028 }, { 5126, 0x8e18 }, { 5132, 0x0041 },
2227 { 5134, 0x6ad0 }, { 5141, 0xca10 }, { 5146, 0xf210 }, { 5152, 0x4b00 },
2228 /* 0x8700 */
2229 { 5156, 0x274d }, { 5164, 0x1506 }, { 5169, 0x0220 }, { 5171, 0x8890 },
2230 { 5175, 0x5a00 }, { 5179, 0x82a8 }, { 5184, 0x4549 }, { 5190, 0x8150 },
2231 { 5194, 0x2004 }, { 5196, 0x8000 }, { 5197, 0x8804 }, { 5200, 0x2c08 },
2232 { 5204, 0x08d1 }, { 5209, 0x0005 }, { 5211, 0x8001 }, { 5213, 0x4ac4 },
2233 /* 0x8800 */
```



```

2321 { 6438, 0x0000 }, { 6438, 0x03c0 }, { 6442, 0x7120 }, { 6447, 0x1018 },
2322 { 6450, 0x0172 }, { 6455, 0xa927 }, { 6463, 0x6004 }, { 6466, 0x8906 },
2323 /* 0x9a00 */
2324 { 6471, 0xc022 }, { 6475, 0x020c }, { 6478, 0x0900 }, { 6480, 0x4081 },
2325 { 6483, 0x202d }, { 6488, 0x8ca0 }, { 6493, 0x0e34 }, { 6499, 0x0000 },
2326 { 6499, 0x0000 }, { 6499, 0x0000 }, { 6499, 0x2100 }, { 6501, 0x1101 },
2327 { 6504, 0x8011 }, { 6507, 0xc11a }, { 6513, 0xec4c }, { 6521, 0x0892 },
2328 /* 0x9b00 */
2329 { 6525, 0x0040 }, { 6526, 0x8500 }, { 6529, 0xc7ac }, { 6538, 0x1806 },
2330 { 6542, 0xe03e }, { 6550, 0x0512 }, { 6554, 0x8000 }, { 6555, 0x0010 },
2331 { 6556, 0x4008 }, { 6558, 0x80ce }, { 6564, 0x6d01 }, { 6570, 0x0210 },
2332 { 6572, 0x8641 }, { 6577, 0x0856 }, { 6582, 0x011e }, { 6587, 0x0027 },
2333 /* 0x9c00 */
2334 { 6591, 0x3750 }, { 6598, 0x083d }, { 6604, 0xe032 }, { 6610, 0x4e05 },
2335 { 6616, 0x01c0 }, { 6619, 0x0484 }, { 6622, 0x0081 }, { 6624, 0x0140 },
2336 { 6626, 0x0000 }, { 6626, 0x0000 }, { 6626, 0x0000 }, { 6626, 0x0000 },
2337 { 6626, 0x0000 }, { 6626, 0x0000 }, { 6626, 0x1aa0 }, { 6631, 0x0059 },
2338 /* 0x9d00 */
2339 { 6635, 0x43c8 }, { 6641, 0x8824 }, { 6645, 0x1d48 }, { 6651, 0xc800 },
2340 { 6654, 0x0152 }, { 6658, 0x7203 }, { 6664, 0x9013 }, { 6669, 0x0404 },
2341 { 6671, 0x8280 }, { 6674, 0x0400 }, { 6675, 0x8a10 }, { 6679, 0x0d14 },
2342 { 6684, 0x8056 }, { 6689, 0x0208 }, { 6691, 0xa040 }, { 6694, 0x2704 },
2343 /* 0x9e00 */
2344 { 6699, 0x0000 }, { 6699, 0x4c00 }, { 6702, 0x0000 }, { 6702, 0x0000 },
2345 { 6702, 0x0000 }, { 6702, 0x0000 }, { 6702, 0xa320 },
2346 { 6707, 0x1902 }, { 6711, 0xa0ae }, { 6718, 0x2660 }, { 6723, 0xdf00 },
2347 { 6730, 0xf010 }, { 6735, 0x7b15 }, { 6744, 0x8121 }, { 6748, 0x3ad0 },
2348 /* 0x9f00 */
2349 { 6755, 0x4180 }, { 6758, 0x0028 }, { 6760, 0x1003 }, { 6763, 0x4800 },
2350 { 6765, 0xcc00 }, { 6769, 0x8014 }, { 6772, 0x14cf }, { 6780, 0x00c4 },
2351 { 6783, 0x2000 }, { 6784, 0x3020 }, { 6787, 0x0001 },
2352 };
2353 static const Summary16 jisx0208_uni2indx_pageff[15] = {
2354 /* 0xff00 */
2355 { 6788, 0xdf7a }, { 6800, 0xffff }, { 6816, 0xffff }, { 6832, 0xefff },
2356 { 6847, 0xffff }, { 6863, 0x3fff }, { 6877, 0x0000 }, { 6877, 0x0000 },
2357 { 6877, 0x0000 }, { 6877, 0x0000 }, { 6877, 0x0000 }, { 6877, 0x0000 },
2358 { 6877, 0x0000 }, { 6877, 0x0000 }, { 6877, 0x0028 },
2359 };
2360
2361 static int
2362 jisx0208_wctomb (conv_t conv, unsigned char *, ucs4_t wc, int n)
2363 {
2364     if (n >= 2) {
2365         const Summary16 *summary = NULL;
2366         if (wc < 0x0100)
2367             summary = &jisx0208_uni2indx_page00[(wc>4)];
2368         else if (wc >= 0x0300 && wc < 0x0460)
2369             summary = &jisx0208_uni2indx_page03[(wc>4)-0x030];
2370         else if (wc >= 0x2000 && wc < 0x2320)
2371             summary = &jisx0208_uni2indx_page20[(wc>4)-0x200];
2372         else if (wc >= 0x2500 && wc < 0x2670)
2373             summary = &jisx0208_uni2indx_page25[(wc>4)-0x250];
2374         else if (wc >= 0x3000 && wc < 0x3100)
2375             summary = &jisx0208_uni2indx_page30[(wc>4)-0x300];
2376         else if (wc >= 0x4e00 && wc < 0x9fb0)
2377             summary = &jisx0208_uni2indx_page4e[(wc>4)-0x4e0];
2378         else if (wc >= 0xffff && wc < 0xffff0)
2379             summary = &jisx0208_uni2indx_pageff[(wc>4)-0xff0];
2380         if (summary) {
2381             unsigned short used = summary->used;
2382             unsigned int i = wc & 0x0f;
2383             if (used & ((unsigned short) 1 << i)) {
2384                 unsigned short c;
2385                 /* Keep in 'used' only the bits 0..i-1. */
2386                 used &= ((unsigned short) 1 << i) - 1;
2387                 /* Add 'summary->indx' and the number of bits set in 'used'. */
2388                 used = (used & 0x5555) + ((used & 0xaaaa) >> 1);
2389                 used = (used & 0x3333) + ((used & 0xcccc) >> 2);
2390                 used = (used & 0x0f0f) + ((used & 0xf0f0) >> 4);
2391                 used = (used & 0x00ff) + (used >> 8);
2392                 c = jisx0208_2charset[summary->indx + used];
2393                 r[0] = (c >> 8); r[1] = (c & 0xff);
2394                 return 2;
2395             }
2396         }
2397         return RET_ILSEQ;
2398     }
2399     return RET_TOOSMALL;
2400 }
2401 #endif /* NEED_TOMB */

```

32.229 jisx0212.h

```
1 /* $XFree86: xc/lib/X11/lcUniConv/jisx0212.h,v 1.5 2003/05/27 22:26:31 tsi Exp $ */
```

```
2
3 /*
4  * JISX0212.1990-0
5  */
6 #ifdef NEED_TOWC
7
8 static const unsigned short jisx0212_2uni_page22[81] = {
9     /* 0x22 */
10    0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
11    0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x02d8, 0x02c7,
12    0x00b8, 0x02d9, 0x02dd, 0x00af, 0x02db, 0x02da, 0x007e, 0x0384,
13    0x0385, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
14    0xffff, 0x00a1, 0x00a6, 0x00bf, 0xffff, 0xffff, 0xffff, 0xffff,
15    0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
16    0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
17    0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
18    0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
19    0xffff, 0xffff, 0x00ba, 0x00aa, 0x00a9, 0x00ae, 0x2122, 0x00a4,
20    0x2116,
21 };
22 static const unsigned short jisx0212_2uni_page26[188] = {
23     /* 0x26 */
24    0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
25    0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
26    0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
27    0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
28    0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
29    0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
30    0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
31    0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
32    0x0386, 0x0388, 0x0389, 0x038a, 0x03aa, 0xffff, 0x038c, 0xffff,
33    0x038e, 0x03ab, 0xffff, 0x038f, 0xffff, 0xffff, 0xffff, 0xffff,
34    0x03ac, 0x03ad, 0x03ae, 0x03af, 0x03ca, 0x0390, 0x03cc, 0x03c2,
35    0x03cd, 0x03cb, 0x03b0, 0x03ce, 0xffff, 0xffff,
36     /* 0x27 */
37    0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
38    0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
39    0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
40    0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
41    0xffff, 0x0402, 0x0403, 0x0404, 0x0405, 0x0406, 0x0407, 0x0408,
42    0x0409, 0x040a, 0x040b, 0x040c, 0x040e, 0x040f, 0xffff, 0xffff,
43    0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
44    0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
45    0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
46    0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
47    0xffff, 0x0452, 0x0453, 0x0454, 0x0455, 0x0456, 0x0457, 0x0458,
48    0x0459, 0x045a, 0x045b, 0x045c, 0x045e, 0x045f,
49 };
50 static const unsigned short jisx0212_2uni_page29[275] = {
51     /* 0x29 */
52    0x00c6, 0x0110, 0xffff, 0x0126, 0xffff, 0x0132, 0xffff, 0x0141,
53    0x013f, 0xffff, 0x014a, 0x00d8, 0x0152, 0xffff, 0x0166, 0x00de,
54    0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
55    0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
56    0x00e6, 0x0111, 0x00f0, 0x0127, 0x0131, 0x0133, 0x0138, 0x0142,
57    0x0140, 0x0149, 0x014b, 0x00f8, 0x0153, 0x00df, 0x0167, 0x00fe,
58    0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
59    0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
60    0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
61    0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
62    0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
63    0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
64     /* 0x2a */
65    0x00c1, 0x00c0, 0x00c4, 0x00c2, 0x0102, 0x01cd, 0x0100, 0x0104,
66    0x00c5, 0x00c3, 0x0106, 0x0108, 0x010c, 0x00c7, 0x010a, 0x010e,
67    0x00c9, 0x00cb, 0x00ca, 0x011a, 0x0116, 0x0112, 0x0118,
68    0xffff, 0x011c, 0x011e, 0x0122, 0x0120, 0x0124, 0x00cd, 0x00cc,
69    0x00cf, 0x00ce, 0x01cf, 0x0130, 0x012a, 0x012e, 0x0128, 0x0134,
70    0x0136, 0x0139, 0x013d, 0x013b, 0x0143, 0x0147, 0x0145, 0x00d1,
71    0x00d3, 0x00d2, 0x00d6, 0x00d4, 0x01d1, 0x0150, 0x014c, 0x00d5,
72    0x0154, 0x0158, 0x0156, 0x015a, 0x015c, 0x0160, 0x015e, 0x0164,
73    0x0162, 0x00da, 0x00d9, 0x00dc, 0x00db, 0x016c, 0x01d3, 0x0170,
74    0x016a, 0x0172, 0x016e, 0x0168, 0x01d7, 0x01db, 0x01d9, 0x01d5,
75    0x0174, 0x00dd, 0x0178, 0x0176, 0x0179, 0x017d, 0x017b, 0xffff,
76    0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
77     /* 0x2b */
78    0x00e1, 0x00e0, 0x00e4, 0x00e2, 0x0103, 0x01ce, 0x0101, 0x0105,
79    0x00e5, 0x00e3, 0x0107, 0x0109, 0x010d, 0x00e7, 0x010b, 0x010f,
80    0x00e9, 0x00eb, 0x00ea, 0x011b, 0x0117, 0x0113, 0x0119,
81    0x01f5, 0x011d, 0x011f, 0xffff, 0x0121, 0x0125, 0x00ed, 0x00ec,
82    0x00ef, 0x00ee, 0x01d0, 0xffff, 0x012b, 0x012f, 0x0129, 0x0135,
83    0x0137, 0x013a, 0x013e, 0x013c, 0x0144, 0x0148, 0x0146, 0x00f1,
84    0x00f3, 0x00f2, 0x00f6, 0x00f4, 0x01d2, 0x0151, 0x014d, 0x00f5,
85    0x0155, 0x0159, 0x0157, 0x015b, 0x015d, 0x0161, 0x015f, 0x0165,
86    0x0163, 0x00fa, 0x00f9, 0x00fc, 0x00fb, 0x016d, 0x01d4, 0x0171,
87    0x016b, 0x0173, 0x016f, 0x0169, 0x01d8, 0x01dc, 0x01da, 0x01d6,
88    0x0175, 0x00fd, 0x00ff, 0x0177, 0x017a, 0x017e, 0x017c,
```



```
176 0x5671, 0x5672, 0x5675, 0x5684, 0x5685, 0x5688, 0x568b, 0x568c,
177 0x5695, 0x5699, 0x569a, 0x569d, 0x569e, 0x569f, 0x56a6, 0x56a7,
178 0x56a8, 0x56a9, 0x56ab, 0x56ac, 0x56ad, 0x56b1, 0x56b3, 0x56b7,
179 0x56be, 0x56c5, 0x56c9, 0x56ca, 0x56cb, 0x56cf, 0x56d0, 0x56cc,
180 0x56cd, 0x56d9, 0x56dc, 0x56dd, 0x56df, 0x56e1, 0x56e4, 0x56e5,
181 0x56e6, 0x56e7, 0x56e8, 0x56f1, 0x56eb, 0x56ed,
182 /* 0x37 */
183 0x56f6, 0x56f7, 0x5701, 0x5702, 0x5707, 0x570a, 0x570c, 0x5711,
184 0x5715, 0x571a, 0x571b, 0x571d, 0x5720, 0x5722, 0x5723, 0x5724,
185 0x5725, 0x5729, 0x572a, 0x572c, 0x572e, 0x572f, 0x5733, 0x5734,
186 0x573d, 0x573e, 0x573f, 0x5745, 0x5746, 0x574c, 0x574d, 0x5752,
187 0x5762, 0x5765, 0x5767, 0x5768, 0x576b, 0x576d, 0x576e, 0x576f,
188 0x5770, 0x5771, 0x5773, 0x5774, 0x5775, 0x5777, 0x5779, 0x577a,
189 0x577b, 0x577c, 0x577e, 0x5781, 0x5783, 0x578c, 0x5794, 0x5797,
190 0x5799, 0x579a, 0x579c, 0x579d, 0x579e, 0x579f, 0x57a1, 0x5795,
191 0x57a7, 0x57a8, 0x57a9, 0x57ac, 0x57b8, 0x57bd, 0x57c7, 0x57c8,
192 0x57cc, 0x57cf, 0x57d5, 0x57dd, 0x57de, 0x57e4, 0x57e6, 0x57e7,
193 0x57e9, 0x57ed, 0x57f0, 0x57f5, 0x57f6, 0x57f8, 0x57fd, 0x57fe,
194 0x57ff, 0x5803, 0x5804, 0x5808, 0x5809, 0x57e1,
195 /* 0x38 */
196 0x580c, 0x580d, 0x581b, 0x581e, 0x581f, 0x5820, 0x5826, 0x5827,
197 0x582d, 0x5832, 0x5839, 0x583f, 0x5849, 0x584c, 0x584d, 0x584f,
198 0x5850, 0x5855, 0x585f, 0x5861, 0x5864, 0x5867, 0x5868, 0x5878,
199 0x587c, 0x587f, 0x5880, 0x5881, 0x5887, 0x5888, 0x5889, 0x588a,
200 0x588c, 0x588d, 0x588f, 0x5890, 0x5894, 0x5896, 0x589d, 0x58a0,
201 0x58a1, 0x58a2, 0x58a6, 0x58a9, 0x58b1, 0x58b2, 0x58c4, 0x58bc,
202 0x58c2, 0x58c8, 0x58cd, 0x58ce, 0x58d0, 0x58d2, 0x58d4, 0x58d6,
203 0x58da, 0x58dd, 0x58e1, 0x58e2, 0x58e9, 0x58f3, 0x5905, 0x5906,
204 0x590b, 0x590c, 0x5912, 0x5913, 0x5914, 0x8641, 0x591d, 0x5921,
205 0x5923, 0x5924, 0x5928, 0x592f, 0x5930, 0x5933, 0x5935, 0x5936,
206 0x593f, 0x5943, 0x5946, 0x5952, 0x5953, 0x5959, 0x595b, 0x595d,
207 0x595e, 0x595f, 0x5961, 0x5963, 0x596b, 0x596d,
208 /* 0x39 */
209 0x596f, 0x5972, 0x5975, 0x5976, 0x5979, 0x597b, 0x597c, 0x598b,
210 0x598c, 0x598e, 0x5992, 0x5995, 0x5997, 0x599f, 0x59a4, 0x59a7,
211 0x59ad, 0x59ae, 0x59af, 0x59b0, 0x59b3, 0x59b7, 0x59ba, 0x59bc,
212 0x59c1, 0x59c3, 0x59c4, 0x59c8, 0x59ca, 0x59cd, 0x59d2, 0x59dd,
213 0x59de, 0x59df, 0x59e3, 0x59e4, 0x59e7, 0x59ee, 0x59ef, 0x59f1,
214 0x59f2, 0x59f4, 0x59f7, 0x5a00, 0x5a04, 0x5a0c, 0x5a0d, 0x5a0e,
215 0x5a12, 0x5a13, 0x5a1e, 0x5a23, 0x5a24, 0x5a27, 0x5a28, 0x5a2a,
216 0x5a2d, 0x5a30, 0x5a44, 0x5a45, 0x5a47, 0x5a48, 0x5a4c, 0x5a50,
217 0x5a55, 0x5a5e, 0x5a63, 0x5a65, 0x5a67, 0x5a6d, 0x5a77, 0x5a7a,
218 0x5a7b, 0x5a7e, 0x5a8b, 0x5a90, 0x5a93, 0x5a96, 0x5a99, 0x5a9c,
219 0x5a9e, 0x5a9f, 0x5aa0, 0x5aa2, 0x5aa7, 0x5aac, 0x5ab1, 0x5ab2,
220 0x5ab3, 0x5ab5, 0x5ab8, 0x5aba, 0x5abb, 0x5abf,
221 /* 0x3a */
222 0x5ac4, 0x5ac6, 0x5ac8, 0x5acf, 0x5ada, 0x5adc, 0x5ae0, 0x5ae5,
223 0x5aea, 0x5aee, 0x5af5, 0x5af6, 0x5afd, 0x5b00, 0x5b01, 0x5b08,
224 0x5b17, 0x5b34, 0x5b39, 0x5b3b, 0x5b3d, 0x5b21, 0x5b25, 0x5b2d,
225 0x5b38, 0x5b41, 0x5b4b, 0x5b4c, 0x5b52, 0x5b56, 0x5b5e, 0x5b68,
226 0x5b6e, 0x5b6f, 0x5b7c, 0x5b7d, 0x5b7e, 0x5b7f, 0x5b81, 0x5b84,
227 0x5b86, 0x5b8a, 0x5b8e, 0x5b90, 0x5b91, 0x5b93, 0x5b94, 0x5b96,
228 0x5ba8, 0x5ba9, 0x5bac, 0x5bad, 0x5baf, 0x5bb1, 0x5bb2, 0x5bb7,
229 0x5bba, 0x5bbc, 0x5bcb, 0x5bc0, 0x5bc1, 0x5bcd, 0x5bce, 0x5bd6, 0x5bd7,
230 0x5bd8, 0x5bd9, 0x5bda, 0x5be0, 0x5bef, 0x5bf1, 0x5bf4, 0x5bfd,
231 0x5c0c, 0x5c17, 0x5c1e, 0x5c1f, 0x5c23, 0x5c26, 0x5c29, 0x5c2b,
232 0x5c2c, 0x5c2e, 0x5c30, 0x5c32, 0x5c35, 0x5c36, 0x5c59, 0x5c5a,
233 0x5c5c, 0x5c62, 0x5c63, 0x5c67, 0x5c68, 0x5c69,
234 /* 0x3b */
235 0x5c6d, 0x5c70, 0x5c74, 0x5c75, 0x5c7a, 0x5c7b, 0x5c7c, 0x5c7d,
236 0x5c87, 0x5c88, 0x5c8a, 0x5c8f, 0x5c92, 0x5c9d, 0x5c9f, 0x5ca0,
237 0x5ca2, 0x5ca3, 0x5ca6, 0x5caa, 0x5cb2, 0x5cb4, 0x5cb5, 0x5cba,
238 0x5cc9, 0x5ccb, 0x5ccd, 0x5cd2, 0x5cd7, 0x5cee, 0x5cf1, 0x5cf2,
239 0x5cf4, 0x5d01, 0x5d06, 0x5d0d, 0x5d12, 0x5d2b, 0x5d23, 0x5d24,
240 0x5d26, 0x5d27, 0x5d31, 0x5d34, 0x5d39, 0x5d3d, 0x5d3f, 0x5d42,
241 0x5d43, 0x5d46, 0x5d48, 0x5d4e, 0x5d51, 0x5d59, 0x5d4a, 0x5d5f,
242 0x5d60, 0x5d61, 0x5d62, 0x5d64, 0x5d6a, 0x5d6d, 0x5d70, 0x5d79,
243 0x5d7a, 0x5d7e, 0x5d7f, 0x5d81, 0x5d83, 0x5d88, 0x5d8a, 0x5d92,
244 0x5d93, 0x5d94, 0x5d95, 0x5d99, 0x5d9b, 0x5d9f, 0x5da0, 0x5da7,
245 0x5dab, 0x5db0, 0x5db4, 0x5db8, 0x5db9, 0x5dc3, 0x5dc7, 0x5dcb,
246 0x5dd0, 0x5dce, 0x5dd8, 0x5dd9, 0x5de0, 0x5de4,
247 /* 0x3c */
248 0x5de9, 0x5df8, 0x5df9, 0x5e00, 0x5e07, 0x5e0d, 0x5e12, 0x5e14,
249 0x5e15, 0x5e18, 0x5e1f, 0x5e20, 0x5e2e, 0x5e28, 0x5e32, 0x5e35,
250 0x5e3e, 0x5e4b, 0x5e50, 0x5e49, 0x5e51, 0x5e56, 0x5e58, 0x5e5b,
251 0x5e5c, 0x5e5e, 0x5e68, 0x5e6a, 0x5e6b, 0x5e6c, 0x5e6d, 0x5e6e,
252 0x5e70, 0x5e80, 0x5e8b, 0x5e8e, 0x5ea2, 0x5ea4, 0x5ea5, 0x5ea8,
253 0x5eaa, 0x5eac, 0x5eb1, 0x5eb3, 0x5ebd, 0x5ebe, 0x5ebf, 0x5ec6,
254 0x5ecc, 0x5ecb, 0x5ece, 0x5ed1, 0x5ed2, 0x5ed4, 0x5ed5, 0x5edc,
255 0x5ede, 0x5ee5, 0x5eeb, 0x5ef0, 0x5ef6, 0x5ef7, 0x5ef8, 0x5ef0e,
256 0x5f19, 0x5f1c, 0x5f1d, 0x5f21, 0x5f22, 0x5f23, 0x5f24, 0x5f28,
257 0x5f2b, 0x5f2c, 0x5f2e, 0x5f30, 0x5f34, 0x5f36, 0x5f3b, 0x5f3d,
258 0x5f3f, 0x5f40, 0x5f44, 0x5f45, 0x5f47, 0x5f4d, 0x5f50, 0x5f54,
259 0x5f58, 0x5f5b, 0x5f60, 0x5f63, 0x5f64, 0x5f67,
260 /* 0x3d */
261 0x5f6f, 0x5f72, 0x5f74, 0x5f75, 0x5f78, 0x5f7a, 0x5f7d, 0x5f7e,
262 0x5f89, 0x5f8d, 0x5f8f, 0x5f96, 0x5f9c, 0x5f9d, 0x5fa2, 0x5fa7,
```

```
263 0x5fab, 0x5fa4, 0x5fac, 0x5faf, 0x5fb0, 0x5fbl, 0x5fb8, 0x5fc4,
264 0x5fc7, 0x5fc8, 0x5fc9, 0x5fcb, 0x5fd0, 0x5fd1, 0x5fd2, 0x5fd3,
265 0x5fd4, 0x5fde, 0x5fee, 0x5fef, 0x5ff2, 0x5ff3, 0x5ff6, 0x5ffa, 0x5ffc,
266 0x5fed, 0x5fee, 0x5fef, 0x5ff2, 0x5ff3, 0x5ff6, 0x5ffa, 0x5ffc,
267 0x6007, 0x600a, 0x600d, 0x6013, 0x6014, 0x6017, 0x6018, 0x601a,
268 0x601f, 0x6024, 0x602d, 0x6033, 0x6035, 0x6040, 0x6047, 0x6048,
269 0x6049, 0x604c, 0x6051, 0x6054, 0x6056, 0x6057, 0x605d, 0x6061,
270 0x6067, 0x6071, 0x607e, 0x607f, 0x6082, 0x6086, 0x6088, 0x608a,
271 0x608e, 0x6091, 0x6093, 0x6095, 0x6098, 0x609d, 0x609e, 0x60a2,
272 0x60a4, 0x60a5, 0x60a8, 0x60b0, 0x60b1, 0x60b7,
273 /* 0x3e */
274 0x60bb, 0x60be, 0x60c2, 0x60c4, 0x60c8, 0x60c9, 0x60ca, 0x60cb,
275 0x60ce, 0x60cf, 0x60d4, 0x60d5, 0x60d9, 0x60db, 0x60dd, 0x60de,
276 0x60e2, 0x60e5, 0x60f2, 0x60f5, 0x60f8, 0x60fc, 0x60fd, 0x6102,
277 0x6107, 0x610a, 0x610c, 0x6110, 0x6111, 0x6112, 0x6113, 0x6114,
278 0x6116, 0x6117, 0x6119, 0x611c, 0x611e, 0x6122, 0x612a, 0x612b,
279 0x6130, 0x6131, 0x6135, 0x6136, 0x6137, 0x6139, 0x6141, 0x6145,
280 0x6146, 0x6149, 0x615e, 0x6160, 0x616c, 0x6172, 0x6178, 0x617b,
281 0x617c, 0x617f, 0x6180, 0x6181, 0x6183, 0x6184, 0x618b, 0x618d,
282 0x6192, 0x6193, 0x6197, 0x6198, 0x619c, 0x619d, 0x619f, 0x61a0,
283 0x61a5, 0x61a8, 0x61aa, 0x61ad, 0x61b8, 0x61b9, 0x61bc, 0x61c0,
284 0x61c1, 0x61c2, 0x61ce, 0x61cf, 0x61d5, 0x61dc, 0x61dd, 0x61de,
285 0x61df, 0x61e1, 0x61e2, 0x61e7, 0x61e9, 0x61e5,
286 /* 0x3f */
287 0x61ec, 0x61ed, 0x61ef, 0x6201, 0x6203, 0x6204, 0x6207, 0x6213,
288 0x6215, 0x621c, 0x6220, 0x6222, 0x6223, 0x6227, 0x6229, 0x622b,
289 0x6239, 0x623d, 0x6242, 0x6243, 0x6244, 0x6246, 0x624c, 0x6250,
290 0x6251, 0x6252, 0x6254, 0x6256, 0x625a, 0x625c, 0x6264, 0x626d,
291 0x626e, 0x6273, 0x627a, 0x627d, 0x628d, 0x628e, 0x628f, 0x6290,
292 0x62a6, 0x62a8, 0x62b3, 0x62b6, 0x62b7, 0x62ba, 0x62be, 0x62bf,
293 0x62c4, 0x62ce, 0x62d5, 0x62d6, 0x62da, 0x62ea, 0x62f2, 0x62f4,
294 0x62fc, 0x62fd, 0x6303, 0x6304, 0x630a, 0x630b, 0x630d, 0x6310,
295 0x6313, 0x6316, 0x6318, 0x6329, 0x632a, 0x632d, 0x6335, 0x6336,
296 0x6339, 0x633c, 0x6341, 0x6342, 0x6343, 0x6344, 0x6346, 0x634a,
297 0x634b, 0x634e, 0x6352, 0x6353, 0x6354, 0x6358, 0x635b, 0x6365,
298 0x6366, 0x636c, 0x636d, 0x6371, 0x6374, 0x6375,
299 /* 0x40 */
300 0x6378, 0x637c, 0x637d, 0x637f, 0x6382, 0x6384, 0x6387, 0x638a,
301 0x6390, 0x6394, 0x6395, 0x6399, 0x639a, 0x639e, 0x63a4, 0x63a6,
302 0x63ad, 0x63ae, 0x63af, 0x63bd, 0x63c1, 0x63c5, 0x63c8, 0x63ce,
303 0x63d1, 0x63d3, 0x63d4, 0x63d5, 0x63dc, 0x63e0, 0x63e5, 0x63ea,
304 0x63ec, 0x63f2, 0x63f3, 0x63f5, 0x63f8, 0x63f9, 0x6409, 0x640a,
305 0x6410, 0x6412, 0x6414, 0x6418, 0x641e, 0x6420, 0x6422, 0x6424,
306 0x6425, 0x6429, 0x642a, 0x642f, 0x6430, 0x6435, 0x643d, 0x643f,
307 0x644b, 0x644f, 0x6451, 0x6452, 0x6453, 0x6454, 0x645a, 0x645b,
308 0x645c, 0x645d, 0x645f, 0x6460, 0x6461, 0x6463, 0x646d, 0x6473,
309 0x6474, 0x647b, 0x647d, 0x6485, 0x6487, 0x648f, 0x6490, 0x6491,
310 0x6498, 0x6499, 0x649b, 0x649d, 0x649f, 0x64a1, 0x64a3, 0x64a6,
311 0x64a8, 0x64ac, 0x64b3, 0x64bd, 0x64be, 0x64bf,
312 /* 0x41 */
313 0x64c4, 0x64c9, 0x64ca, 0x64cb, 0x64cc, 0x64ce, 0x64d0, 0x64d1,
314 0x64d5, 0x64d7, 0x64e4, 0x64e5, 0x64e9, 0x64ea, 0x64ed, 0x64f0,
315 0x64f5, 0x64f7, 0x64fb, 0x64ff, 0x6501, 0x6504, 0x6508, 0x6509,
316 0x650a, 0x650f, 0x6513, 0x6514, 0x6516, 0x6519, 0x651b, 0x651e,
317 0x651f, 0x6522, 0x6526, 0x6529, 0x652e, 0x6531, 0x653a, 0x653c,
318 0x653d, 0x6543, 0x6547, 0x6549, 0x6550, 0x6552, 0x6554, 0x655f,
319 0x6560, 0x6567, 0x656b, 0x657a, 0x657d, 0x6581, 0x6585, 0x658a,
320 0x6592, 0x6595, 0x6598, 0x659d, 0x65a0, 0x65a3, 0x65a6, 0x65ae,
321 0x65b2, 0x65b3, 0x65b4, 0x65bf, 0x65c2, 0x65c8, 0x65c9, 0x65ce,
322 0x65d0, 0x65d4, 0x65d6, 0x65d8, 0x65df, 0x65f0, 0x65f2, 0x65f4,
323 0x65f5, 0x65f9, 0x65fe, 0x65ff, 0x6600, 0x6604, 0x6608, 0x6609,
324 0x660d, 0x6611, 0x6612, 0x6615, 0x6616, 0x661d,
325 /* 0x42 */
326 0x661e, 0x6621, 0x6622, 0x6623, 0x6624, 0x6626, 0x6629, 0x662a,
327 0x662b, 0x662c, 0x662e, 0x6630, 0x6631, 0x6633, 0x6639, 0x6637,
328 0x6640, 0x6645, 0x6646, 0x664a, 0x664c, 0x6651, 0x664e, 0x6657,
329 0x6658, 0x6659, 0x665b, 0x665c, 0x6660, 0x6661, 0x666b, 0x666a,
330 0x666b, 0x666c, 0x667e, 0x6673, 0x6675, 0x667f, 0x6677, 0x6678,
331 0x6679, 0x667b, 0x6680, 0x667c, 0x668b, 0x668c, 0x668d, 0x6690,
332 0x6692, 0x6699, 0x669a, 0x669b, 0x669c, 0x669f, 0x66a0, 0x66a4,
333 0x66ad, 0x66b1, 0x66b2, 0x66b5, 0x66bb, 0x66bf, 0x66c0, 0x66c2,
334 0x66c3, 0x66c8, 0x66cc, 0x66ce, 0x66cf, 0x66d4, 0x66db, 0x66df,
335 0x66e8, 0x66eb, 0x66ec, 0x66ee, 0x66fa, 0x6705, 0x6707, 0x670e,
336 0x6713, 0x6719, 0x671c, 0x6722, 0x6722, 0x6733, 0x673e, 0x6745,
337 0x6747, 0x6748, 0x674c, 0x6754, 0x6755, 0x675d,
338 /* 0x43 */
339 0x6766, 0x676c, 0x676e, 0x6774, 0x6776, 0x677b, 0x6781, 0x6784,
340 0x678e, 0x678f, 0x6791, 0x6793, 0x6796, 0x6798, 0x6799, 0x679b,
341 0x67b0, 0x67b1, 0x67b2, 0x67b5, 0x67bb, 0x67bc, 0x67bd, 0x67f9,
342 0x67c0, 0x67c2, 0x67c3, 0x67c5, 0x67c8, 0x67c9, 0x67d2, 0x67d7,
343 0x67d9, 0x67dc, 0x67e1, 0x67e6, 0x67f0, 0x67f2, 0x67f6, 0x67f7,
344 0x6852, 0x6814, 0x6819, 0x681d, 0x681f, 0x6822, 0x6827, 0x682c,
345 0x682d, 0x682f, 0x6830, 0x6831, 0x6833, 0x683b, 0x683f, 0x6844,
346 0x6845, 0x684a, 0x684c, 0x6855, 0x6857, 0x6858, 0x685b, 0x686b,
347 0x686e, 0x686f, 0x6870, 0x6871, 0x6872, 0x6875, 0x6879, 0x687a,
348 0x687b, 0x687c, 0x6882, 0x6884, 0x6886, 0x6888, 0x6896, 0x6898,
349 0x689a, 0x689c, 0x68a1, 0x68a3, 0x68a5, 0x68a9, 0x68aa, 0x68ae,
```

```
350 0x68b2, 0x68bb, 0x68c5, 0x68c8, 0x68cc, 0x68cf,
351 /* 0x44 */
352 0x68d0, 0x68d1, 0x68d3, 0x68d6, 0x68d9, 0x68dc, 0x68dd, 0x68e5,
353 0x68e8, 0x68ea, 0x68eb, 0x68ec, 0x68ed, 0x68f0, 0x68f1, 0x68f5,
354 0x68f6, 0x68fb, 0x68fc, 0x68fd, 0x6906, 0x6909, 0x690a, 0x6910,
355 0x6911, 0x6913, 0x6916, 0x691b, 0x6917, 0x6931, 0x6933, 0x6935, 0x6938,
356 0x693b, 0x6942, 0x6945, 0x6949, 0x694e, 0x6957, 0x695b, 0x6963,
357 0x6964, 0x6965, 0x6966, 0x6968, 0x6969, 0x696c, 0x6970, 0x6971,
358 0x6972, 0x697a, 0x697b, 0x697f, 0x6980, 0x698d, 0x6992, 0x6996,
359 0x6998, 0x69a1, 0x69a5, 0x69a6, 0x69a8, 0x69ab, 0x69ad, 0x69af,
360 0x69b7, 0x69b8, 0x69ba, 0x69bc, 0x69c5, 0x69c8, 0x69d1, 0x69d6,
361 0x69d7, 0x69e2, 0x69e5, 0x69ee, 0x69ef, 0x69f1, 0x69f3, 0x69f5,
362 0x69fe, 0x6a00, 0x6a01, 0x6a03, 0x6a0f, 0x6a11, 0x6a15, 0x6a1a,
363 0x6a1d, 0x6a20, 0x6a24, 0x6a28, 0x6a30, 0x6a32,
364 /* 0x45 */
365 0x6a34, 0x6a37, 0x6a3b, 0x6a3e, 0x6a3f, 0x6a45, 0x6a46, 0x6a49,
366 0x6a4a, 0x6a4e, 0x6a50, 0x6a51, 0x6a52, 0x6a55, 0x6a56, 0x6a5b,
367 0x6a64, 0x6a67, 0x6a6a, 0x6a71, 0x6a73, 0x6a7e, 0x6a81, 0x6a83,
368 0x6a86, 0x6a87, 0x6a89, 0x6a8b, 0x6a91, 0x6a9b, 0x6a9d, 0x6a9e,
369 0x6a9f, 0x6aa5, 0x6aab, 0x6aaf, 0x6ab0, 0x6ab1, 0x6ab4, 0x6abd,
370 0x6abe, 0x6abf, 0x6ac6, 0x6ac9, 0x6ac8, 0x6acc, 0x6ad0, 0x6ad4,
371 0x6ad5, 0x6ad6, 0x6adc, 0x6add, 0x6ae4, 0x6ae7, 0x6aec, 0x6af0,
372 0x6af1, 0x6af2, 0x6afc, 0x6afd, 0x6b02, 0x6b03, 0x6b06, 0x6b07,
373 0x6b09, 0x6b0f, 0x6b10, 0x6b11, 0x6b17, 0x6b1b, 0x6b1e, 0x6b24,
374 0x6b28, 0x6b2b, 0x6b2c, 0x6b2f, 0x6b35, 0x6b36, 0x6b3b, 0x6b3f,
375 0x6b46, 0x6b4a, 0x6b4d, 0x6b52, 0x6b56, 0x6b58, 0x6b5d, 0x6b60,
376 0x6b67, 0x6b6b, 0x6b6e, 0x6b70, 0x6b75, 0x6b7d,
377 /* 0x46 */
378 0x6b7e, 0x6b82, 0x6b85, 0x6b97, 0x6b9b, 0x6b9f, 0x6ba0, 0x6ba2,
379 0x6ba3, 0x6ba8, 0x6ba9, 0x6bac, 0x6bad, 0x6bae, 0x6bb0, 0x6bb8,
380 0x6bb9, 0x6bbd, 0x6bbe, 0x6bc3, 0x6bc4, 0x6bc9, 0x6bcc, 0x6bd6,
381 0x6bda, 0x6be1, 0x6be3, 0x6be6, 0x6be7, 0x6bee, 0x6bf1, 0x6bf7,
382 0x6bf9, 0x6bff, 0x6c02, 0x6c04, 0x6c05, 0x6c09, 0x6c0d, 0x6c0e,
383 0x6c10, 0x6c12, 0x6c19, 0x6c1f, 0x6c26, 0x6c27, 0x6c28, 0x6c2c,
384 0x6c2e, 0x6c33, 0x6c35, 0x6c36, 0x6c3a, 0x6c3b, 0x6c3f, 0x6c4a,
385 0x6c4b, 0x6c4d, 0x6c4f, 0x6c52, 0x6c54, 0x6c59, 0x6c5b, 0x6c5c,
386 0x6c6b, 0x6c6d, 0x6c6f, 0x6c74, 0x6c76, 0x6c78, 0x6c79, 0x6c7b,
387 0x6c85, 0x6c86, 0x6c87, 0x6c89, 0x6c94, 0x6c95, 0x6c97, 0x6c98,
388 0x6c9c, 0x6c9f, 0x6cb0, 0x6cb2, 0x6cb4, 0x6ccb, 0x6ccb, 0x6ccd,
389 0x6ccf, 0x6cd0, 0x6cd1, 0x6cd2, 0x6cd4, 0x6cd6,
390 /* 0x47 */
391 0x6cda, 0x6cdc, 0x6cec, 0x6ce0, 0x6ce7, 0x6ce9, 0x6ceb, 0x6cec, 0x6cee,
392 0x6cf2, 0x6cf4, 0x6d04, 0x6d07, 0x6d0a, 0x6d0e, 0x6d0f, 0x6d11,
393 0x6d13, 0x6d1a, 0x6d26, 0x6d27, 0x6d28, 0x6d2e, 0x6d2f, 0x6d2f,
394 0x6d31, 0x6d39, 0x6d3c, 0x6d3f, 0x6d57, 0x6d5e, 0x6d5f, 0x6d61,
395 0x6d65, 0x6d67, 0x6d6f, 0x6d70, 0x6d7c, 0x6d82, 0x6d87, 0x6d91,
396 0x6d92, 0x6d94, 0x6d96, 0x6d97, 0x6d98, 0x6daa, 0x6dac, 0x6db4,
397 0x6db7, 0x6db9, 0x6dbd, 0x6dbf, 0x6dc4, 0x6dc8, 0x6dca, 0x6dce,
398 0x6dcf, 0x6dd6, 0x6ddb, 0x6ddd, 0x6ddf, 0x6de0, 0x6de2, 0x6de5,
399 0x6de9, 0x6def, 0x6df0, 0x6df4, 0x6df6, 0x6dfc, 0x6e00, 0x6e04,
400 0x6e1e, 0x6e22, 0x6e27, 0x6e32, 0x6e36, 0x6e39, 0x6e3b, 0x6e3c,
401 0x6e44, 0x6e45, 0x6e48, 0x6e49, 0x6e4b, 0x6e4f, 0x6e51, 0x6e52,
402 0x6e53, 0x6e54, 0x6e57, 0x6e5c, 0x6e5d, 0x6e5e,
403 /* 0x48 */
404 0x6e62, 0x6e63, 0x6e68, 0x6e73, 0x6e7b, 0x6e7d, 0x6e8d, 0x6e93,
405 0x6e99, 0x6ea0, 0x6ea7, 0x6ead, 0x6eae, 0x6eb1, 0x6eb3, 0x6ebb,
406 0x6ebf, 0x6ec0, 0x6ec1, 0x6ec3, 0x6ec7, 0x6ec8, 0x6eca, 0x6ecd,
407 0x6ece, 0x6ecf, 0x6eeb, 0x6eed, 0x6eee, 0x6ef9, 0x6efb, 0x6efd,
408 0x6f04, 0x6f08, 0x6f0a, 0x6f0c, 0x6f0d, 0x6f16, 0x6f18, 0x6f1a,
409 0x6f1b, 0x6f26, 0x6f29, 0x6f2a, 0x6f2f, 0x6f30, 0x6f33, 0x6f36,
410 0x6f3b, 0x6f3c, 0x6f2d, 0x6f4f, 0x6f51, 0x6f52, 0x6f53, 0x6f57,
411 0x6f59, 0x6f5a, 0x6f5d, 0x6f5e, 0x6f61, 0x6f62, 0x6f68, 0x6f6c,
412 0x6f7d, 0x6f7e, 0x6f83, 0x6f87, 0x6f88, 0x6f8b, 0x6f8c, 0x6f8d,
413 0x6f90, 0x6f92, 0x6f93, 0x6f94, 0x6f96, 0x6f9a, 0x6f9f, 0x6fa0,
414 0x6fa5, 0x6fa6, 0x6fa7, 0x6fa8, 0x6fae, 0x6faf, 0x6fb0, 0x6fb5,
415 0x6fb6, 0x6fbc, 0x6fc5, 0x6fc7, 0x6fc8, 0x6fca,
416 /* 0x49 */
417 0x6fda, 0x6fde, 0x6fe8, 0x6fe9, 0x6ff0, 0x6ff5, 0x6ff9, 0x6ffc,
418 0x6ffd, 0x7000, 0x7005, 0x7006, 0x7007, 0x700d, 0x7017, 0x7020,
419 0x7023, 0x702f, 0x7034, 0x7037, 0x7039, 0x703c, 0x7043, 0x7044,
420 0x7048, 0x7049, 0x704a, 0x704b, 0x7054, 0x7055, 0x705d, 0x705e,
421 0x704e, 0x7064, 0x7065, 0x706c, 0x706e, 0x7075, 0x7076, 0x707e,
422 0x7081, 0x7085, 0x7086, 0x7094, 0x7095, 0x7096, 0x7097, 0x7098,
423 0x709b, 0x70a4, 0x70ab, 0x70b0, 0x70b1, 0x70b4, 0x70b7, 0x70ca,
424 0x70d1, 0x70d3, 0x70d4, 0x70d5, 0x70d6, 0x70d8, 0x70dc, 0x70e4,
425 0x70fa, 0x7103, 0x7104, 0x7105, 0x7106, 0x7107, 0x710b, 0x710c,
426 0x710f, 0x711e, 0x7120, 0x712d, 0x712f, 0x712f, 0x7130, 0x7131,
427 0x7138, 0x7141, 0x7145, 0x7146, 0x7147, 0x714a, 0x714b, 0x7150,
428 0x7152, 0x7157, 0x715a, 0x715c, 0x715e, 0x7160,
429 /* 0x4a */
430 0x7168, 0x7179, 0x7180, 0x7185, 0x7187, 0x718c, 0x7192, 0x719a,
431 0x719b, 0x71a0, 0x71a2, 0x71af, 0x71b0, 0x71b2, 0x71b3, 0x71ba,
432 0x71bf, 0x71c0, 0x71c1, 0x71c4, 0x71cb, 0x71cc, 0x71d3, 0x71d6,
433 0x71d9, 0x71da, 0x71dc, 0x71f8, 0x71fe, 0x7200, 0x7207, 0x7208,
434 0x7209, 0x7213, 0x7217, 0x721a, 0x721d, 0x721f, 0x7224, 0x722b,
435 0x722f, 0x7234, 0x7238, 0x7239, 0x7241, 0x7242, 0x7243, 0x7245,
436 0x724e, 0x724f, 0x7250, 0x7253, 0x7255, 0x7256, 0x725a, 0x725c,
```

```

437 0x725e, 0x7260, 0x7263, 0x7268, 0x726b, 0x726e, 0x726f, 0x7271,
438 0x7277, 0x7278, 0x727b, 0x727c, 0x727f, 0x7284, 0x7289, 0x728d,
439 0x728e, 0x7293, 0x729b, 0x72a8, 0x72ad, 0x72ae, 0x72b1, 0x72b4,
440 0x72be, 0x72c1, 0x72c7, 0x72c9, 0x72cc, 0x72d5, 0x72d6, 0x72d8,
441 0x72df, 0x72e5, 0x72f3, 0x72f4, 0x72fa, 0x72fb,
442 /* 0x4b */
443 0x72fe, 0x7302, 0x7304, 0x7305, 0x7307, 0x730b, 0x730d, 0x7312,
444 0x7313, 0x7318, 0x7319, 0x731e, 0x7322, 0x7324, 0x7327, 0x7328,
445 0x732c, 0x7331, 0x7332, 0x7335, 0x733a, 0x733b, 0x733d, 0x7343,
446 0x734d, 0x7350, 0x7352, 0x7356, 0x7358, 0x735d, 0x735e, 0x735f,
447 0x7360, 0x7366, 0x7367, 0x7369, 0x736b, 0x736c, 0x736e, 0x736f,
448 0x7371, 0x7377, 0x7379, 0x737c, 0x7380, 0x7381, 0x7383, 0x7385,
449 0x7386, 0x738e, 0x7390, 0x7393, 0x7395, 0x7397, 0x7398, 0x739c,
450 0x739e, 0x739f, 0x73a0, 0x73a2, 0x73a5, 0x73a6, 0x73aa, 0x73ab,
451 0x73ad, 0x73b5, 0x73b7, 0x73b9, 0x73bc, 0x73bd, 0x73bf, 0x73c5,
452 0x73c6, 0x73c9, 0x73cb, 0x73cc, 0x73cf, 0x73d2, 0x73d3, 0x73d6,
453 0x73d9, 0x73dd, 0x73e1, 0x73e3, 0x73e6, 0x73e7, 0x73e9, 0x73f4,
454 0x73f5, 0x73f7, 0x73f9, 0x73fa, 0x73fb, 0x73fd,
455 /* 0x4c */
456 0x73ff, 0x7400, 0x7401, 0x7404, 0x7407, 0x740a, 0x7411, 0x741a,
457 0x741b, 0x7424, 0x7426, 0x7428, 0x7429, 0x742a, 0x742b, 0x742c,
458 0x742d, 0x742e, 0x742f, 0x7430, 0x7431, 0x7439, 0x7440, 0x7443,
459 0x7444, 0x7446, 0x7447, 0x744b, 0x744d, 0x7451, 0x7452, 0x7457,
460 0x745d, 0x7462, 0x7466, 0x7467, 0x7468, 0x746b, 0x746d, 0x746e,
461 0x7471, 0x7472, 0x7480, 0x7481, 0x7485, 0x7486, 0x7487, 0x7489,
462 0x748f, 0x7490, 0x7491, 0x7492, 0x7498, 0x7499, 0x749a, 0x749c,
463 0x749f, 0x74a0, 0x74a1, 0x74a3, 0x74a6, 0x74a8, 0x74a9, 0x74aa,
464 0x74ab, 0x74ae, 0x74af, 0x74b1, 0x74b2, 0x74b5, 0x74b9, 0x74bb,
465 0x74bf, 0x74c8, 0x74c9, 0x74cc, 0x74d0, 0x74d3, 0x74d8, 0x74da,
466 0x74db, 0x74de, 0x74df, 0x74e4, 0x74e8, 0x74ea, 0x74eb, 0x74ef,
467 0x74f4, 0x74fa, 0x74fb, 0x74fc, 0x74ff, 0x7506,
468 /* 0x4d */
469 0x7512, 0x7516, 0x7517, 0x7520, 0x7521, 0x7524, 0x7527, 0x7529,
470 0x752a, 0x752f, 0x7536, 0x7539, 0x753d, 0x753e, 0x753f, 0x7540,
471 0x7543, 0x7547, 0x7548, 0x754e, 0x7550, 0x7552, 0x7557, 0x755e,
472 0x755f, 0x7561, 0x756f, 0x7571, 0x7579, 0x757a, 0x757b, 0x757c,
473 0x757d, 0x757e, 0x7581, 0x7585, 0x7590, 0x7592, 0x7593, 0x7595,
474 0x7599, 0x759c, 0x75a2, 0x75a4, 0x75b4, 0x75ba, 0x75bf, 0x75c0,
475 0x75c1, 0x75c4, 0x75c6, 0x75cc, 0x75ce, 0x75cf, 0x75d7, 0x75dc,
476 0x75df, 0x75e0, 0x75e1, 0x75e4, 0x75e7, 0x75ec, 0x75ee, 0x75ef,
477 0x75f1, 0x75f9, 0x7600, 0x7602, 0x7603, 0x7604, 0x7607, 0x7608,
478 0x760a, 0x760c, 0x760f, 0x7612, 0x7613, 0x7615, 0x7616, 0x7619,
479 0x761b, 0x761c, 0x761d, 0x761e, 0x7623, 0x7625, 0x7626, 0x7629,
480 0x762d, 0x7632, 0x7633, 0x7635, 0x7638, 0x7639,
481 /* 0x4e */
482 0x763a, 0x763c, 0x764a, 0x7640, 0x7641, 0x7643, 0x7644, 0x7645,
483 0x7649, 0x764b, 0x7655, 0x7659, 0x765f, 0x7664, 0x7665, 0x766d,
484 0x766e, 0x766f, 0x7671, 0x7674, 0x7681, 0x7685, 0x768c, 0x768d,
485 0x7695, 0x769b, 0x769c, 0x769d, 0x769f, 0x76a0, 0x76a2, 0x76a3,
486 0x76a4, 0x76a5, 0x76a6, 0x76a7, 0x76a8, 0x76aa, 0x76ad, 0x76bd,
487 0x76c1, 0x76c5, 0x76c9, 0x76cb, 0x76cc, 0x76ce, 0x76d4, 0x76d9,
488 0x76e0, 0x76e6, 0x76e8, 0x76ec, 0x76f0, 0x76f1, 0x76f6, 0x76f9,
489 0x76fc, 0x7700, 0x7706, 0x770a, 0x770e, 0x7712, 0x7714, 0x7715,
490 0x7717, 0x7719, 0x771a, 0x771c, 0x7722, 0x7728, 0x772d, 0x772e,
491 0x772f, 0x7734, 0x7735, 0x7736, 0x7739, 0x773d, 0x773e, 0x7742,
492 0x7745, 0x7746, 0x774a, 0x774d, 0x774e, 0x774f, 0x7752, 0x7756,
493 0x7757, 0x775c, 0x775e, 0x775f, 0x7760, 0x7762,
494 /* 0x4f */
495 0x7764, 0x7767, 0x776a, 0x776c, 0x7770, 0x7772, 0x7773, 0x7774,
496 0x777a, 0x777d, 0x7780, 0x7784, 0x778c, 0x778d, 0x7794, 0x7795,
497 0x7796, 0x779a, 0x779f, 0x77a2, 0x77a7, 0x77aa, 0x77ae, 0x77af,
498 0x77b1, 0x77b5, 0x77be, 0x77c3, 0x77c9, 0x77d1, 0x77d2, 0x77d5,
499 0x77d9, 0x77de, 0x77df, 0x77e0, 0x77e4, 0x77e6, 0x77ea, 0x77ec,
500 0x77f0, 0x77f1, 0x77f4, 0x77f8, 0x77fb, 0x7805, 0x7806, 0x7809,
501 0x780d, 0x780e, 0x7811, 0x781d, 0x7821, 0x7822, 0x7823, 0x782d,
502 0x782e, 0x7830, 0x7835, 0x7837, 0x7843, 0x7844, 0x7847, 0x7848,
503 0x784c, 0x784e, 0x7852, 0x785c, 0x785e, 0x7860, 0x7861, 0x7863,
504 0x7864, 0x7868, 0x786a, 0x786e, 0x787a, 0x787e, 0x788a, 0x788f,
505 0x7894, 0x7898, 0x78a1, 0x789d, 0x789e, 0x789f, 0x78a4, 0x78a8,
506 0x78ac, 0x78ad, 0x78b0, 0x78b1, 0x78b2, 0x78b3,
507 /* 0x50 */
508 0x78bb, 0x78bd, 0x78bf, 0x78c7, 0x78c8, 0x78c9, 0x78cc, 0x78ce,
509 0x78d2, 0x78d3, 0x78d5, 0x78d6, 0x78e4, 0x78db, 0x78df, 0x78e0,
510 0x78e1, 0x78e6, 0x78ea, 0x78f2, 0x78f3, 0x7900, 0x78f6, 0x78f7,
511 0x78fa, 0x78fb, 0x78ff, 0x7906, 0x790c, 0x7910, 0x791a, 0x791c,
512 0x791e, 0x791f, 0x7920, 0x7925, 0x7927, 0x7929, 0x792d, 0x7931,
513 0x7934, 0x7935, 0x793b, 0x793d, 0x793f, 0x7944, 0x7945, 0x7946,
514 0x794a, 0x794b, 0x794f, 0x7951, 0x7954, 0x7958, 0x795b, 0x795c,
515 0x7967, 0x7969, 0x796b, 0x7972, 0x7979, 0x797b, 0x797c, 0x797e,
516 0x798b, 0x798c, 0x7991, 0x7993, 0x7994, 0x7995, 0x7996, 0x7998,
517 0x799b, 0x799c, 0x79a1, 0x79a8, 0x79a9, 0x79ab, 0x79af, 0x79b1,
518 0x79b4, 0x79b8, 0x79bb, 0x79c2, 0x79c4, 0x79c7, 0x79c8, 0x79ca,
519 0x79cf, 0x79d4, 0x79d6, 0x79da, 0x79dd, 0x79de,
520 /* 0x51 */
521 0x79e0, 0x79e2, 0x79e5, 0x79ea, 0x79eb, 0x79ed, 0x79f1, 0x79f8,
522 0x79fc, 0x7a02, 0x7a03, 0x7a07, 0x7a09, 0x7a0a, 0x7a0c, 0x7a11,
523 0x7a15, 0x7a1b, 0x7a1e, 0x7a21, 0x7a27, 0x7a2b, 0x7a2d, 0x7a2f,

```



```

611 /* 0x58 */
612 0x837d, 0x837f, 0x8380, 0x8382, 0x8384, 0x8386, 0x838d, 0x8392,
613 0x8394, 0x8395, 0x8398, 0x839b, 0x839c, 0x839d, 0x83a6,
614 0x83a7, 0x83a9, 0x83ac, 0x83be, 0x83bf, 0x83c0, 0x83c7, 0x83c9,
615 0x83cf, 0x83d0, 0x83d1, 0x83d4, 0x83dd, 0x8353, 0x83e8, 0x83ea,
616 0x83f6, 0x83f8, 0x83f9, 0x83f9, 0x83fc, 0x8401, 0x8406, 0x840a, 0x840f,
617 0x8411, 0x8415, 0x8419, 0x83ad, 0x842f, 0x8439, 0x8445, 0x8447,
618 0x8448, 0x844a, 0x844d, 0x844f, 0x8451, 0x8452, 0x8456, 0x8458,
619 0x8459, 0x845a, 0x845c, 0x845c, 0x8460, 0x8464, 0x8465, 0x8467, 0x846a,
620 0x8470, 0x8473, 0x8474, 0x8476, 0x8478, 0x847c, 0x847d, 0x8481,
621 0x8485, 0x8492, 0x8493, 0x8495, 0x849e, 0x84a6, 0x84a8, 0x84a9,
622 0x84aa, 0x84af, 0x84b1, 0x84b4, 0x84ba, 0x84bd, 0x84be, 0x84c0,
623 0x84c2, 0x84c7, 0x84c8, 0x84cc, 0x84cf, 0x84d3,
624 /* 0x59 */
625 0x84dc, 0x84e7, 0x84ea, 0x84ef, 0x84f0, 0x84f1, 0x84f2, 0x84f7,
626 0x8532, 0x84fa, 0x84fb, 0x84fd, 0x8502, 0x8503, 0x8507, 0x850c,
627 0x850e, 0x8510, 0x851c, 0x851e, 0x8522, 0x8523, 0x8524, 0x8525,
628 0x8527, 0x8528, 0x852b, 0x852f, 0x8533, 0x8534, 0x8536, 0x853f,
629 0x8546, 0x854f, 0x8550, 0x8551, 0x8552, 0x8553, 0x8556, 0x8559,
630 0x855c, 0x855d, 0x855e, 0x855f, 0x8560, 0x8561, 0x8562, 0x8564,
631 0x856b, 0x856f, 0x8579, 0x8579, 0x857a, 0x857b, 0x857d, 0x857f, 0x8581,
632 0x8585, 0x8586, 0x8589, 0x858b, 0x858c, 0x858f, 0x8593, 0x8598,
633 0x859d, 0x859f, 0x85a0, 0x85a2, 0x85a5, 0x85a7, 0x85b4, 0x85b6,
634 0x85b7, 0x85b8, 0x85bc, 0x85bd, 0x85be, 0x85bf, 0x85c2, 0x85c7,
635 0x85ca, 0x85cb, 0x85ce, 0x85ad, 0x85d8, 0x85da, 0x85df, 0x85e0,
636 0x85e6, 0x85e8, 0x85ed, 0x85f3, 0x85f6, 0x85fc,
637 /* 0x5a */
638 0x85ff, 0x8600, 0x8604, 0x8605, 0x860d, 0x860e, 0x8610, 0x8611,
639 0x8612, 0x8618, 0x8619, 0x861b, 0x861e, 0x8621, 0x8627, 0x8629,
640 0x8636, 0x8638, 0x863a, 0x863c, 0x863d, 0x8640, 0x8642, 0x8646,
641 0x8652, 0x8653, 0x8656, 0x8657, 0x8658, 0x8659, 0x866d, 0x8660,
642 0x8661, 0x8662, 0x8663, 0x8664, 0x8669, 0x866c, 0x866f, 0x8675,
643 0x8676, 0x8677, 0x867a, 0x867a, 0x868d, 0x8691, 0x8696, 0x8698, 0x869a,
644 0x869c, 0x86a1, 0x86a6, 0x86a7, 0x86a8, 0x86ad, 0x86b1, 0x86b3,
645 0x86b4, 0x86b5, 0x86b7, 0x86b8, 0x86b9, 0x86bf, 0x86c0, 0x86c1,
646 0x86c3, 0x86c5, 0x86d1, 0x86d2, 0x86d5, 0x86d7, 0x86da, 0x86dc,
647 0x86e0, 0x86e3, 0x86e5, 0x86e7, 0x86e8, 0x86fa, 0x86fc, 0x86fd,
648 0x8704, 0x8705, 0x8707, 0x870b, 0x870e, 0x870f, 0x8710, 0x8713,
649 0x8714, 0x8719, 0x871e, 0x871f, 0x8721, 0x8723,
650 /* 0x5b */
651 0x8728, 0x872e, 0x872f, 0x8731, 0x8732, 0x8739, 0x873a, 0x873c,
652 0x873d, 0x873e, 0x8740, 0x8743, 0x8745, 0x874d, 0x8758, 0x875d,
653 0x8761, 0x8764, 0x8765, 0x876f, 0x8771, 0x8772, 0x877b, 0x8783,
654 0x8784, 0x8785, 0x8786, 0x8787, 0x8788, 0x8789, 0x878b, 0x878c,
655 0x8790, 0x8793, 0x8795, 0x8795, 0x8797, 0x8798, 0x8799, 0x879e, 0x87a0,
656 0x87a3, 0x87a7, 0x87ac, 0x87ad, 0x87ae, 0x87b1, 0x87b5, 0x87be,
657 0x87bf, 0x87c1, 0x87c8, 0x87c9, 0x87ca, 0x87ce, 0x87d5, 0x87d6,
658 0x87d9, 0x87da, 0x87dc, 0x87df, 0x87e2, 0x87e3, 0x87e4, 0x87ea,
659 0x87eb, 0x87ed, 0x87f1, 0x87f3, 0x87f8, 0x87fa, 0x87ff, 0x8801,
660 0x8803, 0x8806, 0x8809, 0x880a, 0x880b, 0x8810, 0x8819, 0x8812,
661 0x8813, 0x8814, 0x8818, 0x8818, 0x881b, 0x881c, 0x881e, 0x881f,
662 0x8828, 0x882d, 0x882e, 0x8830, 0x8832, 0x8835,
663 /* 0x5c */
664 0x883a, 0x883c, 0x8841, 0x8843, 0x8845, 0x8848, 0x8849, 0x884a,
665 0x884b, 0x884e, 0x8851, 0x8855, 0x8856, 0x8858, 0x885a, 0x885c,
666 0x885f, 0x8860, 0x8864, 0x8869, 0x8871, 0x8879, 0x887b, 0x8880,
667 0x8898, 0x889a, 0x889b, 0x889c, 0x889f, 0x88a0, 0x88a8, 0x88aa,
668 0x88ba, 0x88bd, 0x88be, 0x88c0, 0x88ca, 0x88cb, 0x88cc, 0x88cd,
669 0x88ce, 0x88d1, 0x88d2, 0x88d3, 0x88db, 0x88de, 0x88e7, 0x88ef,
670 0x88f0, 0x88f1, 0x88f5, 0x88f7, 0x8901, 0x8906, 0x890d, 0x890e,
671 0x890f, 0x8915, 0x8916, 0x8918, 0x8919, 0x891a, 0x891c, 0x8920,
672 0x8926, 0x8927, 0x8928, 0x8930, 0x8931, 0x8932, 0x8935, 0x8939,
673 0x893a, 0x893e, 0x8940, 0x8940, 0x8942, 0x8945, 0x8946, 0x894f,
674 0x8952, 0x8957, 0x895a, 0x895b, 0x895c, 0x8961, 0x8962, 0x8963,
675 0x896b, 0x896e, 0x8970, 0x8973, 0x8975, 0x897a,
676 /* 0x5d */
677 0x897b, 0x897c, 0x897d, 0x8989, 0x898d, 0x8990, 0x8994, 0x8995,
678 0x899b, 0x899c, 0x899f, 0x89a0, 0x89a5, 0x89b0, 0x89b4, 0x89b5,
679 0x89b6, 0x89b7, 0x89bc, 0x89d4, 0x89d5, 0x89d6, 0x89d7, 0x89d8,
680 0x89e5, 0x89e9, 0x89eb, 0x89ed, 0x89f1, 0x89f3, 0x89f6, 0x89f9,
681 0x89fd, 0x89ff, 0x8a04, 0x8a05, 0x8a07, 0x8a0f, 0x8a11, 0x8a12,
682 0x8a14, 0x8a15, 0x8a1e, 0x8a20, 0x8a22, 0x8a24, 0x8a26, 0x8a2b,
683 0x8a2c, 0x8a2f, 0x8a35, 0x8a37, 0x8a3d, 0x8a3e, 0x8a40, 0x8a43,
684 0x8a45, 0x8a47, 0x8a49, 0x8a4d, 0x8a4e, 0x8a53, 0x8a56, 0x8a57,
685 0x8a58, 0x8a5c, 0x8a5d, 0x8a61, 0x8a65, 0x8a67, 0x8a75, 0x8a76,
686 0x8a77, 0x8a79, 0x8a7a, 0x8a7b, 0x8a7e, 0x8a7f, 0x8a80, 0x8a83,
687 0x8a86, 0x8a8b, 0x8a8f, 0x8a90, 0x8a92, 0x8a96, 0x8a97, 0x8a99,
688 0x8a9f, 0x8aa7, 0x8aa9, 0x8aae, 0x8aab,
689 /* 0x5e */
690 0x8ab6, 0x8ab7, 0x8abb, 0x8abe, 0x8ac3, 0x8ac6, 0x8ac8, 0x8ac9,
691 0x8aca, 0x8ad1, 0x8ad3, 0x8ad4, 0x8ad5, 0x8ad7, 0x8add, 0x8adf,
692 0x8aec, 0x8af0, 0x8af4, 0x8af5, 0x8af6, 0x8afc, 0x8aff, 0x8b05,
693 0x8b06, 0x8b0b, 0x8b11, 0x8b1c, 0x8b1e, 0x8b1f, 0x8b0a, 0x8b2d,
694 0x8b30, 0x8b37, 0x8b3c, 0x8b42, 0x8b43, 0x8b44, 0x8b45, 0x8b46,
695 0x8b48, 0x8b52, 0x8b53, 0x8b54, 0x8b59, 0x8b64, 0x8b5e, 0x8b63,
696 0x8b6d, 0x8b76, 0x8b78, 0x8b78, 0x8b7c, 0x8b7e, 0x8b81, 0x8b84,
697 0x8b85, 0x8b8b, 0x8b8d, 0x8b8f, 0x8b94, 0x8b95, 0x8b9c, 0x8b9e,

```

```
698 0x8b9f, 0x8c38, 0x8c39, 0x8c3d, 0x8c3e, 0x8c45, 0x8c47, 0x8c49,
699 0x8c4b, 0x8c4f, 0x8c51, 0x8c53, 0x8c54, 0x8c57, 0x8c58, 0x8c5b,
700 0x8c5d, 0x8c59, 0x8c63, 0x8c64, 0x8c66, 0x8c68, 0x8c69, 0x8c6d,
701 0x8c73, 0x8c75, 0x8c76, 0x8c7b, 0x8c7e, 0x8c86,
702 /* 0x5f */
703 0x8c87, 0x8c8b, 0x8c90, 0x8c92, 0x8c93, 0x8c99, 0x8c9b, 0x8c9c,
704 0x8ca4, 0x8cb9, 0x8cba, 0x8cc5, 0x8cc6, 0x8cc9, 0x8ccb, 0x8ccf,
705 0x8cd6, 0x8cd5, 0x8cd9, 0x8cdd, 0x8ce1, 0x8ce8, 0x8cec, 0x8cef,
706 0x8cf0, 0x8cf2, 0x8cf5, 0x8cf7, 0x8cf8, 0x8cfe, 0x8cff, 0x8d01,
707 0x8d03, 0x8d09, 0x8d12, 0x8d17, 0x8d1b, 0x8d65, 0x8d69, 0x8d6c,
708 0x8d6e, 0x8d7f, 0x8d82, 0x8d84, 0x8d88, 0x8d8d, 0x8d90, 0x8d91,
709 0x8d95, 0x8d9e, 0x8d9f, 0x8da0, 0x8da6, 0x8dab, 0x8dac, 0x8daf,
710 0x8db2, 0x8db5, 0x8db7, 0x8db9, 0x8dbb, 0x8dc0, 0x8dc5, 0x8dc6,
711 0x8dc7, 0x8dc8, 0x8dca, 0x8dce, 0x8dd1, 0x8ddd, 0x8dd5, 0x8dd7,
712 0x8dd9, 0x8de4, 0x8de5, 0x8de7, 0x8dec, 0x8df0, 0x8dbc, 0x8df1,
713 0x8df2, 0x8df4, 0x8dfd, 0x8e01, 0x8e04, 0x8e05, 0x8e06, 0x8e0b,
714 0x8e11, 0x8e14, 0x8e16, 0x8e20, 0x8e21, 0x8e22,
715 /* 0x60 */
716 0x8e23, 0x8e26, 0x8e27, 0x8e31, 0x8e33, 0x8e36, 0x8e37, 0x8e38,
717 0x8e39, 0x8e3d, 0x8e40, 0x8e41, 0x8e4b, 0x8e4d, 0x8e4e, 0x8e4f,
718 0x8e54, 0x8e5b, 0x8e5c, 0x8e5d, 0x8e5e, 0x8e61, 0x8e62, 0x8e69,
719 0x8e6c, 0x8e6d, 0x8e6f, 0x8e70, 0x8e71, 0x8e79, 0x8e7a, 0x8e7b,
720 0x8e82, 0x8e83, 0x8e89, 0x8e90, 0x8e92, 0x8e95, 0x8e9a, 0x8e9b,
721 0x8e9d, 0x8e9e, 0x8ea2, 0x8ea7, 0x8ea9, 0x8ead, 0x8eae, 0x8eb3,
722 0x8eb5, 0x8eba, 0x8ebb, 0x8ec0, 0x8ec1, 0x8ec3, 0x8ec4, 0x8ec7,
723 0x8ecf, 0x8ed1, 0x8ed4, 0x8edc, 0x8ee8, 0x8eee, 0x8ef0, 0x8ef1,
724 0x8ef7, 0x8ef9, 0x8efa, 0x8eed, 0x8f00, 0x8f02, 0x8f07, 0x8f08,
725 0x8f0f, 0x8f10, 0x8f16, 0x8f17, 0x8f18, 0x8f1e, 0x8f20, 0x8f21,
726 0x8f23, 0x8f25, 0x8f27, 0x8f28, 0x8f2c, 0x8f2d, 0x8f2e, 0x8f34,
727 0x8f35, 0x8f36, 0x8f37, 0x8f3a, 0x8f40, 0x8f41,
728 /* 0x61 */
729 0x8f43, 0x8f47, 0x8f4f, 0x8f51, 0x8f52, 0x8f53, 0x8f54, 0x8f55,
730 0x8f58, 0x8f5d, 0x8f5e, 0x8f65, 0x8f9d, 0x8fa0, 0x8fa1, 0x8fa4,
731 0x8fa5, 0x8fa6, 0x8fb5, 0x8fb6, 0x8fb8, 0x8fbe, 0x8fc0, 0x8fc1,
732 0x8fc6, 0x8fca, 0x8fcb, 0x8fcd, 0x8fd0, 0x8fd2, 0x8fd3, 0x8fd5,
733 0x8fe0, 0x8fe3, 0x8fe4, 0x8fe8, 0x8fee, 0x8ff1, 0x8ff5, 0x8ff6,
734 0x8ffb, 0x8ffe, 0x9002, 0x9004, 0x9008, 0x900c, 0x9018, 0x901b,
735 0x9028, 0x9029, 0x902f, 0x902a, 0x902c, 0x902d, 0x9033, 0x9034,
736 0x9037, 0x903f, 0x9043, 0x9044, 0x904c, 0x905b, 0x905d, 0x9062,
737 0x9066, 0x9067, 0x906c, 0x9070, 0x9074, 0x9079, 0x9085, 0x9088,
738 0x908b, 0x908c, 0x908e, 0x9090, 0x9095, 0x9097, 0x9098, 0x9099,
739 0x909b, 0x90a0, 0x90a1, 0x90a2, 0x90a5, 0x90b0, 0x90b2, 0x90b3,
740 0x90b4, 0x90b6, 0x90bd, 0x90cc, 0x90be, 0x90c3,
741 /* 0x62 */
742 0x90c4, 0x90c5, 0x90c7, 0x90c8, 0x90d5, 0x90d7, 0x90d8, 0x90d9,
743 0x90dc, 0x90dd, 0x90df, 0x90e5, 0x90d2, 0x90f6, 0x90eb, 0x90ef,
744 0x90f0, 0x90f4, 0x90fe, 0x90ff, 0x9100, 0x9104, 0x9105, 0x9106,
745 0x9108, 0x910d, 0x9110, 0x9114, 0x9116, 0x9117, 0x9118, 0x911a,
746 0x911c, 0x911e, 0x9120, 0x9125, 0x9122, 0x9123, 0x9127, 0x9129,
747 0x912e, 0x912f, 0x9131, 0x9134, 0x9136, 0x9137, 0x9139, 0x913a,
748 0x913c, 0x913d, 0x9143, 0x9147, 0x9148, 0x914f, 0x9153, 0x9157,
749 0x9159, 0x915a, 0x915b, 0x9161, 0x9164, 0x9167, 0x916d, 0x9174,
750 0x9179, 0x917a, 0x917b, 0x9181, 0x9183, 0x9185, 0x9186, 0x918a,
751 0x918e, 0x9191, 0x9193, 0x9194, 0x9195, 0x9198, 0x919e, 0x91a1,
752 0x91a6, 0x91a8, 0x91ac, 0x91ad, 0x91ae, 0x91b0, 0x91b1, 0x91b2,
753 0x91b3, 0x91b6, 0x91bb, 0x91bc, 0x91bd, 0x91bf,
754 /* 0x63 */
755 0x91c2, 0x91c3, 0x91c5, 0x91d3, 0x91d4, 0x91d7, 0x91d9, 0x91da,
756 0x91de, 0x91e4, 0x91e5, 0x91e9, 0x91ea, 0x91ec, 0x91ed, 0x91ee,
757 0x91ef, 0x91f0, 0x91f1, 0x91f7, 0x91f9, 0x91fb, 0x91fd, 0x9200,
758 0x9201, 0x9204, 0x9205, 0x9206, 0x9207, 0x9209, 0x920a, 0x920c,
759 0x9210, 0x9212, 0x9213, 0x9216, 0x9218, 0x921c, 0x921d, 0x9223,
760 0x9224, 0x9225, 0x9226, 0x9228, 0x922e, 0x922f, 0x9230, 0x9233,
761 0x9235, 0x9236, 0x9238, 0x9239, 0x923a, 0x923c, 0x923e, 0x9240,
762 0x9242, 0x9243, 0x9246, 0x9247, 0x924a, 0x924d, 0x924e, 0x924f,
763 0x9251, 0x9258, 0x9259, 0x925c, 0x925d, 0x9260, 0x9261, 0x9265,
764 0x9267, 0x9268, 0x9269, 0x926e, 0x926f, 0x9270, 0x9275, 0x9276,
765 0x9277, 0x9278, 0x9279, 0x927b, 0x927c, 0x927d, 0x927f, 0x9288,
766 0x9289, 0x928a, 0x928d, 0x928e, 0x9292, 0x9297,
767 /* 0x64 */
768 0x9299, 0x929f, 0x92a0, 0x92a4, 0x92a5, 0x92a7, 0x92a8, 0x92ab,
769 0x92af, 0x92b2, 0x92b6, 0x92b8, 0x92ba, 0x92bb, 0x92bc, 0x92bd,
770 0x92bf, 0x92c0, 0x92c1, 0x92c2, 0x92c3, 0x92c5, 0x92c6, 0x92c7,
771 0x92c8, 0x92cb, 0x92cc, 0x92cd, 0x92ce, 0x92d0, 0x92d3, 0x92d5,
772 0x92d7, 0x92d8, 0x92d9, 0x92dc, 0x92dd, 0x92de, 0x92e0, 0x92e1,
773 0x92e3, 0x92e5, 0x92e7, 0x92e8, 0x92ec, 0x92ee, 0x92f0, 0x92f9,
774 0x92fb, 0x92ff, 0x9300, 0x9302, 0x9308, 0x930d, 0x9311, 0x9314,
775 0x9315, 0x931c, 0x931d, 0x931e, 0x931f, 0x9321, 0x9324, 0x9325,
776 0x9327, 0x9329, 0x932a, 0x9333, 0x9334, 0x9336, 0x9337, 0x9347,
777 0x9348, 0x9349, 0x9350, 0x9351, 0x9352, 0x9355, 0x9357, 0x9358,
778 0x935a, 0x935e, 0x9364, 0x9365, 0x9367, 0x9369, 0x936a, 0x936d,
779 0x936f, 0x9370, 0x9371, 0x9373, 0x9374, 0x9376,
780 /* 0x65 */
781 0x937a, 0x937d, 0x937f, 0x9380, 0x9381, 0x9382, 0x9388, 0x938a,
782 0x938b, 0x938d, 0x938f, 0x9392, 0x9395, 0x9398, 0x939b, 0x939e,
783 0x93a1, 0x93a3, 0x93a4, 0x93a6, 0x93a8, 0x93ab, 0x93ba, 0x93b5,
784 0x93b6, 0x93ba, 0x93a9, 0x93c1, 0x93c4, 0x93c5, 0x93c6, 0x93c7,
```

```

785 0x93c9, 0x93ca, 0x93cb, 0x93cc, 0x93cd, 0x93d3, 0x93d9, 0x93dc,
786 0x93de, 0x93df, 0x93e2, 0x93e6, 0x93e7, 0x93f9, 0x93f7, 0x93f8,
787 0x93fa, 0x93fb, 0x93fd, 0x9401, 0x9402, 0x9404, 0x9408, 0x9409,
788 0x940d, 0x940e, 0x940f, 0x9415, 0x9416, 0x9417, 0x941f, 0x942e,
789 0x942f, 0x9431, 0x9432, 0x9433, 0x9434, 0x943b, 0x943f, 0x943d,
790 0x9443, 0x9445, 0x9448, 0x944a, 0x944c, 0x9455, 0x9459, 0x945c,
791 0x945f, 0x9461, 0x9463, 0x9468, 0x946b, 0x946d, 0x946e, 0x946f,
792 0x9471, 0x9472, 0x9484, 0x9483, 0x9578, 0x9579,
793 /* 0x66 */
794 0x957e, 0x9584, 0x9588, 0x958c, 0x958d, 0x958e, 0x959d, 0x959e,
795 0x959f, 0x95a1, 0x95a6, 0x95a9, 0x95ab, 0x95ac, 0x95b4, 0x95b6,
796 0x95ba, 0x95bd, 0x95bf, 0x95c6, 0x95c8, 0x95c9, 0x95cb, 0x95d0,
797 0x95d1, 0x95d2, 0x95d3, 0x95d9, 0x95da, 0x95dd, 0x95de, 0x95df,
798 0x95e0, 0x95e4, 0x95e6, 0x961d, 0x961e, 0x9622, 0x9624, 0x9625,
799 0x9626, 0x962c, 0x9631, 0x9633, 0x9637, 0x9638, 0x9639, 0x963a,
800 0x963c, 0x963d, 0x9641, 0x9652, 0x9654, 0x9656, 0x9657, 0x9658,
801 0x9661, 0x966e, 0x9674, 0x967b, 0x967c, 0x967e, 0x967f, 0x9681,
802 0x9682, 0x9683, 0x9684, 0x9689, 0x9691, 0x9696, 0x969a, 0x969d,
803 0x969f, 0x96a4, 0x96a5, 0x96a6, 0x96a9, 0x96ae, 0x96af, 0x96b3,
804 0x96ba, 0x96ca, 0x96d2, 0x5db2, 0x96d8, 0x96da, 0x96dd, 0x96de,
805 0x96df, 0x96e9, 0x96ef, 0x96f1, 0x96fa, 0x9702,
806 /* 0x67 */
807 0x9703, 0x9705, 0x9709, 0x971a, 0x971b, 0x971d, 0x9721, 0x9722,
808 0x9723, 0x9728, 0x9728, 0x9731, 0x9733, 0x9741, 0x9743, 0x974a, 0x974e,
809 0x974f, 0x9755, 0x9757, 0x9758, 0x975a, 0x975b, 0x9763, 0x9767,
810 0x976a, 0x976e, 0x9773, 0x9776, 0x9777, 0x9778, 0x977b, 0x977d,
811 0x977f, 0x9780, 0x9789, 0x9795, 0x9796, 0x9797, 0x9799, 0x979a,
812 0x979e, 0x979f, 0x97a2, 0x97ac, 0x97ae, 0x97b1, 0x97b2, 0x97b5,
813 0x97b6, 0x97b8, 0x97b9, 0x97ba, 0x97bc, 0x97be, 0x97bf, 0x97c1,
814 0x97c4, 0x97c5, 0x97c7, 0x97c9, 0x97ca, 0x97cc, 0x97cd, 0x97ce,
815 0x97d0, 0x97d1, 0x97d4, 0x97d7, 0x97d8, 0x97d9, 0x97dd, 0x97de,
816 0x97e0, 0x97db, 0x97e1, 0x97e4, 0x97ef, 0x97f1, 0x97f4, 0x97f7,
817 0x97f8, 0x97fa, 0x9807, 0x980a, 0x9819, 0x980d, 0x980e, 0x9814,
818 0x9816, 0x981c, 0x981e, 0x9820, 0x9823, 0x9826,
819 /* 0x68 */
820 0x982b, 0x982e, 0x982f, 0x9830, 0x9832, 0x9833, 0x9835, 0x9825,
821 0x983e, 0x9844, 0x9847, 0x984a, 0x9851, 0x9852, 0x9853, 0x9856,
822 0x9857, 0x9859, 0x985a, 0x9862, 0x9863, 0x9865, 0x9866, 0x986a,
823 0x986c, 0x98ab, 0x98ad, 0x98ae, 0x98b0, 0x98b4, 0x98b7, 0x98b8,
824 0x98ba, 0x98bb, 0x98bf, 0x98c2, 0x98c5, 0x98c8, 0x98cc, 0x98e1,
825 0x98e3, 0x98e5, 0x98e6, 0x98e7, 0x98ea, 0x98f3, 0x98f6, 0x9902,
826 0x9907, 0x9908, 0x9911, 0x9915, 0x9916, 0x9917, 0x991a, 0x991b,
827 0x991c, 0x991f, 0x9922, 0x9926, 0x9927, 0x992b, 0x9931, 0x9932,
828 0x9933, 0x9934, 0x9935, 0x9939, 0x993a, 0x993b, 0x993c, 0x9940,
829 0x9941, 0x9946, 0x9947, 0x9948, 0x994d, 0x994e, 0x9954, 0x9958,
830 0x9959, 0x995b, 0x995c, 0x995e, 0x995f, 0x9960, 0x999b, 0x999d,
831 0x999f, 0x99a6, 0x99b0, 0x99b1, 0x99b2, 0x99b5,
832 /* 0x69 */
833 0x99b9, 0x99ba, 0x99bd, 0x99bf, 0x99c3, 0x99c9, 0x99d3, 0x99d4,
834 0x99d9, 0x99da, 0x99dc, 0x99de, 0x99e7, 0x99ea, 0x99eb, 0x99ec,
835 0x99f0, 0x99f4, 0x99f5, 0x99f9, 0x99fd, 0x99fe, 0x9a02, 0x9a03,
836 0x9a04, 0x9a0b, 0x9a0c, 0x9a10, 0x9a11, 0x9a16, 0x9a1e, 0x9a20,
837 0x9a22, 0x9a23, 0x9a24, 0x9a27, 0x9a2d, 0x9a2e, 0x9a33, 0x9a35,
838 0x9a36, 0x9a38, 0x9a47, 0x9a41, 0x9a44, 0x9a4a, 0x9a4b, 0x9a4c,
839 0x9a4e, 0x9a51, 0x9a54, 0x9a56, 0x9a5d, 0x9aaa, 0x9aac, 0x9aae,
840 0x9aa f, 0x9ab2, 0x9ab4, 0x9ab5, 0x9ab6, 0x9ab9, 0x9abb, 0x9abe,
841 0x9abf, 0x9ac1, 0x9ac3, 0x9ac6, 0x9ac8, 0x9ace, 0x9ad0, 0x9ad2,
842 0x9ad5, 0x9ad6, 0x9ad7, 0x9adb, 0x9adc, 0x9ae0, 0x9ae4, 0x9ae5,
843 0x9ae7, 0x9ae9, 0x9aec, 0x9af2, 0x9af3, 0x9af5, 0x9af9, 0x9afa,
844 0x9afd, 0x9aff, 0x9b00, 0x9b01, 0x9b02, 0x9b03,
845 /* 0x6a */
846 0x9b04, 0x9b05, 0x9b08, 0x9b09, 0x9b0b, 0x9b0c, 0x9b0d, 0x9b0e,
847 0x9b10, 0x9b12, 0x9b16, 0x9b19, 0x9b1b, 0x9b1c, 0x9b20, 0x9b26,
848 0x9b2b, 0x9b2d, 0x9b33, 0x9b34, 0x9b35, 0x9b37, 0x9b39, 0x9b3a,
849 0x9b3d, 0x9b4b, 0x9b4c, 0x9b55, 0x9b56, 0x9b57, 0x9b5b,
850 0x9b5e, 0x9b61, 0x9b63, 0x9b65, 0x9b66, 0x9b68, 0x9b6a, 0x9b6b,
851 0x9b6c, 0x9b6d, 0x9b6e, 0x9b73, 0x9b75, 0x9b77, 0x9b78, 0x9b79,
852 0x9b7f, 0x9b80, 0x9b84, 0x9b85, 0x9b86, 0x9b87, 0x9b89, 0x9b8a,
853 0x9b8b, 0x9b8d, 0x9b8f, 0x9b90, 0x9b94, 0x9b9a, 0x9b9d, 0x9b9e,
854 0x9ba6, 0x9ba7, 0x9ba9, 0x9bac, 0x9bb0, 0x9bb1, 0x9bb2, 0x9bb7,
855 0x9bb8, 0x9bbb, 0x9bbc, 0x9bbe, 0x9bbf, 0x9bc1, 0x9bc7, 0x9bc8,
856 0x9bce, 0x9bd0, 0x9bd7, 0x9bd8, 0x9bdd, 0x9bdf, 0x9be5, 0x9be7,
857 0x9bea, 0x9beb, 0x9bef, 0x9bf3, 0x9bf7, 0x9bf8,
858 /* 0x6b */
859 0x9bf9, 0x9bfa, 0x9bfd, 0x9bff, 0x9c00, 0x9c02, 0x9c0b, 0x9c0f,
860 0x9c11, 0x9c16, 0x9c18, 0x9c19, 0x9c1a, 0x9c1c, 0x9c1e, 0x9c22,
861 0x9c23, 0x9c26, 0x9c27, 0x9c28, 0x9c29, 0x9c2a, 0x9c31, 0x9c35,
862 0x9c36, 0x9c37, 0x9c3d, 0x9c41, 0x9c43, 0x9c44, 0x9c45, 0x9c49,
863 0x9c4a, 0x9c4e, 0x9c4f, 0x9c50, 0x9c53, 0x9c54, 0x9c56, 0x9c58,
864 0x9c5b, 0x9c5d, 0x9c5e, 0x9c5f, 0x9c63, 0x9c69, 0x9c6a, 0x9c5c,
865 0x9c6b, 0x9c68, 0x9c6e, 0x9c70, 0x9c72, 0x9c75, 0x9c77, 0x9c7b,
866 0x9ce6, 0x9cf2, 0x9cf7, 0x9cf9, 0x9d0b, 0x9d02, 0x9d11, 0x9d17,
867 0x9d18, 0x9d1c, 0x9d1d, 0x9d1e, 0x9d2f, 0x9d30, 0x9d32, 0x9d33,
868 0x9d34, 0x9d3a, 0x9d3c, 0x9d45, 0x9d3d, 0x9d42, 0x9d43, 0x9d47,
869 0x9d4a, 0x9d53, 0x9d54, 0x9d5f, 0x9d63, 0x9d62, 0x9d65, 0x9d69,
870 0x9d6a, 0x9d6b, 0x9d70, 0x9d76, 0x9d77, 0x9d7b,
871 /* 0x6c */

```



```
872 0x9d7c, 0x9d7e, 0x9d83, 0x9d84, 0x9d86, 0x9d8a, 0x9d8d, 0x9d8e,
873 0x9d92, 0x9d93, 0x9d95, 0x9d96, 0x9d97, 0x9d98, 0x9da1, 0x9daa,
874 0x9dac, 0x9dae, 0x9db1, 0x9db5, 0x9db9, 0x9dbc, 0x9dbf, 0x9dc3,
875 0x9dc7, 0x9dc9, 0x9dca, 0x9dd4, 0x9dd5, 0x9dd6, 0x9dd7, 0x9dda,
876 0x9dde, 0x9ddf, 0x9de0, 0x9de5, 0x9de7, 0x9de9, 0x9deb, 0x9dee,
877 0x9df0, 0x9df3, 0x9df4, 0x9dfe, 0x9e0a, 0x9e02, 0x9e07, 0x9e0e,
878 0x9e10, 0x9e11, 0x9e12, 0x9e15, 0x9e16, 0x9e19, 0x9e1c, 0x9e1d,
879 0x9e7a, 0x9e7b, 0x9e7c, 0x9e80, 0x9e82, 0x9e83, 0x9e84, 0x9e85,
880 0x9e87, 0x9e8e, 0x9e8f, 0x9e96, 0x9e98, 0x9e9b, 0x9e9e, 0x9ea4,
881 0x9ea8, 0x9eac, 0x9eae, 0x9eaf, 0x9eb0, 0x9eb3, 0x9eb4, 0x9eb5,
882 0x9ec6, 0x9ec8, 0x9ecb, 0x9ed5, 0x9edf, 0x9ee4, 0x9ee7, 0x9eec,
883 0x9eed, 0x9eee, 0x9ef0, 0x9ef1, 0x9ef2, 0x9ef5,
884 /* 0x6d */
885 0x9ef8, 0x9eff, 0x9f02, 0x9f03, 0x9f09, 0x9f0f, 0x9f10, 0x9f11,
886 0x9f12, 0x9f14, 0x9f16, 0x9f17, 0x9f19, 0x9f1a, 0x9f1b, 0x9f1f,
887 0x9f22, 0x9f26, 0x9f2a, 0x9f2b, 0x9f2f, 0x9f31, 0x9f32, 0x9f34,
888 0x9f37, 0x9f39, 0x9f3a, 0x9f3c, 0x9f3d, 0x9f3f, 0x9f41, 0x9f43,
889 0x9f44, 0x9f45, 0x9f46, 0x9f47, 0x9f53, 0x9f55, 0x9f56, 0x9f57,
890 0x9f58, 0x9f5a, 0x9f5d, 0x9f5e, 0x9f68, 0x9f69, 0x9f6d, 0x9f6e,
891 0x9f6f, 0x9f70, 0x9f71, 0x9f73, 0x9f75, 0x9f7a, 0x9f7d, 0x9f8f,
892 0x9f90, 0x9f91, 0x9f92, 0x9f94, 0x9f96, 0x9f97, 0x9f9e, 0x9fa1,
893 0x9fa2, 0x9fa3, 0x9fa5,
894 };
895
896 static int
897 jisx0212_mbtowc (conv_t conv, ucs4_t *pwc, const unsigned char *s, int n)
898 {
899     unsigned char c1 = (s[0] & 0x7F);
900     if ((c1 == 0x22) || (c1 >= 0x26 && c1 <= 0x27) || (c1 >= 0x29 && c1 <= 0x2b) || (c1 >= 0x30 && c1 <=
0x6d)) {
901         if (n >= 2) {
902             unsigned char c2 = (s[1] & 0x7F);
903             if (c2 >= 0x21 && c2 < 0x7f) {
904                 unsigned int i = 94 * (c1 - 0x21) + (c2 - 0x21);
905                 unsigned short wc = 0xffff;
906                 if (i < 470) {
907                     if (i < 175)
908                         wc = jisx0212_2uni_page22[i-94];
909                     } else if (i < 752) {
910                         if (i < 658)
911                             wc = jisx0212_2uni_page26[i-470];
912                     } else if (i < 1410) {
913                         if (i < 1027)
914                             wc = jisx0212_2uni_page29[i-752];
915                     } else {
916                         if (i < 7211)
917                             wc = jisx0212_2uni_page30[i-1410];
918                     }
919                     if (wc != 0xffff) {
920                         *pwc = (ucs4_t) wc;
921                         return 2;
922                     }
923                 }
924                 return RET_ILSEQ;
925             }
926             return RET_TOOFEW(0);
927         }
928         return RET_ILSEQ;
929     }
930 #endif /* NEED_TOWC */
931
932 #ifndef NEED_TOMB
933 static const unsigned short jisx0212_2charset[6067] = {
934 0x2237, 0x2242, 0x2270, 0x2243, 0x226d, 0x226c, 0x226e, 0x2234,
935 0x2231, 0x226b, 0x2244, 0x2a22, 0x2a21, 0x2a24, 0x2a2a, 0x2a23,
936 0x2a29, 0x2921, 0x2a2e, 0x2a32, 0x2a31, 0x2a34, 0x2a33, 0x2a40,
937 0x2a3f, 0x2a42, 0x2a41, 0x2a50, 0x2a52, 0x2a51, 0x2a54, 0x2a58,
938 0x2a53, 0x292c, 0x2a63, 0x2a62, 0x2a65, 0x2a64, 0x2a72, 0x2930,
939 0x294e, 0x2b22, 0x2b21, 0x2b24, 0x2b2a, 0x2b23, 0x2b29, 0x2941,
940 0x2b2e, 0x2b32, 0x2b31, 0x2b34, 0x2b33, 0x2b40, 0x2b3f, 0x2b42,
941 0x2b41, 0x2943, 0x2b50, 0x2b52, 0x2b51, 0x2b54, 0x2b58, 0x2b53,
942 0x294c, 0x2b63, 0x2b62, 0x2b65, 0x2b64, 0x2b72, 0x2950, 0x2b73,
943 0x2a27, 0x2b27, 0x2a25, 0x2b25, 0x2a28, 0x2b28, 0x2a2b, 0x2b2b,
944 0x2a2c, 0x2b2c, 0x2a2f, 0x2b2f, 0x2a2d, 0x2b2d, 0x2a30, 0x2b30,
945 0x2922, 0x2942, 0x2a37, 0x2b37, 0x2a36, 0x2b36, 0x2a38, 0x2b38,
946 0x2a35, 0x2b35, 0x2a3a, 0x2b3a, 0x2a3b, 0x2b3b, 0x2a3d, 0x2b3d,
947 0x2a3c, 0x2a3e, 0x2b3e, 0x2924, 0x2944, 0x2a47, 0x2b47, 0x2a45,
948 0x2b45, 0x2a46, 0x2b46, 0x2a44, 0x2945, 0x2926, 0x2946, 0x2a48,
949 0x2b48, 0x2a49, 0x2b49, 0x2947, 0x2a4a, 0x2b4a, 0x2a4c, 0x2b4c,
950 0x2a4b, 0x2b4b, 0x2929, 0x2928, 0x2948, 0x2a4d, 0x2b4d,
951 0x2a4f, 0x2b4f, 0x2a4e, 0x2b4e, 0x294a, 0x292b, 0x294b, 0x2a57,
952 0x2b57, 0x2a56, 0x2b56, 0x292d, 0x294d, 0x2a59, 0x2b59, 0x2a5b,
953 0x2b5b, 0x2a5a, 0x2b5a, 0x2a5c, 0x2b5c, 0x2a5d, 0x2b5d, 0x2a5f,
954 0x2b5f, 0x2a5e, 0x2b5e, 0x2a61, 0x2b61, 0x2a60, 0x2b60, 0x292f,
955 0x294f, 0x2a6c, 0x2b6c, 0x2a69, 0x2b69, 0x2a66, 0x2b66, 0x2a6b,
956 0x2b6b, 0x2a68, 0x2b68, 0x2a6a, 0x2b6a, 0x2a71, 0x2b71, 0x2a74,
957 0x2b74, 0x2a73, 0x2a75, 0x2b75, 0x2a77, 0x2b77, 0x2a76, 0x2b76,
```

958 0x2a26, 0x2b26, 0x2a43, 0x2b43, 0x2a55, 0x2b55, 0x2a67, 0x2b67,
959 0x2a70, 0x2b70, 0x2a6d, 0x2b6d, 0x2a6f, 0x2b6f, 0x2a6e, 0x2b6e,
960 0x2b39, 0x2230, 0x2230, 0x222f, 0x2232, 0x2236, 0x2235, 0x2233, 0x2238,
961 0x2239, 0x2661, 0x2662, 0x2663, 0x2664, 0x2667, 0x2669, 0x266c,
962 0x2676, 0x2665, 0x266a, 0x2671, 0x2672, 0x2673, 0x2674, 0x267b,
963 0x2678, 0x2675, 0x267a, 0x2677, 0x2679, 0x267c, 0x2742, 0x2743,
964 0x2744, 0x2745, 0x2746, 0x2747, 0x2748, 0x2749, 0x274a, 0x274b,
965 0x274c, 0x274d, 0x274e, 0x2772, 0x2773, 0x2774, 0x2775, 0x2776,
966 0x2777, 0x2778, 0x2779, 0x277a, 0x277b, 0x277c, 0x277d, 0x277e,
967 0x2271, 0x226f, 0x3021, 0x3022, 0x3023, 0x3024, 0x3025, 0x3026,
968 0x3027, 0x3028, 0x3029, 0x302a, 0x302b, 0x302c, 0x302d, 0x302e,
969 0x302f, 0x3030, 0x3031, 0x3032, 0x3033, 0x3034, 0x3035, 0x3036,
970 0x3037, 0x3038, 0x3039, 0x303a, 0x303b, 0x303c, 0x303d, 0x303e,
971 0x303f, 0x3040, 0x3041, 0x3042, 0x3043, 0x3044, 0x3045, 0x3046,
972 0x3047, 0x3048, 0x3049, 0x304a, 0x304b, 0x304c, 0x304d, 0x304e,
973 0x304f, 0x3050, 0x3051, 0x3052, 0x3053, 0x3054, 0x3055, 0x3056,
974 0x3057, 0x3058, 0x3059, 0x305a, 0x305b, 0x305c, 0x305d, 0x305e,
975 0x3060, 0x3061, 0x3062, 0x3063, 0x3064, 0x3065, 0x3066, 0x3067,
976 0x3068, 0x3069, 0x306a, 0x306b, 0x306c, 0x306d, 0x306e, 0x306f,
977 0x3070, 0x305f, 0x3071, 0x3072, 0x3073, 0x3074, 0x3075, 0x3076,
978 0x3077, 0x3078, 0x3079, 0x307a, 0x307b, 0x307c, 0x307d, 0x307e,
979 0x3121, 0x3122, 0x3123, 0x3124, 0x3125, 0x3126, 0x3127, 0x3128,
980 0x3129, 0x312a, 0x312b, 0x312c, 0x312d, 0x312e, 0x312f, 0x3130,
981 0x3131, 0x3132, 0x3133, 0x3134, 0x3135, 0x3136, 0x3137, 0x3138,
982 0x3139, 0x313a, 0x313b, 0x313c, 0x313d, 0x313e, 0x313f, 0x3140,
983 0x3141, 0x3142, 0x3143, 0x3144, 0x3145, 0x3146, 0x3147, 0x3148,
984 0x3149, 0x314a, 0x314b, 0x314c, 0x314d, 0x314e, 0x314f, 0x3150,
985 0x3151, 0x3152, 0x3153, 0x3154, 0x3155, 0x3156, 0x3157, 0x3158,
986 0x3159, 0x315a, 0x315b, 0x315c, 0x315d, 0x315e, 0x315f, 0x3160,
987 0x3161, 0x3162, 0x3163, 0x3164, 0x3165, 0x3166, 0x3167,
988 0x3168, 0x3169, 0x316a, 0x316b, 0x316c, 0x316d, 0x316e, 0x316f,
989 0x3170, 0x3171, 0x3172, 0x3173, 0x3174, 0x3175, 0x3176, 0x3177,
990 0x3178, 0x3179, 0x317a, 0x317b, 0x317c, 0x317d, 0x317e, 0x3221, 0x3222,
991 0x3223, 0x3224, 0x3225, 0x3226, 0x3227, 0x3228, 0x3229, 0x322a,
992 0x322b, 0x322c, 0x322d, 0x322e, 0x322f, 0x3230, 0x3231, 0x3232,
993 0x3233, 0x3234, 0x3235, 0x3236, 0x3237, 0x3238, 0x3239, 0x323a,
994 0x323b, 0x323c, 0x323d, 0x323e, 0x323f, 0x3240, 0x3241, 0x3242,
995 0x3243, 0x3244, 0x3245, 0x3246, 0x3247, 0x3248, 0x3249,
996 0x324a, 0x324b, 0x324c, 0x324d, 0x324e, 0x324f, 0x3250, 0x3251,
997 0x3252, 0x3253, 0x3254, 0x3255, 0x3256, 0x3257, 0x3258, 0x3259, 0x325a,
998 0x325b, 0x325c, 0x325d, 0x325e, 0x325f, 0x3260, 0x3261, 0x3262,
999 0x3263, 0x3264, 0x3265, 0x3266, 0x3267, 0x3268, 0x3269, 0x326a,
1000 0x326b, 0x326c, 0x326d, 0x326e, 0x326f, 0x3270, 0x3271, 0x3272,
1001 0x3273, 0x3274, 0x3275, 0x3276, 0x3277, 0x3278, 0x3279, 0x327a,
1002 0x327b, 0x327c, 0x327d, 0x327e, 0x3321, 0x3322, 0x3323, 0x3324,
1003 0x3325, 0x3326, 0x3327, 0x3328, 0x3329, 0x332a, 0x332b, 0x332c,
1004 0x332d, 0x332e, 0x332f, 0x3330, 0x3331, 0x3332, 0x3333, 0x3334,
1005 0x3335, 0x3336, 0x3337, 0x3338, 0x3339, 0x333a, 0x333b, 0x333c,
1006 0x333d, 0x333e, 0x333f, 0x3340, 0x3341, 0x3342, 0x3343, 0x3344,
1007 0x3345, 0x3346, 0x3347, 0x3348, 0x3349, 0x334a, 0x334b, 0x334c,
1008 0x334d, 0x334e, 0x334f, 0x3350, 0x3351, 0x3352, 0x3353, 0x3354,
1009 0x3355, 0x3356, 0x3357, 0x3358, 0x3359, 0x335a, 0x335b, 0x335c,
1010 0x335d, 0x335e, 0x335f, 0x3360, 0x3361, 0x3362, 0x3363, 0x3364,
1011 0x3365, 0x3366, 0x3367, 0x3368, 0x3369, 0x336a, 0x336b, 0x336c,
1012 0x336d, 0x336e, 0x336f, 0x3370, 0x3371, 0x3372, 0x3373, 0x3374,
1013 0x3375, 0x3376, 0x3377, 0x3378, 0x3379, 0x337a, 0x337b, 0x337c,
1014 0x337d, 0x337e, 0x3421, 0x3422, 0x3423, 0x3424, 0x3425, 0x3426,
1015 0x3427, 0x3428, 0x3429, 0x342a, 0x342b, 0x342c, 0x342d, 0x342e,
1016 0x342f, 0x3430, 0x3431, 0x3432, 0x3433, 0x3434, 0x3435, 0x3436,
1017 0x3437, 0x3438, 0x3439, 0x343a, 0x343b, 0x343c, 0x343d, 0x343e,
1018 0x343f, 0x3440, 0x3441, 0x3442, 0x3443, 0x3444, 0x3445, 0x3446,
1019 0x3447, 0x3448, 0x3449, 0x344a, 0x344b, 0x344c, 0x344d, 0x344e,
1020 0x344f, 0x3450, 0x3451, 0x3452, 0x3453, 0x3454, 0x3455, 0x3456,
1021 0x3457, 0x3458, 0x3459, 0x345a, 0x345b, 0x345c, 0x345d, 0x345e,
1022 0x345f, 0x3460, 0x3461, 0x3462, 0x3463, 0x3464, 0x3465, 0x3466,
1023 0x3467, 0x3468, 0x3469, 0x346a, 0x346b, 0x346c, 0x346d, 0x346e,
1024 0x346f, 0x3470, 0x3471, 0x3472, 0x3473, 0x3474, 0x3475, 0x3476,
1025 0x3477, 0x3478, 0x3479, 0x347a, 0x347b, 0x347c, 0x347d, 0x347e,
1026 0x3521, 0x3522, 0x3523, 0x3524, 0x3525, 0x3526, 0x3527, 0x3528,
1027 0x3529, 0x352a, 0x352b, 0x352c, 0x352d, 0x352e, 0x352f, 0x3530,
1028 0x3531, 0x3532, 0x3533, 0x3534, 0x3535, 0x3536, 0x3537, 0x3538,
1029 0x3539, 0x353a, 0x353b, 0x353c, 0x353d, 0x353e, 0x353f, 0x3540,
1030 0x3541, 0x3542, 0x3543, 0x3544, 0x3545, 0x3546, 0x3547, 0x3548,
1031 0x3549, 0x354a, 0x354b, 0x354c, 0x354d, 0x354e, 0x354f, 0x3550,
1032 0x3551, 0x3552, 0x3553, 0x3554, 0x3555, 0x3556, 0x3557, 0x3558,
1033 0x3559, 0x355a, 0x355b, 0x355c, 0x355d, 0x355e, 0x355f, 0x3560,
1034 0x3561, 0x3562, 0x3563, 0x3564, 0x3565, 0x3566, 0x3567, 0x3568,
1035 0x3569, 0x356a, 0x356b, 0x356c, 0x356d, 0x356e, 0x356f, 0x3570,
1036 0x3571, 0x3572, 0x3573, 0x3574, 0x3575, 0x3576, 0x3577, 0x3578,
1037 0x3579, 0x357a, 0x357b, 0x357c, 0x357d, 0x357e, 0x3621, 0x3622,
1038 0x3623, 0x3624, 0x3625, 0x3626, 0x3627, 0x3628, 0x3629, 0x362a,
1039 0x362b, 0x362c, 0x362d, 0x362e, 0x362f, 0x3630, 0x3631, 0x3632,
1040 0x3633, 0x3634, 0x3635, 0x3636, 0x3637, 0x3638, 0x3639, 0x363a,
1041 0x363b, 0x363c, 0x363d, 0x363e, 0x363f, 0x3640, 0x3641, 0x3642,
1042 0x3643, 0x3644, 0x3645, 0x3646, 0x3647, 0x3648, 0x3649, 0x364a,
1043 0x364b, 0x364c, 0x364d, 0x364e, 0x364f, 0x3650, 0x3651, 0x3652,
1044 0x3653, 0x3654, 0x3655, 0x3656, 0x3657, 0x3658, 0x3659, 0x365a,

1045 0x365b, 0x365c, 0x365d, 0x365e, 0x365f, 0x3660, 0x3661, 0x3662,
1046 0x3663, 0x3664, 0x3665, 0x3666, 0x3667, 0x3668, 0x3669, 0x366a,
1047 0x366b, 0x366c, 0x366d, 0x366e, 0x3670, 0x3671, 0x366e, 0x366f, 0x3672,
1048 0x3673, 0x3674, 0x3675, 0x3676, 0x3677, 0x3678, 0x3679, 0x367a,
1049 0x367b, 0x367d, 0x367e, 0x367f, 0x367c, 0x3721, 0x3722, 0x3723, 0x3724,
1050 0x3725, 0x3726, 0x3727, 0x3728, 0x3729, 0x372a, 0x372b, 0x372c,
1051 0x372d, 0x372e, 0x372f, 0x3730, 0x3731, 0x3732, 0x3733, 0x3734,
1052 0x3735, 0x3736, 0x3737, 0x3738, 0x3739, 0x373a, 0x373b, 0x373c,
1053 0x373d, 0x373e, 0x373f, 0x3740, 0x3741, 0x3742, 0x3743, 0x3744,
1054 0x3745, 0x3746, 0x3747, 0x3748, 0x3749, 0x374a, 0x374b, 0x374c,
1055 0x374d, 0x374e, 0x374f, 0x3750, 0x3751, 0x3752, 0x3753, 0x3754,
1056 0x3755, 0x3756, 0x3757, 0x3758, 0x3759, 0x375a, 0x375b,
1057 0x375c, 0x375d, 0x375e, 0x375f, 0x3761, 0x3762, 0x3763, 0x3764,
1058 0x3765, 0x3766, 0x3767, 0x3768, 0x3769, 0x376a, 0x376b, 0x376c,
1059 0x376d, 0x377e, 0x376e, 0x376f, 0x3770, 0x3771, 0x3772, 0x3773,
1060 0x3774, 0x3775, 0x3776, 0x3777, 0x3778, 0x3779, 0x377a, 0x377b,
1061 0x377c, 0x377d, 0x3821, 0x3822, 0x3823, 0x3824, 0x3825, 0x3826,
1062 0x3827, 0x3828, 0x3829, 0x382a, 0x382b, 0x382c, 0x382d, 0x382e,
1063 0x382f, 0x3830, 0x3831, 0x3832, 0x3833, 0x3834, 0x3835, 0x3836,
1064 0x3837, 0x3838, 0x3839, 0x383a, 0x383b, 0x383c, 0x383d, 0x383e,
1065 0x383f, 0x3840, 0x3841, 0x3842, 0x3843, 0x3844, 0x3845, 0x3846,
1066 0x3847, 0x3848, 0x3849, 0x384a, 0x384b, 0x384c, 0x384d, 0x384e,
1067 0x3850, 0x3851, 0x384f, 0x3852, 0x3853, 0x3854, 0x3855, 0x3856,
1068 0x3857, 0x3858, 0x3859, 0x385a, 0x385b, 0x385c, 0x385d, 0x385e,
1069 0x385f, 0x3860, 0x3861, 0x3862, 0x3863, 0x3864, 0x3865, 0x3867,
1070 0x3868, 0x3869, 0x386a, 0x386b, 0x386c, 0x386d, 0x386e, 0x386f,
1071 0x3870, 0x3871, 0x3872, 0x3873, 0x3874, 0x3875, 0x3876, 0x3877,
1072 0x3878, 0x3879, 0x387a, 0x387b, 0x387c, 0x387d, 0x387e, 0x3921,
1073 0x3922, 0x3923, 0x3924, 0x3925, 0x3926, 0x3927, 0x3928, 0x3929,
1074 0x392a, 0x392b, 0x392c, 0x392d, 0x392e, 0x392f, 0x3930, 0x3931,
1075 0x3932, 0x3933, 0x3934, 0x3935, 0x3936, 0x3937, 0x3938, 0x3939,
1076 0x393a, 0x393b, 0x393c, 0x393d, 0x393e, 0x393f, 0x3940, 0x3941,
1077 0x3942, 0x3943, 0x3944, 0x3945, 0x3946, 0x3947, 0x3948, 0x3949,
1078 0x394a, 0x394b, 0x394c, 0x394d, 0x394e, 0x394f, 0x3950, 0x3951,
1079 0x3952, 0x3953, 0x3954, 0x3955, 0x3956, 0x3957, 0x3958, 0x3959,
1080 0x395a, 0x395b, 0x395c, 0x395d, 0x395e, 0x395f, 0x3960, 0x3961,
1081 0x3962, 0x3963, 0x3964, 0x3965, 0x3966, 0x3967, 0x3968, 0x3969,
1082 0x396a, 0x396b, 0x396c, 0x396d, 0x396e, 0x396f, 0x3970, 0x3971,
1083 0x3972, 0x3973, 0x3974, 0x3975, 0x3976, 0x3977, 0x3978, 0x3979,
1084 0x397a, 0x397b, 0x397c, 0x397d, 0x397e, 0x3a21, 0x3a22, 0x3a23,
1085 0x3a24, 0x3a25, 0x3a26, 0x3a27, 0x3a28, 0x3a29, 0x3a2a, 0x3a2b,
1086 0x3a2c, 0x3a2d, 0x3a2e, 0x3a2f, 0x3a30, 0x3a31, 0x3a32, 0x3a33,
1087 0x3a34, 0x3a35, 0x3a36, 0x3a37, 0x3a38, 0x3a39, 0x3a3a, 0x3a3b,
1088 0x3a3c, 0x3a3d, 0x3a3e, 0x3a3f, 0x3a40, 0x3a41, 0x3a42, 0x3a43,
1089 0x3a44, 0x3a45, 0x3a46, 0x3a47, 0x3a48, 0x3a49, 0x3a4a, 0x3a4b,
1090 0x3a4c, 0x3a4d, 0x3a4e, 0x3a4f, 0x3a50, 0x3a51, 0x3a52, 0x3a53,
1091 0x3a54, 0x3a55, 0x3a56, 0x3a57, 0x3a58, 0x3a59, 0x3a5a, 0x3a5b,
1092 0x3a5c, 0x3a5d, 0x3a5e, 0x3a5f, 0x3a60, 0x3a61, 0x3a62, 0x3a63,
1093 0x3a64, 0x3a65, 0x3a66, 0x3a67, 0x3a68, 0x3a69, 0x3a6a, 0x3a6b,
1094 0x3a6c, 0x3a6d, 0x3a6e, 0x3a6f, 0x3a70, 0x3a71, 0x3a72, 0x3a73,
1095 0x3a74, 0x3a75, 0x3a76, 0x3a77, 0x3a78, 0x3a79, 0x3a7a, 0x3a7b,
1096 0x3a7c, 0x3a7d, 0x3a7e, 0x3b21, 0x3b22, 0x3b23, 0x3b24, 0x3b25,
1097 0x3b26, 0x3b27, 0x3b28, 0x3b29, 0x3b2a, 0x3b2b, 0x3b2c, 0x3b2d,
1098 0x3b2e, 0x3b2f, 0x3b30, 0x3b31, 0x3b32, 0x3b33, 0x3b34, 0x3b35,
1099 0x3b36, 0x3b37, 0x3b38, 0x3b39, 0x3b3a, 0x3b3b, 0x3b3c, 0x3b3d,
1100 0x3b3e, 0x3b3f, 0x3b40, 0x3b41, 0x3b42, 0x3b43, 0x3b44, 0x3b45,
1101 0x3b47, 0x3b48, 0x3b49, 0x3b4a, 0x3b4b, 0x3b4c, 0x3b4d,
1102 0x3b4e, 0x3b4f, 0x3b50, 0x3b51, 0x3b52, 0x3b53, 0x3b54, 0x3b55,
1103 0x3b56, 0x3b57, 0x3b58, 0x3b59, 0x3b5a, 0x3b5b, 0x3b5c, 0x3b5d,
1104 0x3b5e, 0x3b5f, 0x3b60, 0x3b61, 0x3b62, 0x3b63, 0x3b64, 0x3b65,
1105 0x3b66, 0x3b67, 0x3b68, 0x3b69, 0x3b6a, 0x3b6b, 0x3b6c, 0x3b6d,
1106 0x3b6e, 0x3b6f, 0x3b70, 0x3b71, 0x3b72, 0x6674, 0x3b73, 0x3b74,
1107 0x3b75, 0x3b76, 0x3b77, 0x3b78, 0x3b79, 0x3b7a, 0x3b7b, 0x3b7c,
1108 0x3b7d, 0x3b7e, 0x3c21, 0x3c22, 0x3c23, 0x3c24, 0x3c25, 0x3c26,
1109 0x3c27, 0x3c28, 0x3c29, 0x3c2a, 0x3c2b, 0x3c2c, 0x3c2e, 0x3c2d, 0x3c2f,
1110 0x3c30, 0x3c31, 0x3c32, 0x3c33, 0x3c34, 0x3c35, 0x3c36, 0x3c37,
1111 0x3c38, 0x3c39, 0x3c3a, 0x3c3b, 0x3c3c, 0x3c3d, 0x3c3e,
1112 0x3c3f, 0x3c40, 0x3c41, 0x3c42, 0x3c43, 0x3c44, 0x3c45, 0x3c46,
1113 0x3c47, 0x3c48, 0x3c49, 0x3c4a, 0x3c4b, 0x3c4c, 0x3c4d, 0x3c4e,
1114 0x3c4f, 0x3c50, 0x3c51, 0x3c52, 0x3c53, 0x3c54, 0x3c55, 0x3c56,
1115 0x3c57, 0x3c58, 0x3c59, 0x3c5a, 0x3c5b, 0x3c5c, 0x3c5d, 0x3c5e,
1116 0x3c5f, 0x3c60, 0x3c61, 0x3c62, 0x3c63, 0x3c64, 0x3c65, 0x3c66,
1117 0x3c67, 0x3c68, 0x3c69, 0x3c6a, 0x3c6b, 0x3c6c, 0x3c6d, 0x3c6e,
1118 0x3c6f, 0x3c70, 0x3c71, 0x3c72, 0x3c73, 0x3c74, 0x3c75, 0x3c76,
1119 0x3c77, 0x3c78, 0x3c79, 0x3c7a, 0x3c7b, 0x3c7c, 0x3c7d, 0x3c7e,
1120 0x3d21, 0x3d22, 0x3d23, 0x3d24, 0x3d25, 0x3d26, 0x3d27, 0x3d28,
1121 0x3d29, 0x3d2a, 0x3d2b, 0x3d2c, 0x3d2d, 0x3d2e, 0x3d2f, 0x3d30,
1122 0x3d31, 0x3d32, 0x3d33, 0x3d34, 0x3d35, 0x3d36, 0x3d37, 0x3d38,
1123 0x3d39, 0x3d3a, 0x3d3b, 0x3d3c, 0x3d3d, 0x3d3e, 0x3d3f, 0x3d40,
1124 0x3d41, 0x3d42, 0x3d43, 0x3d44, 0x3d45, 0x3d46, 0x3d47, 0x3d48,
1125 0x3d49, 0x3d4a, 0x3d4b, 0x3d4c, 0x3d4d, 0x3d4e, 0x3d4f, 0x3d50,
1126 0x3d51, 0x3d52, 0x3d53, 0x3d54, 0x3d55, 0x3d56, 0x3d57, 0x3d58,
1127 0x3d59, 0x3d5a, 0x3d5b, 0x3d5c, 0x3d5d, 0x3d5e, 0x3d5f, 0x3d60,
1128 0x3d61, 0x3d62, 0x3d63, 0x3d64, 0x3d65, 0x3d66, 0x3d67, 0x3d68,
1129 0x3d69, 0x3d6a, 0x3d6b, 0x3d6c, 0x3d6d, 0x3d6e, 0x3d6f, 0x3d70,
1130 0x3d71, 0x3d72, 0x3d73, 0x3d74, 0x3d75, 0x3d76, 0x3d77, 0x3d78,
1131 0x3d79, 0x3d7a, 0x3d7b, 0x3d7c, 0x3d7d, 0x3d7e, 0x3e21, 0x3e22,

1132 0x3e23, 0x3e24, 0x3e25, 0x3e26, 0x3e27, 0x3e28, 0x3e29, 0x3e2a,
1133 0x3e2b, 0x3e2c, 0x3e2d, 0x3e2e, 0x3e2f, 0x3e30, 0x3e31, 0x3e32,
1134 0x3e33, 0x3e34, 0x3e35, 0x3e36, 0x3e37, 0x3e38, 0x3e39, 0x3e3a,
1135 0x3e3b, 0x3e3c, 0x3e3d, 0x3e3e, 0x3e3f, 0x3e40, 0x3e41, 0x3e42,
1136 0x3e43, 0x3e44, 0x3e45, 0x3e46, 0x3e47, 0x3e48, 0x3e49, 0x3e4a,
1137 0x3e4b, 0x3e4c, 0x3e4d, 0x3e4e, 0x3e4f, 0x3e50, 0x3e51, 0x3e52,
1138 0x3e53, 0x3e54, 0x3e55, 0x3e56, 0x3e57, 0x3e58, 0x3e59, 0x3e5a,
1139 0x3e5b, 0x3e5c, 0x3e5d, 0x3e5e, 0x3e5f, 0x3e60, 0x3e61, 0x3e62,
1140 0x3e63, 0x3e64, 0x3e65, 0x3e66, 0x3e67, 0x3e68, 0x3e69, 0x3e6a,
1141 0x3e6b, 0x3e6c, 0x3e6d, 0x3e6e, 0x3e6f, 0x3e70, 0x3e71, 0x3e72,
1142 0x3e73, 0x3e74, 0x3e75, 0x3e76, 0x3e77, 0x3e78, 0x3e79, 0x3e7a,
1143 0x3e7b, 0x3e7c, 0x3e7d, 0x3e7e, 0x3e7f, 0x3f21, 0x3f22, 0x3f23, 0x3f24,
1144 0x3f25, 0x3f26, 0x3f27, 0x3f28, 0x3f29, 0x3f2a, 0x3f2b, 0x3f2c,
1145 0x3f2d, 0x3f2e, 0x3f2f, 0x3f30, 0x3f31, 0x3f32, 0x3f33, 0x3f34,
1146 0x3f35, 0x3f36, 0x3f37, 0x3f38, 0x3f39, 0x3f3a, 0x3f3b, 0x3f3c,
1147 0x3f3d, 0x3f3e, 0x3f3f, 0x3f40, 0x3f41, 0x3f42, 0x3f43, 0x3f44,
1148 0x3f45, 0x3f46, 0x3f47, 0x3f48, 0x3f49, 0x3f4a, 0x3f4b, 0x3f4c,
1149 0x3f4d, 0x3f4e, 0x3f4f, 0x3f50, 0x3f51, 0x3f52, 0x3f53, 0x3f54,
1150 0x3f55, 0x3f56, 0x3f57, 0x3f58, 0x3f59, 0x3f5a, 0x3f5b, 0x3f5c,
1151 0x3f5d, 0x3f5e, 0x3f5f, 0x3f60, 0x3f61, 0x3f62, 0x3f63, 0x3f64,
1152 0x3f65, 0x3f66, 0x3f67, 0x3f68, 0x3f69, 0x3f6a, 0x3f6b, 0x3f6c,
1153 0x3f6d, 0x3f6e, 0x3f6f, 0x3f70, 0x3f71, 0x3f72, 0x3f73, 0x3f74,
1154 0x3f75, 0x3f76, 0x3f77, 0x3f78, 0x3f79, 0x3f7a, 0x3f7b, 0x3f7c,
1155 0x3f7d, 0x3f7e, 0x4021, 0x4022, 0x4023, 0x4024, 0x4025, 0x4026,
1156 0x4027, 0x4028, 0x4029, 0x402a, 0x402b, 0x402c, 0x402d, 0x402e,
1157 0x402f, 0x4030, 0x4031, 0x4032, 0x4033, 0x4034, 0x4035, 0x4036,
1158 0x4037, 0x4038, 0x4039, 0x403a, 0x403b, 0x403c, 0x403d, 0x403e,
1159 0x403f, 0x4040, 0x4041, 0x4042, 0x4043, 0x4044, 0x4045, 0x4046,
1160 0x4047, 0x4048, 0x4049, 0x404a, 0x404b, 0x404c, 0x404d, 0x404e,
1161 0x404f, 0x4050, 0x4051, 0x4052, 0x4053, 0x4054, 0x4055, 0x4056,
1162 0x4057, 0x4058, 0x4059, 0x405a, 0x405b, 0x405c, 0x405d, 0x405e,
1163 0x405f, 0x4060, 0x4061, 0x4062, 0x4063, 0x4064, 0x4065, 0x4066,
1164 0x4067, 0x4068, 0x4069, 0x406a, 0x406b, 0x406c, 0x406d, 0x406e,
1165 0x406f, 0x4070, 0x4071, 0x4072, 0x4073, 0x4074, 0x4075, 0x4076,
1166 0x4077, 0x4078, 0x4079, 0x407a, 0x407b, 0x407c, 0x407d, 0x407e,
1167 0x4121, 0x4122, 0x4123, 0x4124, 0x4125, 0x4126, 0x4127, 0x4128,
1168 0x4129, 0x412a, 0x412b, 0x412c, 0x412d, 0x412e, 0x412f, 0x4130,
1169 0x4131, 0x4132, 0x4133, 0x4134, 0x4135, 0x4136, 0x4137, 0x4138,
1170 0x4139, 0x413a, 0x413b, 0x413c, 0x413d, 0x413e, 0x413f, 0x4140,
1171 0x4141, 0x4142, 0x4143, 0x4144, 0x4145, 0x4146, 0x4147, 0x4148,
1172 0x4149, 0x414a, 0x414b, 0x414c, 0x414d, 0x414e, 0x414f, 0x4150,
1173 0x4151, 0x4152, 0x4153, 0x4154, 0x4155, 0x4156, 0x4157, 0x4158,
1174 0x4159, 0x415a, 0x415b, 0x415c, 0x415d, 0x415e, 0x415f, 0x4160,
1175 0x4161, 0x4162, 0x4163, 0x4164, 0x4165, 0x4166, 0x4167, 0x4168,
1176 0x4169, 0x416a, 0x416b, 0x416c, 0x416d, 0x416e, 0x416f, 0x4170,
1177 0x4171, 0x4172, 0x4173, 0x4174, 0x4175, 0x4176, 0x4177, 0x4178,
1178 0x4179, 0x417a, 0x417b, 0x417c, 0x417d, 0x417e, 0x4221, 0x4222,
1179 0x4223, 0x4224, 0x4225, 0x4226, 0x4227, 0x4228, 0x4229, 0x422a,
1180 0x422b, 0x422c, 0x422d, 0x422e, 0x4230, 0x422f, 0x4231, 0x4232,
1181 0x4233, 0x4234, 0x4235, 0x4237, 0x4236, 0x4238, 0x4239, 0x423a,
1182 0x423b, 0x423c, 0x423d, 0x423e, 0x4240, 0x4241, 0x4242, 0x4244,
1183 0x4245, 0x4247, 0x4248, 0x4249, 0x424a, 0x424c, 0x4243, 0x4246,
1184 0x424b, 0x424d, 0x424e, 0x424f, 0x4250, 0x4251, 0x4252, 0x4253,
1185 0x4254, 0x4255, 0x4256, 0x4257, 0x4258, 0x4259, 0x425a, 0x425b,
1186 0x425c, 0x425d, 0x425e, 0x425f, 0x4260, 0x4261, 0x4262, 0x4263,
1187 0x4264, 0x4265, 0x4266, 0x4267, 0x4268, 0x4269, 0x426a, 0x426b,
1188 0x426c, 0x426d, 0x423f, 0x426e, 0x426f, 0x4270, 0x4271, 0x4272,
1189 0x4273, 0x4274, 0x4275, 0x4276, 0x4277, 0x4278, 0x4279, 0x427a,
1190 0x427b, 0x427c, 0x427d, 0x427e, 0x4321, 0x4322, 0x4323, 0x4324,
1191 0x4325, 0x4326, 0x4327, 0x4328, 0x4329, 0x432a, 0x432b, 0x432c,
1192 0x432d, 0x432e, 0x432f, 0x4330, 0x4331, 0x4332, 0x4333, 0x4334,
1193 0x4335, 0x4336, 0x4337, 0x4339, 0x433a, 0x433b, 0x433c, 0x433d,
1194 0x433e, 0x433f, 0x4340, 0x4341, 0x4342, 0x4343, 0x4344, 0x4345,
1195 0x4346, 0x4347, 0x4348, 0x4338, 0x434a, 0x434b, 0x434c, 0x434d,
1196 0x434f, 0x434e, 0x4350, 0x4351, 0x4352, 0x4353, 0x4354, 0x4355,
1197 0x4356, 0x4357, 0x4358, 0x4359, 0x435a, 0x435b, 0x4349, 0x435c,
1198 0x435d, 0x435e, 0x435f, 0x4360, 0x4361, 0x4362, 0x4363, 0x4364,
1199 0x4365, 0x4366, 0x4367, 0x4368, 0x4369, 0x436a, 0x436b, 0x436c,
1200 0x436d, 0x436e, 0x436f, 0x4370, 0x4371, 0x4372, 0x4373, 0x4374,
1201 0x4375, 0x4376, 0x4377, 0x4378, 0x4379, 0x437a, 0x437b, 0x437c,
1202 0x437d, 0x437e, 0x4421, 0x4422, 0x4423, 0x4424, 0x4425, 0x4426,
1203 0x4427, 0x4428, 0x4429, 0x442a, 0x442b, 0x442c, 0x442d, 0x442e,
1204 0x442f, 0x4430, 0x4431, 0x4432, 0x4433, 0x4434, 0x4435, 0x4436,
1205 0x4437, 0x4438, 0x4439, 0x443a, 0x443b, 0x443c, 0x443d, 0x443e,
1206 0x443f, 0x4440, 0x4441, 0x4442, 0x4443, 0x4444, 0x4445, 0x4446,
1207 0x4447, 0x4448, 0x4449, 0x444a, 0x444b, 0x444c, 0x444d, 0x444e,
1208 0x444f, 0x4450, 0x4451, 0x4452, 0x4453, 0x4454, 0x4455, 0x4456,
1209 0x4457, 0x4458, 0x4459, 0x445a, 0x445b, 0x445c, 0x445d, 0x445e,
1210 0x445f, 0x4460, 0x4461, 0x4462, 0x4463, 0x4464, 0x4465, 0x4466,
1211 0x4467, 0x4468, 0x4469, 0x446a, 0x446b, 0x446c, 0x446d, 0x446e,
1212 0x446f, 0x4470, 0x4471, 0x4472, 0x4473, 0x4474, 0x4475, 0x4476,
1213 0x4477, 0x4478, 0x4479, 0x447a, 0x447b, 0x447c, 0x447d, 0x447e,
1214 0x4521, 0x4522, 0x4523, 0x4524, 0x4525, 0x4526, 0x4527, 0x4528,
1215 0x4529, 0x452a, 0x452b, 0x452c, 0x452d, 0x452e, 0x452f, 0x4530,
1216 0x4531, 0x4532, 0x4533, 0x4534, 0x4535, 0x4536, 0x4537, 0x4538,
1217 0x4539, 0x453a, 0x453b, 0x453c, 0x453d, 0x453e, 0x453f, 0x4540,
1218 0x4541, 0x4542, 0x4543, 0x4544, 0x4545, 0x4546, 0x4547, 0x4548,


```

1306 0x4c6f, 0x4c70, 0x4c71, 0x4c72, 0x4c73, 0x4c74, 0x4c75, 0x4c76,
1307 0x4c77, 0x4c78, 0x4c79, 0x4c7a, 0x4c7b, 0x4c7c, 0x4c7d, 0x4c7e,
1308 0x4d21, 0x4d22, 0x4d23, 0x4d24, 0x4d25, 0x4d26, 0x4d27, 0x4d28,
1309 0x4d29, 0x4d2a, 0x4d2b, 0x4d2c, 0x4d2d, 0x4d2e, 0x4d2f, 0x4d30,
1310 0x4d31, 0x4d32, 0x4d33, 0x4d34, 0x4d35, 0x4d36, 0x4d37, 0x4d38,
1311 0x4d39, 0x4d3a, 0x4d3b, 0x4d3c, 0x4d3d, 0x4d3e, 0x4d3f, 0x4d40,
1312 0x4d41, 0x4d42, 0x4d43, 0x4d44, 0x4d45, 0x4d46, 0x4d47, 0x4d48,
1313 0x4d49, 0x4d4a, 0x4d4b, 0x4d4c, 0x4d4d, 0x4d4e, 0x4d4f, 0x4d50,
1314 0x4d51, 0x4d52, 0x4d53, 0x4d54, 0x4d55, 0x4d56, 0x4d57, 0x4d58,
1315 0x4d59, 0x4d5a, 0x4d5b, 0x4d5c, 0x4d5d, 0x4d5e, 0x4d5f, 0x4d60,
1316 0x4d61, 0x4d62, 0x4d63, 0x4d64, 0x4d65, 0x4d66, 0x4d67, 0x4d68,
1317 0x4d69, 0x4d6a, 0x4d6b, 0x4d6c, 0x4d6d, 0x4d6e, 0x4d6f, 0x4d70,
1318 0x4d71, 0x4d72, 0x4d73, 0x4d74, 0x4d75, 0x4d76, 0x4d77, 0x4d78,
1319 0x4d79, 0x4d7a, 0x4d7b, 0x4d7c, 0x4d7d, 0x4d7e, 0x4e21, 0x4e22,
1320 0x4e24, 0x4e25, 0x4e26, 0x4e27, 0x4e28, 0x4e29, 0x4e23, 0x4e2a,
1321 0x4e2b, 0x4e2c, 0x4e2d, 0x4e2e, 0x4e2f, 0x4e30, 0x4e31, 0x4e32,
1322 0x4e33, 0x4e34, 0x4e35, 0x4e36, 0x4e37, 0x4e38, 0x4e39, 0x4e3a,
1323 0x4e3b, 0x4e3c, 0x4e3d, 0x4e3e, 0x4e3f, 0x4e40, 0x4e41, 0x4e42,
1324 0x4e43, 0x4e44, 0x4e45, 0x4e46, 0x4e47, 0x4e48, 0x4e49, 0x4e4a,
1325 0x4e4b, 0x4e4c, 0x4e4d, 0x4e4e, 0x4e4f, 0x4e50, 0x4e51, 0x4e52,
1326 0x4e53, 0x4e54, 0x4e55, 0x4e56, 0x4e57, 0x4e58, 0x4e59, 0x4e5a,
1327 0x4e5b, 0x4e5c, 0x4e5d, 0x4e5e, 0x4e5f, 0x4e60, 0x4e61, 0x4e62,
1328 0x4e63, 0x4e64, 0x4e65, 0x4e66, 0x4e67, 0x4e68, 0x4e69, 0x4e6a,
1329 0x4e6b, 0x4e6c, 0x4e6d, 0x4e6e, 0x4e6f, 0x4e70, 0x4e71, 0x4e72,
1330 0x4e73, 0x4e74, 0x4e75, 0x4e76, 0x4e77, 0x4e78, 0x4e79, 0x4e7a,
1331 0x4e7b, 0x4e7c, 0x4e7d, 0x4e7e, 0x4f21, 0x4f22, 0x4f23, 0x4f24,
1332 0x4f25, 0x4f26, 0x4f27, 0x4f28, 0x4f29, 0x4f2a, 0x4f2b, 0x4f2c,
1333 0x4f2d, 0x4f2e, 0x4f2f, 0x4f30, 0x4f31, 0x4f32, 0x4f33, 0x4f34,
1334 0x4f35, 0x4f36, 0x4f37, 0x4f38, 0x4f39, 0x4f3a, 0x4f3b, 0x4f3c,
1335 0x4f3d, 0x4f3e, 0x4f3f, 0x4f40, 0x4f41, 0x4f42, 0x4f43, 0x4f44,
1336 0x4f45, 0x4f46, 0x4f47, 0x4f48, 0x4f49, 0x4f4a, 0x4f4b, 0x4f4c,
1337 0x4f4d, 0x4f4e, 0x4f4f, 0x4f50, 0x4f51, 0x4f52, 0x4f53, 0x4f54,
1338 0x4f55, 0x4f56, 0x4f57, 0x4f58, 0x4f59, 0x4f5a, 0x4f5b, 0x4f5c,
1339 0x4f5d, 0x4f5e, 0x4f5f, 0x4f60, 0x4f61, 0x4f62, 0x4f63, 0x4f64,
1340 0x4f65, 0x4f66, 0x4f67, 0x4f68, 0x4f69, 0x4f6a, 0x4f6b, 0x4f6c,
1341 0x4f6d, 0x4f6e, 0x4f6f, 0x4f70, 0x4f71, 0x4f72, 0x4f74, 0x4f75,
1342 0x4f76, 0x4f77, 0x4f78, 0x4f79, 0x4f7a, 0x4f7b, 0x4f7c,
1343 0x4f7d, 0x4f7e, 0x5021, 0x5022, 0x5023, 0x5024, 0x5025, 0x5026,
1344 0x5027, 0x5028, 0x5029, 0x502a, 0x502b, 0x502c, 0x502e, 0x502f,
1345 0x5030, 0x5031, 0x5032, 0x5033, 0x5034, 0x5035, 0x5037,
1346 0x5038, 0x5039, 0x503a, 0x503b, 0x503c, 0x503d, 0x503e, 0x503f,
1347 0x5040, 0x5041, 0x5042, 0x5043, 0x5044, 0x5045, 0x5046,
1348 0x5047, 0x5048, 0x5049, 0x504a, 0x504b, 0x504c, 0x504d, 0x504e,
1349 0x504f, 0x5050, 0x5051, 0x5052, 0x5053, 0x5054, 0x5055, 0x5056,
1350 0x5057, 0x5058, 0x5059, 0x505a, 0x505b, 0x505c, 0x505d, 0x505e,
1351 0x505f, 0x5060, 0x5061, 0x5062, 0x5063, 0x5064, 0x5065, 0x5066,
1352 0x5067, 0x5068, 0x5069, 0x506a, 0x506b, 0x506c, 0x506d, 0x506e,
1353 0x506f, 0x5070, 0x5071, 0x5072, 0x5073, 0x5074, 0x5075, 0x5076,
1354 0x5077, 0x5078, 0x5079, 0x507a, 0x507b, 0x507c, 0x507d, 0x507e,
1355 0x5121, 0x5122, 0x5123, 0x5124, 0x5125, 0x5126, 0x5127, 0x5128,
1356 0x5129, 0x512a, 0x512b, 0x512c, 0x512d, 0x512e, 0x512f, 0x5130,
1357 0x5131, 0x5132, 0x5133, 0x5134, 0x5135, 0x5136, 0x5137, 0x5138,
1358 0x5139, 0x513a, 0x513b, 0x513c, 0x513d, 0x513e, 0x513f, 0x5140,
1359 0x5141, 0x5142, 0x5143, 0x5144, 0x5145, 0x5146, 0x5147, 0x5148,
1360 0x5149, 0x514a, 0x514b, 0x514c, 0x514d, 0x514e, 0x514f, 0x5150,
1361 0x5151, 0x5152, 0x5153, 0x5154, 0x5155, 0x5156, 0x5157, 0x5158,
1362 0x5159, 0x515a, 0x515b, 0x515c, 0x515d, 0x515e, 0x515f, 0x5160,
1363 0x5161, 0x5162, 0x5163, 0x5164, 0x5165, 0x5166, 0x5167, 0x5168,
1364 0x5169, 0x516a, 0x516b, 0x516c, 0x516d, 0x516e, 0x516f, 0x5170,
1365 0x5171, 0x5172, 0x5173, 0x5174, 0x5175, 0x5176, 0x5177, 0x5178,
1366 0x5179, 0x517a, 0x517b, 0x517c, 0x517d, 0x517e, 0x5221, 0x5222,
1367 0x5223, 0x5224, 0x5225, 0x5226, 0x5227, 0x5228, 0x5229, 0x522a,
1368 0x522b, 0x522c, 0x522d, 0x522e, 0x522f, 0x5230, 0x5231, 0x5232,
1369 0x5233, 0x5234, 0x5235, 0x5236, 0x5237, 0x5238, 0x5239, 0x523a,
1370 0x523b, 0x523c, 0x523d, 0x523e, 0x523f, 0x5240, 0x5241, 0x5242,
1371 0x5243, 0x5244, 0x5245, 0x5246, 0x5247, 0x5248, 0x5249, 0x524a,
1372 0x524b, 0x524c, 0x524d, 0x524e, 0x524f, 0x5250, 0x5251, 0x5252,
1373 0x5253, 0x5254, 0x5255, 0x5256, 0x5257, 0x5258, 0x5259, 0x525a,
1374 0x525b, 0x525c, 0x525d, 0x525e, 0x525f, 0x5260, 0x5261, 0x5262,
1375 0x5263, 0x5264, 0x5265, 0x5266, 0x5267, 0x5268, 0x5269, 0x526a,
1376 0x526b, 0x526c, 0x526d, 0x526e, 0x526f, 0x5270, 0x5271, 0x5272,
1377 0x5273, 0x5274, 0x5275, 0x5276, 0x5277, 0x5278, 0x5279, 0x527a,
1378 0x527b, 0x527c, 0x527d, 0x527e, 0x5321, 0x5322, 0x5323, 0x5324,
1379 0x5325, 0x5326, 0x5327, 0x5328, 0x5329, 0x532a, 0x532b, 0x532c,
1380 0x532d, 0x532e, 0x532f, 0x5330, 0x5331, 0x5332, 0x5333, 0x5334,
1381 0x5335, 0x5336, 0x5337, 0x5338, 0x5339, 0x533a, 0x533b, 0x533c,
1382 0x533d, 0x533e, 0x533f, 0x5340, 0x5341, 0x5342, 0x5343, 0x5344,
1383 0x5345, 0x5346, 0x5347, 0x5348, 0x5349, 0x534a, 0x534b, 0x534c,
1384 0x534d, 0x534e, 0x534f, 0x5350, 0x5351, 0x5352, 0x5353, 0x5354,
1385 0x5355, 0x5356, 0x5357, 0x5358, 0x5359, 0x535a, 0x535b, 0x535c,
1386 0x535d, 0x535e, 0x535f, 0x5360, 0x5361, 0x5362, 0x5363, 0x5364,
1387 0x5365, 0x5366, 0x5367, 0x5368, 0x5369, 0x536a, 0x536b, 0x536c,
1388 0x536d, 0x536e, 0x536f, 0x5370, 0x5371, 0x5372, 0x5373, 0x5374,
1389 0x5375, 0x5376, 0x5377, 0x5378, 0x5379, 0x537a, 0x537b, 0x537c,
1390 0x537d, 0x537e, 0x5421, 0x5422, 0x5423, 0x5424, 0x5425, 0x5426,
1391 0x5427, 0x5428, 0x5429, 0x542a, 0x542b, 0x542c, 0x542d, 0x542e,
1392 0x542f, 0x5430, 0x5431, 0x5432, 0x5433, 0x5434, 0x5435, 0x5436, 0x5437,

```

1393 0x5438, 0x5439, 0x543a, 0x543b, 0x543c, 0x543d, 0x543e, 0x5433,
1394 0x543f, 0x5440, 0x5441, 0x5442, 0x5443, 0x5444, 0x5445, 0x5446,
1395 0x5447, 0x5448, 0x5449, 0x544a, 0x544b, 0x544c, 0x544d, 0x544e,
1396 0x544f, 0x5450, 0x5451, 0x5452, 0x5453, 0x5454, 0x5455, 0x5456,
1397 0x5457, 0x5458, 0x5459, 0x545a, 0x545b, 0x545c, 0x545d, 0x545e,
1398 0x545f, 0x5460, 0x5461, 0x5462, 0x5463, 0x5464, 0x5465, 0x5466,
1399 0x5467, 0x5468, 0x5469, 0x546a, 0x546c, 0x546b, 0x546d, 0x546e,
1400 0x546f, 0x5470, 0x5471, 0x5472, 0x5473, 0x5474, 0x5475, 0x5476,
1401 0x5477, 0x5478, 0x5479, 0x547a, 0x547b, 0x547c, 0x547d, 0x547e,
1402 0x5521, 0x5522, 0x5523, 0x5524, 0x5525, 0x5526, 0x5527, 0x5528,
1403 0x5529, 0x552a, 0x552b, 0x552c, 0x552d, 0x552e, 0x552f, 0x5530,
1404 0x5531, 0x5532, 0x5533, 0x5534, 0x5535, 0x5536, 0x5537, 0x5538,
1405 0x5539, 0x553a, 0x553b, 0x553c, 0x553d, 0x553e, 0x553f, 0x5540,
1406 0x5541, 0x5542, 0x5543, 0x5544, 0x5545, 0x5546, 0x5547, 0x5548,
1407 0x5549, 0x554a, 0x554b, 0x554c, 0x554d, 0x554e, 0x554f, 0x5550,
1408 0x5551, 0x5552, 0x5553, 0x5554, 0x5555, 0x5556, 0x5557, 0x5558,
1409 0x5559, 0x555a, 0x555b, 0x555c, 0x555d, 0x555e, 0x555f, 0x5560,
1410 0x5561, 0x5562, 0x5563, 0x5564, 0x5565, 0x5566, 0x5567, 0x5568,
1411 0x5569, 0x556a, 0x556b, 0x556c, 0x556d, 0x556e, 0x556f, 0x5570,
1412 0x5571, 0x5572, 0x5573, 0x5574, 0x5575, 0x5576, 0x5577, 0x5578,
1413 0x5579, 0x557a, 0x557b, 0x557c, 0x557d, 0x557e, 0x5621, 0x5622,
1414 0x5623, 0x5624, 0x5625, 0x5626, 0x5627, 0x5628, 0x5629, 0x562a,
1415 0x562b, 0x562c, 0x562d, 0x562e, 0x562f, 0x5630, 0x5631, 0x5632,
1416 0x5633, 0x5634, 0x5635, 0x5636, 0x5637, 0x5638, 0x5639, 0x563a,
1417 0x563b, 0x563c, 0x563d, 0x563e, 0x563f, 0x5640, 0x5641, 0x5642,
1418 0x5643, 0x5644, 0x5645, 0x5647, 0x5648, 0x5649, 0x564a, 0x564b,
1419 0x5646, 0x564c, 0x564d, 0x564e, 0x564f, 0x5650, 0x5651, 0x5652,
1420 0x5653, 0x5654, 0x5656, 0x5657, 0x5658, 0x5655, 0x5659, 0x565a,
1421 0x565b, 0x565c, 0x565d, 0x565e, 0x565f, 0x5660, 0x5661, 0x5662,
1422 0x5663, 0x5664, 0x5665, 0x5666, 0x5667, 0x5668, 0x5669, 0x566a,
1423 0x566b, 0x566c, 0x566d, 0x566e, 0x566f, 0x5670, 0x5671, 0x5672,
1424 0x5673, 0x5674, 0x5675, 0x5676, 0x5677, 0x5678, 0x5679, 0x567a,
1425 0x567b, 0x567c, 0x567d, 0x567e, 0x5721, 0x5722, 0x5723, 0x5724,
1426 0x5725, 0x5726, 0x5727, 0x5728, 0x5729, 0x572a, 0x572b, 0x572c,
1427 0x572d, 0x572e, 0x572f, 0x5730, 0x5731, 0x5732, 0x5733, 0x5734,
1428 0x5735, 0x5736, 0x5737, 0x5738, 0x5739, 0x573a, 0x573b, 0x573c,
1429 0x573d, 0x573e, 0x573f, 0x5740, 0x5741, 0x5742, 0x5743, 0x5744,
1430 0x5745, 0x5746, 0x5747, 0x5748, 0x5749, 0x574a, 0x574b, 0x574c,
1431 0x574d, 0x574e, 0x574f, 0x5750, 0x5751, 0x5752, 0x5753, 0x5754,
1432 0x5755, 0x5756, 0x5757, 0x5758, 0x5759, 0x575a, 0x575b, 0x575c,
1433 0x575d, 0x575e, 0x575f, 0x5760, 0x5761, 0x5762, 0x5764, 0x5765,
1434 0x5766, 0x5767, 0x5768, 0x5769, 0x576a, 0x576b, 0x576c, 0x576d,
1435 0x576e, 0x576f, 0x5770, 0x5771, 0x5772, 0x5773, 0x5774, 0x5775,
1436 0x5776, 0x5777, 0x5778, 0x5779, 0x583e, 0x5763, 0x577a, 0x577b,
1437 0x577c, 0x577d, 0x577e, 0x5821, 0x5822, 0x5823, 0x5824, 0x5825,
1438 0x5826, 0x5827, 0x5828, 0x5829, 0x582a, 0x582b, 0x582c, 0x582d,
1439 0x582e, 0x582f, 0x5830, 0x5831, 0x5832, 0x5833, 0x5834, 0x5835,
1440 0x5836, 0x5837, 0x5838, 0x5839, 0x583a, 0x583b, 0x583c,
1441 0x583d, 0x583e, 0x5840, 0x5841, 0x5842, 0x5843, 0x5844, 0x5845,
1442 0x5846, 0x5847, 0x5848, 0x5849, 0x584a, 0x584b, 0x584c, 0x584e,
1443 0x584f, 0x5850, 0x5851, 0x5852, 0x5853, 0x5854, 0x5855, 0x5856,
1444 0x5857, 0x5858, 0x5859, 0x585a, 0x585b, 0x585c, 0x585d, 0x585e,
1445 0x585f, 0x5860, 0x5861, 0x5862, 0x5863, 0x5864, 0x5865, 0x5866,
1446 0x5867, 0x5868, 0x5869, 0x586a, 0x586b, 0x586c, 0x586d, 0x586e,
1447 0x586f, 0x5870, 0x5871, 0x5872, 0x5873, 0x5874, 0x5875, 0x5876,
1448 0x5877, 0x5878, 0x5879, 0x587a, 0x587b, 0x587c, 0x587d, 0x587e,
1449 0x5921, 0x5922, 0x5923, 0x5924, 0x5925, 0x5926, 0x5927, 0x5928,
1450 0x592a, 0x592b, 0x592c, 0x592d, 0x592e, 0x592f, 0x5930, 0x5931,
1451 0x5932, 0x5933, 0x5934, 0x5935, 0x5936, 0x5937, 0x5938, 0x5939,
1452 0x593a, 0x593b, 0x593c, 0x593d, 0x593e, 0x593f, 0x5940,
1453 0x5941, 0x5942, 0x5943, 0x5944, 0x5945, 0x5946, 0x5947, 0x5948,
1454 0x5949, 0x594a, 0x594b, 0x594c, 0x594d, 0x594e, 0x594f, 0x5950,
1455 0x5951, 0x5952, 0x5953, 0x5954, 0x5955, 0x5956, 0x5957, 0x5958,
1456 0x5959, 0x595a, 0x595b, 0x595c, 0x595d, 0x595e, 0x595f, 0x5960,
1457 0x5961, 0x5962, 0x5963, 0x5964, 0x5965, 0x5966, 0x5974, 0x5967,
1458 0x5968, 0x5969, 0x596a, 0x596b, 0x596c, 0x596d, 0x596e, 0x596f,
1459 0x5970, 0x5971, 0x5972, 0x5973, 0x5975, 0x5976, 0x5977, 0x5978,
1460 0x5979, 0x597a, 0x597b, 0x597c, 0x597d, 0x597e, 0x5a21, 0x5a22,
1461 0x5a23, 0x5a24, 0x5a25, 0x5a26, 0x5a27, 0x5a28, 0x5a29, 0x5a2a,
1462 0x5a2b, 0x5a2c, 0x5a2d, 0x5a2e, 0x5a2f, 0x5a30, 0x5a31, 0x5a32,
1463 0x5a33, 0x5a34, 0x5a35, 0x5a36, 0x3866, 0x5a37, 0x5a38, 0x5a39,
1464 0x5a3a, 0x5a3b, 0x5a3c, 0x5a3d, 0x5a3e, 0x5a3f, 0x5a40, 0x5a41,
1465 0x5a42, 0x5a43, 0x5a44, 0x5a45, 0x5a46, 0x5a47, 0x5a48, 0x5a49,
1466 0x5a4a, 0x5a4b, 0x5a4d, 0x5a4c, 0x5a4e, 0x5a4e, 0x5a4f, 0x5a50,
1467 0x5a51, 0x5a52, 0x5a53, 0x5a54, 0x5a55, 0x5a56, 0x5a57, 0x5a58,
1468 0x5a59, 0x5a5a, 0x5a5b, 0x5a5c, 0x5a5d, 0x5a5e, 0x5a5f, 0x5a60,
1469 0x5a61, 0x5a62, 0x5a63, 0x5a64, 0x5a65, 0x5a66, 0x5a67, 0x5a68,
1470 0x5a69, 0x5a6a, 0x5a6b, 0x5a6c, 0x5a6e, 0x5a6f, 0x5a70, 0x5a71,
1471 0x5a72, 0x5a73, 0x5a74, 0x5a75, 0x5a76, 0x5a77, 0x5a78, 0x5a79,
1472 0x5a7a, 0x5a7b, 0x5a7c, 0x5a7d, 0x5a7e, 0x5b21, 0x5b22, 0x5b23,
1473 0x5b24, 0x5b25, 0x5b26, 0x5b27, 0x5b28, 0x5b29, 0x5b2a, 0x5b2b,
1474 0x5b2c, 0x5b2d, 0x5b2e, 0x5b2f, 0x5b30, 0x5b31, 0x5b32, 0x5b33,
1475 0x5b34, 0x5b35, 0x5b36, 0x5b37, 0x5b38, 0x5b39, 0x5b3a, 0x5b3b,
1476 0x5b3c, 0x5b3d, 0x5b3e, 0x5b3f, 0x5b40, 0x5b41, 0x5b42, 0x5b43,
1477 0x5b44, 0x5b45, 0x5b46, 0x5b47, 0x5b48, 0x5b49, 0x5b4a, 0x5b4b,
1478 0x5b4c, 0x5b4d, 0x5b4e, 0x5b4f, 0x5b50, 0x5b51, 0x5b52, 0x5b53,
1479 0x5b54, 0x5b55, 0x5b56, 0x5b57, 0x5b58, 0x5b59, 0x5b5a, 0x5b5b,

1480 0x5b5c, 0x5b5d, 0x5b5e, 0x5b5f, 0x5b60, 0x5b61, 0x5b62, 0x5b63,
1481 0x5b64, 0x5b65, 0x5b66, 0x5b67, 0x5b68, 0x5b69, 0x5b6a, 0x5b6b,
1482 0x5b6c, 0x5b6d, 0x5b6e, 0x5b70, 0x5b71, 0x5b72, 0x5b73, 0x5b6f,
1483 0x5b74, 0x5b75, 0x5b76, 0x5b77, 0x5b78, 0x5b79, 0x5b7a, 0x5b7b,
1484 0x5b7c, 0x5b7d, 0x5b7e, 0x5c21, 0x5c22, 0x5c23, 0x5c24, 0x5c25,
1485 0x5c26, 0x5c27, 0x5c28, 0x5c29, 0x5c2a, 0x5c2b, 0x5c2c, 0x5c2d,
1486 0x5c2e, 0x5c2f, 0x5c30, 0x5c31, 0x5c32, 0x5c33, 0x5c34, 0x5c35,
1487 0x5c36, 0x5c37, 0x5c38, 0x5c39, 0x5c3a, 0x5c3b, 0x5c3c, 0x5c3d,
1488 0x5c3e, 0x5c3f, 0x5c40, 0x5c41, 0x5c42, 0x5c43, 0x5c44, 0x5c45,
1489 0x5c46, 0x5c47, 0x5c48, 0x5c49, 0x5c4a, 0x5c4b, 0x5c4c, 0x5c4d,
1490 0x5c4e, 0x5c4f, 0x5c50, 0x5c51, 0x5c52, 0x5c53, 0x5c54, 0x5c55,
1491 0x5c56, 0x5c57, 0x5c58, 0x5c59, 0x5c5a, 0x5c5b, 0x5c5c, 0x5c5d,
1492 0x5c5e, 0x5c5f, 0x5c60, 0x5c61, 0x5c62, 0x5c63, 0x5c64, 0x5c65,
1493 0x5c66, 0x5c67, 0x5c68, 0x5c69, 0x5c6a, 0x5c6b, 0x5c6c, 0x5c6d,
1494 0x5c6e, 0x5c6f, 0x5c70, 0x5c71, 0x5c72, 0x5c73, 0x5c74, 0x5c75,
1495 0x5c76, 0x5c77, 0x5c78, 0x5c79, 0x5c7a, 0x5c7b, 0x5c7c, 0x5c7d,
1496 0x5c7e, 0x5d21, 0x5d22, 0x5d23, 0x5d24, 0x5d25, 0x5d26, 0x5d27,
1497 0x5d28, 0x5d29, 0x5d2a, 0x5d2b, 0x5d2c, 0x5d2d, 0x5d2e, 0x5d2f,
1498 0x5d30, 0x5d31, 0x5d32, 0x5d33, 0x5d34, 0x5d35, 0x5d36, 0x5d37,
1499 0x5d38, 0x5d39, 0x5d3a, 0x5d3b, 0x5d3c, 0x5d3d, 0x5d3e, 0x5d3f,
1500 0x5d40, 0x5d41, 0x5d42, 0x5d43, 0x5d44, 0x5d45, 0x5d46, 0x5d47,
1501 0x5d48, 0x5d49, 0x5d4a, 0x5d4b, 0x5d4c, 0x5d4d, 0x5d4e, 0x5d4f,
1502 0x5d50, 0x5d51, 0x5d52, 0x5d53, 0x5d54, 0x5d55, 0x5d56, 0x5d57,
1503 0x5d58, 0x5d59, 0x5d5a, 0x5d5b, 0x5d5c, 0x5d5d, 0x5d5e, 0x5d5f,
1504 0x5d60, 0x5d61, 0x5d62, 0x5d63, 0x5d64, 0x5d65, 0x5d66, 0x5d67,
1505 0x5d68, 0x5d69, 0x5d6a, 0x5d6b, 0x5d6c, 0x5d6d, 0x5d6e, 0x5d6f,
1506 0x5d70, 0x5d71, 0x5d72, 0x5d73, 0x5d74, 0x5d75, 0x5d76, 0x5d77,
1507 0x5d78, 0x5d79, 0x5d7a, 0x5d7b, 0x5d7c, 0x5d7d, 0x5d7e, 0x5e21,
1508 0x5e22, 0x5e23, 0x5e24, 0x5e25, 0x5e26, 0x5e27, 0x5e28, 0x5e29,
1509 0x5e2a, 0x5e2b, 0x5e2c, 0x5e2d, 0x5e2e, 0x5e2f, 0x5e30, 0x5e31,
1510 0x5e32, 0x5e33, 0x5e34, 0x5e35, 0x5e36, 0x5e37, 0x5e38, 0x5e39,
1511 0x5e3f, 0x5e3a, 0x5e3b, 0x5e3c, 0x5e3d, 0x5e3e, 0x5e40, 0x5e41,
1512 0x5e42, 0x5e43, 0x5e44, 0x5e45, 0x5e46, 0x5e47, 0x5e48, 0x5e49,
1513 0x5e4e, 0x5e4a, 0x5e4b, 0x5e4c, 0x5e4d, 0x5e4f, 0x5e50, 0x5e51,
1514 0x5e52, 0x5e53, 0x5e54, 0x5e55, 0x5e56, 0x5e57, 0x5e58, 0x5e59,
1515 0x5e5a, 0x5e5b, 0x5e5c, 0x5e5d, 0x5e5e, 0x5e5f, 0x5e60, 0x5e61,
1516 0x5e62, 0x5e63, 0x5e64, 0x5e65, 0x5e66, 0x5e67, 0x5e68, 0x5e69,
1517 0x5e6a, 0x5e6b, 0x5e6c, 0x5e6d, 0x5e6e, 0x5e6f, 0x5e72, 0x5e70,
1518 0x5e71, 0x5e73, 0x5e74, 0x5e75, 0x5e76, 0x5e77, 0x5e78, 0x5e79,
1519 0x5e7a, 0x5e7b, 0x5e7c, 0x5e7d, 0x5e7e, 0x5f21, 0x5f22, 0x5f23,
1520 0x5f24, 0x5f25, 0x5f26, 0x5f27, 0x5f28, 0x5f29, 0x5f2a, 0x5f2b,
1521 0x5f2c, 0x5f2d, 0x5f2e, 0x5f2f, 0x5f30, 0x5f32, 0x5f31, 0x5f33,
1522 0x5f34, 0x5f35, 0x5f36, 0x5f37, 0x5f38, 0x5f39, 0x5f3a, 0x5f3b,
1523 0x5f3c, 0x5f3d, 0x5f3e, 0x5f3f, 0x5f40, 0x5f41, 0x5f42, 0x5f43,
1524 0x5f44, 0x5f45, 0x5f46, 0x5f47, 0x5f48, 0x5f49, 0x5f4a, 0x5f4b,
1525 0x5f4c, 0x5f4d, 0x5f4e, 0x5f4f, 0x5f50, 0x5f51, 0x5f52, 0x5f53,
1526 0x5f54, 0x5f55, 0x5f56, 0x5f57, 0x5f58, 0x5f59, 0x5f5a, 0x5f5b,
1527 0x5f5c, 0x5f5d, 0x5f5e, 0x5f5f, 0x5f60, 0x5f61, 0x5f62,
1528 0x5f63, 0x5f64, 0x5f65, 0x5f66, 0x5f67, 0x5f68, 0x5f69, 0x5f6a,
1529 0x5f6b, 0x5f6c, 0x5f6d, 0x5f6e, 0x5f70, 0x5f71, 0x5f72, 0x5f73,
1530 0x5f74, 0x5f75, 0x5f76, 0x5f77, 0x5f78, 0x5f79, 0x5f7a, 0x5f7b,
1531 0x5f7c, 0x5f7d, 0x5f7e, 0x6021, 0x6022, 0x6023, 0x6024, 0x6025,
1532 0x6026, 0x6027, 0x6028, 0x6029, 0x602a, 0x602b, 0x602c, 0x602d,
1533 0x602e, 0x602f, 0x6030, 0x6031, 0x6032, 0x6033, 0x6034, 0x6035,
1534 0x6036, 0x6037, 0x6038, 0x6039, 0x603a, 0x603b, 0x603c, 0x603d,
1535 0x603e, 0x603f, 0x6040, 0x6041, 0x6042, 0x6043, 0x6044, 0x6045,
1536 0x6046, 0x6047, 0x6048, 0x6049, 0x604a, 0x604b, 0x604c, 0x604d,
1537 0x604e, 0x604f, 0x6050, 0x6051, 0x6052, 0x6053, 0x6054, 0x6055,
1538 0x6056, 0x6057, 0x6058, 0x6059, 0x605a, 0x605b, 0x605c, 0x605d,
1539 0x6064, 0x606e, 0x6065f, 0x6060, 0x6061, 0x6062, 0x6063, 0x6065,
1540 0x6066, 0x6067, 0x6068, 0x6069, 0x606a, 0x606b, 0x606c, 0x606d,
1541 0x606e, 0x606f, 0x6070, 0x6071, 0x6072, 0x6073, 0x6074, 0x6075,
1542 0x6076, 0x6077, 0x6078, 0x6079, 0x607a, 0x607b, 0x607c, 0x607d,
1543 0x607e, 0x6121, 0x6122, 0x6123, 0x6124, 0x6125, 0x6126, 0x6127,
1544 0x6128, 0x6129, 0x612a, 0x612b, 0x612c, 0x612d, 0x612e, 0x612f,
1545 0x6130, 0x6131, 0x6132, 0x6133, 0x6134, 0x6135, 0x6136, 0x6137,
1546 0x6138, 0x6139, 0x613a, 0x613b, 0x613c, 0x613d, 0x613e, 0x613f,
1547 0x6140, 0x6141, 0x6142, 0x6143, 0x6144, 0x6145, 0x6146, 0x6147,
1548 0x6148, 0x6149, 0x614a, 0x614b, 0x614c, 0x614d, 0x614e, 0x614f,
1549 0x6150, 0x6151, 0x6152, 0x6154, 0x6155, 0x6156, 0x6153, 0x6157,
1550 0x6158, 0x6159, 0x615a, 0x615b, 0x615c, 0x615d, 0x615e, 0x615f,
1551 0x6160, 0x6161, 0x6162, 0x6163, 0x6164, 0x6165, 0x6166, 0x6167,
1552 0x6168, 0x6169, 0x616a, 0x616b, 0x616c, 0x616d, 0x616e, 0x616f,
1553 0x6170, 0x6171, 0x6172, 0x6173, 0x6174, 0x6175, 0x6176, 0x6177,
1554 0x6178, 0x6179, 0x617a, 0x617b, 0x617d, 0x617e, 0x6221, 0x6222,
1555 0x6223, 0x6224, 0x617c, 0x622d, 0x6225, 0x6226, 0x6227, 0x6228,
1556 0x6229, 0x622a, 0x622b, 0x622c, 0x622f, 0x6230, 0x6231, 0x6232,
1557 0x622e, 0x6233, 0x6234, 0x6235, 0x6236, 0x6237, 0x6238, 0x6239,
1558 0x623a, 0x623b, 0x623c, 0x623d, 0x623e, 0x623f, 0x6240, 0x6241,
1559 0x6242, 0x6243, 0x6245, 0x6246, 0x6247, 0x6248, 0x6249,
1560 0x624a, 0x624b, 0x624c, 0x624d, 0x624e, 0x624f, 0x6250, 0x6251,
1561 0x6252, 0x6253, 0x6254, 0x6255, 0x6256, 0x6257, 0x6258, 0x6259,
1562 0x625a, 0x625b, 0x625c, 0x625d, 0x625e, 0x625f, 0x6260, 0x6261,
1563 0x6262, 0x6263, 0x6264, 0x6265, 0x6266, 0x6267, 0x6268, 0x6269,
1564 0x626a, 0x626b, 0x626c, 0x626d, 0x626e, 0x626f, 0x6270, 0x6271,
1565 0x6272, 0x6273, 0x6274, 0x6275, 0x6276, 0x6277, 0x6278, 0x6279,
1566 0x627a, 0x627b, 0x627c, 0x627d, 0x627e, 0x6321, 0x6322, 0x6323,

1567 0x6324, 0x6325, 0x6326, 0x6327, 0x6328, 0x6329, 0x632a, 0x632b,
1568 0x632c, 0x632d, 0x632e, 0x632f, 0x6330, 0x6331, 0x6332, 0x6333,
1569 0x6334, 0x6335, 0x6336, 0x6337, 0x6338, 0x6339, 0x633a, 0x633b,
1570 0x633c, 0x633d, 0x633e, 0x633f, 0x6340, 0x6341, 0x6342, 0x6343,
1571 0x6344, 0x6345, 0x6346, 0x6347, 0x6348, 0x6349, 0x634a, 0x634b,
1572 0x634c, 0x634d, 0x634e, 0x634f, 0x6350, 0x6351, 0x6352, 0x6353,
1573 0x6354, 0x6355, 0x6356, 0x6357, 0x6358, 0x6359, 0x635a, 0x635b,
1574 0x635c, 0x635d, 0x635e, 0x635f, 0x6360, 0x6361, 0x6362, 0x6363,
1575 0x6364, 0x6365, 0x6366, 0x6367, 0x6368, 0x6369, 0x636a, 0x636b,
1576 0x636c, 0x636d, 0x636e, 0x636f, 0x6370, 0x6371, 0x6372, 0x6373,
1577 0x6374, 0x6375, 0x6376, 0x6377, 0x6378, 0x6379, 0x637a, 0x637b,
1578 0x637c, 0x637d, 0x637e, 0x637f, 0x6421, 0x6422, 0x6423, 0x6424, 0x6425,
1579 0x6426, 0x6427, 0x6428, 0x6429, 0x642a, 0x642b, 0x642c, 0x642d,
1580 0x642e, 0x642f, 0x6430, 0x6431, 0x6432, 0x6433, 0x6434, 0x6435,
1581 0x6436, 0x6437, 0x6438, 0x6439, 0x643a, 0x643b, 0x643c, 0x643d,
1582 0x643e, 0x643f, 0x6440, 0x6441, 0x6442, 0x6443, 0x6444, 0x6445,
1583 0x6446, 0x6447, 0x6448, 0x6449, 0x644a, 0x644b, 0x644c, 0x644d,
1584 0x644e, 0x644f, 0x6450, 0x6451, 0x6452, 0x6453, 0x6454, 0x6455,
1585 0x6456, 0x6457, 0x6458, 0x6459, 0x645a, 0x645b, 0x645c, 0x645d,
1586 0x645e, 0x645f, 0x6460, 0x6461, 0x6462, 0x6463, 0x6464, 0x6465,
1587 0x6466, 0x6467, 0x6468, 0x6469, 0x646a, 0x646b, 0x646c, 0x646d,
1588 0x646e, 0x646f, 0x6470, 0x6471, 0x6472, 0x6473, 0x6474, 0x6475,
1589 0x6476, 0x6477, 0x6478, 0x6479, 0x647a, 0x647b, 0x647c, 0x647d,
1590 0x647e, 0x6521, 0x6522, 0x6523, 0x6524, 0x6525, 0x6526, 0x6527,
1591 0x6528, 0x6529, 0x652a, 0x652b, 0x652c, 0x652d, 0x652e, 0x652f,
1592 0x6530, 0x6531, 0x6532, 0x6533, 0x6534, 0x6535, 0x653b, 0x653c,
1593 0x6537, 0x6538, 0x6539, 0x653a, 0x653b, 0x653c, 0x653d, 0x653e,
1594 0x6540, 0x6541, 0x6542, 0x6543, 0x6544, 0x6545, 0x6546, 0x6547,
1595 0x6548, 0x6549, 0x654a, 0x654b, 0x654c, 0x654d, 0x654e, 0x654f,
1596 0x654e, 0x6551, 0x6552, 0x6553, 0x6554, 0x6555, 0x6556, 0x6557,
1597 0x6558, 0x6559, 0x655a, 0x655b, 0x655c, 0x655d, 0x655e, 0x655f,
1598 0x6560, 0x6561, 0x6562, 0x6563, 0x6564, 0x6565, 0x6566, 0x6567,
1599 0x6567, 0x6568, 0x6569, 0x656a, 0x656b, 0x656c, 0x656d, 0x656e,
1600 0x6570, 0x6571, 0x6572, 0x6573, 0x6574, 0x6575, 0x6576, 0x6577,
1601 0x6578, 0x6579, 0x657a, 0x657b, 0x657c, 0x657d, 0x657e, 0x6621,
1602 0x6622, 0x6623, 0x6624, 0x6625, 0x6626, 0x6627, 0x6628, 0x6629,
1603 0x662a, 0x662b, 0x662c, 0x662d, 0x662e, 0x662f, 0x6630, 0x6631,
1604 0x6632, 0x6633, 0x6634, 0x6635, 0x6636, 0x6637, 0x6638, 0x6639,
1605 0x663a, 0x663b, 0x663c, 0x663d, 0x663e, 0x663f, 0x6640, 0x6641,
1606 0x6642, 0x6643, 0x6644, 0x6645, 0x6646, 0x6647, 0x6648, 0x6649,
1607 0x664a, 0x664b, 0x664c, 0x664d, 0x664e, 0x664f, 0x6650, 0x6651,
1608 0x6652, 0x6653, 0x6654, 0x6655, 0x6656, 0x6657, 0x6658, 0x6659,
1609 0x665a, 0x665b, 0x665c, 0x665d, 0x665e, 0x665f, 0x6660, 0x6661,
1610 0x6662, 0x6663, 0x6664, 0x6665, 0x6666, 0x6667, 0x6668, 0x6669,
1611 0x666a, 0x666b, 0x666c, 0x666d, 0x666e, 0x666f, 0x6670, 0x6671,
1612 0x6672, 0x6673, 0x6675, 0x6676, 0x6677, 0x6678, 0x6679, 0x667a,
1613 0x667b, 0x667c, 0x667d, 0x667e, 0x6721, 0x6722, 0x6723, 0x6724,
1614 0x6725, 0x6726, 0x6727, 0x6728, 0x6729, 0x672a, 0x672b, 0x672c,
1615 0x672d, 0x672e, 0x672f, 0x6730, 0x6731, 0x6732, 0x6733, 0x6734,
1616 0x6735, 0x6736, 0x6737, 0x6738, 0x6739, 0x673a, 0x673b, 0x673c,
1617 0x673d, 0x673e, 0x673f, 0x6740, 0x6741, 0x6742, 0x6743, 0x6744,
1618 0x6745, 0x6746, 0x6747, 0x6748, 0x6749, 0x674a, 0x674b, 0x674c,
1619 0x674d, 0x674e, 0x674f, 0x6750, 0x6751, 0x6752, 0x6753, 0x6754,
1620 0x6755, 0x6756, 0x6757, 0x6758, 0x6759, 0x675a, 0x675b, 0x675c,
1621 0x675d, 0x675e, 0x675f, 0x6760, 0x6761, 0x6762, 0x6763, 0x6764,
1622 0x6765, 0x6766, 0x676a, 0x6767, 0x6768, 0x6769, 0x676b, 0x676c,
1623 0x676d, 0x676e, 0x676f, 0x6770, 0x6771, 0x6772, 0x6773, 0x6774,
1624 0x6776, 0x6777, 0x6778, 0x6779, 0x6775, 0x677a, 0x677b, 0x677c,
1625 0x677d, 0x6828, 0x677e, 0x6821, 0x6822, 0x6823, 0x6824, 0x6825,
1626 0x6826, 0x6827, 0x6829, 0x682a, 0x682b, 0x682c, 0x682d, 0x682e,
1627 0x682f, 0x6830, 0x6831, 0x6832, 0x6833, 0x6834, 0x6835, 0x6836,
1628 0x6837, 0x6838, 0x6839, 0x683a, 0x683b, 0x683c, 0x683d, 0x683e,
1629 0x683f, 0x6840, 0x6841, 0x6842, 0x6843, 0x6844, 0x6845, 0x6846,
1630 0x6847, 0x6848, 0x6849, 0x684a, 0x684b, 0x684c, 0x684d, 0x684e,
1631 0x684f, 0x6850, 0x6851, 0x6852, 0x6853, 0x6854, 0x6855, 0x6856,
1632 0x6857, 0x6858, 0x6859, 0x685a, 0x685b, 0x685c, 0x685d, 0x685e,
1633 0x685f, 0x6860, 0x6861, 0x6862, 0x6863, 0x6864, 0x6865, 0x6866,
1634 0x6867, 0x6868, 0x6869, 0x686a, 0x686b, 0x686c, 0x686d, 0x686e,
1635 0x686f, 0x6870, 0x6871, 0x6872, 0x6873, 0x6874, 0x6875, 0x6876,
1636 0x6877, 0x6878, 0x6879, 0x687a, 0x687b, 0x687c, 0x687d, 0x687e,
1637 0x6921, 0x6922, 0x6923, 0x6924, 0x6925, 0x6926, 0x6927, 0x6928,
1638 0x6929, 0x692a, 0x692b, 0x692c, 0x692d, 0x692e, 0x692f, 0x6930,
1639 0x6931, 0x6932, 0x6933, 0x6934, 0x6935, 0x6936, 0x6937, 0x6938,
1640 0x6939, 0x693a, 0x693b, 0x693c, 0x693d, 0x693e, 0x693f, 0x6940,
1641 0x6941, 0x6942, 0x6943, 0x6944, 0x6945, 0x6946, 0x6947, 0x6948,
1642 0x6949, 0x694a, 0x694b, 0x694c, 0x694d, 0x694e, 0x694f, 0x6950,
1643 0x6951, 0x6952, 0x6953, 0x6954, 0x6955, 0x6956, 0x6957, 0x6958,
1644 0x6959, 0x695a, 0x695b, 0x695c, 0x695d, 0x695e, 0x695f, 0x6960,
1645 0x6961, 0x6962, 0x6963, 0x6964, 0x6965, 0x6966, 0x6967, 0x6968,
1646 0x6969, 0x696a, 0x696b, 0x696c, 0x696d, 0x696e, 0x696f, 0x6970,
1647 0x6971, 0x6972, 0x6973, 0x6974, 0x6975, 0x6976, 0x6977, 0x6978,
1648 0x6979, 0x697a, 0x697b, 0x697c, 0x697d, 0x697e, 0x6a21, 0x6a22,
1649 0x6a23, 0x6a24, 0x6a25, 0x6a26, 0x6a27, 0x6a28, 0x6a29, 0x6a2a,
1650 0x6a2b, 0x6a2c, 0x6a2d, 0x6a2e, 0x6a2f, 0x6a30, 0x6a31, 0x6a32,
1651 0x6a33, 0x6a34, 0x6a35, 0x6a36, 0x6a37, 0x6a38, 0x6a39, 0x6a3a,
1652 0x6a3b, 0x6a3c, 0x6a3d, 0x6a3e, 0x6a3f, 0x6a40, 0x6a41, 0x6a42,
1653 0x6a43, 0x6a44, 0x6a45, 0x6a46, 0x6a47, 0x6a48, 0x6a49, 0x6a4a,

```

1654 0x6a4b, 0x6a4c, 0x6a4d, 0x6a4e, 0x6a4f, 0x6a50, 0x6a51, 0x6a52,
1655 0x6a53, 0x6a54, 0x6a55, 0x6a56, 0x6a57, 0x6a58, 0x6a59, 0x6a5a,
1656 0x6a5b, 0x6a5c, 0x6a5d, 0x6a5e, 0x6a5f, 0x6a60, 0x6a61, 0x6a62,
1657 0x6a63, 0x6a64, 0x6a65, 0x6a66, 0x6a67, 0x6a68, 0x6a69, 0x6a6a,
1658 0x6a6b, 0x6a6c, 0x6a6d, 0x6a6e, 0x6a6f, 0x6a70, 0x6a71, 0x6a72,
1659 0x6a73, 0x6a74, 0x6a75, 0x6a76, 0x6a77, 0x6a78, 0x6a79, 0x6a7a,
1660 0x6a7b, 0x6a7c, 0x6a7d, 0x6a7e, 0x6b21, 0x6b22, 0x6b23, 0x6b24,
1661 0x6b25, 0x6b26, 0x6b27, 0x6b28, 0x6b29, 0x6b2a, 0x6b2b, 0x6b2c,
1662 0x6b2d, 0x6b2e, 0x6b2f, 0x6b30, 0x6b31, 0x6b32, 0x6b33, 0x6b34,
1663 0x6b35, 0x6b36, 0x6b37, 0x6b38, 0x6b39, 0x6b3a, 0x6b3b, 0x6b3c,
1664 0x6b3d, 0x6b3e, 0x6b3f, 0x6b40, 0x6b41, 0x6b42, 0x6b43, 0x6b44,
1665 0x6b45, 0x6b46, 0x6b47, 0x6b48, 0x6b49, 0x6b50, 0x6b4a, 0x6b4b,
1666 0x6b4c, 0x6b4d, 0x6b52, 0x6b4e, 0x6b4f, 0x6b51, 0x6b53, 0x6b54,
1667 0x6b55, 0x6b56, 0x6b57, 0x6b58, 0x6b59, 0x6b5a, 0x6b5b, 0x6b5c,
1668 0x6b5e, 0x6b5d, 0x6b5f, 0x6b60, 0x6b61, 0x6b62, 0x6b63, 0x6b64,
1669 0x6b65, 0x6b66, 0x6b67, 0x6b68, 0x6b69, 0x6b6a, 0x6b6b, 0x6b6d,
1670 0x6b6e, 0x6b6f, 0x6b6c, 0x6b70, 0x6b71, 0x6b72, 0x6b73, 0x6b74,
1671 0x6b76, 0x6b75, 0x6b77, 0x6b78, 0x6b79, 0x6b7a, 0x6b7b, 0x6b7c,
1672 0x6b7d, 0x6b7e, 0x6c21, 0x6c22, 0x6c23, 0x6c24, 0x6c25, 0x6c26,
1673 0x6c27, 0x6c28, 0x6c29, 0x6c2a, 0x6c2b, 0x6c2c, 0x6c2d, 0x6c2e,
1674 0x6c2f, 0x6c30, 0x6c31, 0x6c32, 0x6c33, 0x6c34, 0x6c35, 0x6c36,
1675 0x6c37, 0x6c38, 0x6c39, 0x6c3a, 0x6c3b, 0x6c3c, 0x6c3d, 0x6c3e,
1676 0x6c3f, 0x6c40, 0x6c41, 0x6c42, 0x6c43, 0x6c44, 0x6c45, 0x6c46,
1677 0x6c47, 0x6c48, 0x6c49, 0x6c4a, 0x6c4b, 0x6c4c, 0x6c4e, 0x6c4f,
1678 0x6c4d, 0x6c50, 0x6c51, 0x6c52, 0x6c53, 0x6c54, 0x6c55, 0x6c56,
1679 0x6c57, 0x6c58, 0x6c59, 0x6c5a, 0x6c5b, 0x6c5c, 0x6c5d, 0x6c5e,
1680 0x6c5f, 0x6c60, 0x6c61, 0x6c62, 0x6c63, 0x6c64, 0x6c65, 0x6c66,
1681 0x6c67, 0x6c68, 0x6c69, 0x6c6a, 0x6c6b, 0x6c6c, 0x6c6d, 0x6c6e,
1682 0x6c6f, 0x6c70, 0x6c71, 0x6c72, 0x6c73, 0x6c74, 0x6c75, 0x6c76,
1683 0x6c77, 0x6c78, 0x6c79, 0x6c7a, 0x6c7b, 0x6c7c, 0x6c7d, 0x6c7e,
1684 0x6d21, 0x6d22, 0x6d23, 0x6d24, 0x6d25, 0x6d26, 0x6d27, 0x6d28,
1685 0x6d29, 0x6d2a, 0x6d2b, 0x6d2c, 0x6d2d, 0x6d2e, 0x6d2f, 0x6d30,
1686 0x6d31, 0x6d32, 0x6d33, 0x6d34, 0x6d35, 0x6d36, 0x6d37, 0x6d38,
1687 0x6d39, 0x6d3a, 0x6d3b, 0x6d3c, 0x6d3d, 0x6d3e, 0x6d3f, 0x6d40,
1688 0x6d41, 0x6d42, 0x6d43, 0x6d44, 0x6d45, 0x6d46, 0x6d47, 0x6d48,
1689 0x6d49, 0x6d4a, 0x6d4b, 0x6d4c, 0x6d4d, 0x6d4e, 0x6d4f, 0x6d50,
1690 0x6d51, 0x6d52, 0x6d53, 0x6d54, 0x6d55, 0x6d56, 0x6d57, 0x6d58,
1691 0x6d59, 0x6d5a, 0x6d5b, 0x6d5c, 0x6d5d, 0x6d5e, 0x6d5f, 0x6d60,
1692 0x6d61, 0x6d62, 0x6d63,
1693 };
1694
1695 static const Summary16 jisx0212_uni2indx_page00[70] = {
1696 /* 0x0000 */
1697 { 0, 0x0000 }, { 0, 0x0000 }, { 0, 0x0000 }, { 0, 0x0000 },
1698 { 0, 0x0000 }, { 0, 0x0000 }, { 0, 0x0000 }, { 0, 0x4000 },
1699 { 1, 0x0000 }, { 1, 0x0000 }, { 1, 0xc652 }, { 8, 0x8500 },
1700 { 11, 0xffff }, { 27, 0xff7e }, { 41, 0xffff }, { 57, 0xff7f },
1701 /* 0x0100 */
1702 { 72, 0xffff }, { 88, 0xffcf }, { 102, 0xcff7 }, { 115, 0xffff },
1703 { 131, 0x3fff }, { 145, 0xffff }, { 161, 0xffff }, { 177, 0x7fff },
1704 { 192, 0x0000 }, { 192, 0x0000 }, { 192, 0x0000 }, { 192, 0x0000 },
1705 { 192, 0xe000 }, { 195, 0x1fff }, { 208, 0x0000 }, { 208, 0x0020 },
1706 /* 0x0200 */
1707 { 209, 0x0000 }, { 209, 0x0000 }, { 209, 0x0000 }, { 209, 0x0000 },
1708 { 209, 0x0000 }, { 209, 0x0000 }, { 209, 0x0000 }, { 209, 0x0000 },
1709 { 209, 0x0000 }, { 209, 0x0000 }, { 209, 0x0000 }, { 209, 0x0000 },
1710 { 209, 0x0080 }, { 210, 0x2f00 }, { 215, 0x0000 }, { 215, 0x0000 },
1711 /* 0x0300 */
1712 { 215, 0x0000 }, { 215, 0x0000 }, { 215, 0x0000 }, { 215, 0x0000 },
1713 { 215, 0x0000 }, { 215, 0x0000 }, { 215, 0x0000 }, { 215, 0x0000 },
1714 { 215, 0xd770 }, { 224, 0x0001 }, { 225, 0xfc00 }, { 231, 0x0001 },
1715 { 232, 0x7c04 }, { 238, 0x0000 }, { 238, 0x0000 }, { 238, 0x0000 },
1716 /* 0x0400 */
1717 { 238, 0xdffc }, { 251, 0x0000 }, { 251, 0x0000 }, { 251, 0x0000 },
1718 { 251, 0x0000 }, { 251, 0xdffc },
1719 };
1720 static const Summary16 jisx0212_uni2indx_page21[3] = {
1721 /* 0x2100 */
1722 { 264, 0x0000 }, { 264, 0x0040 }, { 265, 0x0004 },
1723 };
1724 static const Summary16 jisx0212_uni2indx_page4e[1307] = {
1725 /* 0x4e00 */
1726 { 266, 0x1034 }, { 270, 0x8004 }, { 272, 0xc918 }, { 278, 0x0021 },
1727 { 280, 0x0093 }, { 284, 0x1402 }, { 287, 0x0308 }, { 290, 0x8230 },
1728 { 294, 0x2000 }, { 295, 0x20c0 }, { 298, 0x8000 }, { 299, 0x0200 },
1729 { 300, 0x0008 }, { 301, 0x0c01 }, { 304, 0x8107 }, { 309, 0xe02a },
1730 /* 0x4f00 */
1731 { 315, 0x190d }, { 321, 0x02e4 }, { 326, 0x4000 }, { 327, 0x4aaa },
1732 { 334, 0x1b05 }, { 340, 0x8154 }, { 345, 0x5400 }, { 350, 0x6782 },
1733 { 357, 0x5636 }, { 365, 0xc69d }, { 374, 0x0000 }, { 374, 0x7a84 },
1734 { 381, 0xbb63 }, { 391, 0x1004 }, { 393, 0x0005 }, { 395, 0xb005 },
1735 /* 0x5000 */
1736 { 400, 0x5493 }, { 407, 0x7989 }, { 415, 0x4084 }, { 418, 0x082d },
1737 { 423, 0x5467 }, { 431, 0x828e }, { 437, 0x24cd }, { 444, 0x0003 },
1738 { 446, 0xc45a }, { 453, 0xd85d }, { 462, 0x8407 }, { 467, 0x2601 },
1739 { 471, 0x5099 }, { 477, 0xb119 }, { 484, 0x8354 }, { 490, 0x4446 },
1740 /* 0x5100 */

```

```
1741 { 495, 0x79c8 }, { 503, 0x7a81 }, { 510, 0xb188 }, { 516, 0x033a },
1742 { 522, 0x8404 }, { 525, 0x81a8 }, { 530, 0x0050 }, { 532, 0x4000 },
1743 { 533, 0x4818 }, { 537, 0x2100 }, { 539, 0x200a }, { 542, 0xd500 },
1744 { 547, 0x8104 }, { 550, 0x412e }, { 556, 0x4024 }, { 559, 0x009c },
1745 /* 0x5200 */
1746 { 563, 0x0026 }, { 566, 0x016c }, { 571, 0x0104 }, { 573, 0x1026 },
1747 { 577, 0x0220 }, { 579, 0x95a0 }, { 585, 0x4043 }, { 589, 0x0380 },
1748 { 592, 0x1425 }, { 597, 0x15e8 }, { 604, 0x80f0 }, { 609, 0x2dc1 },
1749 { 616, 0x9151 }, { 622, 0x1852 }, { 627, 0x1722 }, { 633, 0x00d3 },
1750 /* 0x5300 */
1751 { 638, 0x1c09 }, { 643, 0xd90a }, { 650, 0x3ba0 }, { 657, 0x7025 },
1752 { 663, 0x1804 }, { 666, 0x0a00 }, { 668, 0x302a }, { 673, 0x4204 },
1753 { 676, 0x4188 }, { 680, 0x2218 }, { 684, 0x8c12 }, { 689, 0x25b4 },
1754 { 696, 0x8021 }, { 699, 0x642c }, { 705, 0x00c1 }, { 708, 0x0020 },
1755 /* 0x5400 */
1756 { 709, 0x0004 }, { 710, 0x0408 }, { 712, 0x8582 }, { 717, 0x0032 },
1757 { 720, 0xa098 }, { 725, 0x4000 }, { 726, 0x6ad4 }, { 734, 0x8010 },
1758 { 736, 0x232a }, { 742, 0x9062 }, { 747, 0x66c2 }, { 754, 0x8e82 },
1759 { 760, 0x6440 }, { 764, 0x0000 }, { 764, 0x9401 }, { 768, 0xd040 },
1760 /* 0x5500 */
1761 { 772, 0x7323 }, { 780, 0x0020 }, { 781, 0x0c00 }, { 783, 0x3864 },
1762 { 789, 0x2682 }, { 794, 0x4d03 }, { 800, 0x0053 }, { 804, 0x8000 },
1763 { 805, 0xc146 }, { 811, 0x009e }, { 816, 0x2018 }, { 819, 0x8004 },
1764 { 821, 0x5a4a }, { 828, 0x498e }, { 835, 0x0204 }, { 837, 0x8040 },
1765 /* 0x5600 */
1766 { 839, 0xe520 }, { 845, 0x0207 }, { 849, 0x1000 }, { 850, 0xbaa9 },
1767 { 859, 0xaa5b }, { 868, 0x4010 }, { 870, 0xa24f }, { 878, 0x0026 },
1768 { 881, 0x1930 }, { 886, 0xe620 }, { 892, 0x3bc0 }, { 899, 0x408a },
1769 { 903, 0xbe20 }, { 910, 0xb201 }, { 915, 0x29f2 }, { 923, 0x00c2 },
1770 /* 0x5700 */
1771 { 926, 0x1486 }, { 931, 0x2c22 }, { 936, 0xd63d }, { 946, 0xe018 },
1772 { 951, 0x3060 }, { 955, 0x0004 }, { 956, 0xe9a4 }, { 964, 0x5ebb },
1773 { 975, 0x100a }, { 978, 0xf6b0 }, { 987, 0x1382 }, { 992, 0x2100 },
1774 { 994, 0x9180 }, { 998, 0x6020 }, { 1001, 0x22d2 }, { 1007, 0xe161 },
1775 /* 0x5800 */
1776 { 1014, 0x3318 }, { 1020, 0xc800 }, { 1023, 0x20c1 }, { 1027, 0x8204 },
1777 { 1030, 0xb200 }, { 1034, 0x8021 }, { 1037, 0x0192 }, { 1041, 0x9100 },
1778 { 1044, 0xb783 }, { 1053, 0x2051 }, { 1057, 0x0247 }, { 1062, 0x1006 },
1779 { 1065, 0x6114 }, { 1070, 0x2455 }, { 1076, 0x0206 }, { 1079, 0x0008 },
1780 /* 0x5900 */
1781 { 1080, 0x1860 }, { 1084, 0x201c }, { 1088, 0x811a }, { 1093, 0x8069 },
1782 { 1098, 0x0048 }, { 1100, 0xea0c }, { 1107, 0xa80a }, { 1112, 0x1a64 },
1783 { 1118, 0x5800 }, { 1121, 0x80a4 }, { 1125, 0xe090 }, { 1130, 0x1489 },
1784 { 1135, 0x251a }, { 1141, 0xe004 }, { 1145, 0xc098 }, { 1150, 0x0096 },
1785 /* 0x5a00 */
1786 { 1154, 0x7011 }, { 1159, 0x400c }, { 1162, 0x2598 }, { 1168, 0x0001 },
1787 { 1169, 0x11b0 }, { 1174, 0x4021 }, { 1177, 0x20a8 }, { 1181, 0x4c80 },
1788 { 1185, 0x0800 }, { 1186, 0xd249 }, { 1193, 0x1085 }, { 1197, 0x8d2e },
1789 { 1205, 0x8150 }, { 1209, 0x1400 }, { 1211, 0x4421 }, { 1215, 0x2060 },
1790 /* 0x5b00 */
1791 { 1218, 0x0103 }, { 1221, 0x2a80 }, { 1225, 0x2022 }, { 1228, 0x0110 },
1792 { 1230, 0x1802 }, { 1233, 0x4044 }, { 1236, 0xc100 }, { 1239, 0xf000 },
1793 { 1243, 0x4452 }, { 1248, 0x005b }, { 1253, 0xb300 }, { 1258, 0x1486 },
1794 { 1263, 0xa003 }, { 1267, 0x07c0 }, { 1272, 0x8001 }, { 1274, 0x2012 },
1795 /* 0x5c00 */
1796 { 1277, 0x1000 }, { 1278, 0xc080 }, { 1281, 0x5a48 }, { 1287, 0x0065 },
1797 { 1291, 0x0000 }, { 1291, 0x1600 }, { 1294, 0x238c }, { 1300, 0x3c31 },
1798 { 1307, 0x8580 }, { 1311, 0xa004 }, { 1314, 0x044d }, { 1319, 0x0434 },
1799 { 1323, 0x0a00 }, { 1325, 0x2084 }, { 1328, 0x4000 }, { 1329, 0x0016 },
1800 /* 0x5d00 */
1801 { 1332, 0x2042 }, { 1335, 0x0004 }, { 1336, 0x08d8 }, { 1341, 0xa212 },
1802 { 1346, 0x054c }, { 1351, 0x8222 }, { 1355, 0x2417 }, { 1361, 0xc601 },
1803 { 1366, 0x050a }, { 1370, 0x8a3c }, { 1377, 0x0881 }, { 1380, 0x0315 },
1804 { 1385, 0x4888 }, { 1389, 0x0301 }, { 1392, 0x0211 }, { 1395, 0x0300 },
1805 /* 0x5e00 */
1806 { 1397, 0x2081 }, { 1400, 0x8134 }, { 1405, 0x4101 }, { 1408, 0x4024 },
1807 { 1411, 0x0a00 }, { 1413, 0x5943 }, { 1420, 0x7d00 }, { 1426, 0x0001 },
1808 { 1427, 0x4801 }, { 1430, 0x0000 }, { 1430, 0x1534 }, { 1436, 0xe00a },
1809 { 1441, 0x5840 }, { 1445, 0x5036 }, { 1451, 0x0820 }, { 1453, 0x0000 },
1810 /* 0x5f00 */
1811 { 1453, 0x41c4 }, { 1458, 0x3200 }, { 1461, 0x591e }, { 1469, 0xa851 },
1812 { 1475, 0x20b1 }, { 1480, 0x0911 }, { 1484, 0x8099 }, { 1489, 0x6534 },
1813 { 1496, 0xa200 }, { 1499, 0x3040 }, { 1502, 0x9894 }, { 1508, 0x0103 },
1814 { 1511, 0x0b90 }, { 1516, 0x401f }, { 1522, 0xf706 }, { 1531, 0x144c },
1815 /* 0x6000 */
1816 { 1536, 0x2480 }, { 1539, 0x8598 }, { 1545, 0x2010 }, { 1547, 0x0028 },
1817 { 1549, 0x1381 }, { 1554, 0x20d2 }, { 1559, 0x0082 }, { 1561, 0xc002 },
1818 { 1564, 0x4544 }, { 1569, 0x612a }, { 1575, 0x0134 }, { 1579, 0x4883 },
1819 { 1584, 0xcf14 }, { 1592, 0x6a30 }, { 1598, 0x0024 }, { 1600, 0x3124 },
1820 /* 0x6100 */
1821 { 1605, 0x1484 }, { 1609, 0x52df }, { 1619, 0x0c04 }, { 1622, 0x02e3 },
1822 { 1628, 0x0262 }, { 1632, 0x4000 }, { 1633, 0x1001 }, { 1635, 0x9904 },
1823 { 1640, 0x281b }, { 1646, 0xb18c }, { 1653, 0x2521 }, { 1658, 0x1300 },
1824 { 1661, 0xc007 }, { 1666, 0xf020 }, { 1671, 0xb2a6 }, { 1679, 0x0000 },
1825 /* 0x6200 */
1826 { 1679, 0x009a }, { 1683, 0x1028 }, { 1686, 0x0a8d }, { 1692, 0x2200 },
1827 { 1694, 0x105c }, { 1699, 0x1457 }, { 1706, 0xa010 }, { 1709, 0x2408 },
```

```

1828 { 1712, 0xe000 }, { 1715, 0x0001 }, { 1716, 0x0140 }, { 1718, 0xc4c8 },
1829 { 1724, 0x4010 }, { 1726, 0x0460 }, { 1729, 0x0400 }, { 1730, 0x3014 },
1830 /* 0x6300 */
1831 { 1734, 0x2c18 }, { 1739, 0x0149 }, { 1743, 0x2600 }, { 1746, 0x1260 },
1832 { 1750, 0x4c5e }, { 1758, 0x091c }, { 1763, 0x3060 }, { 1767, 0xb132 },
1833 { 1774, 0x0494 }, { 1778, 0x4631 }, { 1784, 0xe050 }, { 1789, 0x2000 },
1834 { 1790, 0x4122 }, { 1794, 0x103a }, { 1799, 0x1421 }, { 1803, 0x032c },
1835 /* 0x6400 */
1836 { 1808, 0x0600 }, { 1810, 0x4115 }, { 1815, 0x8635 }, { 1822, 0xa021 },
1837 { 1826, 0x8800 }, { 1828, 0xbc1e }, { 1837, 0x200b }, { 1841, 0x2818 },
1838 { 1845, 0x80a0 }, { 1848, 0xab03 }, { 1855, 0x114a }, { 1860, 0xe008 },
1839 { 1864, 0x5e10 }, { 1870, 0x00a3 }, { 1874, 0x2630 }, { 1879, 0x88a1 },
1840 /* 0x6500 */
1841 { 1884, 0x8712 }, { 1890, 0xca58 }, { 1897, 0x4244 }, { 1901, 0x3402 },
1842 { 1905, 0x0288 }, { 1908, 0x8015 }, { 1912, 0x0881 }, { 1915, 0x2400 },
1843 { 1917, 0x0422 }, { 1920, 0x2124 }, { 1924, 0x4049 }, { 1928, 0x801c },
1844 { 1932, 0x4304 }, { 1936, 0x8151 }, { 1941, 0x0000 }, { 1941, 0xc235 },
1845 /* 0x6600 */
1846 { 1948, 0x2311 }, { 1953, 0x6066 }, { 1959, 0x5e5e }, { 1969, 0x028b },
1847 { 1974, 0x5461 }, { 1980, 0x1b82 }, { 1986, 0x1c03 }, { 1991, 0xdba8 },
1848 { 2000, 0x3801 }, { 2004, 0x9e05 }, { 2011, 0x2011 }, { 2014, 0x8826 },
1849 { 2019, 0xd10d }, { 2026, 0x8810 }, { 2029, 0x5900 }, { 2033, 0x0c00 },
1850 /* 0x6700 */
1851 { 2035, 0x40a0 }, { 2038, 0x1208 }, { 2041, 0x0005 }, { 2043, 0x4008 },
1852 { 2045, 0x11a0 }, { 2049, 0x2030 }, { 2052, 0x5040 }, { 2055, 0x0850 },
1853 { 2058, 0xc012 }, { 2062, 0x0b4a }, { 2068, 0x0000 }, { 2068, 0x3827 },
1854 { 2075, 0x032d }, { 2081, 0x1284 }, { 2085, 0x0042 }, { 2087, 0x02c5 },
1855 /* 0x6800 */
1856 { 2092, 0x0000 }, { 2092, 0xa210 }, { 2096, 0xb180 }, { 2101, 0x880b },
1857 { 2106, 0x1430 }, { 2110, 0x09a4 }, { 2115, 0xc800 }, { 2118, 0x1e27 },
1858 { 2126, 0x0154 }, { 2130, 0x1540 }, { 2134, 0x462a }, { 2140, 0x0804 },
1859 { 2142, 0x9120 }, { 2146, 0x324b }, { 2153, 0x3d20 }, { 2159, 0x3863 },
1860 /* 0x6900 */
1861 { 2166, 0x0640 }, { 2169, 0x00cb }, { 2174, 0x0000 }, { 2174, 0x092a },
1862 { 2179, 0x4224 }, { 2183, 0x0880 }, { 2185, 0x1378 }, { 2192, 0x8c07 },
1863 { 2198, 0x2001 }, { 2200, 0x0144 }, { 2203, 0xa962 }, { 2210, 0x1580 },
1864 { 2214, 0x0120 }, { 2216, 0x00c2 }, { 2219, 0xc024 }, { 2223, 0x402a },
1865 /* 0x6a00 */
1866 { 2227, 0x800b }, { 2231, 0x2422 }, { 2235, 0x0111 }, { 2238, 0xc895 },
1867 { 2245, 0x4660 }, { 2250, 0x0867 }, { 2256, 0x0490 }, { 2259, 0x400a },
1868 { 2262, 0x0aca }, { 2268, 0xe802 }, { 2273, 0x8820 }, { 2276, 0xe013 },
1869 { 2282, 0x1340 }, { 2286, 0x3071 }, { 2292, 0x1090 }, { 2295, 0x3007 },
1870 /* 0x6b00 */
1871 { 2300, 0x82cc }, { 2306, 0x4883 }, { 2311, 0x9910 }, { 2316, 0x8860 },
1872 { 2320, 0x2440 }, { 2323, 0x2144 }, { 2327, 0x4881 }, { 2331, 0x6021 },
1873 { 2335, 0x0024 }, { 2337, 0x8880 }, { 2340, 0x730d }, { 2348, 0x6301 },
1874 { 2353, 0x1218 }, { 2357, 0x0440 }, { 2359, 0x40ca }, { 2364, 0x8282 },
1875 /* 0x6c00 */
1876 { 2368, 0x6234 }, { 2374, 0x8205 }, { 2378, 0x51c0 }, { 2383, 0x8c68 },
1877 { 2389, 0xac00 }, { 2393, 0x1a14 }, { 2398, 0xa880 }, { 2402, 0x0b50 },
1878 { 2407, 0x02e0 }, { 2411, 0x91b0 }, { 2417, 0x0000 }, { 2417, 0x0015 },
1879 { 2420, 0xa044 }, { 2424, 0x1457 }, { 2431, 0x5a81 }, { 2437, 0x0014 },
1880 /* 0x6d00 */
1881 { 2439, 0xc490 }, { 2444, 0x040a }, { 2447, 0xc1c0 }, { 2452, 0x9202 },
1882 { 2456, 0x0000 }, { 2456, 0xc080 }, { 2459, 0x80a2 }, { 2463, 0x1001 },
1883 { 2465, 0x0084 }, { 2467, 0x01d6 }, { 2473, 0x1400 }, { 2475, 0xa290 },
1884 { 2480, 0xc510 }, { 2485, 0xa840 }, { 2489, 0x8225 }, { 2494, 0x1051 },
1885 /* 0x6e00 */
1886 { 2498, 0x0011 }, { 2500, 0x4000 }, { 2501, 0x0084 }, { 2503, 0x1a44 },
1887 { 2508, 0x8b30 }, { 2514, 0x709e }, { 2522, 0x010c }, { 2525, 0x2808 },
1888 { 2528, 0x2000 }, { 2529, 0x0208 }, { 2531, 0x6081 }, { 2535, 0x880a },
1889 { 2539, 0xe58b }, { 2548, 0x0000 }, { 2548, 0x6800 }, { 2551, 0x2a00 },
1890 /* 0x6f00 */
1891 { 2554, 0x3510 }, { 2559, 0x0d40 }, { 2563, 0xa640 }, { 2568, 0x1849 },
1892 { 2573, 0x8000 }, { 2574, 0x668e }, { 2582, 0x110e }, { 2586, 0x6000 },
1893 { 2588, 0x3988 }, { 2594, 0x845d }, { 2601, 0xc1e1 }, { 2608, 0x1061 },
1894 { 2612, 0x05a0 }, { 2616, 0x4400 }, { 2618, 0x0300 }, { 2620, 0x3221 },
1895 /* 0x7000 */
1896 { 2625, 0x20e1 }, { 2630, 0x0080 }, { 2631, 0x8009 }, { 2634, 0x1290 },
1897 { 2638, 0x4f18 }, { 2645, 0x6030 }, { 2649, 0x5030 }, { 2653, 0x4060 },
1898 { 2656, 0x0062 }, { 2659, 0x09f0 }, { 2665, 0x0810 }, { 2667, 0x0093 },
1899 { 2671, 0x0400 }, { 2672, 0x117a }, { 2679, 0x0010 }, { 2680, 0x0400 },
1900 /* 0x7100 */
1901 { 2681, 0x98f8 }, { 2689, 0x4000 }, { 2690, 0xa801 }, { 2694, 0x0103 },
1902 { 2697, 0x0ce2 }, { 2703, 0x5485 }, { 2709, 0x0101 }, { 2711, 0x0200 },
1903 { 2712, 0x10a1 }, { 2716, 0x0c04 }, { 2719, 0x8005 }, { 2722, 0x840d },
1904 { 2727, 0x1813 }, { 2732, 0x1648 }, { 2737, 0x0000 }, { 2737, 0x4100 },
1905 /* 0x7200 */
1906 { 2739, 0x0381 }, { 2743, 0xa488 }, { 2748, 0x8810 }, { 2751, 0x0310 },
1907 { 2754, 0xc02e }, { 2760, 0x5469 }, { 2767, 0xc909 }, { 2773, 0x9982 },
1908 { 2779, 0x6210 }, { 2783, 0x0808 }, { 2785, 0x6100 }, { 2788, 0x4012 },
1909 { 2791, 0x1282 }, { 2795, 0x8160 }, { 2799, 0x0020 }, { 2800, 0x4c18 },
1910 /* 0x7300 */
1911 { 2805, 0x28b4 }, { 2811, 0x430c }, { 2816, 0x1194 }, { 2821, 0x2c26 },
1912 { 2827, 0x2008 }, { 2829, 0xe145 }, { 2836, 0xda1c }, { 2844, 0x1282 },
1913 { 2848, 0x406b }, { 2854, 0xd1a9 }, { 2862, 0x2c65 }, { 2869, 0xb2a0 },
1914 { 2875, 0x9a60 }, { 2881, 0x224c }, { 2886, 0x02ca }, { 2891, 0xaeb0 },

```

```
1915 /* 0x7400 */
1916 { 2899, 0x0493 }, { 2904, 0x0c02 }, { 2907, 0xff50 }, { 2917, 0x0203 },
1917 { 2920, 0x28d9 }, { 2927, 0x2086 }, { 2931, 0x69c4 }, { 2938, 0x0006 },
1918 { 2940, 0x82e3 }, { 2947, 0x9707 }, { 2955, 0xcf4b }, { 2965, 0x8a26 },
1919 { 2971, 0x1300 }, { 2974, 0xcd09 }, { 2981, 0x8d10 }, { 2986, 0x9c10 },
1920 /* 0x7500 */
1921 { 2991, 0x0040 }, { 2992, 0x00c4 }, { 2995, 0x8693 }, { 3002, 0xe240 },
1922 { 3007, 0x4189 }, { 3012, 0xc085 }, { 3017, 0x8002 }, { 3019, 0x7e02 },
1923 { 3026, 0x0022 }, { 3028, 0x122d }, { 3034, 0x0014 }, { 3036, 0x8410 },
1924 { 3039, 0xd053 }, { 3046, 0x9080 }, { 3049, 0xd093 }, { 3056, 0x0202 },
1925 /* 0x7600 */
1926 { 3058, 0x959d }, { 3067, 0x7a6c }, { 3076, 0x2268 }, { 3081, 0x172c },
1927 { 3088, 0xe3b }, { 3096, 0x8220 }, { 3099, 0xe030 }, { 3104, 0x0012 },
1928 { 3106, 0x3022 }, { 3110, 0xb820 }, { 3115, 0x25fd }, { 3125, 0x2000 },
1929 { 3126, 0x5a22 }, { 3132, 0x0210 }, { 3134, 0x1141 }, { 3138, 0x1243 },
1930 /* 0x7700 */
1931 { 3143, 0x4441 }, { 3147, 0x16b4 }, { 3154, 0xe104 }, { 3159, 0x6270 },
1932 { 3165, 0xe464 }, { 3172, 0xd0c4 }, { 3178, 0x1495 }, { 3184, 0x241d },
1933 { 3190, 0x3011 }, { 3194, 0x8470 }, { 3199, 0xc484 }, { 3204, 0x4022 },
1934 { 3207, 0x0208 }, { 3209, 0xc226 }, { 3215, 0x1451 }, { 3220, 0x0913 },
1935 /* 0x7800 */
1936 { 3225, 0x6260 }, { 3230, 0x2002 }, { 3232, 0x600e }, { 3237, 0x00a1 },
1937 { 3240, 0x5198 }, { 3246, 0x5004 }, { 3249, 0x451b }, { 3256, 0x4400 },
1938 { 3258, 0x8400 }, { 3260, 0xe110 }, { 3265, 0x3112 }, { 3270, 0xa80f },
1939 { 3277, 0x5380 }, { 3282, 0x886c }, { 3288, 0x0453 }, { 3293, 0x8ccc },
1940 /* 0x7900 */
1941 { 3300, 0x1041 }, { 3303, 0xd401 }, { 3308, 0x22a1 }, { 3313, 0xa832 },
1942 { 3319, 0x8c70 }, { 3325, 0x1912 }, { 3330, 0x0a80 }, { 3333, 0x5a04 },
1943 { 3338, 0x1800 }, { 3340, 0x197a }, { 3348, 0x8b02 }, { 3353, 0x0912 },
1944 { 3357, 0x8594 }, { 3363, 0x6450 }, { 3368, 0x2c25 }, { 3374, 0x1102 },
1945 /* 0x7a00 */
1946 { 3377, 0x168c }, { 3383, 0x4822 }, { 3387, 0xa882 }, { 3392, 0x0731 },
1947 { 3398, 0x11b0 }, { 3403, 0xb260 }, { 3409, 0x24a1 }, { 3414, 0x4120 },
1948 { 3417, 0x0c65 }, { 3423, 0x4013 }, { 3427, 0x1009 }, { 3430, 0x1a28 },
1949 { 3435, 0x5240 }, { 3439, 0x0802 }, { 3441, 0x1b00 }, { 3445, 0x6812 },
1950 /* 0x7b00 */
1951 { 3450, 0x0080 }, { 3451, 0x8010 }, { 3453, 0xee88 }, { 3461, 0xa013 },
1952 { 3466, 0x4083 }, { 3470, 0x0020 }, { 3471, 0xa651 }, { 3478, 0x008c },
1953 { 3481, 0x4210 }, { 3484, 0x4843 }, { 3489, 0x9021 }, { 3493, 0x3c65 },
1954 { 3501, 0x0524 }, { 3505, 0x0ed0 }, { 3511, 0x0500 }, { 3513, 0x5734 },
1955 /* 0x7c00 */
1956 { 3521, 0xda5e }, { 3531, 0x0a00 }, { 3533, 0x1161 }, { 3538, 0x065a },
1957 { 3544, 0x0440 }, { 3546, 0x7e2e }, { 3556, 0x628a }, { 3562, 0x3205 },
1958 { 3567, 0x80c0 }, { 3570, 0x4010 }, { 3572, 0x0041 }, { 3574, 0x9cc1 },
1959 { 3581, 0xa390 }, { 3587, 0x26b8 }, { 3594, 0x0a40 }, { 3597, 0x0020 },
1960 /* 0x7d00 */
1961 { 3598, 0x8388 }, { 3603, 0x604e }, { 3609, 0x2448 }, { 3613, 0x7002 },
1962 { 3617, 0x2183 }, { 3622, 0x368a }, { 3629, 0x04a0 }, { 3632, 0x8d01 },
1963 { 3637, 0x396e }, { 3646, 0x60c2 }, { 3651, 0x04c0 }, { 3654, 0x02c8 },
1964 { 3658, 0x707c }, { 3666, 0x0280 }, { 3668, 0x2c64 }, { 3674, 0x0662 },
1965 /* 0x7e00 */
1966 { 3679, 0x0101 }, { 3681, 0x30a3 }, { 3687, 0xb181 }, { 3693, 0x8048 },
1967 { 3696, 0x40b0 }, { 3700, 0x8105 }, { 3704, 0xc826 }, { 3710, 0x4108 },
1968 { 3713, 0x24c2 }, { 3718, 0x6522 }, { 3724, 0x0000 }, { 3724, 0x0000 },
1969 { 3724, 0x0000 }, { 3724, 0x0000 }, { 3724, 0x0000 }, { 3724, 0x0000 },
1970 /* 0x7f00 */
1971 { 3724, 0x0000 }, { 3724, 0x0000 }, { 3724, 0x0000 }, { 3724, 0xf800 },
1972 { 3729, 0x8098 }, { 3733, 0x380c }, { 3738, 0x207a }, { 3744, 0xe002 },
1973 { 3748, 0xa801 }, { 3752, 0x10c3 }, { 3757, 0x244e }, { 3762, 0x9010 },
1974 { 3765, 0xc109 }, { 3770, 0x8800 }, { 3772, 0xd128 }, { 3778, 0xe404 },
1975 /* 0x8000 */
1976 { 3783, 0xe580 }, { 3789, 0xe05a }, { 3796, 0x5051 }, { 3801, 0x56b1 },
1977 { 3809, 0x0011 }, { 3811, 0x0000 }, { 3811, 0x2051 }, { 3815, 0x0022 },
1978 { 3817, 0x4102 }, { 3820, 0x5000 }, { 3822, 0x08c0 }, { 3825, 0x0300 },
1979 { 3827, 0xa100 }, { 3830, 0x01b4 }, { 3835, 0x6001 }, { 3838, 0x464d },
1980 /* 0x8100 */
1981 { 3845, 0x0808 }, { 3847, 0x51c0 }, { 3852, 0x1091 }, { 3856, 0x1421 },
1982 { 3860, 0x14a0 }, { 3864, 0x0084 }, { 3866, 0xa383 }, { 3873, 0x0080 },
1983 { 3874, 0x4872 }, { 3880, 0x4941 }, { 3885, 0x4004 }, { 3887, 0x0814 },
1984 { 3890, 0xcc28 }, { 3896, 0x68a0 }, { 3901, 0x1812 }, { 3905, 0xa367 },
1985 /* 0x8200 */
1986 { 3914, 0x8009 }, { 3917, 0x2618 }, { 3922, 0x0106 }, { 3925, 0x0414 },
1987 { 3928, 0xc878 }, { 3935, 0x1042 }, { 3938, 0x2089 }, { 3942, 0xa810 },
1988 { 3946, 0x469b }, { 3954, 0x0d52 }, { 3960, 0x479b }, { 3969, 0xd495 },
1989 { 3977, 0x0040 }, { 3978, 0x0421 }, { 3981, 0xa515 }, { 3988, 0x60c0 },
1990 /* 0x8300 */
1991 { 3992, 0x0d83 }, { 3998, 0xe800 }, { 4002, 0x7006 }, { 4007, 0x3489 },
1992 { 4013, 0x609c }, { 4019, 0x00fa }, { 4025, 0x0000 }, { 4025, 0xa101 },
1993 { 4029, 0x2055 }, { 4034, 0x3b34 }, { 4042, 0x32c0 }, { 4047, 0xc000 },
1994 { 4049, 0x8281 }, { 4053, 0x2013 }, { 4057, 0x0500 }, { 4059, 0x1340 },
1995 /* 0x8400 */
1996 { 4063, 0x8442 }, { 4067, 0x0222 }, { 4070, 0x8000 }, { 4071, 0x0200 },
1997 { 4072, 0xa5a0 }, { 4078, 0x1746 }, { 4085, 0x04b1 }, { 4090, 0x3159 },
1998 { 4097, 0x0022 }, { 4099, 0x402c }, { 4103, 0x8740 }, { 4108, 0x6412 },
1999 { 4113, 0x9185 }, { 4119, 0x1008 }, { 4121, 0x8480 }, { 4124, 0x2c87 },
2000 /* 0x8500 */
2001 { 4131, 0x508c }, { 4136, 0x5001 }, { 4139, 0x8cbc }, { 4147, 0x805c },
```

```

2002 { 4152, 0x8040 }, { 4154, 0xf24f }, { 4164, 0x8817 }, { 4170, 0xae00 },
2003 { 4175, 0x9a62 }, { 4182, 0xa108 }, { 4186, 0x20a5 }, { 4191, 0xf1d0 },
2004 { 4199, 0x4c84 }, { 4204, 0x8500 }, { 4207, 0x2141 }, { 4211, 0x9048 },
2005 /* 0x8600 */
2006 { 4215, 0x6031 }, { 4220, 0x4b07 }, { 4227, 0x0282 }, { 4230, 0x3540 },
2007 { 4235, 0x0047 }, { 4239, 0x23cc }, { 4246, 0x921f }, { 4254, 0x04e0 },
2008 { 4258, 0x2100 }, { 4260, 0x1542 }, { 4265, 0x21c2 }, { 4270, 0x83ba },
2009 { 4278, 0x002b }, { 4282, 0x14a6 }, { 4288, 0x00a9 }, { 4292, 0x3400 },
2010 /* 0x8700 */
2011 { 4295, 0xc8b0 }, { 4301, 0xc219 }, { 4307, 0xc10a }, { 4312, 0x7606 },
2012 { 4319, 0x2029 }, { 4323, 0x2100 }, { 4325, 0x8032 }, { 4329, 0x0806 },
2013 { 4332, 0x1bf8 }, { 4341, 0x43a9 }, { 4348, 0x7089 }, { 4354, 0xc022 },
2014 { 4358, 0x4702 }, { 4363, 0x9660 }, { 4369, 0x2c1c }, { 4375, 0x850a },
2015 /* 0x8800 */
2016 { 4380, 0x0e4a }, { 4386, 0xdf1d }, { 4397, 0x6100 }, { 4400, 0x1425 },
2017 { 4405, 0x4f2a }, { 4413, 0x9562 }, { 4420, 0x0211 }, { 4423, 0x0a02 },
2018 { 4426, 0x0001 }, { 4427, 0x9d00 }, { 4432, 0x0501 }, { 4435, 0x6400 },
2019 { 4438, 0x7c01 }, { 4444, 0x480e }, { 4449, 0x8080 }, { 4451, 0x00a3 },
2020 /* 0x8900 */
2021 { 4455, 0xe042 }, { 4460, 0x1760 }, { 4466, 0x01c1 }, { 4470, 0x4627 },
2022 { 4477, 0x8265 }, { 4483, 0x1c84 }, { 4488, 0x480e }, { 4493, 0x3c29 },
2023 { 4500, 0x2200 }, { 4502, 0x9831 }, { 4508, 0x0021 }, { 4510, 0x10f1 },
2024 { 4516, 0x0000 }, { 4516, 0x01f0 }, { 4521, 0x2a20 }, { 4525, 0xa24a },
2025 /* 0x8a00 */
2026 { 4531, 0x80b0 }, { 4535, 0x4036 }, { 4540, 0x9855 }, { 4547, 0x60a0 },
2027 { 4551, 0x62a9 }, { 4558, 0x31c8 }, { 4564, 0x00a2 }, { 4567, 0xcee0 },
2028 { 4575, 0x88a9 }, { 4580, 0x82c5 }, { 4586, 0xc280 }, { 4590, 0x48c8 },
2029 { 4595, 0x0748 }, { 4600, 0xa0ba }, { 4607, 0x1000 }, { 4608, 0x9071 },
2030 /* 0x8b00 */
2031 { 4614, 0x0c60 }, { 4618, 0xd002 }, { 4622, 0x2000 }, { 4623, 0x1081 },
2032 { 4626, 0x217c }, { 4633, 0x421c }, { 4638, 0x2008 }, { 4640, 0x5340 },
2033 { 4645, 0xa832 }, { 4651, 0xd030 }, { 4656, 0x0000 }, { 4656, 0x0000 },
2034 { 4656, 0x0000 }, { 4656, 0x0000 }, { 4656, 0x0000 }, { 4656, 0x0000 },
2035 /* 0x8c00 */
2036 { 4656, 0x0000 }, { 4656, 0x0000 }, { 4656, 0x0000 }, { 4656, 0x6300 },
2037 { 4660, 0x8aa0 }, { 4665, 0x2b9a }, { 4673, 0x2358 }, { 4679, 0x4868 },
2038 { 4684, 0x08c0 }, { 4687, 0x1a0d }, { 4693, 0x0010 }, { 4694, 0x0600 },
2039 { 4696, 0x8a60 }, { 4701, 0x2260 }, { 4705, 0x9102 }, { 4709, 0xc1a5 },
2040 /* 0x8d00 */
2041 { 4716, 0x020a }, { 4719, 0x0884 }, { 4722, 0x0000 }, { 4722, 0x0000 },
2042 { 4722, 0x0000 }, { 4722, 0x0000 }, { 4722, 0x5220 }, { 4726, 0x8000 },
2043 { 4727, 0x2114 }, { 4731, 0xc023 }, { 4736, 0x9841 }, { 4741, 0x1aa4 },
2044 { 4747, 0x45e1 }, { 4754, 0x02b2 }, { 4759, 0x10b0 }, { 4763, 0x2017 },
2045 /* 0x8e00 */
2046 { 4768, 0x0872 }, { 4773, 0x0052 }, { 4776, 0x00cf }, { 4782, 0x23ca },
2047 { 4789, 0xe803 }, { 4795, 0x7810 }, { 4800, 0xb206 }, { 4806, 0xe003 },
2048 { 4811, 0x020c }, { 4814, 0x6c25 }, { 4821, 0x6284 }, { 4826, 0x0c28 },
2049 { 4830, 0x809b }, { 4836, 0x1012 }, { 4839, 0x6100 }, { 4842, 0x0683 },
2050 /* 0x8f00 */
2051 { 4847, 0x8185 }, { 4852, 0x41c1 }, { 4857, 0x71ab }, { 4866, 0x04f0 },
2052 { 4871, 0x808b }, { 4876, 0x613e }, { 4884, 0x0020 }, { 4885, 0x0000 },
2053 { 4885, 0x0000 }, { 4885, 0x2000 }, { 4886, 0x0073 }, { 4891, 0x4160 },
2054 { 4895, 0x2c43 }, { 4901, 0x002d }, { 4905, 0x4119 }, { 4910, 0x4862 },
2055 /* 0x9000 */
2056 { 4915, 0x1114 }, { 4919, 0x0900 }, { 4921, 0xb700 }, { 4927, 0x8098 },
2057 { 4931, 0x1018 }, { 4934, 0x2800 }, { 4936, 0x10c4 }, { 4940, 0x0211 },
2058 { 4943, 0x5920 }, { 4948, 0x0ba1 }, { 4954, 0x0027 }, { 4958, 0x605d },
2059 { 4965, 0x11b8 }, { 4971, 0xb3a4 }, { 4979, 0x8820 }, { 4982, 0xc051 },
2060 /* 0x9100 */
2061 { 4987, 0x2171 }, { 4993, 0x55d1 }, { 5001, 0xc2ad }, { 5009, 0x36d2 },
2062 { 5017, 0x8188 }, { 5021, 0x0e88 }, { 5026, 0x2092 }, { 5030, 0x0e10 },
2063 { 5034, 0x446a }, { 5040, 0x413a }, { 5046, 0x7142 }, { 5052, 0xb84f },
2064 { 5061, 0x002c }, { 5064, 0x4698 }, { 5070, 0xf630 }, { 5078, 0x2a83 },
2065 /* 0x9200 */
2066 { 5084, 0x16f3 }, { 5093, 0x314d }, { 5100, 0xc178 }, { 5107, 0x5769 },
2067 { 5116, 0xe4cd }, { 5125, 0x3302 }, { 5130, 0xc3a3 }, { 5138, 0xbbe1 },
2068 { 5148, 0x6700 }, { 5153, 0x8284 }, { 5157, 0x89b1 }, { 5164, 0xbd44 },
2069 { 5172, 0x79ef }, { 5184, 0xb3a9 }, { 5193, 0x51ab }, { 5201, 0x8a01 },
2070 /* 0x9300 */
2071 { 5205, 0x2105 }, { 5209, 0xf032 }, { 5216, 0x06b2 }, { 5222, 0x00d8 },
2072 { 5226, 0x0380 }, { 5229, 0x45a7 }, { 5237, 0xa6b0 }, { 5244, 0xa45b },
2073 { 5252, 0xad07 }, { 5260, 0x4924 }, { 5265, 0x0b5a }, { 5272, 0x0470 },
2074 { 5276, 0x3ef2 }, { 5286, 0xd208 }, { 5291, 0x00c4 }, { 5294, 0x2f80 },
2075 /* 0x9400 */
2076 { 5300, 0xe316 }, { 5308, 0x80e0 }, { 5312, 0xc000 }, { 5314, 0xa81e },
2077 { 5321, 0x1528 }, { 5326, 0x9220 }, { 5330, 0xe90a }, { 5337, 0x0006 },
2078 { 5339, 0x0018 }, { 5341, 0x0000 }, { 5341, 0x0000 }, { 5341, 0x0000 },
2079 { 5341, 0x0000 }, { 5341, 0x0000 }, { 5341, 0x0000 }, { 5341, 0x0000 },
2080 /* 0x9500 */
2081 { 5341, 0x0000 }, { 5341, 0x0000 }, { 5341, 0x0000 }, { 5341, 0x0000 },
2082 { 5341, 0x0000 }, { 5341, 0x0000 }, { 5341, 0x0000 }, { 5341, 0x4300 },
2083 { 5344, 0x7110 }, { 5349, 0xe000 }, { 5352, 0x1a42 }, { 5357, 0xa450 },
2084 { 5362, 0x0b40 }, { 5366, 0xe60f }, { 5375, 0x0051 }, { 5378, 0x0000 },
2085 /* 0x9600 */
2086 { 5378, 0x0000 }, { 5378, 0x6000 }, { 5380, 0x1074 }, { 5385, 0x378a },
2087 { 5393, 0x0002 }, { 5394, 0x01d4 }, { 5399, 0x4002 }, { 5401, 0xd810 },
2088 { 5406, 0x021e }, { 5411, 0xa442 }, { 5416, 0xc270 }, { 5422, 0x0408 },

```

```

2089 { 5424, 0x0400 }, { 5425, 0xe504 }, { 5431, 0x8200 }, { 5433, 0x0402 },
2090 /* 0x9700 */
2091 { 5435, 0x022c }, { 5439, 0x2c00 }, { 5442, 0x010e }, { 5446, 0x000a },
2092 { 5448, 0xc40a }, { 5453, 0x0da0 }, { 5458, 0x4488 }, { 5462, 0xa9c8 },
2093 { 5469, 0x0201 }, { 5471, 0xc6e0 }, { 5478, 0x5004 }, { 5481, 0xd766 },
2094 { 5491, 0x76b2 }, { 5500, 0x6b93 }, { 5509, 0x8013 }, { 5513, 0x0592 },
2095 /* 0x9800 */
2096 { 5518, 0x6480 }, { 5522, 0x5250 }, { 5527, 0xc869 }, { 5534, 0x402d },
2097 { 5539, 0x0490 }, { 5542, 0x06ce }, { 5549, 0x146c }, { 5555, 0x0000 },
2098 { 5555, 0x0000 }, { 5555, 0x0000 }, { 5555, 0x6800 }, { 5558, 0x8d91 },
2099 { 5565, 0x1124 }, { 5569, 0x0000 }, { 5569, 0x04ea }, { 5575, 0x0048 },
2100 /* 0x9900 */
2101 { 5577, 0x0184 }, { 5580, 0x9ce2 }, { 5588, 0x08c4 }, { 5592, 0x1e3e },
2102 { 5601, 0x61c3 }, { 5608, 0xdb10 }, { 5615, 0x0001 }, { 5616, 0x0000 },
2103 { 5616, 0x0000 }, { 5616, 0xa800 }, { 5619, 0x0040 }, { 5620, 0xa627 },
2104 { 5628, 0x0208 }, { 5630, 0x5618 }, { 5636, 0x1c80 }, { 5640, 0x6231 },
2105 /* 0x9a00 */
2106 { 5646, 0x181c }, { 5651, 0x4043 }, { 5655, 0x609d }, { 5662, 0x0168 },
2107 { 5666, 0x5c92 }, { 5673, 0x2052 }, { 5677, 0x0000 }, { 5677, 0x0000 },
2108 { 5677, 0x0000 }, { 5677, 0x0000 }, { 5677, 0xd400 }, { 5681, 0xca74 },
2109 { 5689, 0x414a }, { 5694, 0x18e5 }, { 5701, 0x12b1 }, { 5707, 0xa62c },
2110 /* 0x9b00 */
2111 { 5714, 0x7b3f }, { 5726, 0x1a45 }, { 5732, 0x2841 }, { 5736, 0x26b8 },
2112 { 5743, 0x1900 }, { 5746, 0x48e0 }, { 5751, 0x7d6a }, { 5761, 0x83a8 },
2113 { 5767, 0xae11 }, { 5777, 0x6411 }, { 5782, 0x12c0 }, { 5786, 0xd987 },
2114 { 5795, 0x4182 }, { 5799, 0xa181 }, { 5804, 0x8ca0 }, { 5809, 0xa788 },
2115 /* 0x9c00 */
2116 { 5816, 0x8805 }, { 5820, 0x5742 }, { 5827, 0x07cc }, { 5834, 0x20e2 },
2117 { 5839, 0xc63a }, { 5847, 0xf959 }, { 5857, 0x4f08 }, { 5863, 0x08a5 },
2118 { 5868, 0x0000 }, { 5868, 0x0000 }, { 5868, 0x0000 }, { 5868, 0x0000 },
2119 { 5868, 0x0000 }, { 5868, 0x0000 }, { 5868, 0x0040 }, { 5869, 0x0284 },
2120 /* 0x9d00 */
2121 { 5872, 0x0804 }, { 5874, 0x7182 }, { 5880, 0x8000 }, { 5881, 0x341d },
2122 { 5888, 0x04ac }, { 5893, 0x8018 }, { 5896, 0x0e2c }, { 5902, 0x58c1 },
2123 { 5908, 0x6458 }, { 5914, 0x01ec }, { 5920, 0x5402 }, { 5924, 0x9222 },
2124 { 5929, 0x0688 }, { 5933, 0xc4f0 }, { 5940, 0x4aa1 }, { 5946, 0x4019 },
2125 /* 0x9e00 */
2126 { 5950, 0x4484 }, { 5954, 0x3267 }, { 5962, 0x0000 }, { 5962, 0x0000 },
2127 { 5962, 0x0000 }, { 5962, 0x0000 }, { 5962, 0x0000 }, { 5962, 0x1c00 },
2128 { 5965, 0xc0bd }, { 5973, 0x4940 }, { 5977, 0xd110 }, { 5982, 0x0039 },
2129 { 5986, 0x0940 }, { 5989, 0x8020 }, { 5991, 0x7090 }, { 5996, 0x8127 },
2130 /* 0x9f00 */
2131 { 6002, 0x820c }, { 6006, 0x8ed7 }, { 6016, 0x8c44 }, { 6021, 0xb696 },
2132 { 6030, 0x00fa }, { 6036, 0x65e8 }, { 6044, 0xe300 }, { 6049, 0x242b },
2133 { 6055, 0x8000 }, { 6056, 0x40d7 }, { 6063, 0x002e },
2134 };
2135
2136 static int
2137 jisx0212_wctomb (conv_t conv, unsigned char *, ucs4_t wc, int n)
2138 {
2139     if (n >= 2) {
2140         const Summary16 *summary = NULL;
2141         if (wc < 0x0460)
2142             summary = &jisx0212_uni2indx_page00[(wc>>4)];
2143         else if (wc >= 0x2100 && wc < 0x2130)
2144             summary = &jisx0212_uni2indx_page21[(wc>>4)-0x210];
2145         else if (wc >= 0x4e00 && wc < 0x9fb0)
2146             summary = &jisx0212_uni2indx_page4e[(wc>>4)-0x4e0];
2147         if (summary) {
2148             unsigned short used = summary->used;
2149             unsigned int i = wc & 0x0f;
2150             if (used & ((unsigned short) 1 << i)) {
2151                 unsigned short c;
2152                 /* Keep in 'used' only the bits 0..i-1. */
2153                 used &= ((unsigned short) 1 << i) - 1;
2154                 /* Add 'summary->indx' and the number of bits set in 'used'. */
2155                 used = (used & 0x5555) + ((used & 0xaaaa) >> 1);
2156                 used = (used & 0x3333) + ((used & 0xcccc) >> 2);
2157                 used = (used & 0x0f0f) + ((used & 0xf0f0) >> 4);
2158                 used = (used & 0x00ff) + (used >> 8);
2159                 c = jisx0212_2charset[summary->indx + used];
2160                 r[0] = (c >> 8); r[1] = (c & 0xff);
2161                 return 2;
2162             }
2163         }
2164         return RET_ILSEQ;
2165     }
2166     return RET_TOOSMALL;
2167 }
2168 #endif /* NEED_TOMB */

```

32.230 koi8_c.h

```

1 /* $XFree86: xc/lib/X11/lcUniConv/koi8_c.h,v 1.2 2000/11/28 16:10:29 dawes Exp $ */
2

```

```

3 /*
4  * KOI8-C
5 */
6
7 static const unsigned short koi8_c_2uni[128] = {
8     /* 0x80 */
9     0x0493, 0x0497, 0x049b, 0x049d, 0x04a3, 0x04af, 0x04b1, 0x04b3,
10    0x04b7, 0x04b9, 0x04bb, 0x2580, 0x04d9, 0x04e3, 0x04e9, 0x04ef,
11    /* 0x90 */
12    0x0492, 0x0496, 0x049a, 0x049c, 0x04a2, 0x04ae, 0x04b0, 0x04b2,
13    0x04b6, 0x04b8, 0x04ba, 0x2321, 0x04d8, 0x04e2, 0x04e8, 0x04ee,
14    /* 0xa0 */
15    0x00a0, 0x0452, 0x0453, 0x0451, 0x0454, 0x0455, 0x0456, 0x0457,
16    0x0458, 0x0459, 0x045a, 0x045b, 0x045c, 0x0491, 0x045e, 0x045f,
17    /* 0xb0 */
18    0x2116, 0x0402, 0x0403, 0x0401, 0x0404, 0x0405, 0x0406, 0x0407,
19    0x0486, 0x0409, 0x040a, 0x040b, 0x040c, 0x0490, 0x040e, 0x040f,
20    /* 0xc0 */
21    0x044e, 0x0430, 0x0431, 0x0446, 0x0434, 0x0435, 0x0444, 0x0433,
22    0x0445, 0x0438, 0x0439, 0x043a, 0x043b, 0x043c, 0x043d, 0x043e,
23    /* 0xd0 */
24    0x043f, 0x044f, 0x0440, 0x0441, 0x0442, 0x0443, 0x0436, 0x0432,
25    0x044c, 0x044b, 0x0437, 0x0448, 0x044d, 0x0449, 0x0447, 0x044a,
26    /* 0xe0 */
27    0x042e, 0x0410, 0x0411, 0x0426, 0x0414, 0x0415, 0x0424, 0x0413,
28    0x0425, 0x0418, 0x0419, 0x041a, 0x041b, 0x041c, 0x041d, 0x041e,
29    /* 0xf0 */
30    0x041f, 0x042f, 0x0420, 0x0421, 0x0422, 0x0423, 0x0416, 0x0412,
31    0x042c, 0x042b, 0x0417, 0x0428, 0x042d, 0x0429, 0x0427, 0x042a,
32 };
33
34 static int
35 koi8_c_mbtowc (conv_t conv, ucs4_t *pwc, const unsigned char *s, int n)
36 {
37     unsigned char c = *s;
38     if (c < 0x80)
39         *pwc = (ucs4_t) c;
40     else
41         *pwc = (ucs4_t) koi8_c_2uni[c-0x80];
42     return 1;
43 }
44
45 static const unsigned char koi8_c_page00[1] = {
46     0xa0, /* 0xa0-0xa7 */
47 };
48 static const unsigned char koi8_c_page04[240] = {
49     0x00, 0xb3, 0xb1, 0xb2, 0xb4, 0xb5, 0xb6, 0xb7, /* 0x00-0x07 */
50     0xb8, 0xb9, 0xba, 0xbb, 0xbc, 0x00, 0xbe, 0xbf, /* 0x08-0x0f */
51     0xe1, 0xe2, 0xf7, 0xe7, 0xe4, 0xe5, 0xf6, 0xfa, /* 0x10-0x17 */
52     0xe9, 0xea, 0xeb, 0xec, 0xed, 0xee, 0xef, 0xf0, /* 0x18-0x1f */
53     0xf2, 0xf3, 0xf4, 0xf5, 0xe6, 0xe8, 0xe3, 0xfe, /* 0x20-0x27 */
54     0xfb, 0xfd, 0xff, 0xf9, 0xf8, 0xfc, 0xe0, 0xf1, /* 0x28-0x2f */
55     0xc1, 0xc2, 0xd7, 0xc7, 0xc4, 0xc5, 0xd6, 0xda, /* 0x30-0x37 */
56     0xc9, 0xca, 0xcb, 0xcc, 0xcd, 0xce, 0xcf, 0xd0, /* 0x38-0x3f */
57     0xd2, 0xd3, 0xd4, 0xd5, 0xc6, 0xc8, 0xc3, 0xde, /* 0x40-0x47 */
58     0xdb, 0xdd, 0xdf, 0xd9, 0xd8, 0xdc, 0xc0, 0xd1, /* 0x48-0x4f */
59     0x00, 0xa3, 0xa1, 0xa2, 0xa4, 0xa5, 0xa6, 0xa7, /* 0x50-0x57 */
60     0xa8, 0xa9, 0xaa, 0xab, 0xac, 0x00, 0xae, 0xaf, /* 0x58-0x5f */
61     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x60-0x67 */
62     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x68-0x6f */
63     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x70-0x77 */
64     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x78-0x7f */
65     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x80-0x87 */
66     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x88-0x8f */
67     0xbd, 0xad, 0x90, 0x80, 0x00, 0x00, 0x91, 0x81, /* 0x90-0x97 */
68     0x00, 0x00, 0x92, 0x82, 0x93, 0x83, 0x00, 0x00, /* 0x98-0x9f */
69     0x00, 0x00, 0x94, 0x84, 0x00, 0x00, 0x00, 0x00, /* 0xa0-0xa7 */
70     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x95, 0x85, /* 0xa8-0xaf */
71     0x96, 0x86, 0x97, 0x87, 0x00, 0x00, 0x98, 0x88, /* 0xb0-0xb7 */
72     0x99, 0x89, 0x9a, 0x8a, 0x00, 0x00, 0x00, 0x00, /* 0xb8-0xbf */
73     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xc0-0xc7 */
74     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xc8-0xcf */
75     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xd0-0xd7 */
76     0x9c, 0x8c, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xd8-0xdf */
77     0x00, 0x00, 0x9d, 0x8d, 0x00, 0x00, 0x00, 0x00, /* 0xe0-0xe7 */
78     0x9e, 0x8e, 0x00, 0x00, 0x00, 0x00, 0x9f, 0x8f, /* 0xe8-0xef */
79 };
80 static const unsigned char koi8_c_page22[1] = {
81     0xb0, /* 0x16-0x16 */
82 };
83
84 static int
85 koi8_c_wctomb (conv_t conv, unsigned char *r, ucs4_t wc, int n)
86 {
87     unsigned char c = 0;
88     if (wc < 0x0080) {
89         *r = wc;

```



```

90     return 1;
91 }
92 else if (wc >= 0x00a0 && wc < 0x00a1)
93     c = koi8_c_page00[wc-0x00a0];
94 else if (wc >= 0x0400 && wc < 0x04ef)
95     c = koi8_c_page04[wc-0x0400];
96 else if (wc >= 0x2216 && wc < 0x2217)
97     c = koi8_c_page22[wc-0x2216];
98 if (c != 0) {
99     *r = c;
100    return 1;
101 }
102 return RET_ILSEQ;
103 }

```

32.231 koi8_r.h

```

1 /* $XFree86: xc/lib/X11/lcUniConv/koi8_r.h,v 1.3 2000/11/29 17:40:34 dawes Exp $ */
2
3 /*
4  * KOI8-R
5  */
6
7 /* Specification: RFC 1489 */
8
9 #ifdef NEED_TOWC
10 static const unsigned short koi8_r_2uni[128] = {
11     /* 0x80 */
12     0x2500, 0x2502, 0x250c, 0x2510, 0x2514, 0x2518, 0x251c, 0x2524,
13     0x252c, 0x2534, 0x253c, 0x2580, 0x2584, 0x2588, 0x258c, 0x2590,
14     /* 0x90 */
15     0x2591, 0x2592, 0x2593, 0x2320, 0x25a0, 0x2219, 0x221a, 0x2248,
16     0x2264, 0x2265, 0x00a0, 0x2321, 0x00b0, 0x00b2, 0x00b7, 0x00f7,
17     /* 0xa0 */
18     0x2550, 0x2551, 0x2552, 0x0451, 0x2553, 0x2554, 0x2555, 0x2556,
19     0x2557, 0x2558, 0x2559, 0x255a, 0x255b, 0x255c, 0x255d, 0x255e,
20     /* 0xb0 */
21     0x255f, 0x2560, 0x2561, 0x0401, 0x2562, 0x2563, 0x2564, 0x2565,
22     0x2566, 0x2567, 0x2568, 0x2569, 0x256a, 0x256b, 0x256c, 0x00a9,
23     /* 0xc0 */
24     0x044e, 0x0430, 0x0431, 0x0446, 0x0434, 0x0435, 0x0444, 0x0433,
25     0x0445, 0x0438, 0x0439, 0x043a, 0x043b, 0x043c, 0x043d, 0x043e,
26     /* 0xd0 */
27     0x043f, 0x044f, 0x0440, 0x0441, 0x0442, 0x0443, 0x0436, 0x0432,
28     0x044c, 0x044b, 0x0437, 0x0448, 0x044d, 0x0449, 0x0447, 0x044a,
29     /* 0xe0 */
30     0x042e, 0x0410, 0x0411, 0x0426, 0x0414, 0x0415, 0x0424, 0x0413,
31     0x0425, 0x0418, 0x0419, 0x041a, 0x041b, 0x041c, 0x041d, 0x041e,
32     /* 0xf0 */
33     0x041f, 0x042f, 0x0420, 0x0421, 0x0422, 0x0423, 0x0416, 0x0412,
34     0x042c, 0x042b, 0x0417, 0x0428, 0x042d, 0x0429, 0x0427, 0x042a,
35 };
36
37 static int
38 koi8_r_mbtowc (conv_t conv, ucs4_t *pwc, const unsigned char *s, int n)
39 {
40     unsigned char c = *s;
41     if (c < 0x80)
42         *pwc = (ucs4_t) c;
43     else
44         *pwc = (ucs4_t) koi8_r_2uni[c-0x80];
45     return 1;
46 }
47 #endif /* NEED_TOWC */
48
49 #ifdef NEED_TOMB
50 static const unsigned char koi8_r_page00[88] = {
51     0x9a, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xa0-0xa7 */
52     0x00, 0xbf, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xa8-0xaf */
53     0x9c, 0x00, 0x9d, 0x00, 0x00, 0x00, 0x00, 0x9e, /* 0xb0-0xb7 */
54     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xb8-0xbf */
55     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xc0-0xc7 */
56     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xc8-0xcf */
57     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xd0-0xd7 */
58     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xd8-0xdf */
59     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xe0-0xe7 */
60     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xe8-0xef */
61     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x9f, /* 0xf0-0xf7 */
62 };
63 static const unsigned char koi8_r_page04[88] = {
64     0x00, 0xb3, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x00-0x07 */
65     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x08-0x0f */
66     0xe1, 0xe2, 0xf7, 0xe7, 0xe4, 0xe5, 0xf6, 0xfa, /* 0x10-0x17 */
67     0xe9, 0xea, 0xeb, 0xec, 0xed, 0xee, 0xef, 0xf0, /* 0x18-0x1f */
68     0xf2, 0xf3, 0xf4, 0xf5, 0xe6, 0xe8, 0xe3, 0xfe, /* 0x20-0x27 */

```

```

69  0xfb, 0xfd, 0xff, 0xf9, 0xf8, 0xfc, 0xe0, 0xf1, /* 0x28-0x2f */
70  0xc1, 0xc2, 0xd7, 0xc7, 0xc4, 0xc5, 0xd6, 0xda, /* 0x30-0x37 */
71  0xc9, 0xca, 0xcb, 0xcc, 0xcd, 0xce, 0xcf, 0xd0, /* 0x38-0x3f */
72  0xd2, 0xd3, 0xd4, 0xd5, 0xc6, 0xc8, 0xc3, 0xde, /* 0x40-0x47 */
73  0xdb, 0xdd, 0xdf, 0xd9, 0xd8, 0xdc, 0xc0, 0xd1, /* 0x48-0x4f */
74  0x00, 0xa3, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x50-0x57 */
75 };
76 static const unsigned char koi8_r_page22[80] = {
77  0x00, 0x95, 0x96, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x18-0x1f */
78  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x20-0x27 */
79  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x28-0x2f */
80  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x30-0x37 */
81  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x38-0x3f */
82  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x40-0x47 */
83  0x97, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x48-0x4f */
84  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x50-0x57 */
85  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x58-0x5f */
86  0x00, 0x00, 0x00, 0x00, 0x98, 0x99, 0x00, 0x00, /* 0x60-0x67 */
87 };
88 static const unsigned char koi8_r_page23[8] = {
89  0x93, 0x9b, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x20-0x27 */
90 };
91 static const unsigned char koi8_r_page25[168] = {
92  0x80, 0x00, 0x81, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x00-0x07 */
93  0x00, 0x00, 0x00, 0x00, 0x82, 0x00, 0x00, 0x00, /* 0x08-0x0f */
94  0x83, 0x00, 0x00, 0x00, 0x84, 0x00, 0x00, 0x00, /* 0x10-0x17 */
95  0x85, 0x00, 0x00, 0x00, 0x86, 0x00, 0x00, 0x00, /* 0x18-0x1f */
96  0x00, 0x00, 0x00, 0x00, 0x87, 0x00, 0x00, 0x00, /* 0x20-0x27 */
97  0x00, 0x00, 0x00, 0x00, 0x88, 0x00, 0x00, 0x00, /* 0x28-0x2f */
98  0x00, 0x00, 0x00, 0x00, 0x89, 0x00, 0x00, 0x00, /* 0x30-0x37 */
99  0x00, 0x00, 0x00, 0x00, 0x8a, 0x00, 0x00, 0x00, /* 0x38-0x3f */
100  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x40-0x47 */
101  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x48-0x4f */
102  0xa0, 0xa1, 0xa2, 0xa4, 0xa5, 0xa6, 0xa7, 0xa8, /* 0x50-0x57 */
103  0xa9, 0xaa, 0xab, 0xac, 0xad, 0xae, 0xaf, 0xb0, /* 0x58-0x5f */
104  0xb1, 0xb2, 0xb4, 0xb5, 0xb6, 0xb7, 0xb8, 0xb9, /* 0x60-0x67 */
105  0xba, 0xbb, 0xbc, 0xbd, 0xbe, 0x00, 0x00, 0x00, /* 0x68-0x6f */
106  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x70-0x77 */
107  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x78-0x7f */
108  0x8b, 0x00, 0x00, 0x00, 0x8c, 0x00, 0x00, 0x00, /* 0x80-0x87 */
109  0x8d, 0x00, 0x00, 0x00, 0x8e, 0x00, 0x00, 0x00, /* 0x88-0x8f */
110  0x8f, 0x90, 0x91, 0x92, 0x00, 0x00, 0x00, 0x00, /* 0x90-0x97 */
111  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x98-0x9f */
112  0x94, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xa0-0xa7 */
113 };
114
115 static int
116 koi8_r_wctomb (conv_t conv, unsigned char *r, ucs4_t wc, int n)
117 {
118     unsigned char c = 0;
119     if (wc < 0x0080) {
120         *r = wc;
121         return 1;
122     }
123     else if (wc >= 0x00a0 && wc < 0x00f8)
124         c = koi8_r_page00[wc-0x00a0];
125     else if (wc >= 0x0400 && wc < 0x0458)
126         c = koi8_r_page04[wc-0x0400];
127     else if (wc >= 0x2218 && wc < 0x2268)
128         c = koi8_r_page22[wc-0x2218];
129     else if (wc >= 0x2320 && wc < 0x2328)
130         c = koi8_r_page23[wc-0x2320];
131     else if (wc >= 0x2500 && wc < 0x25a8)
132         c = koi8_r_page25[wc-0x2500];
133     if (c != 0) {
134         *r = c;
135         return 1;
136     }
137     return RET_ILSEQ;
138 }
139 #endif /* NEED_TOMB */

```

32.232 koi8_u.h

```

1 /* $XFree86: xc/lib/X11/lcUniConv/koi8_u.h,v 1.3 2000/11/29 17:40:34 dawes Exp $ */
2
3 /*
4  * KOI8-U
5  */
6
7 /* Specification: RFC 2319 */
8 #ifdef NEED_TOWC
9 static const unsigned short koi8_u_2uni[128] = {
10 /* 0x80 */
11  0x2500, 0x2502, 0x250c, 0x2510, 0x2514, 0x2518, 0x251c, 0x2524,

```

```
12 0x252c, 0x2534, 0x253c, 0x2580, 0x2584, 0x2588, 0x258c, 0x2590,
13 /* 0x90 */
14 0x2591, 0x2592, 0x2593, 0x2320, 0x25a0, 0x2219, 0x221a, 0x2248,
15 0x2264, 0x2265, 0x00a0, 0x2321, 0x00b0, 0x00b2, 0x00b7, 0x00f7,
16 /* 0xa0 */
17 0x2550, 0x2551, 0x2552, 0x0451, 0x0454, 0x2554, 0x0456, 0x0457,
18 0x2557, 0x2558, 0x2559, 0x255a, 0x255b, 0x0491, 0x255d, 0x255e,
19 /* 0xb0 */
20 0x255f, 0x2560, 0x2561, 0x0401, 0x0404, 0x2563, 0x0406, 0x0407,
21 0x2566, 0x2567, 0x2568, 0x2569, 0x256a, 0x0490, 0x256c, 0x00a9,
22 /* 0xc0 */
23 0x044e, 0x0430, 0x0431, 0x0446, 0x0434, 0x0435, 0x0444, 0x0433,
24 0x0445, 0x0438, 0x0439, 0x043a, 0x043b, 0x043c, 0x043d, 0x043e,
25 /* 0xd0 */
26 0x043f, 0x044f, 0x0440, 0x0441, 0x0442, 0x0443, 0x0436, 0x0432,
27 0x044c, 0x044b, 0x0437, 0x0448, 0x044d, 0x0449, 0x0447, 0x044a,
28 /* 0xe0 */
29 0x042e, 0x0410, 0x0411, 0x0426, 0x0414, 0x0415, 0x0424, 0x0413,
30 0x0425, 0x0418, 0x0419, 0x041a, 0x041b, 0x041c, 0x041d, 0x041e,
31 /* 0xf0 */
32 0x041f, 0x042f, 0x0420, 0x0421, 0x0422, 0x0423, 0x0416, 0x0412,
33 0x042c, 0x042b, 0x0417, 0x0428, 0x042d, 0x0429, 0x0427, 0x042a,
34 };
35
36 static int
37 koi8_u_mbtowc (conv_t conv, ucs4_t *pwc, const unsigned char *s, int n)
38 {
39     unsigned char c = *s;
40     if (c < 0x80)
41         *pwc = (ucs4_t) c;
42     else
43         *pwc = (ucs4_t) koi8_u_2uni[c-0x80];
44     return 1;
45 }
46 #endif /* NEED_TOWC */
47
48 #ifdef NEED_TOMB
49 static const unsigned char koi8_u_page00[88] = {
50 0x9a, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xa0-0xa7 */
51 0x00, 0xbf, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xa8-0xaf */
52 0x9c, 0x00, 0x9d, 0x00, 0x00, 0x00, 0x00, 0x9e, /* 0xb0-0xb7 */
53 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xb8-0xbf */
54 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xc0-0xc7 */
55 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xc8-0xcf */
56 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xd0-0xd7 */
57 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xd8-0xdf */
58 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xe0-0xe7 */
59 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xe8-0xef */
60 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x9f, /* 0xf0-0xf7 */
61 };
62 static const unsigned char koi8_u_page04[152] = {
63 0x00, 0xb3, 0x00, 0x00, 0xb4, 0x00, 0xb6, 0xb7, /* 0x00-0x07 */
64 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x08-0x0f */
65 0xe1, 0xe2, 0xf7, 0xe7, 0xe4, 0xe5, 0xf6, 0xfa, /* 0x10-0x17 */
66 0xe9, 0xea, 0xeb, 0xec, 0xed, 0xee, 0xef, 0xf0, /* 0x18-0x1f */
67 0xf2, 0xf3, 0xf4, 0xf5, 0xe6, 0xe8, 0xe3, 0xfe, /* 0x20-0x27 */
68 0xfb, 0xfd, 0xff, 0xf9, 0xf8, 0xfc, 0xe0, 0xf1, /* 0x28-0x2f */
69 0xc1, 0xc2, 0xd7, 0xc7, 0xc4, 0xc5, 0xd6, 0xda, /* 0x30-0x37 */
70 0xc9, 0xca, 0xcb, 0xcc, 0xcd, 0xce, 0xcf, 0xd0, /* 0x38-0x3f */
71 0xd2, 0xd3, 0xd4, 0xd5, 0xc6, 0xc8, 0xc3, 0xde, /* 0x40-0x47 */
72 0xdb, 0xdd, 0xdf, 0xd9, 0xd8, 0xdc, 0xc0, 0xd1, /* 0x48-0x4f */
73 0x00, 0xa3, 0x00, 0x00, 0xa4, 0x00, 0xa6, 0xa7, /* 0x50-0x57 */
74 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x58-0x5f */
75 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x60-0x67 */
76 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x68-0x6f */
77 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x70-0x77 */
78 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x78-0x7f */
79 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x80-0x87 */
80 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x88-0x8f */
81 0xbd, 0xad, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x90-0x97 */
82 };
83 static const unsigned char koi8_u_page22[80] = {
84 0x00, 0x95, 0x96, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x18-0x1f */
85 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x20-0x27 */
86 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x28-0x2f */
87 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x30-0x37 */
88 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x38-0x3f */
89 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x40-0x47 */
90 0x97, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x48-0x4f */
91 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x50-0x57 */
92 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x58-0x5f */
93 0x00, 0x00, 0x00, 0x00, 0x98, 0x99, 0x00, 0x00, /* 0x60-0x67 */
94 };
95 static const unsigned char koi8_u_page23[8] = {
96 0x93, 0x9b, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x20-0x27 */
97 };
98 static const unsigned char koi8_u_page25[168] = {
```

```

99 0x80, 0x00, 0x81, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x00-0x07 */
100 0x00, 0x00, 0x00, 0x00, 0x82, 0x00, 0x00, 0x00, /* 0x08-0x0f */
101 0x83, 0x00, 0x00, 0x00, 0x00, 0x84, 0x00, 0x00, 0x00, /* 0x10-0x17 */
102 0x85, 0x00, 0x00, 0x00, 0x86, 0x00, 0x00, 0x00, /* 0x18-0x1f */
103 0x00, 0x00, 0x00, 0x00, 0x87, 0x00, 0x00, 0x00, /* 0x20-0x27 */
104 0x00, 0x00, 0x00, 0x00, 0x88, 0x00, 0x00, 0x00, /* 0x28-0x2f */
105 0x00, 0x00, 0x00, 0x00, 0x89, 0x00, 0x00, 0x00, /* 0x30-0x37 */
106 0x00, 0x00, 0x00, 0x00, 0x8a, 0x00, 0x00, 0x00, /* 0x38-0x3f */
107 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x40-0x47 */
108 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x48-0x4f */
109 0xa0, 0xa1, 0xa2, 0x00, 0xa5, 0x00, 0x00, 0xa8, /* 0x50-0x57 */
110 0xa9, 0xaa, 0xab, 0xac, 0x00, 0xae, 0xaf, 0xb0, /* 0x58-0x5f */
111 0xb1, 0xb2, 0x00, 0xb5, 0x00, 0x00, 0xb8, 0xb9, /* 0x60-0x67 */
112 0xba, 0xbb, 0xbc, 0x00, 0xbe, 0x00, 0x00, 0x00, /* 0x68-0x6f */
113 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x70-0x77 */
114 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x78-0x7f */
115 0x8b, 0x00, 0x00, 0x00, 0x8c, 0x00, 0x00, 0x00, /* 0x80-0x87 */
116 0x8d, 0x00, 0x00, 0x00, 0x8e, 0x00, 0x00, 0x00, /* 0x88-0x8f */
117 0x8f, 0x90, 0x91, 0x92, 0x00, 0x00, 0x00, 0x00, /* 0x90-0x97 */
118 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x98-0x9f */
119 0x94, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xa0-0xa7 */
120 };
121
122 static int
123 koi8_u_wctomb (conv_t conv, unsigned char *r, ucs4_t wc, int n)
124 {
125     unsigned char c = 0;
126     if (wc < 0x0080) {
127         *r = wc;
128         return 1;
129     }
130     else if (wc >= 0x00a0 && wc < 0x00f8)
131         c = koi8_u_page00[wc-0x00a0];
132     else if (wc >= 0x0400 && wc < 0x0498)
133         c = koi8_u_page04[wc-0x0400];
134     else if (wc >= 0x2218 && wc < 0x2268)
135         c = koi8_u_page22[wc-0x2218];
136     else if (wc >= 0x2320 && wc < 0x2328)
137         c = koi8_u_page23[wc-0x2320];
138     else if (wc >= 0x2500 && wc < 0x25a8)
139         c = koi8_u_page25[wc-0x2500];
140     if (c != 0) {
141         *r = c;
142         return 1;
143     }
144     return RET_ILSEQ;
145 }
146 #endif /* NEED_TOMB */

```

32.233 ksc5601.h

```

1 /* $XFree86: xc/lib/X11/lcUniConv/ksc5601.h,v 1.5 2003/05/27 22:26:34 tsi Exp $ */
2
3 /*
4 * KSC5601.1987-0
5 */
6 #ifdef NEED_TOWC
7 static const unsigned short ksc5601_2uni_page21[1115] = {
8     /* 0x21 */
9     0x3000, 0x3001, 0x3002, 0x00b7, 0x2025, 0x2026, 0x00a8, 0x3003,
10     0x00ad, 0x2015, 0x2225, 0xff3c, 0x223c, 0x2018, 0x2019, 0x201c,
11     0x201d, 0x3014, 0x3015, 0x3008, 0x3009, 0x300a, 0x300b, 0x300c,
12     0x300d, 0x300e, 0x300f, 0x3010, 0x3011, 0x00b1, 0x00d7, 0x00f7,
13     0x2260, 0x2264, 0x2265, 0x221e, 0x2234, 0x00b0, 0x2032, 0x2033,
14     0x2103, 0x212b, 0xffe0, 0xffe1, 0xffe5, 0x2642, 0x2640, 0x2220,
15     0x22a5, 0x2312, 0x2202, 0x2207, 0x2261, 0x2252, 0x00a7, 0x203b,
16     0x2606, 0x2605, 0x25cb, 0x25cf, 0x25ce, 0x25c7, 0x25c6, 0x25a1,
17     0x25a0, 0x25b3, 0x25b2, 0x25bd, 0x25bc, 0x2192, 0x2190, 0x2191,
18     0x2193, 0x2194, 0x3013, 0x226a, 0x226b, 0x221a, 0x223d, 0x221d,
19     0x2235, 0x222b, 0x222c, 0x2208, 0x220b, 0x2286, 0x2287, 0x2282,
20     0x2283, 0x222a, 0x2229, 0x2227, 0x2228, 0xffe2,
21     /* 0x22 */
22     0x21d2, 0x21d4, 0x2200, 0x2203, 0x00b4, 0xff5e, 0x02c7, 0x02d8,
23     0x02dd, 0x02da, 0x02d9, 0x00b8, 0x02db, 0x00a1, 0x00bf, 0x02d0,
24     0x222e, 0x2211, 0x220f, 0x00a4, 0x2109, 0x2030, 0x25c1, 0x25c0,
25     0x25b7, 0x25b6, 0x2664, 0x2660, 0x2661, 0x2665, 0x2667, 0x2663,
26     0x2299, 0x25c8, 0x25a3, 0x25d0, 0x25d1, 0x2592, 0x25a4, 0x25a5,
27     0x25a8, 0x25a7, 0x25a6, 0x25a9, 0x2668, 0x260f, 0x260e, 0x261c,
28     0x261e, 0x00b6, 0x2020, 0x2021, 0x2195, 0x2197, 0x2199, 0x2196,
29     0x2198, 0x266d, 0x2669, 0x266a, 0x266c, 0x327f, 0x321c, 0x2116,
30     0x33c7, 0x2122, 0x33c2, 0x33d8, 0x2121, 0xffff, 0xffff, 0xffff,
31     0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
32     0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
33     0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
34     /* 0x23 */

```

```
35 0xff01, 0xff02, 0xff03, 0xff04, 0xff05, 0xff06, 0xff07, 0xff08,
36 0xff09, 0xff0a, 0xff0b, 0xff0c, 0xff0d, 0xff0e, 0xff0f, 0xff10,
37 0xff11, 0xff12, 0xff13, 0xff14, 0xff15, 0xff16, 0xff17, 0xff18,
38 0xff19, 0xff1a, 0xff1b, 0xff1c, 0xff1d, 0xff1e, 0xff1f, 0xff20,
39 0xff21, 0xff22, 0xff23, 0xff24, 0xff25, 0xff26, 0xff27, 0xff28,
40 0xff29, 0xff2a, 0xff2b, 0xff2c, 0xff2d, 0xff2e, 0xff2f, 0xff30,
41 0xff31, 0xff32, 0xff33, 0xff34, 0xff35, 0xff36, 0xff37, 0xff38,
42 0xff39, 0xff3a, 0xff3b, 0xff3c, 0xff3d, 0xff3e, 0xff3f, 0xff40,
43 0xff41, 0xff42, 0xff43, 0xff44, 0xff45, 0xff46, 0xff47, 0xff48,
44 0xff49, 0xff4a, 0xff4b, 0xff4c, 0xff4d, 0xff4e, 0xff4f, 0xff50,
45 0xff51, 0xff52, 0xff53, 0xff54, 0xff55, 0xff56, 0xff57, 0xff58,
46 0xff59, 0xff5a, 0xff5b, 0xff5c, 0xff5d, 0xff5e, 0xff5f,
47 /* 0x24 */
48 0x3131, 0x3132, 0x3133, 0x3134, 0x3135, 0x3136, 0x3137, 0x3138,
49 0x3139, 0x313a, 0x313b, 0x313c, 0x313d, 0x313e, 0x313f, 0x3140,
50 0x3141, 0x3142, 0x3143, 0x3144, 0x3145, 0x3146, 0x3147, 0x3148,
51 0x3149, 0x314a, 0x314b, 0x314c, 0x314d, 0x314e, 0x314f, 0x3150,
52 0x3151, 0x3152, 0x3153, 0x3154, 0x3155, 0x3156, 0x3157, 0x3158,
53 0x3159, 0x315a, 0x315b, 0x315c, 0x315d, 0x315e, 0x315f, 0x3160,
54 0x3161, 0x3162, 0x3163, 0x3164, 0x3165, 0x3166, 0x3167, 0x3168,
55 0x3169, 0x316a, 0x316b, 0x316c, 0x316d, 0x316e, 0x316f, 0x3170,
56 0x3171, 0x3172, 0x3173, 0x3174, 0x3175, 0x3176, 0x3177, 0x3178,
57 0x3179, 0x317a, 0x317b, 0x317c, 0x317d, 0x317e, 0x317f, 0x3180,
58 0x3181, 0x3182, 0x3183, 0x3184, 0x3185, 0x3186, 0x3187, 0x3188,
59 0x3189, 0x318a, 0x318b, 0x318c, 0x318d, 0x318e,
60 /* 0x25 */
61 0x2170, 0x2171, 0x2172, 0x2173, 0x2174, 0x2175, 0x2176, 0x2177,
62 0x2178, 0x2179, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x2160,
63 0x2161, 0x2162, 0x2163, 0x2164, 0x2165, 0x2166, 0x2167, 0x2168,
64 0x2169, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
65 0x0391, 0x0392, 0x0393, 0x0394, 0x0395, 0x0396, 0x0397, 0x0398,
66 0x0399, 0x039a, 0x039b, 0x039c, 0x039d, 0x039e, 0x039f, 0x03a0,
67 0x03a1, 0x03a2, 0x03a3, 0x03a4, 0x03a5, 0x03a6, 0x03a7, 0x03a8, 0x03a9,
68 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
69 0x03b1, 0x03b2, 0x03b3, 0x03b4, 0x03b5, 0x03b6, 0x03b7, 0x03b8,
70 0x03b9, 0x03ba, 0x03bb, 0x03bc, 0x03bd, 0x03be, 0x03bf, 0x03c0,
71 0x03c1, 0x03c2, 0x03c3, 0x03c4, 0x03c5, 0x03c6, 0x03c7, 0x03c8, 0x03c9,
72 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
73 /* 0x26 */
74 0x2500, 0x2502, 0x250c, 0x2510, 0x2518, 0x2514, 0x251c, 0x252c,
75 0x2524, 0x2534, 0x253c, 0x2501, 0x2503, 0x250f, 0x2513, 0x251b,
76 0x2517, 0x2523, 0x2533, 0x252b, 0x2525, 0x254b, 0x2520, 0x252f,
77 0x2528, 0x2537, 0x253f, 0x251d, 0x2530, 0x2525, 0x2538, 0x2542,
78 0x2512, 0x2511, 0x251a, 0x2519, 0x2516, 0x2515, 0x250e, 0x250d,
79 0x251e, 0x251f, 0x2521, 0x2522, 0x2526, 0x2527, 0x2529, 0x252a,
80 0x252d, 0x252e, 0x2531, 0x2532, 0x2535, 0x2536, 0x2539, 0x253a,
81 0x253d, 0x253e, 0x2540, 0x2541, 0x2543, 0x2544, 0x2545, 0x2546,
82 0x2547, 0x2548, 0x2549, 0x254a, 0xffff, 0xffff, 0xffff, 0xffff,
83 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
84 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
85 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
86 /* 0x27 */
87 0x3395, 0x3396, 0x3397, 0x2113, 0x3398, 0x33c4, 0x33a3, 0x33a4,
88 0x33a5, 0x33a6, 0x3399, 0x339a, 0x339b, 0x339c, 0x339d, 0x339e,
89 0x339f, 0x33a0, 0x33a1, 0x33a2, 0x33ca, 0x338d, 0x338e, 0x338f,
90 0x33cf, 0x3388, 0x3389, 0x33c8, 0x33a7, 0x33a8, 0x33b0, 0x33b1,
91 0x33b2, 0x33b3, 0x33b4, 0x33b5, 0x33b6, 0x33b7, 0x33b8, 0x33b9,
92 0x3380, 0x3381, 0x3382, 0x3383, 0x3384, 0x33ba, 0x33bb, 0x33bc,
93 0x33bd, 0x33be, 0x33bf, 0x3390, 0x3391, 0x3392, 0x3393, 0x3394,
94 0x2126, 0x33c0, 0x33c1, 0x338a, 0x338b, 0x338c, 0x33d6, 0x33c5,
95 0x33ad, 0x33ae, 0x33af, 0x33db, 0x33a9, 0x33aa, 0x33ab, 0x33ac,
96 0x33dd, 0x33d0, 0x33d3, 0x33c3, 0x33c9, 0x33dc, 0x33c6, 0xffff,
97 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
98 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
99 /* 0x28 */
100 0x00c6, 0x00d0, 0x00aa, 0x0126, 0xffff, 0x0132, 0xffff, 0x013f,
101 0x0141, 0x00d8, 0x0152, 0x00ba, 0x00de, 0x0166, 0x014a, 0xffff,
102 0x3260, 0x3261, 0x3262, 0x3263, 0x3264, 0x3265, 0x3266, 0x3267,
103 0x3268, 0x3269, 0x326a, 0x326b, 0x326c, 0x326d, 0x326e, 0x326f,
104 0x3270, 0x3271, 0x3272, 0x3273, 0x3274, 0x3275, 0x3276, 0x3277,
105 0x3278, 0x3279, 0x327a, 0x327b, 0x24d0, 0x24d1, 0x24d2, 0x24d3,
106 0x24d4, 0x24d5, 0x24d6, 0x24d7, 0x24d8, 0x24d9, 0x24da, 0x24db,
107 0x24dc, 0x24dd, 0x24de, 0x24df, 0x24e0, 0x24e1, 0x24e2, 0x24e3,
108 0x24e4, 0x24e5, 0x24e6, 0x24e7, 0x24e8, 0x24e9, 0x2460, 0x2461,
109 0x2462, 0x2463, 0x2464, 0x2465, 0x2466, 0x2467, 0x2468, 0x2469,
110 0x246a, 0x246b, 0x246c, 0x246d, 0x246e, 0x00bd, 0x2153, 0x2154,
111 0x00bc, 0x00be, 0x215b, 0x215c, 0x215d, 0x215e,
112 /* 0x29 */
113 0x00e6, 0x0111, 0x00f0, 0x0127, 0x0131, 0x0133, 0x0138, 0x0140,
114 0x0142, 0x00f8, 0x0153, 0x00df, 0x00fe, 0x0167, 0x014b, 0x0149,
115 0x3200, 0x3201, 0x3202, 0x3203, 0x3204, 0x3205, 0x3206, 0x3207,
116 0x3208, 0x3209, 0x320a, 0x320b, 0x320c, 0x320d, 0x320e, 0x320f,
117 0x3210, 0x3211, 0x3212, 0x3213, 0x3214, 0x3215, 0x3216, 0x3217,
118 0x3218, 0x3219, 0x321a, 0x321b, 0x249c, 0x249d, 0x249e, 0x249f,
119 0x24a0, 0x24a1, 0x24a2, 0x24a3, 0x24a4, 0x24a5, 0x24a6, 0x24a7,
120 0x24a8, 0x24a9, 0x24aa, 0x24ab, 0x24ac, 0x24ad, 0x24ae, 0x24af,
121 0x24b0, 0x24b1, 0x24b2, 0x24b3, 0x24b4, 0x24b5, 0x2474, 0x2475,
```

```

122 0x2476, 0x2477, 0x2478, 0x2479, 0x247a, 0x247b, 0x247c, 0x247d,
123 0x247e, 0x247f, 0x2480, 0x2481, 0x2482, 0x00b9, 0x00b2, 0x00b3,
124 0x2074, 0x207f, 0x2081, 0x2082, 0x2083, 0x2084,
125 /* 0x2a */
126 0x3041, 0x3042, 0x3043, 0x3044, 0x3045, 0x3046, 0x3047, 0x3048,
127 0x3049, 0x304a, 0x304b, 0x304c, 0x304d, 0x304e, 0x304f, 0x3050,
128 0x3051, 0x3052, 0x3053, 0x3054, 0x3055, 0x3056, 0x3057, 0x3058,
129 0x3059, 0x305a, 0x305b, 0x305c, 0x305d, 0x305e, 0x305f, 0x3060,
130 0x3061, 0x3062, 0x3063, 0x3064, 0x3065, 0x3066, 0x3067, 0x3068,
131 0x3069, 0x306a, 0x306b, 0x306c, 0x306d, 0x306e, 0x306f, 0x3070,
132 0x3071, 0x3072, 0x3073, 0x3074, 0x3075, 0x3076, 0x3077, 0x3078,
133 0x3079, 0x307a, 0x307b, 0x307c, 0x307d, 0x307e, 0x307f, 0x3080,
134 0x3081, 0x3082, 0x3083, 0x3084, 0x3085, 0x3086, 0x3087, 0x3088,
135 0x3089, 0x308a, 0x308b, 0x308c, 0x308d, 0x308e, 0x308f, 0x3090,
136 0x3091, 0x3092, 0x3093, 0xffff, 0xffff, 0xffff,
137 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
138 /* 0x2b */
139 0x30a1, 0x30a2, 0x30a3, 0x30a4, 0x30a5, 0x30a6, 0x30a7, 0x30a8,
140 0x30a9, 0x30aa, 0x30ab, 0x30ac, 0x30ad, 0x30ae, 0x30af, 0x30b0,
141 0x30b1, 0x30b2, 0x30b3, 0x30b4, 0x30b5, 0x30b6, 0x30b7, 0x30b8,
142 0x30b9, 0x30ba, 0x30bb, 0x30bc, 0x30bd, 0x30be, 0x30bf, 0x30c0,
143 0x30c1, 0x30c2, 0x30c3, 0x30c4, 0x30c5, 0x30c6, 0x30c7, 0x30c8,
144 0x30c9, 0x30ca, 0x30cb, 0x30cc, 0x30cd, 0x30ce, 0x30cf, 0x30d0,
145 0x30d1, 0x30d2, 0x30d3, 0x30d4, 0x30d5, 0x30d6, 0x30d7, 0x30d8,
146 0x30d9, 0x30da, 0x30db, 0x30dc, 0x30dd, 0x30de, 0x30df, 0x30e0,
147 0x30e1, 0x30e2, 0x30e3, 0x30e4, 0x30e5, 0x30e6, 0x30e7, 0x30e8,
148 0x30e9, 0x30ea, 0x30eb, 0x30ec, 0x30ed, 0x30ee, 0x30ef, 0x30f0,
149 0x30f1, 0x30f2, 0x30f3, 0x30f4, 0x30f5, 0x30f6, 0xffff, 0xffff,
150 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
151 /* 0x2c */
152 0x0410, 0x0411, 0x0412, 0x0413, 0x0414, 0x0415, 0x0401, 0x0416,
153 0x0417, 0x0418, 0x0419, 0x041a, 0x041b, 0x041c, 0x041d, 0x041e,
154 0x041f, 0x0420, 0x0421, 0x0422, 0x0423, 0x0424, 0x0425, 0x0426,
155 0x0427, 0x0428, 0x0429, 0x042a, 0x042b, 0x042c, 0x042d, 0x042e,
156 0x042f, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
157 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
158 0x0430, 0x0431, 0x0432, 0x0433, 0x0434, 0x0435, 0x0451, 0x0436,
159 0x0437, 0x0438, 0x0439, 0x043a, 0x043b, 0x043c, 0x043d, 0x043e,
160 0x043f, 0x0440, 0x0441, 0x0442, 0x0443, 0x0444, 0x0445, 0x0446,
161 0x0447, 0x0448, 0x0449, 0x044a, 0x044b, 0x044c, 0x044d, 0x044e,
162 0x044f,
163 };
164 static const unsigned short ksc5601_2uni_page30[2350] = {
165 /* 0x30 */
166 0xac00, 0xac01, 0xac04, 0xac07, 0xac08, 0xac09, 0xac0a, 0xac10,
167 0xac11, 0xac12, 0xac13, 0xac14, 0xac15, 0xac16, 0xac17, 0xac19,
168 0xac1a, 0xac1b, 0xac1c, 0xac1d, 0xac20, 0xac24, 0xac2c, 0xac2d,
169 0xac2f, 0xac30, 0xac31, 0xac38, 0xac39, 0xac3c, 0xac40, 0xac4b,
170 0xac4d, 0xac54, 0xac58, 0xac5c, 0xac70, 0xac71, 0xac74, 0xac77,
171 0xac78, 0xac7a, 0xac80, 0xac81, 0xac83, 0xac84, 0xac85, 0xac86,
172 0xac89, 0xac8a, 0xac8b, 0xac8c, 0xac90, 0xac94, 0xac9c, 0xac9d,
173 0xac9f, 0xaca0, 0xaca1, 0xaca8, 0xaca9, 0xaca, 0xacac, 0xacaf,
174 0xacb0, 0xacb8, 0xacb9, 0xacbb, 0xacbc, 0xacbd, 0xacc1, 0xacc4,
175 0xacc8, 0xacc, 0xacd5, 0xacd7, 0xace0, 0xace1, 0xace4, 0xace7,
176 0xace8, 0xace, 0xacec, 0xacef, 0xacf0, 0xacf1, 0xacf3, 0xacf5,
177 0xacf6, 0xacf, 0xacfd, 0xad00, 0xad04, 0xad06,
178 /* 0x31 */
179 0xad0c, 0xad0d, 0xad0f, 0xad11, 0xad18, 0xad1c, 0xad20, 0xad29,
180 0xad2c, 0xad2d, 0xad34, 0xad35, 0xad38, 0xad3c, 0xad44, 0xad45,
181 0xad47, 0xad49, 0xad50, 0xad54, 0xad58, 0xad61, 0xad63, 0xad6c,
182 0xad6d, 0xad70, 0xad73, 0xad74, 0xad75, 0xad76, 0xad7b, 0xad7c,
183 0xad7d, 0xad7f, 0xad81, 0xad82, 0xad88, 0xad89, 0xad8c, 0xad90,
184 0xad9c, 0xad9d, 0xada4, 0xadb7, 0xadc0, 0xadc1, 0xadc4, 0xadc8,
185 0xadd0, 0xadd1, 0xadd3, 0xaddc, 0xade0, 0xade4, 0xadf8, 0xadf9,
186 0xadfc, 0xadff, 0xae00, 0xae01, 0xae08, 0xae09, 0xae0b, 0xae0d,
187 0xae14, 0xae30, 0xae31, 0xae34, 0xae37, 0xae38, 0xae3a, 0xae40,
188 0xae41, 0xae43, 0xae45, 0xae46, 0xae4a, 0xae4c, 0xae4d, 0xae4e,
189 0xae50, 0xae54, 0xae56, 0xae5c, 0xae5d, 0xae5f, 0xae60, 0xae61,
190 0xae65, 0xae68, 0xae69, 0xae6c, 0xae70, 0xae78,
191 /* 0x32 */
192 0xae79, 0xae7b, 0xae7c, 0xae7d, 0xae84, 0xae85, 0xae8c, 0xaebc,
193 0xaebd, 0xaebe, 0xaec0, 0xaec4, 0xaecc, 0xaecd, 0xaecf, 0xaed0,
194 0xaed1, 0xaed8, 0xaed9, 0xaedc, 0xae, 0xae, 0xaed, 0xaef4,
195 0xaef8, 0xaef, 0xaf07, 0xaf08, 0xaf0d, 0xaf10, 0xaf2c, 0xaf2d,
196 0xaf30, 0xaf32, 0xaf34, 0xaf3c, 0xaf3d, 0xaf3f, 0xaf41, 0xaf42,
197 0xaf43, 0xaf48, 0xaf49, 0xaf50, 0xaf5c, 0xaf5d, 0xaf64, 0xaf65,
198 0xaf79, 0xaf80, 0xaf84, 0xaf88, 0xaf90, 0xaf91, 0xaf95, 0xaf9c,
199 0xafb8, 0xafb9, 0xafbc, 0xafc0, 0xafc7, 0xafc8, 0xafc9, 0xafcb,
200 0xafcd, 0xafce, 0xafd4, 0xafdc, 0xaf, 0xaf, 0xaf, 0xaf,
201 0xaf, 0xaf, 0xb000, 0xb001, 0xb004, 0xb00c, 0xb010, 0xb014,
202 0xb01c, 0xb01d, 0xb028, 0xb044, 0xb045, 0xb048, 0xb04a, 0xb04c,
203 0xb04e, 0xb053, 0xb054, 0xb055, 0xb057, 0xb059,
204 /* 0x33 */
205 0xb05d, 0xb07c, 0xb07d, 0xb080, 0xb084, 0xb08c, 0xb08d, 0xb08f,
206 0xb091, 0xb098, 0xb099, 0xb09a, 0xb09c, 0xb09f, 0xb0a0, 0xb0a1,
207 0xb0a2, 0xb0a8, 0xb0a9, 0xb0ab, 0xb0ac, 0xb0ad, 0xb0ae, 0xb0af,
208 0xb0b1, 0xb0b3, 0xb0b4, 0xb0b5, 0xb0b8, 0xb0bc, 0xb0c4, 0xb0c5,

```

```

209 0xb0c7, 0xb0c8, 0xb0c9, 0xb0d0, 0xb0d1, 0xb0d4, 0xb0d8, 0xb0e0,
210 0xb0e5, 0xb108, 0xb109, 0xb10b, 0xb10c, 0xb110, 0xb112, 0xb113,
211 0xb118, 0xb119, 0xb11b, 0xb11c, 0xb11d, 0xb123, 0xb124, 0xb125,
212 0xb128, 0xb12c, 0xb134, 0xb135, 0xb137, 0xb138, 0xb139, 0xb140,
213 0xb141, 0xb144, 0xb148, 0xb150, 0xb151, 0xb154, 0xb155, 0xb158,
214 0xb15c, 0xb160, 0xb178, 0xb179, 0xb17c, 0xb180, 0xb182, 0xb188,
215 0xb189, 0xb18b, 0xb18d, 0xb192, 0xb193, 0xb194, 0xb198, 0xb19c,
216 0xb1a8, 0xb1cc, 0xb1d0, 0xb1d4, 0xb1dc, 0xb1dd,
217 /* 0x34 */
218 0xb1df, 0xb1e8, 0xb1e9, 0xb1ec, 0xb1f0, 0xb1f9, 0xb1fb, 0xb1fd,
219 0xb204, 0xb205, 0xb208, 0xb20b, 0xb20c, 0xb214, 0xb215, 0xb217,
220 0xb219, 0xb220, 0xb234, 0xb23c, 0xb258, 0xb25c, 0xb260, 0xb268,
221 0xb269, 0xb274, 0xb275, 0xb27c, 0xb284, 0xb285, 0xb289, 0xb290,
222 0xb291, 0xb294, 0xb298, 0xb299, 0xb29a, 0xb2a0, 0xb2a1, 0xb2a3,
223 0xb2a5, 0xb2a6, 0xb2aa, 0xb2ac, 0xb2b0, 0xb2b4, 0xb2c8, 0xb2c9,
224 0xb2cc, 0xb2d0, 0xb2d2, 0xb2d8, 0xb2d9, 0xb2db, 0xb2dd, 0xb2e2,
225 0xb2e4, 0xb2e5, 0xb2e6, 0xb2e8, 0xb2eb, 0xb2ec, 0xb2ed, 0xb2ee,
226 0xb2ef, 0xb2f3, 0xb2f4, 0xb2f5, 0xb2f7, 0xb2f8, 0xb2f9, 0xb2fa,
227 0xb2fb, 0xb2ff, 0xb300, 0xb301, 0xb304, 0xb308, 0xb310, 0xb311,
228 0xb313, 0xb314, 0xb315, 0xb31c, 0xb354, 0xb355, 0xb356, 0xb358,
229 0xb35b, 0xb35c, 0xb35e, 0xb35f, 0xb364, 0xb365,
230 /* 0x35 */
231 0xb367, 0xb369, 0xb36b, 0xb36e, 0xb370, 0xb371, 0xb374, 0xb378,
232 0xb380, 0xb381, 0xb383, 0xb384, 0xb385, 0xb38c, 0xb390, 0xb394,
233 0xb3a0, 0xb3a1, 0xb3a8, 0xb3ac, 0xb3c4, 0xb3c5, 0xb3c8, 0xb3cb,
234 0xb3cc, 0xb3ce, 0xb3d0, 0xb3d4, 0xb3d5, 0xb3d7, 0xb3d9, 0xb3db,
235 0xb3dd, 0xb3e0, 0xb3e4, 0xb3e8, 0xb3fc, 0xb410, 0xb418, 0xb41c,
236 0xb420, 0xb428, 0xb429, 0xb42b, 0xb434, 0xb450, 0xb451, 0xb454,
237 0xb458, 0xb460, 0xb461, 0xb463, 0xb465, 0xb46c, 0xb480, 0xb488,
238 0xb49d, 0xb4a4, 0xb4a8, 0xb4ac, 0xb4b5, 0xb4b7, 0xb4b9, 0xb4c0,
239 0xb4c4, 0xb4c8, 0xb4d0, 0xb4d5, 0xb4dc, 0xb4dd, 0xb4e0, 0xb4e3,
240 0xb4e4, 0xb4e6, 0xb4ec, 0xb4ed, 0xb4ef, 0xb4f1, 0xb4f8, 0xb514,
241 0xb515, 0xb518, 0xb51b, 0xb51c, 0xb524, 0xb525, 0xb527, 0xb528,
242 0xb529, 0xb52a, 0xb530, 0xb531, 0xb534, 0xb538,
243 /* 0x36 */
244 0xb540, 0xb541, 0xb543, 0xb544, 0xb545, 0xb54b, 0xb54c, 0xb54d,
245 0xb550, 0xb554, 0xb55c, 0xb55d, 0xb55f, 0xb560, 0xb561, 0xb5a0,
246 0xb5a1, 0xb5a4, 0xb5a8, 0xb5aa, 0xb5ab, 0xb5b0, 0xb5b1, 0xb5b3,
247 0xb5b4, 0xb5b5, 0xb5bb, 0xb5bc, 0xb5bd, 0xb5c0, 0xb5c4, 0xb5cc,
248 0xb5cd, 0xb5cf, 0xb5d0, 0xb5d1, 0xb5d8, 0xb5ec, 0xb610, 0xb611,
249 0xb614, 0xb618, 0xb625, 0xb62c, 0xb634, 0xb648, 0xb664, 0xb668,
250 0xb69c, 0xb69d, 0xb6a0, 0xb6a4, 0xb6ab, 0xb6ac, 0xb6b1, 0xb6d4,
251 0xb6f0, 0xb6f4, 0xb6f8, 0xb700, 0xb701, 0xb705, 0xb728, 0xb729,
252 0xb72c, 0xb72f, 0xb730, 0xb738, 0xb739, 0xb73b, 0xb744, 0xb748,
253 0xb74c, 0xb754, 0xb755, 0xb760, 0xb764, 0xb768, 0xb770, 0xb771,
254 0xb773, 0xb775, 0xb77c, 0xb77d, 0xb780, 0xb784, 0xb78c, 0xb78d,
255 0xb78f, 0xb790, 0xb791, 0xb792, 0xb796, 0xb797,
256 /* 0x37 */
257 0xb798, 0xb799, 0xb79c, 0xb7a0, 0xb7a8, 0xb7a9, 0xb7ab, 0xb7ac,
258 0xb7ad, 0xb7b4, 0xb7b5, 0xb7b8, 0xb7c7, 0xb7c9, 0xb7ec, 0xb7ed,
259 0xb7f0, 0xb7f4, 0xb7fc, 0xb7fd, 0xb7ff, 0xb800, 0xb801, 0xb807,
260 0xb808, 0xb809, 0xb80c, 0xb810, 0xb818, 0xb819, 0xb81b, 0xb81d,
261 0xb824, 0xb825, 0xb828, 0xb82c, 0xb834, 0xb835, 0xb837, 0xb838,
262 0xb839, 0xb840, 0xb844, 0xb851, 0xb853, 0xb85c, 0xb85d, 0xb860,
263 0xb864, 0xb86c, 0xb86d, 0xb86f, 0xb871, 0xb878, 0xb87c, 0xb88d,
264 0xb8a8, 0xb8b0, 0xb8b4, 0xb8b8, 0xb8c0, 0xb8c1, 0xb8c3, 0xb8c5,
265 0xb8cc, 0xb8d0, 0xb8d4, 0xb8dd, 0xb8df, 0xb8e1, 0xb8e8, 0xb8e9,
266 0xb8ec, 0xb8f0, 0xb8f8, 0xb8f9, 0xb8fb, 0xb8fd, 0xb904, 0xb918,
267 0xb920, 0xb93c, 0xb93d, 0xb940, 0xb944, 0xb94c, 0xb94f, 0xb951,
268 0xb958, 0xb959, 0xb95c, 0xb960, 0xb968, 0xb969,
269 /* 0x38 */
270 0xb96b, 0xb96d, 0xb974, 0xb975, 0xb978, 0xb97c, 0xb984, 0xb985,
271 0xb987, 0xb989, 0xb98a, 0xb98d, 0xb98e, 0xb99c, 0xb9ad, 0xb9b0,
272 0xb9b4, 0xb9bc, 0xb9bd, 0xb9bf, 0xb9c1, 0xb9c8, 0xb9c9, 0xb9cc,
273 0xb9ce, 0xb9cf, 0xb9d0, 0xb9d1, 0xb9d2, 0xb9d8, 0xb9d9, 0xb9db,
274 0xb9dd, 0xb9de, 0xb9e1, 0xb9e3, 0xb9e4, 0xb9e5, 0xb9e8, 0xb9ec,
275 0xb9f4, 0xb9f5, 0xb9f7, 0xb9f8, 0xb9f9, 0xb9fa, 0xba00, 0xba01,
276 0xba08, 0xba15, 0xba38, 0xba39, 0xba3c, 0xba40, 0xba42, 0xba48,
277 0xba49, 0xba4b, 0xba4d, 0xba4e, 0xba53, 0xba54, 0xba55, 0xba58,
278 0xba5c, 0xba64, 0xba65, 0xba67, 0xba68, 0xba69, 0xba70, 0xba71,
279 0xba74, 0xba78, 0xba83, 0xba84, 0xba85, 0xba87, 0xba8c, 0xbaa8,
280 0xbaa9, 0bbaab, 0bbaac, 0bab0, 0bab2, 0bab8, 0bab9, 0babbb,
281 0xbabd, 0xbac4, 0xbac8, 0bad8, 0bad9, 0bafcf,
282 /* 0x39 */
283 0xbb00, 0xbb04, 0xbb0d, 0xbb0f, 0xbb11, 0xbb18, 0xbb1c, 0xbb20,
284 0xbb29, 0xbb2b, 0xbb34, 0xbb35, 0xbb36, 0xbb38, 0xbb3b, 0xbb3c,
285 0xbb3d, 0xbb3e, 0xbb44, 0xbb45, 0xbb47, 0xbb49, 0xbb4d, 0xbb4f,
286 0xbb50, 0xbb54, 0xbb58, 0xbb61, 0xbb63, 0xbb6c, 0xbb88, 0xbb8c,
287 0xbb90, 0xbaa4, 0xbba8, 0bbbac, 0bbb4, 0bbb7, 0xbbc0, 0xbbc4,
288 0xbbc8, 0xbbd0, 0bbdd3, 0bbdf8, 0bbff9, 0bbffc, 0bbfff, 0x bc00,
289 0x bc02, 0x bc08, 0x bc09, 0x bc0b, 0x bc0c, 0x bc0d, 0x bc0f, 0x bc11,
290 0x bc14, 0x bc15, 0x bc16, 0x bc17, 0x bc18, 0x bc1b, 0x bc1c, 0x bc1d,
291 0x bc1e, 0x bc1f, 0x bc24, 0x bc25, 0x bc27, 0x bc29, 0x bc2d, 0x bc30,
292 0x bc31, 0x bc34, 0x bc38, 0x bc40, 0x bc41, 0x bc43, 0x bc44, 0x bc45,
293 0x bc49, 0x bc4c, 0x bc4d, 0x bc50, 0x bc5d, 0x bc84, 0x bc85, 0x bc88,
294 0x bc8b, 0x bc8c, 0x bc8e, 0x bc94, 0x bc95, 0x bc97,
295 /* 0x3a */

```

```
296 0xbc99, 0xbc9a, 0xbca0, 0xbca1, 0xbca4, 0xbca7, 0xbca8, 0xbcb0,
297 0xbcb1, 0xbcb3, 0xbcb4, 0xbcb5, 0xbcbc, 0xbcbd, 0xbcc0, 0xbcc4,
298 0xbccd, 0xbccf, 0xbcd0, 0xbcd1, 0xbcd5, 0xbcd8, 0xbcdc, 0xbcf4,
299 0xbcf5, 0xbcf6, 0xbcf8, 0xbcf9, 0xbd04, 0xbd05, 0xbd07, 0xbd09,
300 0xbd10, 0xbd14, 0xbd24, 0xbd2c, 0xbd40, 0xbd48, 0xbd49, 0xbd4c,
301 0xbd50, 0xbd58, 0xbd59, 0xbd64, 0xbd68, 0xbd80, 0xbd81, 0xbd84,
302 0xbd87, 0xbd88, 0xbd89, 0xbd8a, 0xbd90, 0xbd91, 0xbd93, 0xbd95,
303 0xbd99, 0xbd9a, 0xbd9c, 0xbda4, 0xbdb0, 0xbdb8, 0xbdd4, 0xbdd5,
304 0xbdd8, 0xbddc, 0xbde9, 0xbdf0, 0xbdf4, 0xbdf8, 0xbe00, 0xbe03,
305 0xbe05, 0xbe0c, 0xbe0d, 0xbe10, 0xbe14, 0xbe1c, 0xbe1d, 0xbe1f,
306 0xbe44, 0xbe45, 0xbe48, 0xbe4c, 0xbe4e, 0xbe54, 0xbe55, 0xbe57,
307 0xbe59, 0xbe5a, 0xbe5b, 0xbe60, 0xbe61, 0xbe64,
308 /* 0x3b */
309 0xbe68, 0xbe6a, 0xbe70, 0xbe71, 0xbe73, 0xbe74, 0xbe75, 0xbe7b,
310 0xbe7c, 0xbe7d, 0xbe80, 0xbe84, 0xbe8c, 0xbe8d, 0xbe8f, 0xbe90,
311 0xbe91, 0xbe98, 0xbe99, 0xbea8, 0xbed0, 0xbed1, 0xbed4, 0xbed7,
312 0xbed8, 0xee0, 0xee3, 0xee4, 0xee5, 0xee6, 0xbf01, 0xbf08,
313 0xbf09, 0xbf18, 0xbf19, 0xbf1b, 0xbf1c, 0xbf1d, 0xbf40, 0xbf41,
314 0xbf44, 0xbf48, 0xbf50, 0xbf51, 0xbf55, 0xbf94, 0xbf9b, 0xbfc5,
315 0xbfcc, 0xbfcd, 0xbfd0, 0xbfd4, 0xbfdc, 0xbfdf, 0xbfe1, 0xc03c,
316 0xc051, 0xc058, 0xc05c, 0xc060, 0xc068, 0xc069, 0xc090, 0xc091,
317 0xc094, 0xc098, 0xc0a0, 0xc0a1, 0xc0a3, 0xc0a5, 0xc0ac, 0xc0ad,
318 0xc0af, 0xc0b0, 0xc0b3, 0xc0b4, 0xc0b5, 0xc0b6, 0xc0bc, 0xc0bd,
319 0xc0bf, 0xc0c0, 0xc0c1, 0xc0c5, 0xc0c8, 0xc0c9, 0xc0cc, 0xc0d0,
320 0xc0d8, 0xc0d9, 0xc0db, 0xc0dc, 0xc0dd, 0xc0e4,
321 /* 0x3c */
322 0xc0e5, 0xc0e8, 0xc0ec, 0xc0f4, 0xc0f5, 0xc0f7, 0xc0f9, 0xc100,
323 0xc104, 0xc108, 0xc110, 0xc115, 0xc11c, 0xc11d, 0xc11e, 0xc11f,
324 0xc120, 0xc123, 0xc124, 0xc126, 0xc127, 0xc12c, 0xc12d, 0xc12f,
325 0xc130, 0xc131, 0xc136, 0xc138, 0xc139, 0xc13c, 0xc140, 0xc148,
326 0xc149, 0xc14b, 0xc14c, 0xc14d, 0xc154, 0xc155, 0xc158, 0xc15c,
327 0xc164, 0xc165, 0xc167, 0xc168, 0xc169, 0xc170, 0xc174, 0xc178,
328 0xc185, 0xc18c, 0xc18d, 0xc18e, 0xc190, 0xc194, 0xc196, 0xc19c,
329 0xc19d, 0xc19f, 0xc1a1, 0xc1a5, 0xc1a8, 0xc1a9, 0xc1ac, 0xc1b0,
330 0xc1bd, 0xc1c4, 0xc1c8, 0xc1cc, 0xc1d4, 0xc1d7, 0xc1d8, 0xc1e0,
331 0xc1e4, 0xc1e8, 0xc1f0, 0xc1f1, 0xc1f3, 0xc1fc, 0xc1fd, 0xc200,
332 0xc204, 0xc20c, 0xc20d, 0xc20f, 0xc211, 0xc218, 0xc219, 0xc21c,
333 0xc21f, 0xc220, 0xc228, 0xc229, 0xc22b, 0xc22d,
334 /* 0x3d */
335 0xc22f, 0xc231, 0xc232, 0xc234, 0xc248, 0xc250, 0xc251, 0xc254,
336 0xc258, 0xc260, 0xc265, 0xc26c, 0xc26d, 0xc270, 0xc274, 0xc27c,
337 0xc27d, 0xc27f, 0xc281, 0xc288, 0xc289, 0xc290, 0xc298, 0xc29b,
338 0xc29d, 0xc2a4, 0xc2a5, 0xc2a8, 0xc2ac, 0xc2ad, 0xc2b4, 0xc2b5,
339 0xc2b7, 0xc2b9, 0xc2dc, 0xc2dd, 0xc2e0, 0xc2e3, 0xc2e4, 0xc2eb,
340 0xc2ec, 0xc2ed, 0xc2ef, 0xc2f1, 0xc2f6, 0xc2f8, 0xc2f9, 0xc2fb,
341 0xc2fc, 0xc300, 0xc308, 0xc309, 0xc30c, 0xc30d, 0xc313, 0xc314,
342 0xc315, 0xc318, 0xc31c, 0xc324, 0xc325, 0xc328, 0xc329, 0xc345,
343 0xc368, 0xc369, 0xc36c, 0xc370, 0xc372, 0xc378, 0xc379, 0xc37c,
344 0xc37d, 0xc384, 0xc388, 0xc38c, 0xc3c0, 0xc3d8, 0xc3d9, 0xc3dc,
345 0xc3df, 0xc3e0, 0xc3e2, 0xc3e8, 0xc3e9, 0xc3ed, 0xc3f4, 0xc3f5,
346 0xc3f8, 0xc408, 0xc410, 0xc424, 0xc42c, 0xc430,
347 /* 0x3e */
348 0xc434, 0xc43c, 0xc43d, 0xc448, 0xc464, 0xc465, 0xc468, 0xc46c,
349 0xc474, 0xc475, 0xc479, 0xc480, 0xc494, 0xc49c, 0xc4b8, 0xc4bc,
350 0xc4e9, 0xc4f0, 0xc4f1, 0xc4f4, 0xc4f8, 0xc4fa, 0xc4ff, 0xc500,
351 0xc501, 0xc50c, 0xc510, 0xc514, 0xc51c, 0xc528, 0xc529, 0xc52c,
352 0xc530, 0xc538, 0xc539, 0xc53b, 0xc53d, 0xc544, 0xc545, 0xc548,
353 0xc549, 0xc54a, 0xc54c, 0xc54d, 0xc54e, 0xc553, 0xc554, 0xc555,
354 0xc557, 0xc558, 0xc559, 0xc55d, 0xc55e, 0xc560, 0xc561, 0xc564,
355 0xc568, 0xc570, 0xc571, 0xc573, 0xc574, 0xc575, 0xc57c, 0xc57d,
356 0xc580, 0xc584, 0xc587, 0xc58c, 0xc58d, 0xc58f, 0xc591, 0xc595,
357 0xc597, 0xc598, 0xc59c, 0xc5a0, 0xc5a9, 0xc5b4, 0xc5b5, 0xc5b8,
358 0xc5b9, 0xc5bb, 0xc5bc, 0xc5bd, 0xc5be, 0xc5c4, 0xc5c5, 0xc5c6,
359 0xc5c7, 0xc5c8, 0xc5c9, 0xc5ca, 0xc5cc, 0xc5ce,
360 /* 0x3f */
361 0xc5d0, 0xc5d1, 0xc5d4, 0xc5d8, 0xc5e0, 0xc5e1, 0xc5e3, 0xc5e5,
362 0xc5ec, 0xc5ed, 0xc5ee, 0xc5f0, 0xc5f4, 0xc5f6, 0xc5f7, 0xc5fc,
363 0xc5fd, 0xc5fe, 0xc5ff, 0xc600, 0xc601, 0xc605, 0xc606, 0xc607,
364 0xc608, 0xc60c, 0xc610, 0xc618, 0xc619, 0xc61b, 0xc61c, 0xc624,
365 0xc625, 0xc628, 0xc62c, 0xc62d, 0xc62e, 0xc630, 0xc633, 0xc634,
366 0xc635, 0xc637, 0xc639, 0xc63b, 0xc640, 0xc641, 0xc644, 0xc648,
367 0xc650, 0xc651, 0xc653, 0xc654, 0xc655, 0xc65c, 0xc65d, 0xc660,
368 0xc66c, 0xc66f, 0xc671, 0xc678, 0xc679, 0xc67c, 0xc680, 0xc688,
369 0xc689, 0xc68b, 0xc68d, 0xc694, 0xc695, 0xc698, 0xc69c, 0xc6a4,
370 0xc6a5, 0xc6a7, 0xc6a9, 0xc6b0, 0xc6b1, 0xc6b4, 0xc6b8, 0xc6b9,
371 0xc6ba, 0xc6c0, 0xc6c1, 0xc6c3, 0xc6c5, 0xc6cc, 0xc6cd, 0xc6d0,
372 0xc6d4, 0xc6dc, 0xc6dd, 0xc6e0, 0xc6e1, 0xc6e8,
373 /* 0x40 */
374 0xc6e9, 0xc6ec, 0xc6f0, 0xc6f8, 0xc6f9, 0xc6fd, 0xc704, 0xc705,
375 0xc708, 0xc70c, 0xc714, 0xc715, 0xc717, 0xc719, 0xc720, 0xc721,
376 0xc724, 0xc728, 0xc730, 0xc731, 0xc733, 0xc735, 0xc737, 0xc73c,
377 0xc73d, 0xc740, 0xc744, 0xc74a, 0xc74c, 0xc74d, 0xc74f, 0xc751,
378 0xc752, 0xc753, 0xc754, 0xc755, 0xc756, 0xc757, 0xc758, 0xc75c,
379 0xc760, 0xc768, 0xc76b, 0xc774, 0xc775, 0xc778, 0xc77c, 0xc77d,
380 0xc77e, 0xc783, 0xc784, 0xc785, 0xc787, 0xc788, 0xc789, 0xc78a,
381 0xc78e, 0xc790, 0xc791, 0xc794, 0xc796, 0xc797, 0xc798, 0xc79a,
382 0xc7a0, 0xc7a1, 0xc7a3, 0xc7a4, 0xc7a5, 0xc7a6, 0xc7ac, 0xc7ad,
```



```
383 0xc7b0, 0xc7b4, 0xc7bc, 0xc7bd, 0xc7bf, 0xc7c0, 0xc7c1, 0xc7c8,
384 0xc7c9, 0xc7cc, 0xc7ce, 0xc7d0, 0xc7d8, 0xc7dd, 0xc7e4, 0xc7e8,
385 0xc7ec, 0xc800, 0xc801, 0xc804, 0xc808, 0xc80a,
386 /* 0x41 */
387 0xc810, 0xc811, 0xc813, 0xc815, 0xc816, 0xc81c, 0xc81d, 0xc820,
388 0xc824, 0xc82c, 0xc82d, 0xc82f, 0xc831, 0xc838, 0xc83c, 0xc840,
389 0xc848, 0xc849, 0xc84c, 0xc84d, 0xc854, 0xc870, 0xc871, 0xc874,
390 0xc878, 0xc87a, 0xc880, 0xc881, 0xc883, 0xc885, 0xc886, 0xc887,
391 0xc88b, 0xc88c, 0xc88d, 0xc894, 0xc89d, 0xc89f, 0xc8a1, 0xc8a8,
392 0xc8bc, 0xc8bd, 0xc8c4, 0xc8c8, 0xc8cc, 0xc8d4, 0xc8d5, 0xc8d7,
393 0xc8d9, 0xc8e0, 0xc8e1, 0xc8e4, 0xc8f5, 0xc8fc, 0xc8fd, 0xc900,
394 0xc904, 0xc905, 0xc906, 0xc90c, 0xc90d, 0xc90f, 0xc911, 0xc918,
395 0xc92c, 0xc934, 0xc950, 0xc951, 0xc954, 0xc958, 0xc960, 0xc961,
396 0xc963, 0xc96c, 0xc970, 0xc974, 0xc97c, 0xc988, 0xc989, 0xc98c,
397 0xc990, 0xc998, 0xc999, 0xc99b, 0xc99d, 0xc9c0, 0xc9c1, 0xc9c4,
398 0xc9c7, 0xc9c8, 0xc9ca, 0xc9d0, 0xc9d1, 0xc9d3,
399 /* 0x42 */
400 0xc9d5, 0xc9d6, 0xc9d9, 0xc9da, 0xc9dc, 0xc9dd, 0xc9e0, 0xc9e2,
401 0xc9e4, 0xc9e7, 0xc9ec, 0xc9ed, 0xc9ef, 0xc9f0, 0xc9f1, 0xc9f8,
402 0xc9f9, 0xc9fc, 0xca00, 0xca08, 0xca09, 0xca0b, 0xca0c, 0xca0d,
403 0xca14, 0xca18, 0xca18, 0xca29, 0xca4c, 0xca4d, 0xca50, 0xca54, 0xca5c,
404 0xca5d, 0xca5f, 0xca60, 0xca61, 0xca68, 0xca7d, 0xca84, 0xca98,
405 0xcabc, 0xcabd, 0xcac0, 0xcac4, 0xcacc, 0xcacd, 0xcacf, 0xcad1,
406 0xcad3, 0xcad8, 0xcad9, 0xcae0, 0xcaec, 0xcaf4, 0xcb08, 0xcb10,
407 0xcb14, 0xcb18, 0xcb20, 0xcb21, 0xcb41, 0xcb48, 0xcb49, 0xcb4c,
408 0xcb50, 0xcb58, 0xcb59, 0xcb5d, 0xcb64, 0xcb78, 0xcb79, 0xcb9c,
409 0xcbb8, 0xcbd4, 0xcbe4, 0xcbe7, 0xcbe9, 0xcc0c, 0xcc0d, 0xcc10,
410 0xcc14, 0xcc1c, 0xcc1d, 0xcc21, 0xcc22, 0xcc27, 0xcc28, 0xcc29,
411 0xcc2c, 0xcc2e, 0xcc30, 0xcc38, 0xcc39, 0xcc3b,
412 /* 0x43 */
413 0xcc3c, 0xcc3d, 0xcc3e, 0xcc44, 0xcc45, 0xcc48, 0xcc4c, 0xcc54,
414 0xcc55, 0xcc57, 0xcc58, 0xcc59, 0xcc60, 0xcc64, 0xcc66, 0xcc68,
415 0xcc70, 0xcc75, 0xcc98, 0xcc99, 0xcc9c, 0xccea, 0xccea9,
416 0xccab, 0xccac, 0xccad, 0ccb4, 0ccb5, 0ccb8, 0ccbcb, 0ccc4,
417 0cccc5, 0ccc7, 0cccc9, 0xcd0, 0xcd4, 0xcce4, 0xccec, 0ccf0,
418 0xcd01, 0xcd08, 0xcd09, 0xcd0c, 0xcd10, 0xcd18, 0xcd19, 0xcd1b,
419 0xcd1d, 0xcd24, 0xcd28, 0xcd2c, 0xcd39, 0xcd5c, 0xcd60, 0xcd64,
420 0xcd6c, 0xcd6d, 0xcd6f, 0xcd71, 0xcd78, 0xcd88, 0xcd94, 0xcd95,
421 0xcd98, 0xcd9c, 0xcda4, 0xcda5, 0xcda7, 0xcda9, 0xcdb0, 0xcdbc4,
422 0xcdcc, 0xcdc0, 0xcde8, 0xcdec, 0cdf0, 0cdf8, 0cdf9, 0cdfb,
423 0xcdfd, 0xce04, 0xce08, 0xce0c, 0xce14, 0xce19, 0xce20, 0xce21,
424 0xce24, 0xce28, 0xce30, 0xce31, 0xce33, 0xce35,
425 /* 0x44 */
426 0xce58, 0xce59, 0xce5c, 0xce5f, 0xce60, 0xce61, 0xce68, 0xce69,
427 0xce6b, 0xce6d, 0xce74, 0xce75, 0xce78, 0xce7c, 0xce84, 0xce85,
428 0xce87, 0xce89, 0xce90, 0xce91, 0xce94, 0xce98, 0xcea0, 0xcea1,
429 0xcea3, 0xcea4, 0xcea5, 0xceac, 0xcead, 0xcec1, 0xcee4, 0xcee5,
430 0xcee8, 0xceeb, 0xceec, 0xcef4, 0xcef5, 0xcef7, 0xcef8, 0xcef9,
431 0xcf00, 0xcf01, 0xcf04, 0xcf08, 0xcf10, 0xcf11, 0xcf13, 0xcf15,
432 0xcf1c, 0xcf20, 0xcf24, 0xcf2c, 0xcf2d, 0xcf2f, 0xcf30, 0xcf31,
433 0xcf38, 0xcf54, 0xcf55, 0xcf58, 0xcf5c, 0xcf64, 0xcf65, 0xcf67,
434 0xcf69, 0xcf70, 0xcf71, 0xcf74, 0xcf78, 0xcf80, 0xcf85, 0xcf8c,
435 0xcfa1, 0xcfa8, 0xcfb0, 0xcfc4, 0xcfe0, 0xcfel, 0xcfe4, 0xcfe8,
436 0xcff0, 0xcff1, 0xcff3, 0xcff5, 0xcffc, 0xd000, 0xd004, 0xd011,
437 0xd018, 0xd02d, 0xd034, 0xd035, 0xd038, 0xd03c,
438 /* 0x45 */
439 0xd044, 0xd045, 0xd047, 0xd049, 0xd050, 0xd054, 0xd058, 0xd060,
440 0xd06c, 0xd06d, 0xd070, 0xd074, 0xd07c, 0xd07d, 0xd081, 0xd0a4,
441 0xd0a5, 0xd0a8, 0xd0ac, 0xd0b4, 0xd0b5, 0xd0b7, 0xd0b9, 0xd0c0,
442 0xd0c1, 0xd0c4, 0xd0c8, 0xd0c9, 0xd0d0, 0xd0d1, 0xd0d3, 0xd0d4,
443 0xd0d5, 0xd0dc, 0xd0dd, 0xd0e0, 0xd0e4, 0xd0ec, 0xd0ed, 0xd0ef,
444 0xd0f0, 0xd0f1, 0xd0f8, 0xd10d, 0xd130, 0xd131, 0xd134, 0xd138,
445 0xd13a, 0xd140, 0xd141, 0xd141, 0xd143, 0xd144, 0xd145, 0xd14c, 0xd14d,
446 0xd150, 0xd154, 0xd15c, 0xd15d, 0xd15f, 0xd161, 0xd168, 0xd16c,
447 0xd17c, 0xd184, 0xd188, 0xd1a0, 0xd1a1, 0xd1a4, 0xd1a8, 0xd1b0,
448 0xd1b1, 0xd1b3, 0xd1b5, 0xd1ba, 0xd1bc, 0xd1c0, 0xd1d8, 0xd1f4,
449 0xd1f8, 0xd207, 0xd209, 0xd210, 0xd22c, 0xd22d, 0xd230, 0xd234,
450 0xd23c, 0xd23d, 0xd23f, 0xd241, 0xd248, 0xd25c,
451 /* 0x46 */
452 0xd264, 0xd280, 0xd281, 0xd284, 0xd288, 0xd290, 0xd291, 0xd295,
453 0xd29c, 0xd2a0, 0xd2a4, 0xd2ac, 0xd2b1, 0xd2b8, 0xd2b9, 0xd2bc,
454 0xd2bf, 0xd2c0, 0xd2c2, 0xd2c8, 0xd2c9, 0xd2cb, 0xd2d4, 0xd2d8,
455 0xd2dc, 0xd2e4, 0xd2e5, 0xd2f0, 0xd2f1, 0xd2f4, 0xd2f8, 0xd300,
456 0xd301, 0xd303, 0xd305, 0xd30c, 0xd30d, 0xd30e, 0xd310, 0xd314,
457 0xd316, 0xd31c, 0xd31d, 0xd31f, 0xd320, 0xd321, 0xd325, 0xd328,
458 0xd329, 0xd32c, 0xd330, 0xd338, 0xd339, 0xd33b, 0xd33c, 0xd33d,
459 0xd344, 0xd345, 0xd37c, 0xd37d, 0xd380, 0xd384, 0xd38c, 0xd38d,
460 0xd38f, 0xd390, 0xd391, 0xd398, 0xd399, 0xd39c, 0xd3a0, 0xd3a8,
461 0xd3a9, 0xd3ab, 0xd3ad, 0xd3b4, 0xd3b8, 0xd3bc, 0xd3c4, 0xd3c5,
462 0xd3c8, 0xd3c9, 0xd3d0, 0xd3d8, 0xd3e1, 0xd3e3, 0xd3ec, 0xd3ed,
463 0xd3f0, 0xd3f4, 0xd3fc, 0xd3fd, 0xd3ff, 0xd401,
464 /* 0x47 */
465 0xd408, 0xd41d, 0xd440, 0xd444, 0xd45c, 0xd460, 0xd464, 0xd46d,
466 0xd46f, 0xd478, 0xd479, 0xd47c, 0xd47f, 0xd480, 0xd482, 0xd488,
467 0xd489, 0xd48b, 0xd48d, 0xd494, 0xd4a9, 0xd4cc, 0xd4d0, 0xd4d4,
468 0xd4dc, 0xd4df, 0xd4e8, 0xd4ec, 0xd4f0, 0xd4f8, 0xd4fb, 0xd4fd,
469 0xd504, 0xd508, 0xd50c, 0xd514, 0xd515, 0xd517, 0xd53c, 0xd53d,
```

```
470 0xd540, 0xd544, 0xd54c, 0xd54d, 0xd54f, 0xd551, 0xd558, 0xd559,
471 0xd55c, 0xd560, 0xd565, 0xd568, 0xd569, 0xd56b, 0xd56d, 0xd574,
472 0xd575, 0xd578, 0xd57c, 0xd57c, 0xd584, 0xd585, 0xd587, 0xd588, 0xd589,
473 0xd590, 0xd5a5, 0xd5c8, 0xd5c9, 0xd5cc, 0xd5d0, 0xd5d2, 0xd5d8,
474 0xd5d9, 0xd5db, 0xd5dd, 0xd5e4, 0xd5e5, 0xd5e8, 0xd5ec, 0xd5f4,
475 0xd5f5, 0xd5f7, 0xd5f9, 0xd600, 0xd601, 0xd604, 0xd608, 0xd610,
476 0xd611, 0xd613, 0xd614, 0xd615, 0xd61c, 0xd620,
477 /* 0x48 */
478 0xd624, 0xd62d, 0xd638, 0xd639, 0xd63c, 0xd640, 0xd645, 0xd648,
479 0xd649, 0xd64b, 0xd64d, 0xd651, 0xd654, 0xd655, 0xd658, 0xd65c,
480 0xd667, 0xd669, 0xd670, 0xd671, 0xd674, 0xd683, 0xd685, 0xd68c,
481 0xd68d, 0xd690, 0xd694, 0xd69d, 0xd69f, 0xd6a1, 0xd6a8, 0xd6ac,
482 0xd6b0, 0xd6b9, 0xd6bb, 0xd6c4, 0xd6c5, 0xd6c8, 0xd6cc, 0xd6d1,
483 0xd6d4, 0xd6d7, 0xd6d9, 0xd6e0, 0xd6e4, 0xd6e8, 0xd6f0, 0xd6f5,
484 0xd6fc, 0xd6fd, 0xd700, 0xd704, 0xd711, 0xd711, 0xd718, 0xd719, 0xd71c,
485 0xd720, 0xd728, 0xd729, 0xd72b, 0xd72d, 0xd734, 0xd735, 0xd738,
486 0xd73c, 0xd744, 0xd747, 0xd749, 0xd750, 0xd751, 0xd754, 0xd756,
487 0xd757, 0xd758, 0xd759, 0xd760, 0xd761, 0xd763, 0xd765, 0xd769,
488 0xd76c, 0xd770, 0xd774, 0xd77c, 0xd77d, 0xd781, 0xd788, 0xd789,
489 0xd78c, 0xd790, 0xd798, 0xd799, 0xd79b, 0xd79d,
490 };
491 static const unsigned short ksc5601_2uni_page4a[4888] = {
492 /* 0x4a */
493 0x4f3d, 0x4f73, 0x5047, 0x50f9, 0x52a0, 0x53ef, 0x5475, 0x54e5,
494 0x5609, 0x5ac1, 0x5bb6, 0x6687, 0x67b6, 0x67b7, 0x67ef, 0x6b4c,
495 0x73c2, 0x75c2, 0x7a3c, 0x82db, 0x8304, 0x8857, 0x8888, 0x8a36,
496 0x8cc8, 0x8dcf, 0x8efb, 0x8fe6, 0x99d5, 0x523b, 0x5374, 0x5404,
497 0x606a, 0x6164, 0x6bbc, 0x73cf, 0x811a, 0x89ba, 0x89d2, 0x95a3,
498 0x4f83, 0x520a, 0x58be, 0x5978, 0x59e6, 0x5e72, 0x5e79, 0x61c7,
499 0x63c0, 0x6746, 0x67ec, 0x687f, 0x6f97, 0x764e, 0x770b, 0x78f5,
500 0x7a08, 0x7aff, 0x7c21, 0x809d, 0x826e, 0x8271, 0x8aeb, 0x9593,
501 0x4e6b, 0x559d, 0x66f7, 0x6e34, 0x78a3, 0x7aed, 0x845b, 0x8910,
502 0x874e, 0x97a8, 0x52d8, 0x574e, 0x582a, 0x5d4c, 0x611f, 0x61be,
503 0x6221, 0x6562, 0x67d1, 0x6a44, 0x6e1b, 0x7518, 0x75b3, 0x76e3,
504 0x77b0, 0x7d3a, 0x90af, 0x9451, 0x9452, 0x9f95,
505 /* 0x4b */
506 0x5323, 0x5cac, 0x7532, 0x80db, 0x9240, 0x9598, 0x525b, 0x5808,
507 0x59dc, 0x5ca1, 0x5d17, 0x5eb7, 0x5f3a, 0x5f4a, 0x6177, 0x6c5f,
508 0x757a, 0x7586, 0x7ce0, 0x7d73, 0x7db1, 0x7f8c, 0x8154, 0x8221,
509 0x8591, 0x8941, 0x8b1b, 0x92fc, 0x964d, 0x9c47, 0x4ecb, 0x4ef7,
510 0x500b, 0x51f1, 0x584f, 0x6137, 0x613e, 0x6168, 0x6539, 0x69ea,
511 0x6f11, 0x75a5, 0x7686, 0x76d6, 0x7b87, 0x82a5, 0x84cb, 0xf900,
512 0x93a7, 0x958b, 0x5580, 0x5ba2, 0x5751, 0xf901, 0x7cb3, 0x7fb9,
513 0x91b5, 0x5028, 0x53bb, 0x5c45, 0x5de8, 0x62d2, 0x636e, 0x64da,
514 0x64e7, 0x6e20, 0x70ac, 0x795b, 0x8ddd, 0x8e1e, 0xf902, 0x907d,
515 0x9245, 0x92f8, 0x4e7e, 0x4ef6, 0x5065, 0x5dfe, 0x5efa, 0x6106,
516 0x6957, 0x8171, 0x8654, 0x8e47, 0x9375, 0x9a2b, 0x4e5e, 0x5091,
517 0x6770, 0x6840, 0x5109, 0x528d, 0x5292, 0x6aa2,
518 /* 0x4c */
519 0x77bc, 0x9210, 0x9ed4, 0x52ab, 0x602f, 0x8ff2, 0x5048, 0x61a9,
520 0x63ed, 0x64ca, 0x683c, 0x6a84, 0x6fc0, 0x8188, 0x89a1, 0x9694,
521 0x5805, 0x727d, 0x72ac, 0x7504, 0x7d79, 0x7e6d, 0x80a9, 0x898b,
522 0x8b74, 0x9063, 0x9d51, 0x6289, 0x6c7a, 0x6f54, 0x7d50, 0x7f3a,
523 0x8a23, 0x517c, 0x614a, 0x7b9d, 0x8b19, 0x9257, 0x938c, 0x4eac,
524 0x4fd3, 0x501e, 0x50be, 0x5106, 0x52c1, 0x52cd, 0x537f, 0x5770,
525 0x5883, 0x5e9a, 0x5f91, 0x6176, 0x61ac, 0x64ce, 0x656c, 0x666f,
526 0x66bb, 0x66f4, 0x6897, 0x6d87, 0x7085, 0x70f1, 0x749f, 0x74a5,
527 0x74ca, 0x75d9, 0x786c, 0x78ec, 0x7adf, 0x7af6, 0x7d45, 0x7d93,
528 0x8015, 0x803f, 0x811b, 0x8396, 0x8b66, 0x8f15, 0x9015, 0x93e1,
529 0x9803, 0x9838, 0x9a5a, 0x9be8, 0x4fc2, 0x5553, 0x583a, 0x5951,
530 0x5b63, 0x5c46, 0x60b8, 0x6212, 0x6842, 0x68b0,
531 /* 0x4d */
532 0x68e8, 0x6eaa, 0x754c, 0x7678, 0x78ce, 0x7a3d, 0x7cfb, 0x7e6b,
533 0x7e7c, 0x8a08, 0x8aa1, 0x8c3f, 0x968e, 0x9dc4, 0x53e4, 0x53e9,
534 0x544a, 0x5471, 0x56fa, 0x59d1, 0x5b64, 0x5c3b, 0x5eab, 0x62f7,
535 0x6537, 0x6545, 0x6572, 0x66a0, 0x67af, 0x69c1, 0x6cbd, 0x75fc,
536 0x7690, 0x777e, 0x7a3f, 0x7f94, 0x8003, 0x80a1, 0x818f, 0x82e6,
537 0x82fd, 0x83f0, 0x85c1, 0x8831, 0x88b4, 0x8aa5, 0xf903, 0x8f9c,
538 0x932e, 0x96c7, 0x9867, 0x9ad8, 0x9f13, 0x54ed, 0x659b, 0x66f2,
539 0x688f, 0x7a40, 0x8c37, 0x9d60, 0x56f0, 0x5764, 0x5d11, 0x6606,
540 0x68b1, 0x68cd, 0x6efe, 0x7428, 0x889e, 0x9be4, 0x6c68, 0xf904,
541 0x9aa8, 0x4f9b, 0x516c, 0x5171, 0x529f, 0x5b54, 0x5de5, 0x6050,
542 0x606d, 0x62f1, 0x63a7, 0x653b, 0x73d9, 0x7a7a, 0x86a3, 0x8ca2,
543 0x978f, 0x4e32, 0x5be1, 0x6208, 0x679c, 0x74dc,
544 /* 0x4e */
545 0x79d1, 0x83d3, 0x8a87, 0x8ab2, 0x8de8, 0x904e, 0x934b, 0x9846,
546 0x5ed3, 0x69e8, 0x85ff, 0x90e8, 0xf905, 0x51a0, 0x5b98, 0x5bec,
547 0x6163, 0x68fa, 0x6b3e, 0x704c, 0x742f, 0x74d8, 0x7ba1, 0x7f50,
548 0x83c5, 0x89c0, 0x8cab, 0x95dc, 0x9928, 0x522e, 0x605d, 0x62ec,
549 0x9002, 0x4f8a, 0x5149, 0x5321, 0x58d9, 0x5ee3, 0x66e0, 0x6d38,
550 0x709a, 0x72c2, 0x73d6, 0x7b50, 0x80f1, 0x945b, 0x5366, 0x639b,
551 0x7f6b, 0x4e56, 0x5080, 0x584a, 0x58de, 0x602a, 0x6127, 0x62d0,
552 0x69d0, 0x9b41, 0x5b8f, 0x7d18, 0x80b1, 0x8f5f, 0x4ea4, 0x50d1,
553 0x54ac, 0x55ac, 0x5b0c, 0x5da0, 0x5de7, 0x652a, 0x654e, 0x6821,
554 0x6a4b, 0x72e1, 0x768e, 0x77ef, 0x7d5e, 0x7ff9, 0x81a0, 0x854e,
555 0x86df, 0x8f03, 0x8f4e, 0x90ca, 0x9903, 0x9a55, 0x9bab, 0x4e18,
556 0x4e45, 0x4e5d, 0x4ec7, 0x4ff1, 0x5177, 0x52fe,
```

```

557 /* 0x4f */
558 0x5340, 0x53e3, 0x53e5, 0x548e, 0x5614, 0x5775, 0x57a2, 0x5bc7,
559 0x5d87, 0x5ed0, 0x61fc, 0x62d8, 0x6551, 0x67b8, 0x67e9, 0x69cb,
560 0x6b50, 0x6bc6, 0x6bec, 0x6c42, 0x6e9d, 0x7078, 0x72d7, 0x7396,
561 0x7403, 0x77bf, 0x77e9, 0x7a76, 0x7d7f, 0x8009, 0x81fc, 0x8205,
562 0x820a, 0x82df, 0x8862, 0x8b33, 0x8cfc, 0x8ec0, 0x9011, 0x90b1,
563 0x9264, 0x92b6, 0x99d2, 0x9a45, 0x9ce9, 0x9dd7, 0x9f9c, 0x570b,
564 0x5c40, 0x83ca, 0x97a0, 0x97ab, 0x9eb4, 0x541b, 0x7a98, 0x7fa4,
565 0x88d9, 0x8ecd, 0x90e1, 0x5800, 0x5c48, 0x6398, 0x7a9f, 0x5bae,
566 0x5f13, 0x7a79, 0x7aae, 0x828e, 0x8eac, 0x5026, 0x5238, 0x52f8,
567 0x5377, 0x5708, 0x62f3, 0x6372, 0x6b0a, 0x6dc3, 0x7737, 0x53a5,
568 0x7357, 0x8568, 0x8e76, 0x95d5, 0x673a, 0x6ac3, 0x6f70, 0x8a6d,
569 0x8ecc, 0x994b, 0xf906, 0x6677, 0x6b78, 0x8cb4,
570 /* 0x50 */
571 0x9b3c, 0xf907, 0x53eb, 0x572d, 0x594e, 0x63c6, 0x69fb, 0x73ea,
572 0x7845, 0x7aba, 0x7ac5, 0x7cfe, 0x8475, 0x898f, 0x8d73, 0x9035,
573 0x95a8, 0x52fb, 0x5747, 0x7547, 0x7b60, 0x83cc, 0x921e, 0xf908,
574 0x6a58, 0x514b, 0x524b, 0x5287, 0x621f, 0x68d8, 0x6975, 0x9699,
575 0x50c5, 0x52a4, 0x52e4, 0x61c3, 0x65a4, 0x6839, 0x69ff, 0x747e,
576 0x7b4b, 0x82b9, 0x83eb, 0x89b2, 0x8b39, 0x8fd1, 0x9949, 0xf909,
577 0x4eca, 0x5997, 0x64d2, 0x6611, 0x6a8e, 0x7434, 0x7981, 0x79bd,
578 0x82a9, 0x887e, 0x887f, 0x895f, 0xf90a, 0x9326, 0x4f0b, 0x53ca,
579 0x6025, 0x6271, 0x6c72, 0x7d1a, 0x7d66, 0x4e98, 0x5162, 0x77dc,
580 0x80af, 0x4f01, 0x4f0e, 0x5176, 0x5180, 0x55dc, 0x5668, 0x573b,
581 0x57fa, 0x57fc, 0x5914, 0x5947, 0x5993, 0x5bc4, 0x5c90, 0x5d0e,
582 0x5df1, 0x5e7e, 0x5fcc, 0x6280, 0x65d7, 0x65e3,
583 /* 0x51 */
584 0x671e, 0x671f, 0x675e, 0x68cb, 0x68c4, 0x6a5f, 0x6b3a, 0x6c23,
585 0x6c7d, 0x6c82, 0x6dc7, 0x7398, 0x7426, 0x742a, 0x7482, 0x74a3,
586 0x7578, 0x757f, 0x75f7, 0x7881, 0x78ef, 0x7941, 0x7947, 0x7948, 0x797a,
587 0x7b95, 0x7d00, 0x7dba, 0x7f88, 0x8006, 0x802d, 0x808c, 0x8a18,
588 0x8b4f, 0x8c48, 0x8d77, 0x9321, 0x9324, 0x98e2, 0x9951, 0x9a0e,
589 0x9a0f, 0x9a65, 0x9e92, 0x7dca, 0x4f76, 0x5409, 0x62ee, 0x6854,
590 0x91d1, 0x55ab, 0x513a, 0xf90b, 0xf90c, 0x5a1c, 0x61e6, 0xf90d,
591 0x62cf, 0x62ff, 0xf90e, 0xf90f, 0xf910, 0xf911, 0xf912, 0xf913,
592 0x90a3, 0xf914, 0xf915, 0xf916, 0xf917, 0xf918, 0x8afe, 0xf919,
593 0xf91a, 0xf91b, 0xf91c, 0x6696, 0xf91d, 0x7156, 0xf91e, 0xf91f,
594 0x96e3, 0xf920, 0x634f, 0x637a, 0x5357, 0xf921, 0x678f, 0x6960,
595 0x6e73, 0xf922, 0x7537, 0xf923, 0xf924, 0xf925,
596 /* 0x52 */
597 0x7d0d, 0xf926, 0xf927, 0x8872, 0x56ca, 0x5a18, 0xf928, 0xf929,
598 0xf92a, 0xf92b, 0xf92c, 0x4e43, 0xf92d, 0x5167, 0x5948, 0x67f0,
599 0x8010, 0xf92e, 0x5973, 0x5e74, 0x649a, 0x79ca, 0x5ff5, 0x606c,
600 0x62c8, 0x637b, 0x5be7, 0x5bd7, 0x52aa, 0xf92f, 0x5974, 0x5f29,
601 0x6012, 0xf930, 0xf931, 0xf932, 0x7459, 0xf933, 0xf934, 0xf935,
602 0xf936, 0xf937, 0xf938, 0x99d1, 0xf939, 0xf93a, 0xf93b, 0xf93c,
603 0xf93d, 0xf93e, 0xf93f, 0xf940, 0xf941, 0xf942, 0xf943, 0x6fc3,
604 0xf944, 0xf945, 0x81bf, 0x8fb2, 0x60f1, 0xf946, 0xf947, 0x8166,
605 0xf948, 0xf949, 0x5c3f, 0xf94a, 0xf94b, 0xf94c, 0xf94d, 0xf94e,
606 0xf94f, 0xf950, 0xf951, 0x5ae9, 0x8a25, 0x677b, 0x7d10, 0xf952,
607 0xf953, 0xf954, 0xf955, 0xf956, 0xf957, 0x80fd, 0xf958, 0xf959,
608 0x5c3c, 0x6ce5, 0x533f, 0x6eba, 0x591a, 0x8336,
609 /* 0x53 */
610 0x4e39, 0x4eb6, 0x4f46, 0x55ae, 0x5718, 0x58c7, 0x5f56, 0x65b7,
611 0x65e6, 0x6a80, 0x6bb5, 0x6e4d, 0x77ed, 0x7aef, 0x7c1e, 0x7dde,
612 0x86cb, 0x8892, 0x9132, 0x935b, 0x64bb, 0x6fbe, 0x737a, 0x75b8,
613 0x9054, 0x5556, 0x574d, 0x61ba, 0x64d4, 0x66c7, 0x6de1, 0x6e5b,
614 0x6f6d, 0x6fb9, 0x75f0, 0x8043, 0x81bd, 0x8541, 0x8983, 0x8ac7,
615 0x8b5a, 0x931f, 0x6c93, 0x7553, 0x7b54, 0x8e0f, 0x905d, 0x5510,
616 0x5802, 0x5858, 0x5e62, 0x6207, 0x649e, 0x68e0, 0x7576, 0x7cd6,
617 0x87b3, 0x9ee8, 0x4ee3, 0x5788, 0x576e, 0x5927, 0x5c0d, 0x5cb1,
618 0x5e36, 0x5f85, 0x6234, 0x64e1, 0x73b3, 0x81fa, 0x888b, 0x8cb8,
619 0x968a, 0x9edb, 0x5b85, 0x5fb7, 0x60b3, 0x5012, 0x5200, 0x5230,
620 0x5716, 0x5835, 0x5857, 0x5c0e, 0x5c60, 0x5c6f, 0x5d8b, 0x5ea6,
621 0x5f92, 0x60bc, 0x6311, 0x6389, 0x6417, 0x6843,
622 /* 0x54 */
623 0x68f9, 0x6ac2, 0x6dd8, 0x6e21, 0x6ed4, 0x6fe4, 0x71fe, 0x76dc,
624 0x7779, 0x79b1, 0x7a3b, 0x8404, 0x89a9, 0x8ced, 0x8df3, 0x8e48,
625 0x9003, 0x9014, 0x9053, 0x90fd, 0x934d, 0x9676, 0x97dc, 0x6bd2,
626 0x7006, 0x7258, 0x72a2, 0x7368, 0x7763, 0x79bf, 0x7be4, 0x7e9b,
627 0x8b80, 0x58a9, 0x60c7, 0x6566, 0x65fd, 0x66be, 0x6c8c, 0x711e,
628 0x71c9, 0x8c5a, 0x9813, 0x4e6d, 0x7a81, 0x4edd, 0x51ac, 0x51cd,
629 0x52d5, 0x540c, 0x61a7, 0x6771, 0x6850, 0x68df, 0x6d1e, 0x6f7c,
630 0x75bc, 0x77b3, 0x7ae5, 0x80f4, 0x8463, 0x9285, 0x515c, 0x6597,
631 0x675c, 0x6793, 0x75d8, 0x7ac7, 0x8373, 0xf95a, 0x8c46, 0x9017,
632 0x982d, 0x5c6f, 0x81c0, 0x829a, 0x9041, 0x906f, 0x920d, 0x5f97,
633 0x5d9d, 0x6a59, 0x71c8, 0x767b, 0x7b49, 0x85e4, 0x8b04, 0x9127,
634 0x9a30, 0x5587, 0x61f6, 0xf95b, 0x7669, 0x7f85,
635 /* 0x55 */
636 0x863f, 0x87ba, 0x88f8, 0x908f, 0xf95c, 0x6d1b, 0x70d9, 0x73de,
637 0x7d61, 0x843d, 0xf95d, 0x916a, 0x99f1, 0xf95e, 0x4e82, 0x5375,
638 0x6b04, 0x6b12, 0x703e, 0x721b, 0x862d, 0x9e1e, 0x524c, 0x8fa3,
639 0x5d50, 0x64e5, 0x652c, 0x6b16, 0x6feb, 0x7c43, 0x7e9c, 0x85cd,
640 0x8964, 0x89bd, 0x62c9, 0x81d8, 0x881f, 0x5eca, 0x6717, 0x6d6a,
641 0x72fc, 0x7405, 0x746f, 0x8782, 0x90de, 0x4f86, 0x5d0d, 0x5fa0,
642 0x840a, 0x51b7, 0x63a0, 0x7565, 0x4eae, 0x5006, 0x5169, 0x51c9,
643 0x6881, 0x6a11, 0x7cae, 0x7cb1, 0x7ce7, 0x826f, 0x8ad2, 0x8f1b,

```

```
644 0x91cf, 0x4fb6, 0x5137, 0x52f5, 0x5442, 0x5eec, 0x616e, 0x623e,
645 0x65c5, 0x6ada, 0x6ffe, 0x792a, 0x85dc, 0x8823, 0x95ad, 0x9a62,
646 0x9a6a, 0x9e97, 0x9ece, 0x529b, 0x66c6, 0x6b77, 0x701d, 0x792b,
647 0x8f62, 0x9742, 0x6190, 0x6200, 0x6523, 0x6f23,
648 /* 0x56 */
649 0x7149, 0x7489, 0x7df4, 0x806f, 0x84ee, 0x8f26, 0x9023, 0x934a,
650 0x51bd, 0x5217, 0x52a3, 0x6d0c, 0x70c8, 0x88c2, 0x5ec9, 0x6582,
651 0x6bae, 0x6fc2, 0x7c3e, 0x7375, 0x4ee4, 0x4f36, 0x56f9, 0xf95f,
652 0x5cba, 0x5dba, 0x601c, 0x73b2, 0x7b2d, 0x7f9a, 0x7fce, 0x8046,
653 0x901e, 0x9234, 0x96f6, 0x9748, 0x9818, 0x9f61, 0x4f8b, 0x6fa7,
654 0x79ae, 0x91b4, 0x96b7, 0x52de, 0xf960, 0x6488, 0x64c4, 0x6ad3,
655 0x6f5e, 0x7018, 0x7210, 0x76e7, 0x8001, 0x8606, 0x865c, 0x8def,
656 0x8f05, 0x9732, 0x9b6f, 0x9dfa, 0x9e75, 0x788c, 0x797f, 0x7da0,
657 0x83c9, 0x9304, 0x9e7f, 0x9e93, 0x8ad6, 0x58df, 0x5f04, 0x6727,
658 0x7027, 0x74cf, 0x7c60, 0x807e, 0x5121, 0x7028, 0x7262, 0x78ca,
659 0x8cc2, 0x8cda, 0x8cf4, 0x96f7, 0x4e86, 0x50da, 0x5bee, 0x5ed6,
660 0x6599, 0x71ce, 0x7642, 0x77ad, 0x804a, 0x84fc,
661 /* 0x57 */
662 0x907c, 0x9b27, 0x9f8d, 0x58d8, 0x5a41, 0x5c62, 0x6a13, 0x6dda,
663 0x6f0f, 0x763b, 0x7d2f, 0x7e37, 0x851e, 0x8938, 0x93e4, 0x964b,
664 0x5289, 0x65d2, 0x67f3, 0x69b4, 0x6d41, 0x6e9c, 0x700f, 0x7409,
665 0x7460, 0x7559, 0x7624, 0x786b, 0x8b2c, 0x985e, 0x516d, 0x622e,
666 0x9678, 0x4f96, 0x502b, 0x5d19, 0x6dea, 0x7db8, 0x8f2a, 0x5f8b,
667 0x6144, 0x6817, 0xf961, 0x9686, 0x52d2, 0x808b, 0x51dc, 0x51cc,
668 0x695e, 0x7a1c, 0x7dbe, 0x83f1, 0x9675, 0x4fda, 0x5229, 0x5398,
669 0x540f, 0x550e, 0x5c65, 0x60a7, 0x674e, 0x68a8, 0x6d6c, 0x7281,
670 0x72f8, 0x7406, 0x7483, 0xf962, 0x75e2, 0x7c6c, 0xf7f9, 0x7fb8,
671 0x8389, 0x88cf, 0x88e1, 0x91cc, 0x91d0, 0x96e2, 0x9bc9, 0x541d,
672 0x6f7e, 0x71d0, 0x7498, 0x85fa, 0x8eaa, 0x96a3, 0x9c57, 0x9e9f,
673 0x6797, 0x6dcb, 0x7433, 0x81e8, 0x9716, 0x782c,
674 /* 0x58 */
675 0x7acb, 0x7b20, 0x7c92, 0x6469, 0x746a, 0x75f2, 0x78bc, 0x78e8,
676 0x99ac, 0x9b54, 0x9ebb, 0x5bde, 0x5e55, 0x6f20, 0x819c, 0x83ab,
677 0x9088, 0x4e07, 0x534d, 0x5a29, 0x5dd2, 0x5f4e, 0x6162, 0x633d,
678 0x6669, 0x66fc, 0x6eff, 0x6f2b, 0x7063, 0x779e, 0x842c, 0x8513,
679 0x883b, 0x8f13, 0x9945, 0x9c3b, 0x551c, 0x62b9, 0x672b, 0x6cab,
680 0x8309, 0x896a, 0x977a, 0x4ea1, 0x5984, 0x5fd8, 0x5fd9, 0x671b,
681 0x7db2, 0x7f54, 0x8292, 0x832b, 0x83bd, 0x8f1e, 0x9099, 0x57cb,
682 0x59b9, 0x5a92, 0x5bd0, 0x6627, 0x679a, 0x6885, 0x6bcf, 0x7164,
683 0x7f75, 0x8cb7, 0x8ce3, 0x9081, 0x9b45, 0x8108, 0x8c8a, 0x964c,
684 0x9a40, 0x9ea5, 0x5b5f, 0x6c13, 0x731b, 0x76f2, 0x76df, 0x840c,
685 0x51aa, 0x8993, 0x514d, 0x5195, 0x52c9, 0x68c9, 0x6c94, 0x7704,
686 0x7720, 0x7dbf, 0x7dec, 0x9762, 0x9eb5, 0x6ec5,
687 /* 0x59 */
688 0x8511, 0x51a5, 0x540d, 0x547d, 0x660e, 0x669d, 0x6927, 0x6e9f,
689 0x76bf, 0x7791, 0x8317, 0x84c2, 0x879f, 0x9169, 0x9298, 0x9cf4,
690 0x8882, 0x4fae, 0x5192, 0x52df, 0x59c6, 0x5e3d, 0x6155, 0x6478,
691 0x6479, 0x66ae, 0x67d0, 0x6a21, 0x6bcd, 0x6bdb, 0x725f, 0x7261,
692 0x7441, 0x7738, 0x77db, 0x8017, 0x82bc, 0x8305, 0x8b00, 0x8b28,
693 0x8c8c, 0x6728, 0x6c90, 0x7267, 0x76ee, 0x7766, 0x7a46, 0x9da9,
694 0x6b7f, 0x6c92, 0x5922, 0x6726, 0x8499, 0x536f, 0x5893, 0x5999,
695 0x5edf, 0x63cf, 0x6634, 0x6773, 0x6e3a, 0x732b, 0x7ad7, 0x82d7,
696 0x9328, 0x52d9, 0x5deb, 0x61ae, 0x61cb, 0x620a, 0x62c7, 0x64ab,
697 0x65e0, 0x6959, 0x6b66, 0x6bcb, 0x7121, 0x73f7, 0x755d, 0x7e46,
698 0x821e, 0x8302, 0x856a, 0x8aa3, 0x8cbf, 0x9727, 0x9d61, 0x58a8,
699 0x9ed8, 0x5011, 0x520e, 0x543b, 0x554f, 0x6587,
700 /* 0x5a */
701 0x6c76, 0x7d0a, 0x7d0b, 0x805e, 0x868a, 0x9580, 0x96ef, 0x52ff,
702 0x6c95, 0x7269, 0x5473, 0x5a9a, 0x5c3e, 0x5d4b, 0x5f4c, 0x5fae,
703 0x672a, 0x68b6, 0x6963, 0x6e3c, 0x6e44, 0x7709, 0x7c73, 0x7f8e,
704 0x8587, 0x8b0e, 0x8ff7, 0x9761, 0x9ef4, 0x5cb7, 0x60b6, 0x610d,
705 0x61ab, 0x654f, 0x65fb, 0x65fc, 0x6c11, 0x6cef, 0x739f, 0x73c9,
706 0x7de1, 0x9594, 0x5bcb, 0x871c, 0x8b10, 0x525d, 0x535a, 0x62cd,
707 0x640f, 0x64b2, 0x6734, 0x6a38, 0x6cca, 0x73c0, 0x749e, 0x7b94,
708 0x7c95, 0x7e1b, 0x818a, 0x8236, 0x8584, 0x8feb, 0x96f9, 0x99c1,
709 0x4f34, 0x534a, 0x53cd, 0x53db, 0x62cc, 0x642c, 0x6500, 0x6591,
710 0x69c3, 0x6cee, 0x6f58, 0x73ed, 0x7554, 0x7622, 0x76e4, 0x76fc,
711 0x78d0, 0x78fb, 0x792c, 0x7d46, 0x822c, 0x87e0, 0x8fd4, 0x9812,
712 0x98ef, 0x52c3, 0x62d4, 0x64a5, 0x6e24, 0x6f51,
713 /* 0x5b */
714 0x767c, 0x8dcb, 0x91b1, 0x9262, 0x9aee, 0x9b43, 0x5023, 0x508d,
715 0x574a, 0x59a8, 0x5c28, 0x5e47, 0x5f77, 0x623f, 0x653e, 0x65b9,
716 0x65c1, 0x6609, 0x678b, 0x699c, 0x6ec2, 0x78c5, 0x7d21, 0x80aa,
717 0x8180, 0x822b, 0x82b3, 0x84a1, 0x868c, 0x8a2a, 0x8b17, 0x90a6,
718 0x9632, 0x9f90, 0x500d, 0x4ff3, 0xf963, 0x57f9, 0x5f98, 0x62dc,
719 0x6392, 0x676f, 0x6e43, 0x7119, 0x76c3, 0x80cc, 0x80da, 0x88f7,
720 0x88f5, 0x8919, 0x8ce0, 0x8f29, 0x914d, 0x966a, 0x4f2f, 0x4f70,
721 0x5e1b, 0x67cf, 0x6822, 0x767d, 0x767e, 0x9b44, 0x5e61, 0x6a0a,
722 0x7169, 0x71d4, 0x756a, 0xf964, 0x7e41, 0x8543, 0x85e9, 0x98dc,
723 0x4f10, 0x7b4f, 0x7f70, 0x95a5, 0x51e1, 0x5e06, 0x68b5, 0x6c3e,
724 0x6c4e, 0x6cdb, 0x72af, 0x7bc4, 0x8303, 0x6cd5, 0x743a, 0x50fb,
725 0x5288, 0x58c1, 0x64d8, 0x6a97, 0x74a7, 0x7656,
726 /* 0x5c */
727 0x78a7, 0x8617, 0x95e2, 0x9739, 0xf965, 0x535e, 0x5f01, 0x8b8a,
728 0x8fa8, 0x8faf, 0x908a, 0x5225, 0x77a5, 0x9c49, 0x9f08, 0x4e19,
729 0x5002, 0x5175, 0x5c5b, 0x5e77, 0x661e, 0x663a, 0x67c4, 0x68c5,
730 0x70b3, 0x7501, 0x75c5, 0x79c9, 0x7add, 0x8f27, 0x9920, 0x9a08,
```

```

731 0x4fdd, 0x5821, 0x5831, 0x5bf6, 0x666e, 0x6b65, 0x6d11, 0x6e7a,
732 0x6f7d, 0x73e4, 0x752b, 0x83e9, 0x88dc, 0x8913, 0x8b5c, 0x8f14,
733 0x4f0f, 0x50d5, 0x5310, 0x535c, 0x5b93, 0x5fa9, 0x670d, 0x798f,
734 0x8179, 0x832f, 0x8514, 0x8907, 0x8986, 0x8f39, 0x8f3b, 0x99a5,
735 0x9c12, 0x672c, 0x4e76, 0x4ff8, 0x5949, 0x5c01, 0x5cef, 0x5cf0,
736 0x6367, 0x68d2, 0x70fd, 0x71a2, 0x742b, 0x7e2b, 0x84ec, 0x8702,
737 0x9022, 0x92d2, 0x9cf3, 0x4e0d, 0x4ed8, 0x4fef, 0x5085, 0x5256,
738 0x526f, 0x5426, 0x5490, 0x57e0, 0x592b, 0x5a66,
739 /* 0x5d */
740 0x5b5a, 0x5b75, 0x5bcc, 0x5e9c, 0xf966, 0x6276, 0x6577, 0x65a7,
741 0x6d6e, 0x6ea5, 0x7236, 0x7b26, 0x7c3f, 0x7f36, 0x8150, 0x8151,
742 0x819a, 0x8240, 0x8299, 0x83a9, 0x8a03, 0x8ca0, 0x8ce6, 0x8cfb,
743 0x8d74, 0x8dba, 0x90e8, 0x91dc, 0x961c, 0x9644, 0x99d9, 0x9ce7,
744 0x5317, 0x5206, 0x5429, 0x5674, 0x58b3, 0x5954, 0x596e, 0x5fff,
745 0x61a4, 0x626e, 0x6610, 0x6c7e, 0x711a, 0x76c6, 0x7c89, 0x7cde,
746 0x7d1b, 0x82ac, 0x8cc1, 0x96f0, 0xf967, 0x4f5b, 0x5f17, 0x5f7f,
747 0x62c2, 0x5d29, 0x670b, 0x68da, 0x787c, 0x7e43, 0x9d6c, 0x4e15,
748 0x5099, 0x5315, 0x532a, 0x5351, 0x5983, 0x5a62, 0x5e87, 0x60b2,
749 0x618a, 0x6249, 0x6279, 0x6590, 0x6787, 0x69a7, 0x6bd4, 0x6bd6,
750 0x6bd7, 0x6bd8, 0x6cb8, 0xf968, 0x7435, 0x75fa, 0x7812, 0x7891,
751 0x79d5, 0x79d8, 0x7c83, 0x7dcb, 0x7fe1, 0x80a5,
752 /* 0x5e */
753 0x813e, 0x81c2, 0x83f2, 0x871a, 0x88e8, 0x8ab9, 0x8b6c, 0x8cbb,
754 0x9119, 0x975e, 0x98db, 0x9f3b, 0x56ac, 0x5b2a, 0x5f6c, 0x658c,
755 0x6ab3, 0x6baf, 0x6d5c, 0x6ff1, 0x7015, 0x725d, 0x73ad, 0x8ca7,
756 0x8cd3, 0x983b, 0x6191, 0x6c37, 0x8058, 0x9a01, 0x4e4d, 0x4e8b,
757 0x4e9b, 0x4ed5, 0x4f3a, 0x4f3c, 0x4f7f, 0x4fd9, 0x50ff, 0x53f2,
758 0x53f8, 0x5506, 0x55e3, 0x56db, 0x58eb, 0x5962, 0x5a11, 0x5beb,
759 0x5bfa, 0x5c04, 0x5df3, 0x5e2b, 0x5f99, 0x601d, 0x6368, 0x659c,
760 0x65af, 0x67f6, 0x67fb, 0x68ad, 0x6b7b, 0x6c99, 0x6cd7, 0x6e23,
761 0x7009, 0x7345, 0x7802, 0x793e, 0x7940, 0x7960, 0x79c1, 0x7be9,
762 0x7d17, 0x7d72, 0x8086, 0x820d, 0x838e, 0x84d1, 0x86c7, 0x88df,
763 0x8a50, 0x8a5e, 0x8b1d, 0x8c8c, 0x8d66, 0x8fad, 0x90aa, 0x98fc,
764 0x99df, 0x9e9d, 0x524a, 0xf969, 0x6714, 0xf96a,
765 /* 0x5f */
766 0x5098, 0x522a, 0x5c71, 0x6563, 0x6c55, 0x73ca, 0x7523, 0x759d,
767 0x7b97, 0x849c, 0x9178, 0x9730, 0x4e77, 0x6492, 0x6bba, 0x715e,
768 0x85a9, 0x4e09, 0xf96b, 0x6749, 0x68ee, 0x6e17, 0x829f, 0x8518,
769 0x886b, 0x63f7, 0x6f81, 0x9212, 0x98af, 0x4e0a, 0x50b7, 0x50cf,
770 0x511f, 0x5546, 0x55aa, 0x5617, 0x5b40, 0x5c19, 0x5ce0, 0x5e38,
771 0x5e8a, 0x5ea0, 0x5ec2, 0x60f3, 0x6851, 0x6a61, 0x6e58, 0x723d,
772 0x7240, 0x72c0, 0x76f8, 0x7965, 0x7bb1, 0x7fd4, 0x88f3, 0x89f4,
773 0x8a73, 0x8c61, 0x8cde, 0x971c, 0x585e, 0x74bd, 0x8cfd, 0x55c7,
774 0xf96c, 0x7a61, 0x7d22, 0x8272, 0x7272, 0x751f, 0x7525, 0xf96d,
775 0x7b19, 0x5885, 0x58fb, 0x5dbc, 0x5e8f, 0x5eb6, 0x5f90, 0x6055,
776 0x6292, 0x637f, 0x654d, 0x6691, 0x66d9, 0x66f8, 0x6816, 0x68f2,
777 0x7280, 0x745e, 0x7b6e, 0x7d6e, 0x7dd6, 0x7f72,
778 /* 0x60 */
779 0x80e5, 0x8212, 0x85af, 0x897f, 0x8a93, 0x901d, 0x92e4, 0x9ecd,
780 0x9f20, 0x5915, 0x596d, 0x5e2d, 0x60dc, 0x6614, 0x6673, 0x6790,
781 0x6c50, 0x6dc5, 0x6f5f, 0x77f3, 0x78a9, 0x84c6, 0x91cb, 0x932b,
782 0x4ed9, 0x50ca, 0x5148, 0x5584, 0x5b0b, 0x5ba3, 0x6247, 0x657e,
783 0x65cb, 0x6e32, 0x717d, 0x7401, 0x7444, 0x7487, 0x74bf, 0x766c,
784 0x79aa, 0x7dda, 0x7e55, 0x7fa8, 0x817a, 0x81b3, 0x8239, 0x861a,
785 0x87ec, 0x8a75, 0x8de3, 0x9078, 0x9291, 0x9425, 0x994d, 0x9bae,
786 0x5368, 0x5c51, 0x6954, 0x6cc4, 0x6d29, 0x6e2b, 0x820c, 0x859b,
787 0x893b, 0x8a2d, 0x8aaa, 0x96ea, 0x9f67, 0x5261, 0x66b9, 0x6bb2,
788 0x7e96, 0x87fe, 0x8d0d, 0x9583, 0x965d, 0x651d, 0x6d89, 0x71ee,
789 0xf96e, 0x57ce, 0x59d3, 0x5bac, 0x6027, 0x60fa, 0x6210, 0x661f,
790 0x665f, 0x7329, 0x73f9, 0x76db, 0x7701, 0x7b6c,
791 /* 0x61 */
792 0x8056, 0x8072, 0x8165, 0x8aa0, 0x9192, 0x4e16, 0x52e2, 0x6b72,
793 0x6d17, 0x7a05, 0x7b39, 0x7d30, 0xf96f, 0x8cb0, 0x53ec, 0x562f,
794 0x5851, 0x5bb5, 0x5c0f, 0x5c11, 0x5de2, 0x6240, 0x6383, 0x6414,
795 0x662d, 0x68b3, 0x6cbc, 0x6d88, 0x6eaf, 0x701f, 0x70a4, 0x71d2,
796 0x7526, 0x758f, 0x758e, 0x7619, 0x7b11, 0x7be0, 0x7c2b, 0x7d20,
797 0x7d39, 0x852c, 0x856d, 0x8607, 0x8a34, 0x900d, 0x9061, 0x90b5,
798 0x92b7, 0x97f6, 0x9a37, 0x4fd7, 0x5c6c, 0x675f, 0x6d91, 0x7c9f,
799 0x7e8c, 0x8b16, 0x8d16, 0x901f, 0x5b6b, 0x5dfd, 0x640d, 0x84c0,
800 0x905c, 0x98e1, 0x7387, 0x5b8b, 0x609a, 0x677e, 0x6dde, 0x8a1f,
801 0x8aa6, 0x9001, 0x980c, 0x5237, 0xf970, 0x7051, 0x788e, 0x9396,
802 0x8870, 0x91d7, 0x4fee, 0x53d7, 0x55fd, 0x56da, 0x5782, 0x58fd,
803 0x5ac2, 0x5b88, 0x5cab, 0x5cc0, 0x5e25, 0x6101,
804 /* 0x62 */
805 0x620d, 0x624b, 0x6388, 0x641c, 0x6536, 0x6578, 0x6a39, 0x6b8a,
806 0x6c34, 0x6d19, 0x6f31, 0x71e7, 0x72e9, 0x7378, 0x7407, 0x74b2,
807 0x7626, 0x7761, 0x79c0, 0x7a57, 0x7aea, 0x7cb9, 0x7d8f, 0x7dac,
808 0x7e61, 0x7f9e, 0x8129, 0x8331, 0x8490, 0x84da, 0x85ea, 0x8896,
809 0x8ab0, 0x8b90, 0x8f38, 0x9042, 0x9083, 0x916c, 0x9296, 0x92b9,
810 0x968b, 0x96a7, 0x96a8, 0x96d6, 0x9700, 0x9808, 0x9996, 0x9ad3,
811 0x9b1a, 0x53d4, 0x587e, 0x5919, 0x5b70, 0x5bbb, 0x6dd1, 0x6f5a,
812 0x719f, 0x7421, 0x74b9, 0x8085, 0x83fd, 0x5de1, 0x5f87, 0x5faa,
813 0x6042, 0x65ec, 0x6812, 0x69fe, 0x6a53, 0x6b89, 0x6d35, 0x6df3,
814 0x73e3, 0x76fe, 0x77ac, 0x7b4d, 0x7d14, 0x8123, 0x821c, 0x8340,
815 0x84f4, 0x8563, 0x8a62, 0x8ac4, 0x9187, 0x931e, 0x9806, 0x99b4,
816 0x620c, 0x8853, 0x8ff0, 0x9265, 0x5d07, 0x5d27,
817 /* 0x63 */

```

```
818 0x5d69, 0x745f, 0x819d, 0x8768, 0x6fd5, 0x62fe, 0x7fd2, 0x8936,
819 0x8972, 0x4e1e, 0x4e58, 0x50e7, 0x52dd, 0x5347, 0x627f, 0x6607,
820 0x7e69, 0x8805, 0x965e, 0x4f8d, 0x5319, 0x5636, 0x59cb, 0x5aa4,
821 0x5c38, 0x5c4e, 0x5c4d, 0x5e02, 0x5f11, 0x6043, 0x65bd, 0x662f,
822 0x6642, 0x67be, 0x67f4, 0x731c, 0x77e2, 0x793a, 0x7fc5, 0x8494,
823 0x84cd, 0x8996, 0x8a66, 0x8a69, 0x8ae1, 0x8c55, 0x8c7a, 0x57f4,
824 0x5bd4, 0x5f0f, 0x606f, 0x62ed, 0x690d, 0x6b96, 0x6e5c, 0x7184,
825 0x7bd2, 0x8755, 0x8b58, 0x8efe, 0x98df, 0x98fe, 0x4f38, 0x4f81,
826 0x4fe1, 0x547b, 0x5a20, 0x5bb8, 0x613c, 0x65b0, 0x6668, 0x71fc,
827 0x7533, 0x795e, 0x7d33, 0x814e, 0x81e3, 0x8398, 0x85aa, 0x85ce,
828 0x8703, 0x8a0a, 0x8eab, 0x8f9b, 0xf971, 0x8fc5, 0x5931, 0x5ba4,
829 0x5be6, 0x6089, 0x5be9, 0x5c0b, 0x5fc3, 0x6c81,
830 /* 0x64 */
831 0xf972, 0x6df1, 0x700b, 0x751a, 0x82af, 0x8af6, 0x4ec0, 0x5341,
832 0xf973, 0x96d9, 0x6c0f, 0x4e9e, 0x4fc4, 0x5152, 0x555e, 0x5a25,
833 0x5ce8, 0x6211, 0x7259, 0x82bd, 0x83aa, 0x86fe, 0x8859, 0x8ald,
834 0x963f, 0x96c5, 0x9913, 0x9d09, 0x9d5d, 0x580a, 0x5cb3, 0x5dbd,
835 0x5e44, 0x60e1, 0x6115, 0x63e1, 0x6a02, 0x6e25, 0x9102, 0x9354,
836 0x984e, 0x9c10, 0x9f77, 0x5b89, 0x5cb8, 0x6309, 0x664f, 0x6848,
837 0x773c, 0x96c1, 0x978d, 0x9854, 0x9b9f, 0x65a1, 0x8b01, 0x8ecb,
838 0x95bc, 0x5535, 0x8ca9, 0x5dd6, 0x5eb5, 0x6697, 0x764c, 0x83f4,
839 0x95c7, 0x58d3, 0x62bc, 0x72ce, 0x9d28, 0x4ef0, 0x592e, 0x600f,
840 0x663b, 0x6b83, 0x79e7, 0x9d26, 0x5393, 0x54c0, 0x57c3, 0x5d16,
841 0x611b, 0x66d6, 0x6daf, 0x788d, 0x827e, 0x9698, 0x9744, 0x5384,
842 0x627c, 0x6396, 0x6db2, 0x7e0a, 0x814b, 0x984d,
843 /* 0x65 */
844 0x6afb, 0x7f4c, 0x9daf, 0x9e1a, 0x4e5f, 0x503b, 0x51b6, 0x591c,
845 0x60f9, 0x63f6, 0x6930, 0x723a, 0x8036, 0xf974, 0x91ce, 0x5f31,
846 0xf975, 0xf976, 0x7d04, 0x82e5, 0x846f, 0x84bb, 0x85e5, 0x8e8d,
847 0xf977, 0x4f6f, 0xf978, 0xf979, 0x584e, 0x5b43, 0x6059, 0x63da,
848 0x6518, 0x656d, 0x6698, 0xf97a, 0x694a, 0x6a23, 0x6d0b, 0x7001,
849 0x716c, 0x75d2, 0x760d, 0x79b3, 0x7a70, 0xf97b, 0x7f8a, 0xf97c,
850 0x8944, 0xf97d, 0x8b93, 0x91c0, 0x967d, 0xf97e, 0x990a, 0x5704,
851 0x5fa1, 0x65bc, 0x6f01, 0x7600, 0x79a6, 0x8a9e, 0x99ad, 0x9b5a,
852 0x9f6c, 0x5104, 0x61b6, 0x6291, 0x6a8d, 0x81c6, 0x5043, 0x5830,
853 0x5f66, 0x7109, 0x8a00, 0x8afa, 0x5b7c, 0x8616, 0x4ffa, 0x513c,
854 0x56b4, 0x5944, 0x63a9, 0x6df9, 0x5daa, 0x696d, 0x5186, 0x4e88,
855 0x4f59, 0xf97f, 0xf980, 0xf981, 0x5982, 0xf982,
856 /* 0x66 */
857 0xf983, 0x6b5f, 0x6c5d, 0xf984, 0x74b5, 0x7916, 0xf985, 0x8207,
858 0x8245, 0x8339, 0x8f3f, 0x8f5d, 0xf986, 0x9918, 0xf987, 0xf988,
859 0xf989, 0x4ea6, 0xf98a, 0x57df, 0x5f79, 0x6613, 0xf98b, 0xf98c,
860 0x75ab, 0x7e79, 0x8b6f, 0xf98d, 0x9006, 0x9a5b, 0x56a5, 0x5827,
861 0x59f8, 0x5a1f, 0x5bb4, 0xf98e, 0x5ef6, 0xf98f, 0xf990, 0x6350,
862 0x633b, 0xf991, 0x693d, 0x6c87, 0x6cbf, 0x6d8e, 0x6d93, 0x6df5,
863 0x6f14, 0xf992, 0x70df, 0x7136, 0x7159, 0xf993, 0x71c3, 0x71d5,
864 0xf994, 0x784f, 0x786f, 0xf995, 0x7b75, 0x7de3, 0xf996, 0x7e2f,
865 0xf997, 0x884d, 0x8edf, 0xf998, 0xf999, 0xf99a, 0x925b, 0xf99b,
866 0x9c6f, 0xf99c, 0xf99d, 0xf99e, 0x6085, 0x6d85, 0xf99f, 0x71b1,
867 0xf9a0, 0xf9a1, 0x95b1, 0x53ad, 0xf9a2, 0xf9a3, 0xf9a4, 0x67d3,
868 0xf9a5, 0x708e, 0x7130, 0x7430, 0x8276, 0x82d2,
869 /* 0x67 */
870 0xf9a6, 0x95bb, 0x9ae5, 0x9e7d, 0x66c4, 0xf9a7, 0x71c1, 0x8449,
871 0xf9a8, 0xf9a9, 0x584b, 0xf9ab, 0x5db8, 0x5f71, 0xf9ac,
872 0x6620, 0x668e, 0x6979, 0x69ae, 0x6c38, 0x6cf3, 0x6e36, 0x6f41,
873 0x6fda, 0x701b, 0x702f, 0x7150, 0x71df, 0x7370, 0xf9ad, 0x745b,
874 0xf9ae, 0x74d4, 0x76c8, 0x7a4e, 0x7e93, 0xf9af, 0xf9b0, 0x82f1,
875 0x8a60, 0x8fce, 0xf9b1, 0x9348, 0xf9b2, 0x9719, 0xf9b3, 0xf9b4,
876 0x4e42, 0x502a, 0xf9b5, 0x5208, 0x53e1, 0x66f3, 0x6c6d, 0x6fca,
877 0x730a, 0x777f, 0x7a62, 0x82ae, 0x85dd, 0x8602, 0xf9b6, 0x88d4,
878 0x8a63, 0x8b7d, 0x8c6b, 0xf9b7, 0x92b3, 0xf9b8, 0x9713, 0x9810,
879 0x4e94, 0x4f0d, 0x4fc9, 0x50b2, 0x5348, 0x543e, 0x5433, 0x55da,
880 0x5862, 0x58ba, 0x5967, 0x5a1b, 0x5be4, 0x609f, 0xf9b9, 0x61ca,
881 0x6556, 0x65ff, 0x6664, 0x68a7, 0x6c5a, 0x6fb3,
882 /* 0x68 */
883 0x70cf, 0x71ac, 0x7352, 0x7b7d, 0x8708, 0x8aa4, 0x9c32, 0x9f07,
884 0x5c4b, 0x6c83, 0x7344, 0x7389, 0x923a, 0x6eab, 0x7465, 0x761f,
885 0x7a69, 0x7e15, 0x860a, 0x5140, 0x58c5, 0x64c1, 0x74ee, 0x7515,
886 0x7670, 0x7fc1, 0x9095, 0x96cd, 0x9954, 0x6e26, 0x74e6, 0x7aa9,
887 0x7aaa, 0x81e5, 0x86d9, 0x8778, 0x8a1b, 0x5a49, 0x5b8c, 0x5b9b,
888 0x68a1, 0x6900, 0x6d63, 0x73a9, 0x7413, 0x742c, 0x7897, 0x7de9,
889 0x7feb, 0x8118, 0x8155, 0x839e, 0x8c4c, 0x962e, 0x9811, 0x66f0,
890 0x5f80, 0x65fa, 0x6789, 0x6c6a, 0x738b, 0x502d, 0x5a03, 0x6b6a,
891 0x77ee, 0x5916, 0x5d6c, 0x5dc3, 0x7325, 0x754f, 0xf9ba, 0xf9bb,
892 0x50e5, 0x51f9, 0x582f, 0x592d, 0x5996, 0x59da, 0x5be5, 0xf9bc,
893 0xf9bd, 0x5da2, 0x62d7, 0x6416, 0x6493, 0x64fe, 0xf9be, 0x66dc,
894 0xf9bf, 0x6a48, 0xf9c0, 0x71ff, 0x7464, 0xf9c1,
895 /* 0x69 */
896 0x7a88, 0x7aaf, 0x7e47, 0x7e5e, 0x8000, 0x8170, 0xf9c2, 0x87ef,
897 0x8981, 0x8b20, 0x9059, 0xf9c3, 0x9080, 0x9952, 0x617e, 0x6b32,
898 0x6d74, 0x7e1f, 0x8925, 0x8fb1, 0x4fd1, 0x50ad, 0x5197, 0x52c7,
899 0x57c7, 0x5889, 0x5bb9, 0x5eb8, 0x6142, 0x6995, 0x6d8c, 0x6e67,
900 0x6eb6, 0x7194, 0x7462, 0x7528, 0x752c, 0x8073, 0x8338, 0x84c9,
901 0x8e0a, 0x9394, 0x93de, 0xf9c4, 0x4e8e, 0x4f51, 0x5076, 0x512a,
902 0x53c8, 0x53cb, 0x53f3, 0x5b87, 0x5bd3, 0x5c24, 0x611a, 0x6182,
903 0x65f4, 0x725b, 0x7397, 0x7440, 0x76c2, 0x7950, 0x7991, 0x79b9,
904 0x7d06, 0x7fbd, 0x828b, 0x85d5, 0x865e, 0x8fc2, 0x9047, 0x90f5,
```

```

905 0x91ea, 0x9685, 0x96e8, 0x96e9, 0x52d6, 0x5f67, 0x65ed, 0x6631,
906 0x682f, 0x715c, 0x7a36, 0x90c1, 0x980a, 0x4e91, 0xf9c5, 0x6a52,
907 0x6b9e, 0x6f90, 0x7189, 0x8018, 0x82b8, 0x8553,
908 /* 0x6a */
909 0x904b, 0x9695, 0x96f2, 0x97fb, 0x851a, 0x9b31, 0x4e90, 0x718a,
910 0x96c4, 0x5143, 0x539f, 0x54e1, 0x5713, 0x5712, 0x57a3, 0x5a9b,
911 0x5ac4, 0x5bc3, 0x6028, 0x613f, 0x63f4, 0x6c85, 0x6d39, 0x6e72,
912 0x6e90, 0x7230, 0x733f, 0x7457, 0x82d1, 0x8881, 0x8f45, 0x9060,
913 0xf9c6, 0x9662, 0x9858, 0x9d1b, 0x6708, 0x8d8a, 0x925e, 0x4f4d,
914 0x5049, 0x50de, 0x5371, 0x570d, 0x59d4, 0x5a01, 0x5c09, 0x6170,
915 0x6690, 0x6e2d, 0x7232, 0x744b, 0x7def, 0x80c3, 0x840e, 0x8466,
916 0x853f, 0x875f, 0x885b, 0x8918, 0x8b02, 0x9055, 0x97cb, 0x9b4f,
917 0x4e73, 0x4f91, 0x5112, 0x516a, 0xf9c7, 0x552f, 0x55a9, 0x5b7a,
918 0x5ba5, 0x5e7c, 0x5e7d, 0x5ebe, 0x60a0, 0x60df, 0x6108, 0x6109,
919 0x63c4, 0x6538, 0x6709, 0xf9c8, 0x67d4, 0x67da, 0xf9c9, 0x6961,
920 0x6962, 0x6cb9, 0x6d27, 0xf9ca, 0x6e38, 0xf9cb,
921 /* 0x6b */
922 0x6fe1, 0x7336, 0x7337, 0xf9cc, 0x745c, 0x7531, 0xf9cd, 0x7652,
923 0xf9ce, 0xf9cf, 0x7dad, 0x81fe, 0x8438, 0x88d5, 0x8a98, 0x8adb,
924 0x8aed, 0x8e30, 0x8e42, 0x904a, 0x903e, 0x907a, 0x9149, 0x91c9,
925 0x936e, 0xf9d0, 0xf9d1, 0x5809, 0xf9d2, 0x6bd3, 0x8089, 0x80b2,
926 0xf9d3, 0xf9d4, 0x5141, 0x596b, 0x5c39, 0xf9d5, 0xf9d6, 0x6f64,
927 0x73a7, 0x80e4, 0x8d07, 0xf9d7, 0x9217, 0x9588, 0xf9d8, 0xf9d9,
928 0xf9da, 0xf9db, 0x807f, 0x620e, 0x701c, 0x7d68, 0x878d, 0xf9dc,
929 0x57a0, 0x6069, 0x6147, 0x6bb7, 0x8abe, 0x9280, 0x96b1, 0x4e59,
930 0x541f, 0x6deb, 0x852d, 0x9670, 0x97f3, 0x98ee, 0x63d6, 0x6ce3,
931 0x9091, 0x51dd, 0x61c9, 0x81ba, 0x9df9, 0x4f9d, 0x501a, 0x5100,
932 0x5b9c, 0x610f, 0x61ff, 0x64ec, 0x6905, 0x6bc5, 0x7591, 0x77e3,
933 0x7fa9, 0x8264, 0x858f, 0x87fb, 0x8863, 0x8abc,
934 /* 0x6c */
935 0x8b70, 0x91ab, 0x4e8c, 0x4ee5, 0x4f0a, 0xf9dd, 0xf9de, 0x5937,
936 0x59e8, 0xf9df, 0x5df2, 0x5f1b, 0x5f5b, 0x6021, 0xf9e0, 0xf9e1,
937 0xf9e2, 0xf9e3, 0x723e, 0x73e5, 0xf9e4, 0x7570, 0x75cd, 0xf9e5,
938 0x79fb, 0xf9e6, 0x800c, 0x8033, 0x8084, 0x82e1, 0x8351, 0xf9e7,
939 0xf9e8, 0x8cbd, 0x8cb3, 0x9087, 0xf9e9, 0xf9ea, 0x98f4, 0x990c,
940 0xf9eb, 0xf9ec, 0x7037, 0x76ca, 0x7fca, 0x7fcc, 0x7ffc, 0x8b1a,
941 0x4eba, 0x4ec1, 0x5203, 0x5370, 0xf9ed, 0x54bd, 0x56e0, 0x59fb,
942 0x5bc5, 0x5f15, 0x5fcd, 0x6e6e, 0xf9ee, 0xf9ef, 0x7d6a, 0x8335,
943 0xf9f0, 0x8693, 0x8a8d, 0xf9f1, 0x976d, 0x9777, 0xf9f2, 0xf9f3,
944 0x4e00, 0x4f5a, 0x4f7e, 0x58f9, 0x65e5, 0x6ea2, 0x9038, 0x93b0,
945 0x99b9, 0x4efb, 0x58ec, 0x598a, 0x59d9, 0x6041, 0xf9f4, 0xf9f5,
946 0x7a14, 0xf9f6, 0x834f, 0x8cc3, 0x5165, 0x5344,
947 /* 0x6d */
948 0xf9f7, 0xf9f8, 0xf9f9, 0x4ecd, 0x5269, 0x5b55, 0x82bf, 0x4ed4,
949 0x523a, 0x54a8, 0x59c9, 0x59f9, 0x5b50, 0x5b57, 0x5b5c, 0x6063,
950 0x6148, 0x6ecb, 0x7099, 0x716e, 0x7386, 0x74f7, 0x75b5, 0x78c1,
951 0x7d2b, 0x8005, 0x81ea, 0x8328, 0x8517, 0x85c9, 0x8aee, 0x8cc7,
952 0x96cc, 0x4f5c, 0x52fa, 0x56bc, 0x65ab, 0x6628, 0x707c, 0x70b8,
953 0x7235, 0x7dbd, 0x828d, 0x914c, 0x96c0, 0x9d72, 0x5b71, 0x68e7,
954 0x6b98, 0x6f7a, 0x76de, 0x5c91, 0x66ab, 0x6f5b, 0x7bb4, 0x7c2a,
955 0x8836, 0x96dc, 0x4e08, 0x4ed7, 0x5320, 0x5834, 0x58bb, 0x58ef,
956 0x596c, 0x5c07, 0x5e33, 0x5e84, 0x5f35, 0x638c, 0x66b2, 0x6756,
957 0x6a1f, 0x6aa3, 0x6b0c, 0x6f3f, 0x7246, 0xf9fa, 0x7350, 0x748b,
958 0x7ae0, 0x7ca7, 0x8178, 0x81df, 0x81e7, 0x838a, 0x846c, 0x8523,
959 0x8594, 0x85cf, 0x88dd, 0x8d13, 0x91ac, 0x9577,
960 /* 0x6e */
961 0x969c, 0x518d, 0x54c9, 0x5728, 0x5bb0, 0x624d, 0x6750, 0x683d,
962 0x6893, 0x663d, 0x6ed3, 0x707d, 0x7e21, 0x88c1, 0x8ca1, 0x8f09,
963 0x9f4b, 0x9f4e, 0x722d, 0x7b8f, 0x8acd, 0x931a, 0x4f47, 0x4f4e,
964 0x5132, 0x5480, 0x59d0, 0x5e95, 0x62b5, 0x6775, 0x696e, 0x6a17,
965 0x6cae, 0x6e1a, 0x72d9, 0x732a, 0x75bd, 0x7bb8, 0x7d35, 0x82e7,
966 0x83f9, 0x8457, 0x85f7, 0x8a5b, 0x8caf, 0x8e87, 0x9019, 0x90b8,
967 0x96ce, 0x9f5f, 0x52e3, 0x540a, 0x5ae1, 0x5bc2, 0x6458, 0x6575,
968 0x6ef4, 0x72c4, 0xf9fb, 0x7684, 0x7a4d, 0x7b1b, 0x7c4d, 0x7e3e,
969 0x7fdf, 0x837b, 0x8b2b, 0x8cca, 0x8d64, 0x8del, 0x8e5f, 0x8fea,
970 0x8ff9, 0x9069, 0x93d1, 0x4f43, 0x4f7a, 0x50b3, 0x5168, 0x5178,
971 0x524d, 0x526a, 0x5861, 0x587c, 0x5960, 0x5c08, 0x5c55, 0x5edb,
972 0x609b, 0x6230, 0x6813, 0x6bbf, 0x6c08, 0x6fbl,
973 /* 0x6f */
974 0x714e, 0x7420, 0x7530, 0x7538, 0x7551, 0x7672, 0x7b4c, 0x7b8b,
975 0x7bad, 0x7bc6, 0x7e8f, 0x8a6e, 0x8f3e, 0x8f49, 0x923f, 0x9293,
976 0x9322, 0x942b, 0x96fb, 0x985a, 0x986b, 0x991e, 0x5207, 0x622a,
977 0x6298, 0x6d59, 0x7664, 0x7aca, 0x7bc0, 0x7d76, 0x5360, 0x5cbe,
978 0x5e97, 0x6f38, 0x70b9, 0x7c98, 0x9711, 0x9b8e, 0x9ede, 0x63a5,
979 0x647a, 0x8776, 0x4e01, 0x4e95, 0x4ead, 0x505c, 0x5075, 0x5448,
980 0x59c3, 0x5b9a, 0x5e40, 0x5ead, 0x5ef7, 0x5f81, 0x60c5, 0x633a,
981 0x653f, 0x6574, 0x65cc, 0x6676, 0x6678, 0x67fe, 0x6968, 0x6a89,
982 0x6b63, 0x6c40, 0x6dc0, 0x6de8, 0x6e1f, 0x6e5e, 0x701e, 0x70a1,
983 0x738e, 0x73fd, 0x753a, 0x775b, 0x7887, 0x798e, 0x7a0b, 0x7a7d,
984 0x7cbe, 0x7d8e, 0x8247, 0x8a02, 0x8aea, 0x8c9e, 0x912d, 0x914a,
985 0x91d8, 0x9266, 0x92cc, 0x9320, 0x9706, 0x9756,
986 /* 0x70 */
987 0x975c, 0x9802, 0x9f0e, 0x5236, 0x5291, 0x557c, 0x5824, 0x5e1d,
988 0x5f1f, 0x608c, 0x63d0, 0x68af, 0x6fdf, 0x796d, 0x7b2c, 0x81cd,
989 0x85ba, 0x88fd, 0x8af8, 0x8e44, 0x918d, 0x9664, 0x969b, 0x973d,
990 0x984c, 0x9f4a, 0x4fce, 0x5146, 0x51cb, 0x52a9, 0x5632, 0x5f14,
991 0x5f6b, 0x63aa, 0x64cd, 0x65e9, 0x6641, 0x66fa, 0x66f9, 0x671d,

```

```

992 0x689d, 0x68d7, 0x69fd, 0x6f15, 0x6f6e, 0x7167, 0x71e5, 0x722a,
993 0x74aa, 0x773a, 0x7956, 0x795a, 0x79df, 0x7a20, 0x7a95, 0x7c97,
994 0x7cdf, 0x7d44, 0x7e70, 0x8087, 0x85fb, 0x86a4, 0x8a54, 0x8abf,
995 0x8d99, 0x8e81, 0x9020, 0x906d, 0x91e3, 0x963b, 0x96d5, 0x9ce5,
996 0x65cf, 0x7c07, 0x8db3, 0x93c3, 0x5b58, 0x5c0a, 0x5352, 0x62d9,
997 0x731d, 0x5027, 0x5b97, 0x5f9e, 0x60b0, 0x616b, 0x68d5, 0x6dd9,
998 0x742e, 0x7a2e, 0x7d42, 0x7d9c, 0x7e31, 0x816b,
999 /* 0x71 */
1000 0x8e2a, 0x8e35, 0x937e, 0x9418, 0x4f50, 0x5750, 0x5de6, 0x5ea7,
1001 0x632b, 0x7f6a, 0x4e3b, 0x4f4f, 0x4f8f, 0x505a, 0x59dd, 0x80c4,
1002 0x546a, 0x5468, 0x55fe, 0x594f, 0x5b99, 0x5dde, 0x5eda, 0x665d,
1003 0x6731, 0x67f1, 0x682a, 0x6ce8, 0x6d32, 0x6e4a, 0x6f8d, 0x70b7,
1004 0x73e0, 0x7587, 0x7c4c, 0x7d02, 0x7d2c, 0x7da2, 0x821f, 0x86db,
1005 0x8a3b, 0x8a85, 0x8d70, 0x8e8a, 0x8f33, 0x9031, 0x914e, 0x9152,
1006 0x9444, 0x99d0, 0x7af9, 0x7ca5, 0x4fca, 0x5101, 0x51c6, 0x57c8,
1007 0x5bef, 0x5cfb, 0x6659, 0x6a3d, 0x6d5a, 0x6e96, 0x6fec, 0x710c,
1008 0x756f, 0x7ae3, 0x8822, 0x9021, 0x9075, 0x96cb, 0x99ff, 0x8301,
1009 0x4e2d, 0x4ef2, 0x8846, 0x91cd, 0x537d, 0x6adb, 0x696b, 0x6c41,
1010 0x847a, 0x589e, 0x618e, 0x66fe, 0x62ef, 0x70dd, 0x7511, 0x75c7,
1011 0x7e52, 0x84b8, 0x8b49, 0x8d08, 0x4e4b, 0x53ea,
1012 /* 0x72 */
1013 0x54ab, 0x5730, 0x5740, 0x5fd7, 0x6301, 0x6307, 0x646f, 0x652f,
1014 0x65e8, 0x667a, 0x679d, 0x67b3, 0x6b62, 0x6c60, 0x6c9a, 0x6f2c,
1015 0x77e5, 0x7825, 0x7949, 0x7957, 0x7d19, 0x80a2, 0x8102, 0x81f3,
1016 0x829d, 0x82b7, 0x8718, 0x8a8c, 0xf9fc, 0x8d04, 0x8dbe, 0x9072,
1017 0x76f4, 0x7a19, 0x7a37, 0x7e54, 0x8077, 0x5507, 0x55d4, 0x5875,
1018 0x632f, 0x6422, 0x6649, 0x664b, 0x686d, 0x699b, 0x6b84, 0x6d25,
1019 0x6eb1, 0x73cd, 0x7468, 0x74a1, 0x755b, 0x75b9, 0x76e1, 0x771e,
1020 0x778b, 0x79e6, 0x7e09, 0x7e1d, 0x81fb, 0x852f, 0x8897, 0x8a3a,
1021 0x8cd1, 0x8eeb, 0x8fb0, 0x9032, 0x93ad, 0x9663, 0x9673, 0x9707,
1022 0x4f84, 0x53f1, 0x59ea, 0x5ac9, 0x5e19, 0x684e, 0x74c6, 0x75be,
1023 0x79e9, 0x7a92, 0x81a3, 0x86ed, 0x8cea, 0x8dcc, 0x8fed, 0x659f,
1024 0x6715, 0xf9fd, 0x57f7, 0x7ddd, 0x8f2f,
1025 /* 0x73 */
1026 0x93f6, 0x96c6, 0x5fb5, 0x61f2, 0x6f84, 0x4e14, 0x4f98, 0x501f,
1027 0x53c9, 0x55df, 0x5d6f, 0x5dee, 0x6b21, 0x6b64, 0x78cb, 0x7b9a,
1028 0xf9fe, 0x8e49, 0x8eca, 0x906e, 0x6349, 0x643e, 0x7740, 0x7a84,
1029 0x932f, 0x947f, 0x9f6a, 0x64b0, 0x6faf, 0x71e6, 0x74a8, 0x74da,
1030 0x7ac4, 0x7c12, 0x7e82, 0x7cb2, 0x7e98, 0x8b9a, 0x8d0a, 0x947d,
1031 0x9910, 0x994c, 0x5239, 0x5bdf, 0x64e6, 0x672d, 0x72de, 0x50ed,
1032 0x53c3, 0x5879, 0x6158, 0x6159, 0x61fa, 0x65ac, 0x7ad9, 0x8b92,
1033 0x8b96, 0x5009, 0x5021, 0x5275, 0x5531, 0x5a3c, 0x5ee0, 0x5f70,
1034 0x6134, 0x655e, 0x660c, 0x6636, 0x66a2, 0x69cd, 0x6ec4, 0x6f32,
1035 0x7316, 0x7621, 0x7a93, 0x8139, 0x8259, 0x83d6, 0x84bc, 0x50b5,
1036 0x57f0, 0x5bc0, 0x5be8, 0x5f69, 0x63a1, 0x7826, 0x7db5, 0x83dc,
1037 0x8521, 0x91c7, 0x91f5, 0x518a, 0x67f5, 0x7b56,
1038 /* 0x74 */
1039 0x8cac, 0x51c4, 0x59bb, 0x60bd, 0x8655, 0x501c, 0xf9ff, 0x5254,
1040 0x5c3a, 0x617d, 0x621a, 0x62d3, 0x64f2, 0x65a5, 0x6ecc, 0x7620,
1041 0x810a, 0x8e60, 0x965f, 0x96bb, 0x4edf, 0x5343, 0x5598, 0x5929,
1042 0x5ddd, 0x64c5, 0x6cc9, 0x6dfa, 0x7394, 0x7a7f, 0x821b, 0x85a6,
1043 0x8ce4, 0x8e10, 0x9077, 0x91e7, 0x95e1, 0x9621, 0x97c6, 0x51f8,
1044 0x54f2, 0x5586, 0x5fb9, 0x64a4, 0x6f88, 0x7db4, 0x8f1f, 0x8f4d,
1045 0x9435, 0x50c9, 0x5c16, 0x6cbe, 0x6dfb, 0x751b, 0x77bb, 0x7c3d,
1046 0x7c64, 0x8a79, 0x8ac2, 0x581e, 0x59be, 0x5e16, 0x6377, 0x7252,
1047 0x758a, 0x776b, 0x8adc, 0x8cbc, 0x8f12, 0x5ef3, 0x6674, 0x6df8,
1048 0x807d, 0x83c1, 0x8acb, 0x9751, 0x9bd6, 0xfa00, 0x5243, 0x66ff,
1049 0x6d95, 0x6eef, 0x7de0, 0x8ae6, 0x902e, 0x905e, 0x9ad4, 0x521d,
1050 0x527f, 0x54e8, 0x6194, 0x6284, 0x62db, 0x68a2,
1051 /* 0x75 */
1052 0x6912, 0x695a, 0x6a35, 0x7092, 0x7126, 0x785d, 0x7901, 0x790e,
1053 0x79d2, 0x7a0d, 0x8096, 0x8278, 0x82d5, 0x8349, 0x8549, 0x8c82,
1054 0x8d85, 0x9162, 0x918b, 0x91ae, 0x4fc3, 0x56d1, 0x71ed, 0x77d7,
1055 0x8700, 0x89f8, 0x5bf8, 0x5fd6, 0x6751, 0x90a8, 0x53e2, 0x585a,
1056 0x5bf5, 0x60a4, 0x6181, 0x6460, 0x7e3d, 0x8070, 0x8525, 0x9283,
1057 0x64ae, 0x50ac, 0x5d14, 0x6700, 0x589c, 0x62bd, 0x63a8, 0x690e,
1058 0x6978, 0x6a1e, 0x6e6b, 0x76ba, 0x79cb, 0x82bb, 0x8429, 0x8acf,
1059 0x8da8, 0x8ffd, 0x9112, 0x914b, 0x919c, 0x9310, 0x9318, 0x939a,
1060 0x96db, 0x9a36, 0x9c0d, 0x4e11, 0x755c, 0x795d, 0x7afa, 0x7b51,
1061 0x7bc9, 0x7e2e, 0x84c4, 0x8e59, 0x8e74, 0x8ef8, 0x9010, 0x6625,
1062 0x693f, 0x7443, 0x51fa, 0x672e, 0x9edc, 0x5145, 0x5fe0, 0x6c96,
1063 0x87f2, 0x885d, 0x8877, 0x60b4, 0x81b5, 0x8403,
1064 /* 0x76 */
1065 0x8d05, 0x53d6, 0x5439, 0x5634, 0x5a36, 0x5c31, 0x708a, 0x7fe0,
1066 0x805a, 0x8106, 0x81ed, 0x8da3, 0x9189, 0x9a5f, 0x9df2, 0x5074,
1067 0x4ec4, 0x53a0, 0x60fb, 0x6e2c, 0x5c64, 0x4f88, 0x5024, 0x55e4,
1068 0x5cd9, 0x5e5f, 0x6065, 0x6894, 0x6cbb, 0x6dc4, 0x71be, 0x75d4,
1069 0x75f4, 0x7661, 0x7a1a, 0x7a49, 0x7dc7, 0x7dfb, 0x7f6e, 0x81f4,
1070 0x86a9, 0x8f1c, 0x96c9, 0x99b3, 0x9f52, 0x5247, 0x52c5, 0x98ed,
1071 0x89aa, 0x4e03, 0x67d2, 0x6f06, 0x4fb5, 0x5be2, 0x6795, 0x6c88,
1072 0x6d78, 0x741b, 0x7827, 0x91dd, 0x937c, 0x87c4, 0x79e4, 0x7a31,
1073 0x5feb, 0x4ed6, 0x54a4, 0x553e, 0x58ae, 0x59a5, 0x60f0, 0x6253,
1074 0x62d6, 0x6736, 0x6955, 0x8235, 0x9640, 0x99b1, 0x99dd, 0x502c,
1075 0x5353, 0x5544, 0x577c, 0xfa01, 0x6258, 0xfa02, 0x64e2, 0x666b,
1076 0x67dd, 0x6fc1, 0x6fef, 0x7422, 0x7438, 0x8a17,
1077 /* 0x77 */
1078 0x9438, 0x5451, 0x5606, 0x5766, 0x5f48, 0x619a, 0x6b4e, 0x7058,

```



```

1079 0x70ad, 0x7dbb, 0x8a95, 0x596a, 0x812b, 0x63a2, 0x7708, 0x803d,
1080 0x8caa, 0x5854, 0x642d, 0x69bb, 0x5b95, 0x5e11, 0x6e6f, 0xfa03,
1081 0x8569, 0x514c, 0x53f0, 0x592a, 0x6020, 0x614b, 0x6b86, 0x6c70,
1082 0x6cf0, 0x7b1e, 0x80ce, 0x82d4, 0x8dc6, 0x90b0, 0x98b1, 0xfa04,
1083 0x64c7, 0x6fa4, 0x6491, 0x6504, 0x514e, 0x5410, 0x571f, 0x8a0e,
1084 0x615f, 0x6876, 0xfa05, 0xfa05, 0x75db, 0x7b52, 0x7d71, 0x901a, 0x5806,
1085 0x69cc, 0x817f, 0x892a, 0x9000, 0x9839, 0x5078, 0x5957, 0x59ac,
1086 0x6295, 0x900f, 0x9b2a, 0x615d, 0x7279, 0x95d6, 0x5761, 0x5a46,
1087 0x5df4, 0x628a, 0x64ad, 0x64fa, 0x6777, 0x6ce2, 0x6d3e, 0x722c,
1088 0x7436, 0x7834, 0x7f77, 0x82ad, 0x8ddb, 0x9817, 0x5224, 0x5742,
1089 0x677f, 0x7248, 0x74e3, 0x8ca9, 0x8fa6, 0x9211,
1090 /* 0x78 */
1091 0x962a, 0x516b, 0x53ed, 0x634c, 0x4f69, 0x5504, 0x6096, 0x6557,
1092 0x6c9b, 0x6d7f, 0x724c, 0x72fd, 0x7a17, 0x8987, 0x8c9d, 0x5f6d,
1093 0x6f8e, 0x70f9, 0x81a8, 0x610e, 0x4fbf, 0x504f, 0x6241, 0x7247,
1094 0x7bc7, 0x7de8, 0x7fe9, 0x904d, 0x97ad, 0x9a19, 0x8cb6, 0x576a,
1095 0x5e73, 0x67b0, 0x840d, 0x8a55, 0x5420, 0x5b16, 0x5e63, 0x5ee2,
1096 0x5f0a, 0x6583, 0x80ba, 0x853d, 0x9589, 0x965b, 0x4f48, 0x5305,
1097 0x530d, 0x530f, 0x5486, 0x54fa, 0x5703, 0x5e03, 0x6016, 0x629b,
1098 0x62b1, 0x6355, 0xfa06, 0x6ce1, 0x6d66, 0x75b1, 0x7832, 0x80de,
1099 0x812f, 0x82de, 0x8461, 0x84b2, 0x888d, 0x8912, 0x900b, 0x92ea,
1100 0x98fd, 0x9b91, 0x5e45, 0x66b4, 0x66dd, 0x7011, 0x7206, 0xfa07,
1101 0x4ff5, 0x527d, 0x5f6a, 0x6153, 0x6753, 0x6a19, 0x6f02, 0x74e2,
1102 0x7968, 0x8868, 0x8c79, 0x98c7, 0x98c4, 0x9a43,
1103 /* 0x79 */
1104 0x54c1, 0x7a1f, 0x6953, 0x8af7, 0x8c4a, 0x98a8, 0x99ae, 0x5f7c,
1105 0x62ab, 0x75b2, 0x76ae, 0x88ab, 0x907f, 0x9642, 0x5339, 0x5f3c,
1106 0x5fc5, 0x6ccc, 0x73cc, 0x7562, 0x758b, 0x7b46, 0x82fe, 0x999d,
1107 0x4e4f, 0x903c, 0x4e0b, 0x4f55, 0x53a6, 0x590f, 0x5ec8, 0x6630,
1108 0x6cb3, 0x7455, 0x8377, 0x8766, 0x8cc0, 0x9050, 0x971e, 0x9c15,
1109 0x58d1, 0x5b78, 0x8650, 0x8b14, 0x9db4, 0x5bd2, 0x6068, 0x608d,
1110 0x65f1, 0x6c57, 0x6f22, 0x6fa3, 0x701a, 0x7f55, 0x7ff0, 0x9591,
1111 0x9592, 0x9650, 0x97d3, 0x5272, 0x8f44, 0x51d1, 0x542b, 0x54b8,
1112 0x5563, 0x558a, 0x6abb, 0x6db5, 0x7dd8, 0x8266, 0x929c, 0x9677,
1113 0x9e79, 0x5408, 0x54c8, 0x76d2, 0x86e4, 0x95a4, 0x95d4, 0x965c,
1114 0x4ea2, 0x4f09, 0x59ee, 0x5ae6, 0x5df7, 0x6052, 0x6297, 0x676d,
1115 0x6841, 0x6c86, 0x6e2f, 0x7f38, 0x809b, 0x822a,
1116 /* 0x7a */
1117 0xfa08, 0xfa09, 0x9805, 0x4ea5, 0x5055, 0x54b3, 0x5793, 0x595a,
1118 0x5b69, 0x5bb3, 0x61c8, 0x6977, 0x6d77, 0x7023, 0x87f9, 0x89e3,
1119 0x8a72, 0x8ae7, 0x9082, 0x99ed, 0x9ab8, 0x52be, 0x6838, 0x5016,
1120 0x5e78, 0x674f, 0x8347, 0x884c, 0x4eab, 0x5411, 0x56ae, 0x73e6,
1121 0x9115, 0x97ff, 0x9909, 0x9957, 0x9999, 0x5653, 0x589f, 0x865b,
1122 0x8a31, 0x61b2, 0x6af6, 0x737b, 0x8ed2, 0x6b47, 0x96aa, 0x9a57,
1123 0x5955, 0x7200, 0x8d6b, 0x9769, 0x4fd4, 0x5cfa, 0x5f26, 0x61f8,
1124 0x665b, 0x6ceb, 0x70ab, 0x7384, 0x73b9, 0x73fe, 0x7729, 0x774d,
1125 0x7d43, 0x7d62, 0x7e23, 0x8237, 0x8852, 0xfa0a, 0x8ce2, 0x9249,
1126 0x986f, 0x5b51, 0x7a74, 0x8840, 0x9801, 0x5acc, 0x4fe0, 0x5354,
1127 0x593e, 0x5cfd, 0x633e, 0x6d79, 0x72f9, 0x8105, 0x8107, 0x83a2,
1128 0x92cf, 0x9830, 0x4ea8, 0x5144, 0x5211, 0x578b,
1129 /* 0x7b */
1130 0x5f62, 0x6cc2, 0x6ece, 0x7005, 0x7050, 0x70af, 0x7192, 0x73e9,
1131 0x7469, 0x834a, 0x87a2, 0x8861, 0x9008, 0x90a2, 0x93a3, 0x99a8,
1132 0x516e, 0x5f57, 0x60e0, 0x6167, 0x66b3, 0x8559, 0x8e4a, 0x91af,
1133 0x978b, 0x4e4e, 0x4e92, 0x547c, 0x58d5, 0x58fa, 0x597d, 0x5cb5,
1134 0x5f27, 0x6236, 0x6248, 0x660a, 0x6667, 0x6beb, 0x6d69, 0x6dcf,
1135 0x6e56, 0x6ef8, 0x6f94, 0x6fe0, 0x6fe9, 0x705d, 0x72d0, 0x7425,
1136 0x745a, 0x74e0, 0x7693, 0x795c, 0x7cca, 0x7e1e, 0x80e1, 0x82a6,
1137 0x846b, 0x84bf, 0x864e, 0x865f, 0x8774, 0x8b77, 0x8c6a, 0x93ac,
1138 0x9800, 0x9865, 0x60d1, 0x6216, 0x9177, 0x5a5a, 0x66f0, 0x6df7,
1139 0x6e3e, 0x743f, 0x9b42, 0x5ffd, 0x60da, 0x7b0f, 0x5c4c, 0x5f18,
1140 0x6c5e, 0x6cd3, 0x6d2a, 0x70d8, 0x7d05, 0x8679, 0x8a0c, 0x9d3b,
1141 0x5316, 0x548c, 0x5b05, 0x6a3a, 0x706b, 0x7575,
1142 /* 0x7c */
1143 0x798d, 0x79be, 0x82b1, 0x83ef, 0x8a71, 0x8b41, 0x8ca8, 0x9774,
1144 0xfa0b, 0x64f4, 0x652b, 0x78ba, 0x78bb, 0x7a6b, 0x4e38, 0x559a,
1145 0x5950, 0x5ba6, 0x5e7b, 0x60a3, 0x63db, 0x6b61, 0x6665, 0x6853,
1146 0x6e19, 0x7165, 0x74b0, 0x7d08, 0x9084, 0x9a69, 0x9c25, 0x6d3b,
1147 0x6ed1, 0x733e, 0x8c41, 0x95ca, 0x51f0, 0x5e4c, 0x5fa8, 0x604d,
1148 0x60f6, 0x6130, 0x614c, 0x6643, 0x6644, 0x69a5, 0x6cc1, 0x6e5f,
1149 0x6ec9, 0x6f62, 0x714c, 0x749c, 0x7687, 0x7bc1, 0x7c27, 0x8352,
1150 0x8757, 0x9051, 0x968d, 0x9ec3, 0x532f, 0x56de, 0x5efb, 0x5f8a,
1151 0x6062, 0x6094, 0x61f7, 0x6666, 0x6703, 0x6a9c, 0x6dee, 0x6fae,
1152 0x7070, 0x736a, 0x7e6a, 0x81be, 0x8334, 0x86d4, 0x8aa8, 0x8cc4,
1153 0x5283, 0x7372, 0x5b96, 0x6a6b, 0x9404, 0x54ee, 0x5686, 0x5b5d,
1154 0x6548, 0x6585, 0x66c9, 0x689f, 0x6d8d, 0x6dc6,
1155 /* 0x7d */
1156 0x723b, 0x80b4, 0x9175, 0x9a4d, 0x4faf, 0x5019, 0x539a, 0x540e,
1157 0x543c, 0x5589, 0x55c5, 0x5e3f, 0x5f8c, 0x673d, 0x7166, 0x73dd,
1158 0x9005, 0x52db, 0x52f3, 0x5864, 0x58ce, 0x7104, 0x718f, 0x71fb,
1159 0x85b0, 0x8a13, 0x6688, 0x85a8, 0x55a7, 0x6684, 0x714a, 0x8431,
1160 0x5349, 0x5599, 0x6bc1, 0x5f59, 0x5fbd, 0x63ee, 0x6689, 0x7147,
1161 0x8af1, 0x8f1d, 0x9ebe, 0x4f11, 0x643a, 0x70cb, 0x7566, 0x8667,
1162 0x6064, 0x8b4e, 0x9df8, 0x5147, 0x51f6, 0x5308, 0x6d36, 0x80f8,
1163 0x9ed1, 0x6615, 0x6b23, 0x7098, 0x75d5, 0x5403, 0x5c79, 0x7d07,
1164 0x8a16, 0x6b20, 0x6b3d, 0x6b46, 0x5438, 0x6070, 0x6d3d, 0x7fd5,
1165 0x8208, 0x50d6, 0x51de, 0x559c, 0x566b, 0x56cd, 0x59ec, 0x5b09,

```

```
1166 0x5e0c, 0x6199, 0x6198, 0x6231, 0x665e, 0x66e6, 0x7199, 0x71b9,
1167 0x71ba, 0x72a7, 0x79a7, 0x7a00, 0x7fb2, 0x8a70,
1168 };
1169
1170 static int
1171 ksc5601_mbtowc (conv_t conv, ucs4_t *pwc, const unsigned char *s, int n)
1172 {
1173     unsigned char c1 = (s[0] & 0x7F);
1174     if ((c1 >= 0x21 && c1 <= 0x2c) || (c1 >= 0x30 && c1 <= 0x48) || (c1 >= 0x4a && c1 <= 0x7d)) {
1175         if (n >= 2) {
1176             unsigned char c2 = (s[1] & 0x7F);
1177             if (c2 >= 0x21 && c2 < 0x7f) {
1178                 unsigned int i = 94 * (c1 - 0x21) + (c2 - 0x21);
1179                 unsigned short wc = 0xffffd;
1180                 if (i < 1410) {
1181                     if (i < 1115)
1182                         wc = ksc5601_2uni_page21[i];
1183                     } else if (i < 3854) {
1184                         if (i < 3760)
1185                             wc = ksc5601_2uni_page30[i-1410];
1186                         } else {
1187                             if (i < 8742)
1188                                 wc = ksc5601_2uni_page4a[i-3854];
1189                         }
1190                     if (wc != 0xffffd) {
1191                         *pwc = (ucs4_t) wc;
1192                         return 2;
1193                     }
1194                 }
1195                 return RET_ILSEQ;
1196             }
1197             return RET_TOOFEW(0);
1198         }
1199         return RET_ILSEQ;
1200     }
1201 #endif /* NEED_TOWC */
1202
1203 #ifdef NEED_TOMB
1204 static const unsigned short ksc5601_2charset[8224] = {
1205     0x222e, 0x2234, 0x2157, 0x2127, 0x2823, 0x2129, 0x2146, 0x213e,
1206     0x2977, 0x2978, 0x2225, 0x2252, 0x2124, 0x222c, 0x2976, 0x282c,
1207     0x2879, 0x2876, 0x287a, 0x222f, 0x2821, 0x2822, 0x213f, 0x282a,
1208     0x282d, 0x292c, 0x2921, 0x2923, 0x2140, 0x292a, 0x292d, 0x2922,
1209     0x2824, 0x2924, 0x2925, 0x2826, 0x2926, 0x2927, 0x2828, 0x2928,
1210     0x2829, 0x2929, 0x2930, 0x282f, 0x292f, 0x282b, 0x292b, 0x282e,
1211     0x292e, 0x2227, 0x2230, 0x2228, 0x222b, 0x222a, 0x222d, 0x2229,
1212     0x2541, 0x2542, 0x2543, 0x2544, 0x2545, 0x2546, 0x2547, 0x2548,
1213     0x2549, 0x254a, 0x254b, 0x254c, 0x254d, 0x254e, 0x254f, 0x2550,
1214     0x2551, 0x2552, 0x2553, 0x2554, 0x2555, 0x2556, 0x2557, 0x2558,
1215     0x2561, 0x2562, 0x2563, 0x2564, 0x2565, 0x2566, 0x2567, 0x2568,
1216     0x2569, 0x256a, 0x256b, 0x256c, 0x256d, 0x256e, 0x256f, 0x2570,
1217     0x2571, 0x2572, 0x2573, 0x2574, 0x2575, 0x2576, 0x2577, 0x2578,
1218     0x2c27, 0x2c21, 0x2c22, 0x2c23, 0x2c24, 0x2c25, 0x2c26, 0x2c28,
1219     0x2c29, 0x2c2a, 0x2c2b, 0x2c2c, 0x2c2d, 0x2c2e, 0x2c2f, 0x2c30,
1220     0x2c31, 0x2c32, 0x2c33, 0x2c34, 0x2c35, 0x2c36, 0x2c37, 0x2c38,
1221     0x2c39, 0x2c3a, 0x2c3b, 0x2c3c, 0x2c3d, 0x2c3e, 0x2c3f, 0x2c40,
1222     0x2c41, 0x2c51, 0x2c52, 0x2c53, 0x2c54, 0x2c55, 0x2c56, 0x2c58,
1223     0x2c59, 0x2c5a, 0x2c5b, 0x2c5c, 0x2c5d, 0x2c5e, 0x2c5f, 0x2c60,
1224     0x2c61, 0x2c62, 0x2c63, 0x2c64, 0x2c65, 0x2c66, 0x2c67, 0x2c68,
1225     0x2c69, 0x2c6a, 0x2c6b, 0x2c6c, 0x2c6d, 0x2c6e, 0x2c6f, 0x2c70,
1226     0x2c71, 0x2c57, 0x212a, 0x212e, 0x212f, 0x2130, 0x2131, 0x2253,
1227     0x2254, 0x2125, 0x2126, 0x2236, 0x2147, 0x2148, 0x2158, 0x2979,
1228     0x297a, 0x297b, 0x297c, 0x297d, 0x297e, 0x2149, 0x2235, 0x2724,
1229     0x2260, 0x2265, 0x2262, 0x2759, 0x214a, 0x2877, 0x2878, 0x287b,
1230     0x287c, 0x287d, 0x287e, 0x2530, 0x2531, 0x2532, 0x2533, 0x2534,
1231     0x2535, 0x2536, 0x2537, 0x2538, 0x2539, 0x2521, 0x2522, 0x2523,
1232     0x2524, 0x2525, 0x2526, 0x2527, 0x2528, 0x2529, 0x252a, 0x2167,
1233     0x2168, 0x2166, 0x2169, 0x216a, 0x2255, 0x2258, 0x2256, 0x2259,
1234     0x2257, 0x2221, 0x2222, 0x2223, 0x2153, 0x2224, 0x2154, 0x2174,
1235     0x2175, 0x2233, 0x2232, 0x216e, 0x2170, 0x2144, 0x2150, 0x212b,
1236     0x217c, 0x217d, 0x217b, 0x217a, 0x2172, 0x2173, 0x2231, 0x2145,
1237     0x2171, 0x212d, 0x216f, 0x2156, 0x2141, 0x2155, 0x2142, 0x2143,
1238     0x216c, 0x216d, 0x2178, 0x2179, 0x2176, 0x2177, 0x2241, 0x2151,
1239     0x2152, 0x2867, 0x2868, 0x2869, 0x286a, 0x286b, 0x286c, 0x286d,
1240     0x286e, 0x286f, 0x2870, 0x2871, 0x2872, 0x2873, 0x2874, 0x2875,
1241     0x2967, 0x2968, 0x2969, 0x296a, 0x296b, 0x296c, 0x296d, 0x296e,
1242     0x296f, 0x2970, 0x2971, 0x2972, 0x2973, 0x2974, 0x2975, 0x297d,
1243     0x294e, 0x294f, 0x2950, 0x2951, 0x2952, 0x2953, 0x2954, 0x2955,
1244     0x2956, 0x2957, 0x2958, 0x2959, 0x295a, 0x295b, 0x295c, 0x295d,
1245     0x295e, 0x295f, 0x2960, 0x2961, 0x2962, 0x2963, 0x2964, 0x2965,
1246     0x2966, 0x284d, 0x284e, 0x284f, 0x2850, 0x2851, 0x2852, 0x2853,
1247     0x2854, 0x2855, 0x2856, 0x2857, 0x2858, 0x2859, 0x285a, 0x285b,
1248     0x285c, 0x285d, 0x285e, 0x285f, 0x2860, 0x2861, 0x2862, 0x2863,
1249     0x2864, 0x2865, 0x2866, 0x2621, 0x262c, 0x2622, 0x262d, 0x2623,
1250     0x2648, 0x2647, 0x262e, 0x2624, 0x2642, 0x2641, 0x262f, 0x2626,
1251     0x2646, 0x2645, 0x2631, 0x2625, 0x2644, 0x2643, 0x2630, 0x2627,
1252     0x263c, 0x2649, 0x264a, 0x2637, 0x264b, 0x264c, 0x2632, 0x2629,
```

1253 0x263e, 0x264d, 0x264e, 0x2639, 0x264f, 0x2650, 0x2634, 0x2628,
1254 0x2651, 0x2652, 0x2638, 0x263d, 0x2653, 0x2654, 0x2633, 0x262a,
1255 0x2655, 0x2656, 0x263a, 0x263f, 0x2657, 0x2658, 0x2635, 0x262b,
1256 0x2659, 0x265a, 0x263b, 0x265b, 0x265c, 0x2640, 0x265d, 0x265e,
1257 0x265f, 0x2660, 0x2661, 0x2662, 0x2663, 0x2664, 0x2636, 0x2246,
1258 0x2161, 0x2160, 0x2243, 0x2247, 0x2248, 0x224b, 0x224a, 0x2249,
1259 0x224c, 0x2163, 0x2162, 0x223a, 0x2239, 0x2165, 0x2164, 0x2238,
1260 0x2237, 0x215f, 0x215e, 0x2242, 0x215b, 0x215d, 0x215c, 0x2244,
1261 0x2245, 0x215a, 0x2159, 0x224f, 0x224e, 0x2250, 0x2251, 0x214f,
1262 0x214e, 0x223c, 0x223d, 0x2240, 0x223b, 0x223e, 0x223f, 0x224d,
1263 0x225b, 0x225c, 0x225d, 0x225a, 0x2121, 0x2122, 0x2123, 0x2128,
1264 0x2134, 0x2135, 0x2136, 0x2137, 0x2138, 0x2139, 0x213a, 0x213b,
1265 0x213c, 0x213d, 0x216b, 0x2132, 0x2133, 0x2a21, 0x2a22, 0x2a23,
1266 0x2a24, 0x2a25, 0x2a26, 0x2a27, 0x2a28, 0x2a29, 0x2a2a, 0x2a2b,
1267 0x2a2c, 0x2a2d, 0x2a2e, 0x2a2f, 0x2a30, 0x2a31, 0x2a32, 0x2a33,
1268 0x2a34, 0x2a35, 0x2a36, 0x2a37, 0x2a38, 0x2a39, 0x2a3a, 0x2a3b,
1269 0x2a3c, 0x2a3d, 0x2a3e, 0x2a3f, 0x2a40, 0x2a41, 0x2a42, 0x2a43,
1270 0x2a44, 0x2a45, 0x2a46, 0x2a47, 0x2a48, 0x2a49, 0x2a4a, 0x2a4b,
1271 0x2a4c, 0x2a4d, 0x2a4e, 0x2a4f, 0x2a50, 0x2a51, 0x2a52, 0x2a53,
1272 0x2a54, 0x2a55, 0x2a56, 0x2a57, 0x2a58, 0x2a59, 0x2a5a, 0x2a5b,
1273 0x2a5c, 0x2a5d, 0x2a5e, 0x2a5f, 0x2a60, 0x2a61, 0x2a62, 0x2a63,
1274 0x2a64, 0x2a65, 0x2a66, 0x2a67, 0x2a68, 0x2a69, 0x2a6a, 0x2a6b,
1275 0x2a6c, 0x2a6d, 0x2a6e, 0x2a6f, 0x2a70, 0x2a71, 0x2a72, 0x2a73,
1276 0x2b21, 0x2b22, 0x2b23, 0x2b24, 0x2b25, 0x2b26, 0x2b27, 0x2b28,
1277 0x2b29, 0x2b2a, 0x2b2b, 0x2b2c, 0x2b2d, 0x2b2e, 0x2b2f, 0x2b30,
1278 0x2b31, 0x2b32, 0x2b33, 0x2b34, 0x2b35, 0x2b36, 0x2b37, 0x2b38,
1279 0x2b39, 0x2b3a, 0x2b3b, 0x2b3c, 0x2b3d, 0x2b3e, 0x2b3f, 0x2b40,
1280 0x2b41, 0x2b42, 0x2b43, 0x2b44, 0x2b45, 0x2b46, 0x2b47, 0x2b48,
1281 0x2b49, 0x2b4a, 0x2b4b, 0x2b4c, 0x2b4d, 0x2b4e, 0x2b4f, 0x2b50,
1282 0x2b51, 0x2b52, 0x2b53, 0x2b54, 0x2b55, 0x2b56, 0x2b57, 0x2b58,
1283 0x2b59, 0x2b5a, 0x2b5b, 0x2b5c, 0x2b5d, 0x2b5e, 0x2b5f, 0x2b60,
1284 0x2b61, 0x2b62, 0x2b63, 0x2b64, 0x2b65, 0x2b66, 0x2b67, 0x2b68,
1285 0x2b69, 0x2b6a, 0x2b6b, 0x2b6c, 0x2b6d, 0x2b6e, 0x2b6f, 0x2b70,
1286 0x2b71, 0x2b72, 0x2b73, 0x2b74, 0x2b75, 0x2b76, 0x2421, 0x2422,
1287 0x2423, 0x2424, 0x2425, 0x2426, 0x2427, 0x2428, 0x2429, 0x242a,
1288 0x242b, 0x242c, 0x242d, 0x242e, 0x242f, 0x2430, 0x2431, 0x2432,
1289 0x2433, 0x2434, 0x2435, 0x2436, 0x2437, 0x2438, 0x2439, 0x243a,
1290 0x243b, 0x243c, 0x243d, 0x243e, 0x243f, 0x2440, 0x2441, 0x2442,
1291 0x2443, 0x2444, 0x2445, 0x2446, 0x2447, 0x2448, 0x2449, 0x244a,
1292 0x244b, 0x244c, 0x244d, 0x244e, 0x244f, 0x2450, 0x2451, 0x2452,
1293 0x2453, 0x2454, 0x2455, 0x2456, 0x2457, 0x2458, 0x2459, 0x245a,
1294 0x245b, 0x245c, 0x245d, 0x245e, 0x245f, 0x2460, 0x2461, 0x2462,
1295 0x2463, 0x2464, 0x2465, 0x2466, 0x2467, 0x2468, 0x2469, 0x246a,
1296 0x246b, 0x246c, 0x246d, 0x246e, 0x246f, 0x2470, 0x2471, 0x2472,
1297 0x2473, 0x2474, 0x2475, 0x2476, 0x2477, 0x2478, 0x2479, 0x247a,
1298 0x247b, 0x247c, 0x247d, 0x247e, 0x2931, 0x2932, 0x2933, 0x2934,
1299 0x2935, 0x2936, 0x2937, 0x2938, 0x2939, 0x293a, 0x293b, 0x293c,
1300 0x293d, 0x293e, 0x293f, 0x2940, 0x2941, 0x2942, 0x2943, 0x2944,
1301 0x2945, 0x2946, 0x2947, 0x2948, 0x2949, 0x294a, 0x294b, 0x294c,
1302 0x225f, 0x2831, 0x2832, 0x2833, 0x2834, 0x2835, 0x2836, 0x2837,
1303 0x2838, 0x2839, 0x283a, 0x283b, 0x283c, 0x283d, 0x283e, 0x283f,
1304 0x2840, 0x2841, 0x2842, 0x2843, 0x2844, 0x2845, 0x2846, 0x2847,
1305 0x2848, 0x2849, 0x284a, 0x284b, 0x284c, 0x225e, 0x2749, 0x274a,
1306 0x274b, 0x274c, 0x274d, 0x273a, 0x273b, 0x275c, 0x275d, 0x275e,
1307 0x2736, 0x2737, 0x2738, 0x2754, 0x2755, 0x2756, 0x2757, 0x2758,
1308 0x2721, 0x2722, 0x2723, 0x2725, 0x272b, 0x272c, 0x272d, 0x272e,
1309 0x272f, 0x2730, 0x2731, 0x2732, 0x2733, 0x2734, 0x2727, 0x2728,
1310 0x2729, 0x272a, 0x273d, 0x273e, 0x2765, 0x2766, 0x2767, 0x2768,
1311 0x2761, 0x2762, 0x2763, 0x273f, 0x2740, 0x2741, 0x2742, 0x2743,
1312 0x2744, 0x2745, 0x2746, 0x2747, 0x2748, 0x274e, 0x274f, 0x2750,
1313 0x2751, 0x2752, 0x2753, 0x275a, 0x275b, 0x2263, 0x276c, 0x2726,
1314 0x2760, 0x276f, 0x2261, 0x273c, 0x276d, 0x2735, 0x2739, 0x276a,
1315 0x276b, 0x275f, 0x2264, 0x276e, 0x276e, 0x2769, 0x6c69, 0x6f4b,
1316 0x7652, 0x5832, 0x6d5b, 0x5f32, 0x5f3e, 0x793b, 0x5c74, 0x7564,
1317 0x7326, 0x5d60, 0x6126, 0x4e78, 0x5c30, 0x632a, 0x7169, 0x4d7a,
1318 0x7c2f, 0x5321, 0x712b, 0x6751, 0x522c, 0x4e79, 0x717d, 0x5e3f,
1319 0x7b3a, 0x7939, 0x4e52, 0x632b, 0x6b60, 0x4e7a, 0x4b77, 0x6525,
1320 0x4a61, 0x544c, 0x6a61, 0x5c63, 0x5f2d, 0x4b6b, 0x552f, 0x5675,
1321 0x6578, 0x5e40, 0x6c23, 0x694d, 0x6a27, 0x6976, 0x7b3b, 0x6769,
1322 0x6f4c, 0x5066, 0x5e41, 0x642c, 0x584c, 0x7971, 0x4e5f, 0x7a24,
1323 0x6632, 0x7a7b, 0x7a3d, 0x4c48, 0x6f4d, 0x5555, 0x5322, 0x6c51,
1324 0x6427, 0x6c52, 0x7631, 0x4e7b, 0x5051, 0x4b3f, 0x6d24, 0x6d28,
1325 0x5e42, 0x7662, 0x6d5c, 0x5c75, 0x6039, 0x544e, 0x7435, 0x535b,
1326 0x5635, 0x6c24, 0x6466, 0x716a, 0x4b6c, 0x4b40, 0x6c72, 0x506a,
1327 0x7972, 0x6c25, 0x505f, 0x676a, 0x506b, 0x5c51, 0x5b69, 0x7d4c,
1328 0x5b57, 0x5a61, 0x5636, 0x635f, 0x5e43, 0x5e44, 0x4a21, 0x6e6c,
1329 0x5323, 0x6e37, 0x784f, 0x6a48, 0x6e38, 0x712c, 0x7125, 0x694e,
1330 0x793c, 0x6579, 0x6c6a, 0x5d56, 0x6d42, 0x7825, 0x653a, 0x5b58,
1331 0x4a22, 0x514d, 0x6e6d, 0x6c6b, 0x5e45, 0x6360, 0x4a49, 0x7269,
1332 0x554e, 0x7636, 0x4e42, 0x5647, 0x6334, 0x712d, 0x6a62, 0x5742,
1333 0x7327, 0x4d6a, 0x6b6e, 0x5932, 0x7d25, 0x7655, 0x5562, 0x7835,
1334 0x4c75, 0x7535, 0x642d, 0x676b, 0x7155, 0x703b, 0x6935, 0x4c49,
1335 0x7a55, 0x6154, 0x5756, 0x5c41, 0x5e46, 0x7a6f, 0x6361, 0x6173,
1336 0x5c76, 0x4e7c, 0x5b44, 0x7871, 0x5c64, 0x656f, 0x5c31, 0x5556,
1337 0x735a, 0x4b41, 0x5b43, 0x597a, 0x536e, 0x7a38, 0x7d26, 0x6b6e,
1338 0x7426, 0x4c4a, 0x7328, 0x735b, 0x5b27, 0x7637, 0x4f66, 0x7072,
1339 0x4b5a, 0x6752, 0x5743, 0x7670, 0x685e, 0x6526, 0x6567, 0x4a23,

```
1340 0x4c27, 0x6a49, 0x7836, 0x7a25, 0x712e, 0x6f4e, 0x4b6d, 0x7630,
1341 0x6f4f, 0x694f, 0x775e, 0x4e53, 0x5c77, 0x5b28, 0x4b78, 0x5f21,
1342 0x5d61, 0x754a, 0x6936, 0x676c, 0x6e6e, 0x7370, 0x5f3f, 0x4c4b,
1343 0x5041, 0x7452, 0x603a, 0x5f40, 0x4e60, 0x5c52, 0x7d6a, 0x5676,
1344 0x6a4a, 0x6869, 0x632c, 0x7350, 0x4a24, 0x5b78, 0x5e47, 0xb670,
1345 0x7156, 0x6562, 0x642e, 0x4c4c, 0x4b7b, 0x6a63, 0x5f41, 0x566d, 0x6950,
1346 0x6e39, 0x5563, 0x5153, 0x6570, 0x6834, 0x6b43, 0x6a2a, 0x7a7c,
1347 0x7576, 0x703c, 0x7d54, 0x603b, 0x4e43, 0x503a, 0x773a, 0x5873,
1348 0x774d, 0x642e, 0x545f, 0x5067, 0x6c7d, 0x522e, 0x6e6f, 0x5557,
1349 0x6a64, 0x7822, 0x4d6b, 0x573f, 0x7b31, 0x4d6c, 0x5c32, 0x506c,
1350 0x4e7d, 0x6e70, 0x4c42, 0x506d, 0x6577, 0x737c, 0x6e22, 0x5933,
1351 0x5874, 0x6937, 0x4e2e, 0x5922, 0x5871, 0x544f, 0x6527, 0x5552,
1352 0x5629, 0x7422, 0x7157, 0x5558, 0x703d, 0x5750, 0x5450, 0x574f,
1353 0x6b6a, 0x7d6b, 0x5b6d, 0x7c45, 0x4b42, 0x7d55, 0x7448, 0x686a,
1354 0x7573, 0x795e, 0x536f, 0x6c53, 0x5d42, 0x6f37, 0x6754, 0x4a4a,
1355 0x597b, 0x7a7d, 0x562a, 0x7478, 0x7777, 0x5c2c, 0x5757, 0x5f22,
1356 0x4e3e, 0x5370, 0x7024, 0x616c, 0x4f67, 0x734b, 0x6d29, 0x4a3e,
1357 0x746f, 0x764e, 0x5e7b, 0x503b, 0x5537, 0x6e71, 0x7428, 0x5c78,
1358 0x4b27, 0x5a4e, 0x6066, 0x6d25, 0x6e72, 0x5c79, 0x795c, 0x735c,
1359 0x7872, 0x7479, 0x7c71, 0x503c, 0x5b79, 0x5731, 0x4b7c, 0x7025,
1360 0x4b7d, 0x6574, 0x4d6d, 0x4a25, 0x562b, 0x5042, 0x703e, 0x523d,
1361 0x4c24, 0x7a36, 0x4c4d, 0x5a7a, 0x764f, 0x6938, 0x5875, 0x4c4e,
1362 0x574d, 0x5451, 0x696d, 0x4a6b, 0x5962, 0x7d32, 0x632d, 0x564c,
1363 0x5934, 0x6127, 0x6e53, 0x5043, 0x7d33, 0x5564, 0x4f68, 0x6d43,
1364 0x5032, 0x4e7e, 0x5a28, 0x7850, 0x7d56, 0x7851, 0x7852, 0x5c53,
1365 0x5d62, 0x7b79, 0x5d41, 0x6335, 0x6d5d, 0x4e44, 0x4b21, 0x5d63,
1366 0x7c5d, 0x792f, 0x527b, 0x4f21, 0x6428, 0x7436, 0x6c7e, 0x632e,
1367 0x676d, 0x7d41, 0x5a62, 0x5833, 0x5d64, 0x706f, 0x7671, 0x7a70,
1368 0x5175, 0x5a4f, 0x5c54, 0x5c26, 0x6f3f, 0x4e4f, 0x6059, 0x5956,
1369 0x6c54, 0x6a4b, 0x4a3f, 0x5530, 0x4f69, 0x716d, 0x4c4f, 0x6478,
1370 0x646d, 0x5758, 0x7d27, 0x6a2b, 0x7632, 0x4f70, 0x793d, 0x6674,
1371 0x4b5b, 0x7351, 0x6951, 0x7329, 0x5060, 0x6952, 0x5a63, 0x6252,
1372 0x7622, 0x6174, 0x5a64, 0x6755, 0x753f, 0x4f22, 0x4d2f, 0x4f23,
1373 0x4d30, 0x717e, 0x5023, 0x612f, 0x7823, 0x4a26, 0x773b, 0x726a,
1374 0x5e48, 0x6953, 0x5e49, 0x7d5e, 0x4a40, 0x796a, 0x514e, 0x6e54,
1375 0x5452, 0x5923, 0x7d28, 0x5759, 0x774e, 0x7a3e, 0x4f56, 0x5770,
1376 0x6b61, 0x7845, 0x5c7a, 0x5d43, 0x795f, 0x676f, 0x7d65, 0x7623,
1377 0x597c, 0x7d29, 0x676e, 0x5565, 0x6f50, 0x4d31, 0x7722, 0x7132,
1378 0x7131, 0x4d32, 0x5a2b, 0x4a27, 0x6362, 0x7b3c, 0x5924, 0x6e3a,
1379 0x7853, 0x7b7a, 0x4f24, 0x5c7b, 0x7663, 0x6d2a, 0x7221, 0x4e61,
1380 0x7a26, 0x7960, 0x6c56, 0x646e, 0x7921, 0x7b6f, 0x796b, 0x6e23,
1381 0x6a2c, 0x4a28, 0x747a, 0x4d56, 0x7c76, 0x7449, 0x7854, 0x7826,
1382 0x5e4a, 0x7246, 0x575a, 0x5350, 0x5845, 0x6a66, 0x735d, 0x645a,
1383 0x7664, 0x7672, 0x5f42, 0x597d, 0x4c76, 0x533a, 0x642f, 0x7961,
1384 0x7026, 0x4b53, 0x603c, 0x744a, 0x547a, 0x7d2a, 0x7962, 0x7437,
1385 0x7d42, 0x7c30, 0x7d6c, 0x4a62, 0x7d3d, 0x6a67, 0x5f43, 0x5152,
1386 0x4e62, 0x5324, 0x7d2b, 0x5f60, 0x7247, 0x6770, 0x506e, 0x732a,
1387 0x5e4b, 0x7638, 0x6175, 0x7133, 0x7723, 0x4a29, 0x4f25, 0x5f44,
1388 0x6130, 0x703f, 0x7624, 0x6336, 0x7a46, 0x506f, 0x7d6d, 0x5d44,
1389 0x7c77, 0x663f, 0x5e2d, 0x7a3f, 0x6571, 0x6d44, 0x5225, 0x7d6e,
1390 0x7536, 0x6176, 0x5e4c, 0x7c5e, 0x6c57, 0x4d5d, 0x5637, 0x4d33,
1391 0x7855, 0x6558, 0x4f6a, 0x4f50, 0x6a4c, 0x6a2e, 0x6a2d, 0x5371,
1392 0x5325, 0x774f, 0x6e24, 0x5024, 0x7222, 0x5070, 0x7223, 0x7778,
1393 0x5033, 0x5b29, 0x533b, 0x4a6c, 0x7126, 0x4b55, 0x7767, 0x4d5e,
1394 0x7724, 0x7840, 0x535d, 0x4c50, 0x4f26, 0x7673, 0x6177, 0x535c,
1395 0x7a7e, 0x7a27, 0x6b59, 0x4f27, 0x6a2f, 0x646f, 0x6939, 0x7158,
1396 0x5858, 0x6072, 0x6634, 0x5c7c, 0x7371, 0x6350, 0x727b, 0x5b46,
1397 0x5071, 0x5072, 0x4f5c, 0x5351, 0x4c31, 0x7758, 0x4b28, 0x6b3c,
1398 0x643e, 0x745c, 0x5c42, 0x7027, 0x6640, 0x4a6d, 0x686b, 0x6568,
1399 0x5c43, 0x6d5e, 0x5372, 0x4c77, 0x4e54, 0x672b, 0x4b43, 0x6131,
1400 0x7732, 0x5373, 0x5352, 0x7540, 0x5f5d, 0x6e73, 0x6771, 0x7d34,
1401 0x7248, 0x7352, 0x6e74, 0x6253, 0x4c51, 0x5f6a, 0x693a, 0x5957,
1402 0x754d, 0x7172, 0x7a47, 0x5978, 0x5442, 0x7665, 0x5d45, 0x6772,
1403 0x6d5f, 0x4a4b, 0x5b7a, 0x6835, 0x5326, 0x7d35, 0x7949, 0x6462,
1404 0x7b3d, 0x5724, 0x4e45, 0x4e55, 0x5666, 0x653d, 0x5e4d, 0x6c73,
1405 0x6d60, 0x6c6c, 0x7b3e, 0x5f6b, 0x6178, 0x793e, 0x5073, 0x602a,
1406 0x6862, 0x6254, 0x527d, 0x6528, 0x5953, 0x535e, 0x7438, 0x773c,
1407 0x5c7d, 0x686c, 0x6467, 0x6377, 0x6c28, 0x7a71, 0x6572, 0x5074,
1408 0x522f, 0x5c65, 0x5025, 0x7134, 0x7c31, 0x4c78, 0x5d46, 0x7a51,
1409 0x775f, 0x7a28, 0x6e75, 0x5e4e, 0x6773, 0x772c, 0x6b44, 0x6d61,
1410 0x602b, 0x5d47, 0x5233, 0x523f, 0x4a4c, 0x7b3f, 0x657d, 0x5d65,
1411 0x584d, 0x6c74, 0x5075, 0x686d, 0x5052, 0x5958, 0x7666, 0x5b2a,
1412 0x7760, 0x5859, 0x7423, 0x745d, 0x6f51, 0x5935, 0x6d2b, 0x6337,
1413 0x6e3b, 0x4d34, 0x6073, 0x6a4d, 0x6c75, 0x686e, 0x4b29, 0x712f,
1414 0x4a4d, 0x6c29, 0x726b, 0x7d6f, 0x7973, 0x6641, 0x6c58, 0x6d2c,
1415 0x6a4e, 0x685f, 0x5e4f, 0x5226, 0x6774, 0x5156, 0x6642, 0x6363,
1416 0x6430, 0x5834, 0x7625, 0x735e, 0x5725, 0x7768, 0x6846, 0x7b66,
1417 0x5d66, 0x5c7e, 0x585a, 0x5a2c, 0x6a30, 0x6338, 0x4a2a, 0x6179,
1418 0x6a31, 0x726c, 0x7a6e, 0x6e55, 0x7974, 0x526c, 0x7b7b, 0x7d70,
1419 0x603d, 0x4e63, 0x7846, 0x5e2e, 0x5f45, 0x653e, 0x6d2d, 0x7a6a,
1420 0x4d6e, 0x6d26, 0x6d2e, 0x706d, 0x5d21, 0x6d2f, 0x7c78, 0x586b,
1421 0x4c79, 0x4d35, 0x7a29, 0x615d, 0x6255, 0x6d4f, 0x5d22, 0x794a,
1422 0x6a68, 0x656d, 0x536b, 0x6954, 0x617a, 0x644c, 0x6164, 0x6847,
1423 0x4e5b, 0x5c55, 0x7735, 0x7c73, 0x7073, 0x4e2f, 0x7135, 0x6f52,
1424 0x6848, 0x6b71, 0x4b54, 0x603e, 0x6378, 0x6a69, 0x7c32, 0x6074,
1425 0x4f60, 0x6e25, 0x7a2a, 0x6643, 0x6132, 0x4a2b, 0x6364, 0x693b,
1426 0x6256, 0x7372, 0x6e56, 0x6a32, 0x5076, 0x6c59, 0x5a4b, 0x4f28,
```

```
1427 0x5d23, 0x585b, 0x794e, 0x6955, 0x6351, 0x523c, 0x582c, 0x734c,
1428 0x4d7b, 0x7656, 0x6775, 0x686f, 0x6379, 0x523b, 0x7373, 0x637b,
1429 0x5e50, 0x4e30, 0x5677, 0x7159, 0x7541, 0x5c44, 0x753b, 0x5e51,
1430 0x5c66, 0x5e52, 0x6d62, 0x6e76, 0x6a4f, 0x706e, 0x637c, 0x535f,
1431 0x5374, 0x6133, 0x6134, 0x7453, 0x5f46, 0x6956, 0x5b2b, 0x7626,
1432 0x6339, 0x6b45, 0x7429, 0x4d36, 0x5279, 0x5a2d, 0x5263, 0x4f51,
1433 0x4b5c, 0x4c7a, 0x4f5d, 0x6829, 0x633b, 0x633a, 0x605a, 0x6e77,
1434 0x5c33, 0x5375, 0x5726, 0x7635, 0x575b, 0x6155, 0x546a, 0x5f23,
1435 0x7d5f, 0x5077, 0x6d54, 0x4b2a, 0x645b, 0x617b, 0x4b22, 0x5360,
1436 0x643f, 0x7b40, 0x5a3e, 0x644d, 0x5639, 0x6f40, 0x617c, 0x7639,
1437 0x5f47, 0x6431, 0x5c67, 0x5c68, 0x7a56, 0x5376, 0x715a, 0x7a72,
1438 0x627d, 0x554f, 0x5078, 0x4d5f, 0x754b, 0x6470, 0x4b2b, 0x5744,
1439 0x627e, 0x5d5a, 0x5a2e, 0x4a6e, 0x5539, 0x6321, 0x6863, 0x732b,
1440 0x4f29, 0x5377, 0x5471, 0x4e64, 0x6872, 0x6575, 0x672e, 0x563a,
1441 0x5f6c, 0x6440, 0x6864, 0x5835, 0x645c, 0x7439, 0x7136, 0x625e,
1442 0x6135, 0x4d6f, 0x7127, 0x4e65, 0x4b5d, 0x5963, 0x732c, 0x5079,
1443 0x6c2b, 0x5e53, 0x7769, 0x7975, 0x615e, 0x4b6e, 0x633c, 0x7856,
1444 0x5b6e, 0x7d71, 0x7736, 0x745e, 0x726d, 0x5b59, 0x7028, 0x617d,
1445 0x5e54, 0x602c, 0x6d63, 0x5361, 0x5f48, 0x5936, 0x7d2c, 0x6f53,
1446 0x6441, 0x786b, 0x5b2c, 0x7c46, 0x582d, 0x763a, 0x5b5f, 0x5353,
1447 0x7847, 0x4a4e, 0x5078, 0x7841, 0x5234, 0x5c34, 0x7a39, 0x4a4f, 0x7c33,
1448 0x6a6a, 0x6a6b, 0x507a, 0x6d64, 0x5d67, 0x5f49, 0x5f6d, 0x6e3c,
1449 0x6f41, 0x4c52, 0x5d24, 0x5f4a, 0x5378, 0x7128, 0x4d37, 0x6f54,
1450 0x645d, 0x5f6e, 0x4b2c, 0x693c, 0x6a6c, 0x5f4b, 0x793f, 0x562f,
1451 0x5546, 0x4f2a, 0x4e29, 0x5678, 0x7137, 0x6e78, 0x5959, 0x735f,
1452 0x7848, 0x4e46, 0x5566, 0x7466, 0x6645, 0x6f55, 0x4b6f, 0x7c5f,
1453 0x5c27, 0x5667, 0x7849, 0x6352, 0x633d, 0x4f61, 0x7040, 0x6c5a,
1454 0x5d57, 0x7b70, 0x6c2c, 0x7029, 0x7a57, 0x7b41, 0x5240, 0x6530,
1455 0x6d65, 0x4b2d, 0x7930, 0x7725, 0x4b2e, 0x5a2f, 0x5836, 0x5327,
1456 0x7b32, 0x7d44, 0x6c2d, 0x7b21, 0x6569, 0x696e, 0x7374, 0x7873,
1457 0x7041, 0x5e2f, 0x7830, 0x7360, 0x672f, 0x5b2d, 0x6635, 0x7928,
1458 0x5d58, 0x6859, 0x6f56, 0x5362, 0x625f, 0x7c60, 0x5748, 0x7d2d,
1459 0x5f6f, 0x4c53, 0x5379, 0x5379, 0x5470, 0x5b47, 0x5e55, 0x7074, 0x5550,
1460 0x6559, 0x7c47, 0x5c56, 0x6260, 0x5a30, 0x7323, 0x536c, 0x744b,
1461 0x7d45, 0x637d, 0x7931, 0x507b, 0x6c5b, 0x753c, 0x7224, 0x584e,
1462 0x584f, 0x7577, 0x7661, 0x5237, 0x7b6c, 0x5d48, 0x6468, 0x5241,
1463 0x7857, 0x563b, 0x5e56, 0x773d, 0x6c2e, 0x5061, 0x6075, 0x6a33,
1464 0x4e56, 0x4c25, 0x6c76, 0x6261, 0x633e, 0x7c48, 0x4d70, 0x7976,
1465 0x5f70, 0x653f, 0x4e3f, 0x7c61, 0x6d30, 0x7d51, 0x763b, 0x794f,
1466 0x6b5a, 0x4a41, 0x5238, 0x4d71, 0x6353, 0x7d66, 0x666d, 0x637a,
1467 0x702a, 0x7950, 0x7c62, 0x7827, 0x6165, 0x6e79, 0x6776, 0x6a6d,
1468 0x7c34, 0x7542, 0x575c, 0x7075, 0x5d68, 0x536d, 0x757c, 0x5a3f,
1469 0x4c7b, 0x537a, 0x7424, 0x6f57, 0x5443, 0x7b63, 0x7b6d, 0x602d,
1470 0x6a6e, 0x7b33, 0x6442, 0x7667, 0x525d, 0x5f4c, 0x7c49, 0x6529,
1471 0x6076, 0x7633, 0x617e, 0x4b70, 0x6a6f, 0x6a70, 0x5a40, 0x7834,
1472 0x6b72, 0x6443, 0x6957, 0x6471, 0x4a6f, 0x4e57, 0x7c4a, 0x7361,
1473 0x4b44, 0x6365, 0x4b45, 0x6a34, 0x693d, 0x5749, 0x6b5b, 0x6d31,
1474 0x4c43, 0x773e, 0x7c4b, 0x7874, 0x5937, 0x7353, 0x7354, 0x7764,
1475 0x7751, 0x5837, 0x4e31, 0x4a42, 0x7b34, 0x4b46, 0x7076, 0x5567,
1476 0x6a50, 0x4c54, 0x4b2f, 0x742a, 0x692f, 0x7543, 0x6958, 0x5d69,
1477 0x7173, 0x557b, 0x5e3b, 0x747b, 0x7d73, 0x7d72, 0x7726, 0x5d49,
1478 0x5453, 0x4c28, 0x5a41, 0x4c55, 0x5964, 0x7a4a, 0x6563, 0x533c,
1479 0x4a70, 0x5044, 0x4a50, 0x7a2b, 0x6b6b, 0x6778, 0x5965, 0x5157,
1480 0x7324, 0x547b, 0x7c63, 0x7a58, 0x7355, 0x4f2b, 0x6b73, 0x557c,
1481 0x5354, 0x4d7c, 0x5966, 0x6279, 0x6221, 0x6b54, 0x6077, 0x6432,
1482 0x4c7c, 0x7b64, 0x742b, 0x503d, 0x4a71, 0x6f38, 0x5740, 0x6e7a,
1483 0x7d74, 0x5363, 0x7b42, 0x5568, 0x5b2e, 0x6136, 0x7837, 0x603f,
1484 0x7b43, 0x5d6a, 0x6222, 0x6e26, 0x7668, 0x7675, 0x5d4a, 0x5062,
1485 0x5d26, 0x5d6b, 0x6479, 0x632f, 0x507c, 0x747c, 0x4c3c, 0x776a,
1486 0x6564, 0x5f71, 0x7761, 0x7977, 0x6f39, 0x7858, 0x7929, 0x7859,
1487 0x6e3d, 0x5846, 0x6463, 0x754e, 0x5d59, 0x5967, 0x5239, 0x5543,
1488 0x5a65, 0x5a50, 0x5159, 0x4e58, 0x4b5e, 0x742c, 0x5a7b, 0x7669,
1489 0x6873, 0x4f2c, 0x7070, 0x747d, 0x5b48, 0x4e40, 0x6354, 0x514f,
1490 0x7175, 0x4d72, 0x4f6b, 0x4d38, 0x6326, 0x515a, 0x7225, 0x7226,
1491 0x644e, 0x537b, 0x7129, 0x7249, 0x6f58, 0x6649, 0x5838, 0x7a73,
1492 0x7335, 0x7824, 0x5173, 0x6648, 0x785a, 0x5c69, 0x5e57, 0x4b5f,
1493 0x4f6c, 0x745f, 0x5174, 0x523a, 0x5f72, 0x6137, 0x6223, 0x537c,
1494 0x6d66, 0x5b49, 0x647a, 0x4f5e, 0x4e50, 0x5553, 0x7375, 0x772e,
1495 0x6f48, 0x4d73, 0x754f, 0x6573, 0x7042, 0x4a51, 0x6a71, 0x5026,
1496 0x595a, 0x702b, 0x6b67, 0x6540, 0x7c35, 0x6444, 0x4c29, 0x7d46,
1497 0x6a35, 0x652a, 0x5f3a, 0x615f, 0x5a51, 0x6138, 0x6874, 0x537d,
1498 0x6224, 0x724a, 0x5a66, 0x7733, 0x7d4d, 0x7336, 0x6e57, 0x7544,
1499 0x5824, 0x7227, 0x5938, 0x5939, 0x6f49, 0x564e, 0x774b, 0x5f2e,
1500 0x6875, 0x5235, 0x5355, 0x744c, 0x5a7c, 0x5968, 0x776b, 0x7549,
1501 0x733c, 0x5a52, 0x5335, 0x6836, 0x564f, 0x743a, 0x7749, 0x4c2a,
1502 0x7043, 0x4c56, 0x5053, 0x533d, 0x5b7b, 0x4b60, 0x5364, 0x7677,
1503 0x553a, 0x734d, 0x4b61, 0x6b74, 0x742d, 0x7c2a, 0x776c, 0x6876,
1504 0x5a67, 0x774c, 0x6541, 0x606e, 0x557d, 0x4e66, 0x7c2b, 0x553b,
1505 0x7228, 0x6225, 0x4d39, 0x6a72, 0x4b47, 0x4d74, 0x5b2f, 0x6f59,
1506 0x4d3a, 0x7c79, 0x5f73, 0x4e67, 0x5a42, 0x4f2d, 0x6779, 0x7828,
1507 0x7362, 0x4a72, 0x5f24, 0x5444, 0x4c57, 0x6542, 0x4d3b, 0x6f5a,
1508 0x6e58, 0x5d27, 0x6226, 0x6040, 0x5630, 0x784a, 0x7c7a, 0x597e,
1509 0x5e30, 0x5d6c, 0x5a68, 0x5460, 0x5679, 0x4d57, 0x5e58, 0x7278,
1510 0x6456, 0x5045, 0x742e, 0x5d28, 0x6d45, 0x7356, 0x5e59, 0x6366,
1511 0x5328, 0x5b30, 0x655a, 0x633f, 0x5b31, 0x5569, 0x6041, 0x6f5b,
1512 0x7069, 0x5732, 0x507d, 0x5969, 0x507e, 0x6c6d, 0x5329, 0x7229,
1513 0x7044, 0x6262, 0x696f, 0x7951, 0x6959, 0x685a, 0x5a43, 0x5a44,
```

1514 0x5445, 0x677a, 0x4d60, 0x6330, 0x5b32, 0x7b44, 0x7363, 0x5925,
1515 0x7b67, 0x5d4b, 0x5054, 0x6636, 0x602e, 0x7d5a, 0x5c35, 0x6078,
1516 0x6731, 0x7570, 0x585c, 0x6d46, 0x6139, 0x6340, 0x7940, 0x6970,
1517 0x595b, 0x7364, 0x5c36, 0x6469, 0x7045, 0x6341, 0x7c4c, 0x7c4d,
1518 0x724b, 0x724c, 0x644f, 0x715b, 0x7a59, 0x7138, 0x7d75, 0x6079,
1519 0x677b, 0x7c37, 0x7c64, 0x7c64, 0x7b45, 0x6367, 0x5839, 0x7678, 0x5c45,
1520 0x4c58, 0x602f, 0x7467, 0x6f5c, 0x4f7c, 0x6f5d, 0x722a, 0x7d3e,
1521 0x4a2c, 0x7d3b, 0x7d47, 0x6732, 0x6a51, 0x5f74, 0x516c, 0x645e,
1522 0x6543, 0x5926, 0x4d3c, 0x7365, 0x6d55, 0x593a, 0x6d67, 0x7b35,
1523 0x786c, 0x6067, 0x4c59, 0x5446, 0x6725, 0x5575, 0x533e, 0x7c7b,
1524 0x6472, 0x5f75, 0x6878, 0x786d, 0x4e47, 0x7d76, 0x6858, 0x4d58,
1525 0x6756, 0x4c5a, 0x4a63, 0x5f76, 0x7047, 0x7046, 0x583a, 0x7174,
1526 0x7470, 0x754c, 0x7c65, 0x6a45, 0x6a73, 0x5d5b, 0x5c57, 0x5e7d,
1527 0x7279, 0x5547, 0x5850, 0x7048, 0x5121, 0x5122, 0x5954, 0x5668,
1528 0x594a, 0x5a31, 0x5847, 0x5847, 0x5c62, 0x734e, 0x7574, 0x7139, 0x5a53,
1529 0x766a, 0x4f75, 0x7d2e, 0x4a52, 0x5f34, 0x575d, 0x7a3a, 0x6e27,
1530 0x753d, 0x7875, 0x6d68, 0x5461, 0x5123, 0x6156, 0x7978, 0x5b4a,
1531 0x4b79, 0x5454, 0x595c, 0x6e3e, 0x776d, 0x526e, 0x6166, 0x7779,
1532 0x5d6d, 0x685b, 0x5b33, 0x5177, 0x6030, 0x5462, 0x7657, 0x5779,
1533 0x585d, 0x4d7d, 0x722b, 0x4d3d, 0x7842, 0x722c, 0x4a2d, 0x4a2e,
1534 0x4f2e, 0x6342, 0x5c37, 0x5b5a, 0x593b, 0x4a73, 0x7653, 0x6678,
1535 0x6a75, 0x6a76, 0x7679, 0x4f2f, 0x4a53, 0x4a2f, 0x5230, 0x713a,
1536 0x5733, 0x6343, 0x737d, 0x5e5a, 0x5e5b, 0x6f5e, 0x6263, 0x6e7b,
1537 0x5f77, 0x574a, 0x4e68, 0x5b5b, 0x713b, 0x6971, 0x7a37, 0x5046,
1538 0x4c2b, 0x6e28, 0x4b7a, 0x7979, 0x4c7d, 0x537e, 0x6450, 0x726e,
1539 0x5455, 0x5f4d, 0x7c38, 0x5150, 0x724d, 0x7752, 0x4a54, 0x5559,
1540 0x585e, 0x4d59, 0x6e29, 0x763c, 0x4c5b, 0x7049, 0x7c7c, 0x6849,
1541 0x747e, 0x677c, 0x575e, 0x5e5c, 0x702c, 0x4c7e, 0x4d61, 0x613a,
1542 0x5b6f, 0x5a32, 0x5125, 0x5c38, 0x5876, 0x5124, 0x4d62, 0x5c6a,
1543 0x7077, 0x704a, 0x503e, 0x5d5c, 0x5456, 0x5356, 0x6d50, 0x4d21,
1544 0x5f35, 0x5f78, 0x5421, 0x4e32, 0x684a, 0x6b75, 0x6355, 0x7550,
1545 0x7521, 0x5927, 0x652b, 0x664b, 0x7571, 0x6545, 0x7923, 0x605b,
1546 0x766b, 0x4b71, 0x596a, 0x7522, 0x5751, 0x5178, 0x6a78, 0x6a79,
1547 0x5a33, 0x6f5f, 0x716f, 0x6576, 0x6e3f, 0x6264, 0x503f, 0x7a2c,
1548 0x7551, 0x6733, 0x693e, 0x724e, 0x5b34, 0x7c4e, 0x5d6e, 0x6734,
1549 0x5734, 0x7734, 0x4d3e, 0x5a69, 0x4f30, 0x7759, 0x7366, 0x4e59,
1550 0x4e2a, 0x4b48, 0x5027, 0x704b, 0x5047, 0x6445, 0x5b60, 0x555a,
1551 0x5727, 0x6e40, 0x7876, 0x7552, 0x6d69, 0x593c, 0x6546, 0x7523,
1552 0x5a54, 0x6227, 0x7b7c, 0x715c, 0x4a74, 0x687a, 0x4e69, 0x6978,
1553 0x6265, 0x5039, 0x5472, 0x5126, 0x5f4e, 0x7c74, 0x532a, 0x4c2c,
1554 0x6f60, 0x6565, 0x5055, 0x5b7c, 0x7c66, 0x4b7e, 0x6d6a, 0x5e31,
1555 0x7963, 0x5422, 0x4f76, 0x5650, 0x556a, 0x716e, 0x7a4b, 0x6521,
1556 0x5531, 0x4f6d, 0x6d6b, 0x5532, 0x553c, 0x7d62, 0x732d, 0x7d5b,
1557 0x6930, 0x5127, 0x7d63, 0x4e33, 0x7d64, 0x7a4e, 0x4a30, 0x7727,
1558 0x4f31, 0x6622, 0x7c36, 0x722d, 0x6f61, 0x732e, 0x5c46, 0x596b,
1559 0x6860, 0x6128, 0x5576, 0x4f7d, 0x5e5d, 0x5951, 0x646a, 0x724f,
1560 0x773f, 0x6266, 0x6228, 0x6356, 0x6d51, 0x6979, 0x5631, 0x5e32,
1561 0x6068, 0x532b, 0x6b5c, 0x5f2f, 0x4a43, 0x6e7c, 0x7d43, 0x6b76,
1562 0x4f32, 0x596c, 0x593d, 0x585f, 0x5438, 0x6b3e, 0x5d6f, 0x5d70,
1563 0x5d71, 0x5d72, 0x593e, 0x7b46, 0x4f33, 0x6e7d, 0x642b, 0x5a45,
1564 0x586c, 0x5128, 0x6229, 0x5e3c, 0x6735, 0x5b70, 0x6f62, 0x7170,
1565 0x4f34, 0x5b71, 0x6031, 0x5f25, 0x7952, 0x677d, 0x6623, 0x7b71,
1566 0x4b30, 0x722e, 0x4d67, 0x685c, 0x6757, 0x7740, 0x5063, 0x5a21,
1567 0x4c3d, 0x5129, 0x5d4c, 0x637e, 0x512a, 0x682a, 0x6a36, 0x797a,
1568 0x664c, 0x7658, 0x5447, 0x594b, 0x5952, 0x534b, 0x5877, 0x5a29,
1569 0x7578, 0x5e5e, 0x722f, 0x7829, 0x5848, 0x6e41, 0x7941, 0x5d73,
1570 0x6a7a, 0x763d, 0x613b, 0x4d3f, 0x7454, 0x664d, 0x7c4f, 0x7b22,
1571 0x605c, 0x743b, 0x5a55, 0x7932, 0x7b72, 0x5b76, 0x5e5f, 0x5b72,
1572 0x785c, 0x776e, 0x6b68, 0x527a, 0x713c, 0x7a5a, 0x5a6a, 0x5a46,
1573 0x7741, 0x6736, 0x6547, 0x562c, 0x5c47, 0x6129, 0x622a, 0x5526,
1574 0x5457, 0x7250, 0x6a7b, 0x605d, 0x7b73, 0x713d, 0x6267, 0x7d57,
1575 0x4e48, 0x6a37, 0x7c40, 0x7d67, 0x776f, 0x5735, 0x6f3a, 0x715d,
1576 0x5e33, 0x684b, 0x785d, 0x7b47, 0x5548, 0x575f, 0x5d29, 0x6931,
1577 0x7a2d, 0x7659, 0x7a74, 0x782a, 0x666e, 0x4c5c, 0x613c, 0x606f,
1578 0x693f, 0x7c7d, 0x664e, 0x6157, 0x664f, 0x7471, 0x6473, 0x647b,
1579 0x7964, 0x6f63, 0x4f6e, 0x763e, 0x6032, 0x7c7e, 0x512b, 0x577a,
1580 0x7b48, 0x6257, 0x5423, 0x7078, 0x5728, 0x6167, 0x533f, 0x6f64,
1581 0x5745, 0x6b62, 0x7c67, 0x6422, 0x6268, 0x6650, 0x7b68, 0x7468,
1582 0x6574, 0x743c, 0x7455, 0x5f36, 0x7c39, 0x6e42, 0x4a75, 0x6f65,
1583 0x4b62, 0x5424, 0x5e60, 0x5a7d, 0x6446, 0x683e, 0x605e, 0x7634,
1584 0x6a52, 0x797b, 0x6042, 0x4a64, 0x6737, 0x6a7d, 0x595d, 0x5a34,
1585 0x6e2a, 0x7b69, 0x5b4b, 0x5a35, 0x713e, 0x532c, 0x7b49, 0x5f4f,
1586 0x5340, 0x6357, 0x6f66, 0x7c50, 0x6940, 0x7553, 0x6c5c, 0x7737,
1587 0x6a38, 0x5179, 0x5c48, 0x6a39, 0x715e, 0x5736, 0x4f35, 0x5928,
1588 0x6c6e, 0x5d2a, 0x4d22, 0x682e, 0x613d, 0x7251, 0x6941, 0x527c,
1589 0x5b35, 0x7367, 0x587e, 0x7c51, 0x6d32, 0x742f, 0x7b23, 0x7c41,
1590 0x6e2b, 0x5425, 0x7472, 0x6e59, 0x7b4a, 0x4d63, 0x583b, 0x655b,
1591 0x7877, 0x7654, 0x5729, 0x4b49, 0x6651, 0x704c, 0x582e, 0x7953,
1592 0x557e, 0x583c, 0x7230, 0x622b, 0x7368, 0x6f42, 0x6d6c, 0x6738,
1593 0x5a7e, 0x4c3e, 0x727c, 0x5a6b, 0x6258, 0x6d56, 0x5651, 0x6033,
1594 0x7c52, 0x6b48, 0x5341, 0x704d, 0x4f77, 0x6d52, 0x5458, 0x5c49,
1595 0x5771, 0x5f3b, 0x7325, 0x744d, 0x713f, 0x7831, 0x697a, 0x7b4b,
1596 0x4a55, 0x7954, 0x774a, 0x5648, 0x7c68, 0x733d, 0x6e7e, 0x677e,
1597 0x5342, 0x5336, 0x4c2d, 0x767a, 0x5632, 0x5258, 0x6758, 0x6325,
1598 0x6739, 0x702d, 0x7b4c, 0x6b21, 0x5426, 0x7b4d, 0x553d, 0x715f,
1599 0x767b, 0x5e34, 0x556b, 0x6548, 0x7b24, 0x5439, 0x5e61, 0x6423,
1600 0x5737, 0x786e, 0x5e35, 0x5652, 0x7955, 0x673a, 0x6b55, 0x5577,

1601 0x6f67, 0x613e, 0x7a2e, 0x5669, 0x566e, 0x673b, 0x6c4b, 0x5533,
 1602 0x4e34, 0x7b25, 0x616e, 0x7728, 0x7b4e, 0x583d, 0x7b7d, 0x7c69,
 1603 0x4f36, 0x6d47, 0x6e2c, 0x4c5d, 0x7627, 0x667a, 0x7524, 0x7d5c,
 1604 0x6d33, 0x4e49, 0x6f68, 0x613f, 0x7a5b, 0x4b63, 0x7729, 0x7b26,
 1605 0x5c39, 0x7140, 0x6d48, 0x6f43, 0x562d, 0x7d4e, 0x6821, 0x7b74,
 1606 0x5527, 0x7176, 0x6653, 0x4c5e, 0x7832, 0x5c6b, 0x7d36, 0x656a,
 1607 0x7160, 0x5b4c, 0x5d4d, 0x5448, 0x596d, 0x7525, 0x667b, 0x6654,
 1608 0x7d48, 0x5621, 0x7d3f, 0x7c53, 0x6f21, 0x673c, 0x516e, 0x6655,
 1609 0x6972, 0x5f30, 0x5860, 0x7c3a, 0x7d2f, 0x704e, 0x5b61, 0x6549,
 1610 0x6d34, 0x6043, 0x6358, 0x697b, 0x6a28, 0x7d37, 0x7b27, 0x6942,
 1611 0x7d77, 0x6259, 0x5c6c, 0x6822, 0x6670, 0x7d78, 0x7d79, 0x763f,
 1612 0x6727, 0x6657, 0x5473, 0x5449, 0x567a, 0x5772, 0x6140, 0x5b62,
 1613 0x6658, 0x673d, 0x704f, 0x733e, 0x622c, 0x7537, 0x6070, 0x7d38,
 1614 0x6368, 0x5427, 0x687c, 0x7a52, 0x786f, 0x5653, 0x5534, 0x7050,
 1615 0x7770, 0x6e33, 0x6a3a, 0x6a53, 0x6d49, 0x5d2b, 0x652c, 0x7d21,
 1616 0x5f50, 0x6c33, 0x5f51, 0x6d6d, 0x7838, 0x777a, 0x782b, 0x7460,
 1617 0x543a, 0x6433, 0x695a, 0x5e36, 0x593f, 0x5940, 0x566f, 0x594c,
 1618 0x5a2a, 0x5f65, 0x7765, 0x4c32, 0x5f79, 0x5760, 0x543b, 0x7d7a,
 1619 0x4c33, 0x5b73, 0x5f52, 0x4e4a, 0x6e5a, 0x6464, 0x7b4f, 0x4f37,
 1620 0x6e43, 0x4e6a, 0x622d, 0x5761, 0x7a75, 0x5549, 0x782c, 0x6759,
 1621 0x7369, 0x586d, 0x6344, 0x7071, 0x6865, 0x607a, 0x6e44, 0x595e,
 1622 0x6b22, 0x6b23, 0x7c42, 0x6a3b, 0x682b, 0x5e62, 0x6d6f, 0x6823,
 1623 0x4f71, 0x543c, 0x7c6a, 0x673e, 0x7c72, 0x5634, 0x622e, 0x5337,
 1624 0x7a4c, 0x7a5c, 0x6d35, 0x6163, 0x682c, 0x685d, 0x6f69, 0x743d,
 1625 0x4f38, 0x695b, 0x512c, 0x5a47, 0x6b49, 0x684c, 0x5e37, 0x563c,
 1626 0x5365, 0x7a5d, 0x5a56, 0x4a31, 0x5a48, 0x5f26, 0x7933, 0x7252,
 1627 0x4a44, 0x4e48, 0x4d75, 0x7d30, 0x5528, 0x7141, 0x6269, 0x5c4a,
 1628 0x6c34, 0x7a40, 0x7b28, 0x5028, 0x5a6c, 0x596e, 0x607b, 0x6f6a,
 1629 0x7a5e, 0x6044, 0x4f39, 0x554a, 0x5762, 0x622f, 0x5738, 0x684d,
 1630 0x765a, 0x6f22, 0x625a, 0x767c, 0x7b50, 0x512d, 0x4d64, 0x512e,
 1631 0x5c6d, 0x684e, 0x7079, 0x4e35, 0x667c, 0x577b, 0x5056, 0x5d75,
 1632 0x7771, 0x767d, 0x5b77, 0x7b6a, 0x695c, 0x5941, 0x7572, 0x6045,
 1633 0x6a54, 0x7942, 0x6a3c, 0x5245, 0x7b51, 0x6740, 0x6b25, 0x5f7a,
 1634 0x6322, 0x5739, 0x6943, 0x687d, 0x682f, 0x7253, 0x7b29, 0x5825,
 1635 0x554b, 0x5048, 0x512f, 0x5763, 0x6046, 0x5622, 0x6d70, 0x5773,
 1636 0x7c54, 0x5a57, 0x4c5f, 0x7254, 0x5130, 0x4c60, 0x5b7d, 0x733f,
 1637 0x7051, 0x7c3b, 0x6230, 0x6625, 0x625b, 0x5f5e, 0x6047, 0x726f,
 1638 0x4c61, 0x566a, 0x6742, 0x4e36, 0x7340, 0x4d7e, 0x7b52, 0x7878,
 1639 0x777b, 0x683f, 0x6837, 0x6d36, 0x5c3a, 0x4c34, 0x7177, 0x6838,
 1640 0x4a76, 0x6424, 0x7456, 0x5f66, 0x5f27, 0x5f67, 0x6141, 0x6944,
 1641 0x5c4b, 0x6945, 0x6f23, 0x6b26, 0x4b23, 0x6369, 0x517b, 0x6f24,
 1642 0x6f6b, 0x5034, 0x4d23, 0x6866, 0x6f25, 0x534c, 0x5a6d, 0x573a,
 1643 0x7255, 0x7565, 0x596f, 0x7934, 0x5554, 0x7d4f, 0x5b63, 0x7161,
 1644 0x6c36, 0x7b7e, 0x5357, 0x5131, 0x4b31, 0x5132, 0x4b32, 0x7142,
 1645 0x7461, 0x7935, 0x6143, 0x6142, 0x6b77, 0x5f28, 0x4b4a, 0x6639,
 1646 0x785e, 0x792a, 0x4a77, 0x6d37, 0x5338, 0x7256, 0x5459, 0x6e45,
 1647 0x7270, 0x4a32, 0x5c3b, 0x7178, 0x6c37, 0x654a, 0x7640, 0x7d5d,
 1648 0x5463, 0x4c62, 0x7754, 0x5765, 0x5343, 0x5826, 0x7641, 0x5d76,
 1649 0x4d40, 0x655c, 0x654b, 0x6144, 0x6830, 0x7430, 0x736a, 0x5a6e,
 1650 0x573b, 0x6231, 0x572a, 0x567b, 0x645f, 0x4a56, 0x6b28, 0x5b7e,
 1651 0x7642, 0x6f3b, 0x547d, 0x6048, 0x6839, 0x6f26, 0x4d24, 0x5474,
 1652 0x5b21, 0x5b5c, 0x5b5d, 0x6e5c, 0x4b4b, 0x7c55, 0x4e6b, 0x4d41,
 1653 0x7b53, 0x792b, 0x7554, 0x5929, 0x695d, 0x5b4d, 0x5d4e, 0x6743,
 1654 0x6c4c, 0x796c, 0x4b4c, 0x607c, 0x5428, 0x6d53, 0x586f, 0x7257,
 1655 0x4a78, 0x5a6f, 0x5654, 0x594d, 0x586e, 0x7241, 0x5f53, 0x5a70,
 1656 0x626a, 0x607d, 0x5878, 0x772f, 0x5a36, 0x4a57, 0x7258, 0x5879,
 1657 0x7a5f, 0x4f6f, 0x5942, 0x7052, 0x6451, 0x7337, 0x7a60, 0x6f6c,
 1658 0x6232, 0x543d, 0x594e, 0x7462, 0x5429, 0x4d42, 0x675a, 0x7259,
 1659 0x592a, 0x583e, 0x5c2d, 0x626b, 0x567c, 0x4a79, 0x545a, 0x7457,
 1660 0x4c21, 0x4f3a, 0x7538, 0x5943, 0x5068, 0x6345, 0x6b78, 0x7231,
 1661 0x4f3b, 0x532d, 0x6861, 0x4e6c, 0x6034, 0x5e63, 0x5d77, 0x7232,
 1662 0x7376, 0x765b, 0x577e, 0x785f, 0x7772, 0x5029, 0x665a, 0x7526,
 1663 0x573c, 0x4c63, 0x665b, 0x5d5d, 0x5133, 0x6f6d, 0x565e, 0x6474,
 1664 0x616f, 0x5d78, 0x684f, 0x4a65, 0x5c21, 0x6035, 0x7c2c, 0x7c2d,
 1665 0x5827, 0x6d38, 0x5b36, 0x5670, 0x732f, 0x4d25, 0x5a71, 0x5828,
 1666 0x4c64, 0x5134, 0x4a58, 0x5a72, 0x7527, 0x7528, 0x6626, 0x556c,
 1667 0x5578, 0x5a73, 0x6346, 0x5e64, 0x5e65, 0x5135, 0x5136, 0x5137,
 1668 0x7233, 0x695e, 0x7053, 0x7234, 0x7054, 0x4b64, 0x7b54, 0x7566,
 1669 0x636a, 0x5e66, 0x5f54, 0x7879, 0x702e, 0x5138, 0x565f, 0x5057,
 1670 0x7c21, 0x6f6e, 0x5c58, 0x695f, 0x655d, 0x7d7b, 0x6049, 0x5649,
 1671 0x542a, 0x654c, 0x6960, 0x5058, 0x7c22, 0x543e, 0x6233, 0x5e67,
 1672 0x5c3c, 0x5236, 0x7555, 0x4e21, 0x7529, 0x5d79, 0x5d7a, 0x7055,
 1673 0x765f, 0x725a, 0x646b, 0x7271, 0x6c39, 0x7d7c, 0x612a, 0x4a59,
 1674 0x6f6f, 0x752a, 0x6c79, 0x782d, 0x7242, 0x7643, 0x5752, 0x7922,
 1675 0x7056, 0x707a, 0x7660, 0x6973, 0x7243, 0x542b, 0x4a33, 0x4d26,
 1676 0x4d43, 0x4d5a, 0x594f, 0x7644, 0x6e5d, 0x6744, 0x6234, 0x5f62,
 1677 0x675b, 0x6831, 0x7c2e, 0x654d, 0x7a6b, 0x4f3c, 0x4f62, 0x4d76,
 1678 0x6f70, 0x743e, 0x544d, 0x7338, 0x6921, 0x7272, 0x736b, 0x7057,
 1679 0x4f57, 0x4f5f, 0x6840, 0x6841, 0x4f63, 0x6922, 0x502a, 0x7341,
 1680 0x502b, 0x5464, 0x6f3c, 0x5821, 0x595f, 0x7357, 0x5c3d, 0x4c65,
 1681 0x6d71, 0x7162, 0x545b, 0x6235, 0x4a66, 0x532e, 0x4c66, 0x7153,
 1682 0x7567, 0x4a5a, 0x7b6e, 0x6145, 0x5f69, 0x6e5e, 0x7742, 0x5822,
 1683 0x5d2c, 0x702f, 0x563d, 0x612b, 0x7936, 0x5475, 0x5049, 0x6f27,
 1684 0x626c, 0x5b6a, 0x4e4c, 0x7568, 0x7755, 0x534d, 0x737e, 0x5035,
 1685 0x607e, 0x5f7b, 0x665d, 0x6824, 0x4b4d, 0x6f28, 0x6e34, 0x5a58,
 1686 0x5139, 0x5f29, 0x7330, 0x4c44, 0x4e37, 0x6f29, 0x5f55, 0x6d57,
 1687 0x6e46, 0x6f3d, 0x7c56, 0x5b74, 0x6f2a, 0x7839, 0x7569, 0x6359,

```
1688 0x6146, 0x543f, 0x5e68, 0x706a, 0x7342, 0x532f, 0x4a5b, 0x7c57,
1689 0x6d58, 0x6147, 0x7458, 0x5633, 0x5d2d, 0x553e, 0x7143, 0x6e5f,
1690 0x566b, 0x7459, 0x5766, 0x5a37, 0x5d7b, 0x5d4f, 0x5823, 0x5a59,
1691 0x7058, 0x6f44, 0x6158, 0x7154, 0x6d72, 0x555b, 0x555c, 0x7344,
1692 0x4b57, 0x6236, 0x6f71, 0x7b55, 0x5358, 0x5d50, 0x7059, 0x4b33,
1693 0x555d, 0x4d27, 0x502c, 0x513a, 0x7144, 0x6533, 0x7b75, 0x6961,
1694 0x7d60, 0x7c3c, 0x5a22, 0x5a23, 0x5221, 0x526f, 0x626d, 0x5e69,
1695 0x4e5c, 0x7235, 0x5064, 0x5d51, 0x6148, 0x5b37, 0x5f63, 0x6d39,
1696 0x7145, 0x734f, 0x572b, 0x612c, 0x636b, 0x6e47, 0x6149, 0x4a7a,
1697 0x707b, 0x7a61, 0x705a, 0x4c67, 0x5a74, 0x4c3f, 0x4e6d, 0x5529,
1698 0x7a62, 0x5065, 0x6b56, 0x6c5f, 0x5f7c, 0x7756, 0x5e6a, 0x4b34,
1699 0x6f3e, 0x4c35, 0x4f3d, 0x6f72, 0x6237, 0x4c68, 0x707c, 0x5660,
1700 0x7146, 0x6238, 0x6b2b, 0x4b35, 0x5851, 0x744e, 0x7377, 0x5746,
1701 0x513b, 0x772a, 0x6d4a, 0x5753, 0x587a, 0x7645, 0x514c, 0x5d7c,
1702 0x5f7d, 0x7965, 0x604a, 0x727d, 0x5330, 0x7473, 0x5a49, 0x665e,
1703 0x783a, 0x6850, 0x587b, 0x6a55, 0x5623, 0x7646, 0x725b, 0x647c,
1704 0x6832, 0x5a5a, 0x725c, 0x7b56, 0x6932, 0x6e2d, 0x7a63, 0x5c6e,
1705 0x756a, 0x6660, 0x707d, 0x572c, 0x7545, 0x6e60, 0x5b65, 0x5d5e,
1706 0x5970, 0x6923, 0x7179, 0x7244, 0x604b, 0x6924, 0x6239, 0x6331,
1707 0x7c6b, 0x4d28, 0x4c36, 0x705b, 0x663a, 0x4d29, 0x7343, 0x6159,
1708 0x6f2b, 0x6745, 0x6069, 0x7345, 0x5440, 0x553f, 0x5d2e, 0x797c,
1709 0x4c40, 0x6522, 0x4e38, 0x5852, 0x7956, 0x712a, 0x4e51, 0x7647,
1710 0x5b6b, 0x5f7e, 0x5861, 0x7773, 0x5767, 0x547e, 0x513c, 0x654f,
1711 0x4b36, 0x5a38, 0x4d44, 0x563e, 0x623a, 0x4f58, 0x604c, 0x6b79,
1712 0x7d7d, 0x5768, 0x4b58, 0x6962, 0x683a, 0x6347, 0x6c4d, 0x6c4e,
1713 0x563f, 0x6327, 0x5f56, 0x7d68, 0x6e61, 0x7628, 0x5d7d, 0x783b,
1714 0x6851, 0x7957, 0x4e6e, 0x6c4f, 0x6925, 0x6555, 0x4d45, 0x6d3a,
1715 0x513d, 0x4f3e, 0x6c3b, 0x5231, 0x4c69, 0x5944, 0x697c, 0x513e,
1716 0x6c3c, 0x652d, 0x7730, 0x4c6a, 0x5344, 0x5640, 0x567d, 0x6121,
1717 0x5e3d, 0x7629, 0x5a24, 0x562a, 0x7546, 0x6122, 0x6946, 0x7245,
1718 0x7469, 0x566c, 0x6b53, 0x6c3d, 0x625c, 0x5e6b, 0x705c, 0x6b3f,
1719 0x574e, 0x513f, 0x752b, 0x797d, 0x4a5c, 0x4d46, 0x7236, 0x5d7e,
1720 0x4c37, 0x5b38, 0x5069, 0x4e5d, 0x6b40, 0x7d22, 0x784b, 0x6a56,
1721 0x7130, 0x5b4e, 0x7743, 0x5b4f, 0x4b24, 0x7860, 0x7b57, 0x6b4a,
1722 0x6021, 0x4e4d, 0x545c, 0x7d58, 0x5276, 0x7237, 0x7a76, 0x762a,
1723 0x7a77, 0x5866, 0x7431, 0x6852, 0x4a45, 0x4c6b, 0x626e, 0x623b,
1724 0x772d, 0x7861, 0x736c, 0x5e21, 0x647d, 0x636c, 0x5d2f, 0x5d30,
1725 0x4b37, 0x6853, 0x6123, 0x5260, 0x707e, 0x6926, 0x4b72, 0x6d73,
1726 0x5c59, 0x604d, 0x775a, 0x5b39, 0x4c2e, 0x5a5b, 0x4d47, 0x5d31,
1727 0x582f, 0x6323, 0x4e6f, 0x7273, 0x7833, 0x604e, 0x757d, 0x6b6c,
1728 0x5345, 0x7c6c, 0x525b, 0x546b, 0x5e22, 0x6566, 0x7030, 0x5544,
1729 0x6d74, 0x636d, 0x6842, 0x6d75, 0x577c, 0x6d3b, 0x762b, 0x7238,
1730 0x7648, 0x5366, 0x725d, 0x4f3f, 0x6b2c, 0x4f40, 0x6628, 0x7d69,
1731 0x4f41, 0x605f, 0x5e6c, 0x6022, 0x743f, 0x626f, 0x5971, 0x7147,
1732 0x4b38, 0x797e, 0x5b3a, 0x5a75, 0x766c, 0x5a5c, 0x7a64, 0x604f,
1733 0x5d32, 0x6629, 0x6f73, 0x736d, 0x6b7a, 0x7966, 0x4a5d, 0x555e,
1734 0x4a5e, 0x5f64, 0x667d, 0x752c, 0x6475, 0x6963, 0x6d4b, 0x4f64,
1735 0x5853, 0x5d33, 0x546c, 0x7239, 0x5f37, 0x4b4e, 0x7b58, 0x5059,
1736 0x5d52, 0x7774, 0x675c, 0x6425, 0x7c23, 0x5b3b, 0x723a, 0x697d,
1737 0x504a, 0x7556, 0x5945, 0x6434, 0x6d27, 0x6a3d, 0x667e, 0x7744,
1738 0x752d, 0x5960, 0x4a34, 0x7862, 0x4f42, 0x6c3e, 0x6534, 0x4d48,
1739 0x6e48, 0x6748, 0x4d49, 0x7937, 0x7168, 0x5972, 0x5b75, 0x4a35,
1740 0x5946, 0x5849, 0x592b, 0x6d3c, 0x5854, 0x5c5a, 0x623c, 0x7c6d,
1741 0x6c60, 0x527e, 0x6947, 0x662a, 0x6270, 0x7a3b, 0x752e, 0x7b2a,
1742 0x6c7b, 0x6c3f, 0x7c58, 0x5465, 0x7943, 0x6e62, 0x5769, 0x6d76,
1743 0x5e6d, 0x4c6c, 0x636e, 0x6854, 0x7a78, 0x5d34, 0x6435, 0x5830,
1744 0x5855, 0x746a, 0x4e39, 0x5661, 0x4f52, 0x5036, 0x4e22, 0x736e,
1745 0x7378, 0x5c4c, 0x504b, 0x7c24, 0x4d4a, 0x5754, 0x5e23, 0x6460,
1746 0x6e49, 0x625d, 0x757e, 0x542c, 0x5551, 0x5870, 0x7843, 0x6a57,
1747 0x7557, 0x583f, 0x7d40, 0x6b2d, 0x552a, 0x6728, 0x6e4a, 0x4a67,
1748 0x7863, 0x545d, 0x6a58, 0x7b59, 0x6d77, 0x6535, 0x502d, 0x7171,
1749 0x623d, 0x6348, 0x5955, 0x5f2a, 0x5b3c, 0x7864, 0x717a, 0x6536,
1750 0x736f, 0x7b5a, 0x6160, 0x592c, 0x756b, 0x6036, 0x6948, 0x4b4f,
1751 0x6349, 0x5e6e, 0x623e, 0x5c6f, 0x5625, 0x6271, 0x567e, 0x5921,
1752 0x5840, 0x5c5b, 0x6d3d, 0x5f38, 0x6a25, 0x572d, 0x7379, 0x6d78,
1753 0x7547, 0x614a, 0x6b63, 0x725e, 0x784c, 0x6a59, 0x5346, 0x5b66,
1754 0x752f, 0x4e70, 0x697e, 0x7b36, 0x6272, 0x4f72, 0x7739, 0x5973,
1755 0x614b, 0x5a5d, 0x5a39, 0x6b7b, 0x4b39, 0x6d79, 0x6060, 0x7440,
1756 0x7d3c, 0x5f31, 0x636f, 0x6023, 0x7d39, 0x7031, 0x4d4b, 0x6d3e,
1757 0x5540, 0x6370, 0x6d7a, 0x6964, 0x556d, 0x675d, 0x5476, 0x6537,
1758 0x5b67, 0x623f, 0x6e4b, 0x5774, 0x705d, 0x4e2b, 0x675e, 0x5656,
1759 0x614c, 0x6833, 0x656e, 0x5c22, 0x6050, 0x5535, 0x5521, 0x7b5b,
1760 0x794b, 0x4b73, 0x7425, 0x7a48, 0x5657, 0x6965, 0x7b5c, 0x7d50,
1761 0x7b76, 0x5a25, 0x5b3d, 0x6c62, 0x4d77, 0x705e, 0x7649, 0x5e6f,
1762 0x5331, 0x7c6e, 0x6843, 0x7148, 0x4e71, 0x796d, 0x7274, 0x6436,
1763 0x7539, 0x5c70, 0x6371, 0x6825, 0x723b, 0x5e24, 0x5a4c, 0x4a69,
1764 0x635a, 0x7c59, 0x6a5a, 0x7944, 0x6324, 0x7b5d, 0x6f4a, 0x6844,
1765 0x554c, 0x6b57, 0x592d, 0x7b2b, 0x5359, 0x5522, 0x765e, 0x5a76,
1766 0x6051, 0x6928, 0x7579, 0x7a2f, 0x6b7c, 0x606a, 0x6332, 0x5545,
1767 0x7163, 0x556e, 0x4d4c, 0x6d59, 0x5841, 0x7a6c, 0x716b, 0x7a3c,
1768 0x6662, 0x7a65, 0x627a, 0x4a36, 0x6437, 0x6a5b, 0x757a, 0x7b2c,
1769 0x4f43, 0x6b7d, 0x787a, 0x5f39, 0x6171, 0x5224, 0x757b, 0x505a,
1770 0x505b, 0x6a3e, 0x5931, 0x4a37, 0x5367, 0x7865, 0x5332, 0x6240,
1771 0x725f, 0x4d65, 0x792c, 0x4d4d, 0x6e2e, 0x562e, 0x576a, 0x6760,
1772 0x6b2e, 0x4f59, 0x5c4d, 0x6d7b, 0x5e70, 0x576b, 0x5e25, 0x5f57,
1773 0x5b50, 0x5b51, 0x5523, 0x7032, 0x5c5c, 0x4a68, 0x7866, 0x5c4e,
1774 0x6a5c, 0x5b52, 0x6933, 0x775b, 0x6328, 0x572e, 0x6061, 0x4b3a,
```


1775 0x6551, 0x505c, 0x5541, 0x584a, 0x6329, 0x6024, 0x6929, 0x5347,
1776 0x5c5d, 0x782e, 0x4c38, 0x502e, 0x5872, 0x634a, 0x4c2f, 0x542d,
1777 0x7651, 0x504c, 0x4a46, 0x5542, 0x4e3a, 0x4a47, 0x7a30, 0x5f58,
1778 0x753a, 0x656b, 0x6f74, 0x5d35, 0x4d2a, 0x6372, 0x7b77, 0x7750,
1779 0x7d3a, 0x7d61, 0x767e, 0x5140, 0x6845, 0x6438, 0x6168, 0x4c41,
1780 0x526d, 0x5b3e, 0x6062, 0x7a49, 0x614d, 0x4a38, 0x7260, 0x7149,
1781 0x5e71, 0x705f, 0x7844, 0x6e4c, 0x5e72, 0x6749, 0x6273, 0x6761,
1782 0x634b, 0x634c, 0x4f78, 0x6f2c, 0x7d7e, 0x7c25, 0x7a31, 0x5f59,
1783 0x6052, 0x745a, 0x714a, 0x4e23, 0x723c, 0x6c63, 0x6025, 0x772b,
1784 0x6b2f, 0x655e, 0x6124, 0x4d2b, 0x5974, 0x6826, 0x4d4e, 0x6169,
1785 0x7c6f, 0x6063, 0x6241, 0x4e24, 0x5e26, 0x6b7e, 0x6b5d, 0x7060,
1786 0x745b, 0x6274, 0x5348, 0x746b, 0x6e35, 0x7558, 0x555f, 0x5665,
1787 0x6b30, 0x7463, 0x634d, 0x7474, 0x7a32, 0x6f75, 0x4a5f, 0x6b31,
1788 0x6d3f, 0x7d49, 0x6426, 0x7924, 0x7033, 0x656c, 0x5167, 0x5947,
1789 0x6457, 0x6a5d, 0x5477, 0x5a3a, 0x5a4d, 0x794c, 0x615a, 0x5b3f,
1790 0x4c45, 0x6c50, 0x4b3b, 0x5e73, 0x692a, 0x5948, 0x6e63, 0x573d,
1791 0x4f44, 0x504d, 0x7c26, 0x717b, 0x7d52, 0x5141, 0x635b, 0x5349,
1792 0x5c4f, 0x4c6d, 0x4e27, 0x5e27, 0x663b, 0x6c21, 0x4c39, 0x7b5e, 0x6762,
1793 0x5441, 0x5c28, 0x6242, 0x7358, 0x6553, 0x7359, 0x7346, 0x4d5b,
1794 0x4d2c, 0x7c43, 0x5467, 0x5142, 0x7925, 0x6855, 0x634e, 0x544a,
1795 0x5f5a, 0x7b5f, 0x6763, 0x787b, 0x634f, 0x7530, 0x5867, 0x5949,
1796 0x782f, 0x6f76, 0x5d36, 0x6e2f, 0x4d78, 0x5e38, 0x7c27, 0x777c,
1797 0x7731, 0x4e3b, 0x7421, 0x6e4d, 0x612e, 0x6c43, 0x4f7e, 0x783f,
1798 0x5862, 0x5368, 0x5e28, 0x7464, 0x6c42, 0x5975, 0x7945, 0x5d53,
1799 0x5671, 0x6c7c, 0x7c70, 0x6d40, 0x4a39, 0x6e64, 0x7261, 0x5e39,
1800 0x5672, 0x5e74, 0x5f5b, 0x5b53, 0x7a67, 0x5863, 0x7441, 0x5d37,
1801 0x7275, 0x542e, 0x5673, 0x5d38, 0x4f45, 0x5f5f, 0x723e, 0x7621,
1802 0x6b4b, 0x717c, 0x7347, 0x606b, 0x6d7c, 0x615b, 0x6e65, 0x5e75,
1803 0x7a53, 0x714b, 0x502f, 0x5d39, 0x5143, 0x7531, 0x6a46, 0x7061,
1804 0x762c, 0x7559, 0x706b, 0x5d3a, 0x723f, 0x7745, 0x5b22, 0x727e,
1805 0x4a3a, 0x7775, 0x4b65, 0x6e66, 0x6053, 0x4e25, 0x5658, 0x542f,
1806 0x6949, 0x534e, 0x7442, 0x4b66, 0x7121, 0x6b32, 0x7122, 0x6b33,
1807 0x7034, 0x4b74, 0x5430, 0x7332, 0x7b37, 0x756c, 0x6e67, 0x7432,
1808 0x756d, 0x4f73, 0x7062, 0x6e4e, 0x714c, 0x6538, 0x5775, 0x6373,
1809 0x4f65, 0x4f46, 0x7333, 0x6458, 0x4f79, 0x4f5a, 0x7a4d, 0x6663,
1810 0x7262, 0x756e, 0x4a3b, 0x635c, 0x4e72, 0x5659, 0x6e30, 0x7465,
1811 0x5842, 0x5c50, 0x4c6e, 0x5560, 0x764a, 0x7d4a, 0x5856, 0x744f,
1812 0x5626, 0x5c3e, 0x5b54, 0x5747, 0x727e, 0x714d, 0x6243, 0x5c5e,
1813 0x5c5f, 0x6f2d, 0x662b, 0x795d, 0x6a3f, 0x6f2e, 0x7450, 0x4e73,
1814 0x662c, 0x4e5e, 0x5579, 0x6374, 0x4d50, 0x5538, 0x777d, 0x5c29,
1815 0x5e76, 0x5c2a, 0x7263, 0x6934, 0x525c, 0x6966, 0x6376, 0x674a,
1816 0x504e, 0x5a77, 0x4a3c, 0x6e68, 0x5a5e, 0x7277, 0x627b, 0x4c2e,
1817 0x5a3b, 0x6e69, 0x755a, 0x775c, 0x616a, 0x4e41, 0x5431, 0x7d31,
1818 0x663d, 0x7b2d, 0x7867, 0x614e, 0x7762, 0x756f, 0x4f47, 0x5432,
1819 0x4c6f, 0x5468, 0x6e4f, 0x7757, 0x6026, 0x5641, 0x615c, 0x7063,
1820 0x7164, 0x5c71, 0x5627, 0x7475, 0x714e, 0x7264, 0x5030, 0x6c6f,
1821 0x793a, 0x6b35, 0x546d, 0x6244, 0x6967, 0x6b34, 0x6a21, 0x783c,
1822 0x4e26, 0x7946, 0x7c5a, 0x5433, 0x5339, 0x6a5e, 0x692b, 0x6161,
1823 0x534f, 0x7476, 0x6a40, 0x614f, 0x4c3a, 0x6e6a, 0x7064, 0x7334,
1824 0x546e, 0x7240, 0x7165, 0x7443, 0x6054, 0x6b36, 0x5721, 0x4b68,
1825 0x792d, 0x692d, 0x5864, 0x7a33, 0x6245, 0x7c3d, 0x6c44, 0x5831,
1826 0x5c2b, 0x5524, 0x6b69, 0x683b, 0x5857, 0x7b2e, 0x5161, 0x5b40,
1827 0x753e, 0x5e77, 0x4a7b, 0x7746, 0x4f48, 0x6150, 0x6e50, 0x6974,
1828 0x4e74, 0x554d, 0x4f5b, 0x5d3b, 0x4e2c, 0x6968, 0x5434, 0x6447,
1829 0x755b, 0x7a41, 0x5e29, 0x5478, 0x6f77, 0x5333, 0x6b37, 0x6f78,
1830 0x755c, 0x6d4c, 0x5b55, 0x714f, 0x7150, 0x7532, 0x592e, 0x552c,
1831 0x6246, 0x7d23, 0x7b65, 0x5f2b, 0x6275, 0x762d, 0x7533, 0x7035,
1832 0x6125, 0x755d, 0x6c22, 0x6d7d, 0x7534, 0x7b38, 0x5b23, 0x564a,
1833 0x4b59, 0x6554, 0x737a, 0x6b38, 0x6037, 0x576c, 0x716c, 0x652f,
1834 0x5561, 0x576d, 0x5151, 0x6172, 0x6f79, 0x5d3c, 0x765c, 0x7065,
1835 0x7444, 0x6969, 0x737b, 0x546f, 0x4c22, 0x777e, 0x5f3c, 0x6b4d,
1836 0x5037, 0x5642, 0x682d, 0x6f2f, 0x4b25, 0x4b69, 0x7a68, 0x4c46,
1837 0x6667, 0x6a47, 0x5b24, 0x4f49, 0x627c, 0x6f7a, 0x6b5e, 0x7548,
1838 0x545e, 0x6055, 0x6f30, 0x6247, 0x592f, 0x7967, 0x6765, 0x4f4a,
1839 0x6151, 0x6248, 0x6f7b, 0x7a79, 0x5c72, 0x6027, 0x7868, 0x4b6a,
1840 0x4b3c, 0x5662, 0x755e, 0x755f, 0x6e36, 0x6276, 0x534a, 0x6f7c,
1841 0x5144, 0x6f31, 0x5145, 0x505e, 0x5961, 0x6038, 0x4d51, 0x7339,
1842 0x674c, 0x5628, 0x4e27, 0x5435, 0x6448, 0x5334, 0x6b39, 0x4b75,
1843 0x765d, 0x7123, 0x4c47, 0x694a, 0x6170, 0x7560, 0x7b2f, 0x4b51,
1844 0x7b60, 0x7265, 0x6c70, 0x706c, 0x6e6b, 0x694b, 0x4c70, 0x572f,
1845 0x7321, 0x7c75, 0x7124, 0x6056, 0x6f32, 0x7451, 0x7721, 0x7151,
1846 0x4a7c, 0x4a7d, 0x4e4e, 0x7348, 0x733a, 0x6d7e, 0x5a26, 0x606c,
1847 0x784d, 0x4b52, 0x6b4e, 0x7958, 0x7959, 0x4a60, 0x5a4a, 0x4b26,
1848 0x4a48, 0x796e, 0x5b6c, 0x5031, 0x556f, 0x6673, 0x6722, 0x6459,
1849 0x6461, 0x7c44, 0x796f, 0x4f74, 0x7766, 0x4e3c, 0x7445, 0x5c23,
1850 0x5d3d, 0x7446, 0x7821, 0x6856, 0x5b41, 0x7066, 0x6439, 0x766d,
1851 0x792e, 0x5d3e, 0x5730, 0x5868, 0x4b3d, 0x795a, 0x784e, 0x7970,
1852 0x606d, 0x6333, 0x7433, 0x6a42, 0x7266, 0x7036, 0x5b56, 0x6b64,
1853 0x7267, 0x5755, 0x5436, 0x7968, 0x5741, 0x6555, 0x696a, 0x574c,
1854 0x5369, 0x6249, 0x7c5b, 0x4d2d, 0x4c30, 0x6a22, 0x6476, 0x5040,
1855 0x7037, 0x6e21, 0x5776, 0x624a, 0x624b, 0x7a4f, 0x6b5f, 0x564b,
1856 0x7434, 0x6d4d, 0x6452, 0x6a29, 0x643a, 0x7322, 0x4d52, 0x764b,
1857 0x7166, 0x6d41, 0x683c, 0x6e51, 0x7067, 0x624c, 0x642a, 0x7561,
1858 0x6d5a, 0x576e, 0x5171, 0x696b, 0x696c, 0x6064, 0x5a27, 0x5d54,
1859 0x6a23, 0x5643, 0x5674, 0x5a5f, 0x6f33, 0x624d, 0x6f7d, 0x7268,
1860 0x6f45, 0x6767, 0x577d, 0x674e, 0x5f5c, 0x7947, 0x5976, 0x5f2c,
1861 0x565a, 0x5c24, 0x7038, 0x557a, 0x6477, 0x5644, 0x746c, 0x6f7e,

1862 0x7021, 0x5e2a, 0x5a3c, 0x587c, 0x7a54, 0x6c65, 0x7c28, 0x6c66,
1863 0x584b, 0x7b39, 0x6453, 0x4d79, 0x4f53, 0x4a6a, 0x4f54, 0x783d,
1864 0x7447, 0x6a5f, 0x795b, 0x5437, 0x6b65, 0x6152, 0x6a24, 0x7a42,
1865 0x7b61, 0x7a6d, 0x7022, 0x4c71, 0x7a23, 0x6277, 0x624e, 0x6975,
1866 0x616b, 0x6768, 0x6857, 0x5a78, 0x544b, 0x7776, 0x5645, 0x5469,
1867 0x7a7a, 0x4c72, 0x775d, 0x5e3a, 0x4e28, 0x7039, 0x647e, 0x6449,
1868 0x6454, 0x6a43, 0x6f34, 0x573e, 0x7b62, 0x4d53, 0x6f35, 0x7a69,
1869 0x7926, 0x5f3d, 0x7747, 0x787d, 0x787c, 0x5e2b, 0x5b68, 0x635d,
1870 0x6162, 0x5146, 0x7650, 0x6b66, 0x5a79, 0x6c47, 0x5e78, 0x7869,
1871 0x635e, 0x4e75, 0x7a43, 0x6557, 0x6c48, 0x7349, 0x643b, 0x662e,
1872 0x6f36, 0x5c3f, 0x4e3d, 0x5843, 0x504f, 0x4f7a, 0x734a, 0x6057,
1873 0x5147, 0x692e, 0x683d, 0x7a44, 0x624f, 0x7a45, 0x7938, 0x5c60,
1874 0x7b30, 0x5829, 0x655f, 0x7927, 0x766e, 0x764c, 0x6278, 0x6c71,
1875 0x5a60, 0x7152, 0x524c, 0x4f4b, 0x4a3d, 0x5d3f, 0x766f, 0x5e79,
1876 0x7a34, 0x552d, 0x7167, 0x5e3e, 0x5c40, 0x5148, 0x5149, 0x783e,
1877 0x4b76, 0x5479, 0x7562, 0x6153, 0x5869, 0x787e, 0x4f4c, 0x7d24,
1878 0x4e76, 0x7a50, 0x4c73, 0x6636, 0x762e, 0x5570, 0x514a, 0x7c3e,
1879 0x5571, 0x4d46, 0x7a35, 0x6250, 0x7477, 0x4d54, 0x6723, 0x5b25,
1880 0x6251, 0x5722, 0x7763, 0x6a26, 0x5021, 0x4e5a, 0x7b6b, 0x5b26,
1881 0x5b5e, 0x5865, 0x6a60, 0x582a, 0x6560, 0x565b, 0x6f46, 0x786a,
1882 0x6455, 0x4e77, 0x6058, 0x576f, 0x746d, 0x4d66, 0x4c74, 0x7563,
1883 0x644a, 0x5c61, 0x7948, 0x7c3f, 0x6827, 0x5844, 0x4b3e, 0x5c2e,
1884 0x5777, 0x7068, 0x5d40, 0x4f4d, 0x5c73, 0x5930, 0x6669, 0x643c,
1885 0x6a44, 0x646c, 0x6465, 0x7b78, 0x4c3b, 0x643d, 0x4d5c, 0x5977,
1886 0x5d5f, 0x6d4e, 0x5950, 0x6523, 0x794d, 0x4d2e, 0x4f4e, 0x762f,
1887 0x7d53, 0x6b6d, 0x565c, 0x6524, 0x5536, 0x565d, 0x7969, 0x6724,
1888 0x5663, 0x514b, 0x5664, 0x5572, 0x5e7a, 0x5778, 0x586a, 0x4f55,
1889 0x587d, 0x582b, 0x7d4b, 0x7c5c, 0x6028, 0x5573, 0x7d59, 0x4c23,
1890 0x5979, 0x536a, 0x7575, 0x6f47, 0x535a, 0x5a3d, 0x6828, 0x5c2f,
1891 0x7023, 0x4d55, 0x6029, 0x5e2c, 0x703a, 0x6e31, 0x6e32, 0x764d,
1892 0x6e52, 0x5646, 0x6065, 0x733b, 0x6561, 0x644b, 0x5723, 0x5b42,
1893 0x4a7e, 0x4f4f, 0x3021, 0x3022, 0x3023, 0x3024, 0x3025, 0x3026,
1894 0x3027, 0x3028, 0x3029, 0x302a, 0x302b, 0x302c, 0x302d, 0x302e,
1895 0x302f, 0x3030, 0x3031, 0x3032, 0x3033, 0x3034, 0x3035, 0x3036,
1896 0x3037, 0x3038, 0x3039, 0x303a, 0x303b, 0x303c, 0x303d, 0x303e,
1897 0x303f, 0x3040, 0x3041, 0x3042, 0x3043, 0x3044, 0x3045, 0x3046,
1898 0x3047, 0x3048, 0x3049, 0x304a, 0x304b, 0x304c, 0x304d, 0x304e,
1899 0x304f, 0x3050, 0x3051, 0x3052, 0x3053, 0x3054, 0x3055, 0x3056,
1900 0x3057, 0x3058, 0x3059, 0x305a, 0x305b, 0x305c, 0x305d, 0x305e,
1901 0x305f, 0x3060, 0x3061, 0x3062, 0x3063, 0x3064, 0x3065, 0x3066,
1902 0x3067, 0x3068, 0x3069, 0x306a, 0x306b, 0x306c, 0x306d, 0x306e,
1903 0x306f, 0x3070, 0x3071, 0x3072, 0x3073, 0x3074, 0x3075, 0x3076,
1904 0x3077, 0x3078, 0x3079, 0x307a, 0x307b, 0x307c, 0x307d, 0x307e,
1905 0x3121, 0x3122, 0x3123, 0x3124, 0x3125, 0x3126, 0x3127, 0x3128,
1906 0x3129, 0x312a, 0x312b, 0x312c, 0x312d, 0x312e, 0x312f, 0x3130,
1907 0x3131, 0x3132, 0x3133, 0x3134, 0x3135, 0x3136, 0x3137, 0x3138,
1908 0x3139, 0x313a, 0x313b, 0x313c, 0x313d, 0x313e, 0x313f, 0x3140,
1909 0x3141, 0x3142, 0x3143, 0x3144, 0x3145, 0x3146, 0x3147, 0x3148,
1910 0x3149, 0x314a, 0x314b, 0x314c, 0x314d, 0x314e, 0x314f, 0x3150,
1911 0x3151, 0x3152, 0x3153, 0x3154, 0x3155, 0x3156, 0x3157, 0x3158,
1912 0x3159, 0x315a, 0x315b, 0x315c, 0x315d, 0x315e, 0x315f, 0x3160,
1913 0x3161, 0x3162, 0x3163, 0x3164, 0x3165, 0x3166, 0x3167, 0x3168,
1914 0x3169, 0x316a, 0x316b, 0x316c, 0x316d, 0x316e, 0x316f, 0x3170,
1915 0x3171, 0x3172, 0x3173, 0x3174, 0x3175, 0x3176, 0x3177, 0x3178,
1916 0x3179, 0x317a, 0x317b, 0x317c, 0x317d, 0x317e, 0x3221, 0x3222,
1917 0x3223, 0x3224, 0x3225, 0x3226, 0x3227, 0x3228, 0x3229, 0x322a,
1918 0x322b, 0x322c, 0x322d, 0x322e, 0x322f, 0x3230, 0x3231, 0x3232,
1919 0x3233, 0x3234, 0x3235, 0x3236, 0x3237, 0x3238, 0x3239, 0x323a,
1920 0x323b, 0x323c, 0x323d, 0x323e, 0x323f, 0x3240, 0x3241, 0x3242,
1921 0x3243, 0x3244, 0x3245, 0x3246, 0x3247, 0x3248, 0x3249, 0x324a,
1922 0x324b, 0x324c, 0x324d, 0x324e, 0x324f, 0x3250, 0x3251, 0x3252,
1923 0x3253, 0x3254, 0x3255, 0x3256, 0x3257, 0x3258, 0x3259, 0x325a,
1924 0x325b, 0x325c, 0x325d, 0x325e, 0x325f, 0x3260, 0x3261, 0x3262,
1925 0x3263, 0x3264, 0x3265, 0x3266, 0x3267, 0x3268, 0x3269, 0x326a,
1926 0x326b, 0x326c, 0x326d, 0x326e, 0x326f, 0x3270, 0x3271, 0x3272,
1927 0x3273, 0x3274, 0x3275, 0x3276, 0x3277, 0x3278, 0x3279, 0x327a,
1928 0x327b, 0x327c, 0x327d, 0x327e, 0x3321, 0x3322, 0x3323, 0x3324,
1929 0x3325, 0x3326, 0x3327, 0x3328, 0x3329, 0x332a, 0x332b, 0x332c,
1930 0x332d, 0x332e, 0x332f, 0x3330, 0x3331, 0x3332, 0x3333, 0x3334,
1931 0x3335, 0x3336, 0x3337, 0x3338, 0x3339, 0x333a, 0x333b, 0x333c,
1932 0x333d, 0x333e, 0x333f, 0x3340, 0x3341, 0x3342, 0x3343, 0x3344,
1933 0x3345, 0x3346, 0x3347, 0x3348, 0x3349, 0x334a, 0x334b, 0x334c,
1934 0x334d, 0x334e, 0x334f, 0x3350, 0x3351, 0x3352, 0x3353, 0x3354,
1935 0x3355, 0x3356, 0x3357, 0x3358, 0x3359, 0x335a, 0x335b, 0x335c,
1936 0x335d, 0x335e, 0x335f, 0x3360, 0x3361, 0x3362, 0x3363, 0x3364,
1937 0x3365, 0x3366, 0x3367, 0x3368, 0x3369, 0x336a, 0x336b, 0x336c,
1938 0x336d, 0x336e, 0x336f, 0x3370, 0x3371, 0x3372, 0x3373, 0x3374,
1939 0x3375, 0x3376, 0x3377, 0x3378, 0x3379, 0x337a, 0x337b, 0x337c,
1940 0x337d, 0x337e, 0x3421, 0x3422, 0x3423, 0x3424, 0x3425, 0x3426,
1941 0x3427, 0x3428, 0x3429, 0x342a, 0x342b, 0x342c, 0x342d, 0x342e,
1942 0x342f, 0x3430, 0x3431, 0x3432, 0x3433, 0x3434, 0x3435, 0x3436,
1943 0x3437, 0x3438, 0x3439, 0x343a, 0x343b, 0x343c, 0x343d, 0x343e,
1944 0x343f, 0x3440, 0x3441, 0x3442, 0x3443, 0x3444, 0x3445, 0x3446,
1945 0x3447, 0x3448, 0x3449, 0x344a, 0x344b, 0x344c, 0x344d, 0x344e,
1946 0x344f, 0x3450, 0x3451, 0x3452, 0x3453, 0x3454, 0x3455, 0x3456,
1947 0x3457, 0x3458, 0x3459, 0x345a, 0x345b, 0x345c, 0x345d, 0x345e,
1948 0x345f, 0x3460, 0x3461, 0x3462, 0x3463, 0x3464, 0x3465, 0x3466,

1949 0x3467, 0x3468, 0x3469, 0x346a, 0x346b, 0x346c, 0x346d, 0x346e,
1950 0x346f, 0x3470, 0x3471, 0x3472, 0x3473, 0x3474, 0x3475, 0x3476,
1951 0x3477, 0x3478, 0x3479, 0x347a, 0x347b, 0x347c, 0x347d, 0x347e,
1952 0x3521, 0x3522, 0x3523, 0x3524, 0x3525, 0x3526, 0x3527, 0x3528,
1953 0x3529, 0x352a, 0x352b, 0x352c, 0x352d, 0x352e, 0x352f, 0x3530,
1954 0x3531, 0x3532, 0x3533, 0x3534, 0x3535, 0x3536, 0x3537, 0x3538,
1955 0x3539, 0x353a, 0x353b, 0x353c, 0x353d, 0x353e, 0x353f, 0x3540,
1956 0x3541, 0x3542, 0x3543, 0x3544, 0x3545, 0x3546, 0x3547, 0x3548,
1957 0x3549, 0x354a, 0x354b, 0x354c, 0x354d, 0x354e, 0x354f, 0x3550,
1958 0x3551, 0x3552, 0x3553, 0x3554, 0x3555, 0x3556, 0x3557, 0x3558,
1959 0x3559, 0x355a, 0x355b, 0x355c, 0x355d, 0x355e, 0x355f, 0x3560,
1960 0x3561, 0x3562, 0x3563, 0x3564, 0x3565, 0x3566, 0x3567, 0x3568,
1961 0x3569, 0x356a, 0x356b, 0x356c, 0x356d, 0x356e, 0x356f, 0x3570,
1962 0x3571, 0x3572, 0x3573, 0x3574, 0x3575, 0x3576, 0x3577, 0x3578,
1963 0x3579, 0x357a, 0x357b, 0x357c, 0x357d, 0x357e, 0x357f, 0x3580,
1964 0x3623, 0x3624, 0x3625, 0x3626, 0x3627, 0x3628, 0x3629, 0x362a,
1965 0x362b, 0x362c, 0x362d, 0x362e, 0x362f, 0x3630, 0x3631, 0x3632,
1966 0x3633, 0x3634, 0x3635, 0x3636, 0x3637, 0x3638, 0x3639, 0x363a,
1967 0x363b, 0x363c, 0x363d, 0x363e, 0x363f, 0x3640, 0x3641, 0x3642,
1968 0x3643, 0x3644, 0x3645, 0x3646, 0x3647, 0x3648, 0x3649, 0x364a,
1969 0x364b, 0x364c, 0x364d, 0x364e, 0x364f, 0x3650, 0x3651, 0x3652,
1970 0x3653, 0x3654, 0x3655, 0x3656, 0x3657, 0x3658, 0x3659, 0x365a,
1971 0x365b, 0x365c, 0x365d, 0x365e, 0x365f, 0x3660, 0x3661, 0x3662,
1972 0x3663, 0x3664, 0x3665, 0x3666, 0x3667, 0x3668, 0x3669, 0x366a,
1973 0x366b, 0x366c, 0x366d, 0x366e, 0x366f, 0x3670, 0x3671, 0x3672,
1974 0x3673, 0x3674, 0x3675, 0x3676, 0x3677, 0x3678, 0x3679, 0x367a,
1975 0x367b, 0x367c, 0x367d, 0x367e, 0x367f, 0x3680, 0x3681, 0x3682,
1976 0x3725, 0x3726, 0x3727, 0x3728, 0x3729, 0x372a, 0x372b, 0x372c,
1977 0x372d, 0x372e, 0x372f, 0x3730, 0x3731, 0x3732, 0x3733, 0x3734,
1978 0x3735, 0x3736, 0x3737, 0x3738, 0x3739, 0x373a, 0x373b, 0x373c,
1979 0x373d, 0x373e, 0x373f, 0x3740, 0x3741, 0x3742, 0x3743, 0x3744,
1980 0x3745, 0x3746, 0x3747, 0x3748, 0x3749, 0x374a, 0x374b, 0x374c,
1981 0x374d, 0x374e, 0x374f, 0x3750, 0x3751, 0x3752, 0x3753, 0x3754,
1982 0x3755, 0x3756, 0x3757, 0x3758, 0x3759, 0x375a, 0x375b, 0x375c,
1983 0x375d, 0x375e, 0x375f, 0x3760, 0x3761, 0x3762, 0x3763, 0x3764,
1984 0x3765, 0x3766, 0x3767, 0x3768, 0x3769, 0x376a, 0x376b, 0x376c,
1985 0x376d, 0x376e, 0x376f, 0x3770, 0x3771, 0x3772, 0x3773, 0x3774,
1986 0x3775, 0x3776, 0x3777, 0x3778, 0x3779, 0x377a, 0x377b, 0x377c,
1987 0x377d, 0x377e, 0x377f, 0x3780, 0x3781, 0x3782, 0x3783, 0x3784,
1988 0x3827, 0x3828, 0x3829, 0x382a, 0x382b, 0x382c, 0x382d, 0x382e,
1989 0x382f, 0x3830, 0x3831, 0x3832, 0x3833, 0x3834, 0x3835, 0x3836,
1990 0x3837, 0x3838, 0x3839, 0x383a, 0x383b, 0x383c, 0x383d, 0x383e,
1991 0x383f, 0x3840, 0x3841, 0x3842, 0x3843, 0x3844, 0x3845, 0x3846,
1992 0x3847, 0x3848, 0x3849, 0x384a, 0x384b, 0x384c, 0x384d, 0x384e,
1993 0x384f, 0x3850, 0x3851, 0x3852, 0x3853, 0x3854, 0x3855, 0x3856,
1994 0x3857, 0x3858, 0x3859, 0x385a, 0x385b, 0x385c, 0x385d, 0x385e,
1995 0x385f, 0x3860, 0x3861, 0x3862, 0x3863, 0x3864, 0x3865, 0x3866,
1996 0x3867, 0x3868, 0x3869, 0x386a, 0x386b, 0x386c, 0x386d, 0x386e,
1997 0x386f, 0x3870, 0x3871, 0x3872, 0x3873, 0x3874, 0x3875, 0x3876,
1998 0x3877, 0x3878, 0x3879, 0x387a, 0x387b, 0x387c, 0x387d, 0x387e,
1999 0x3921, 0x3922, 0x3923, 0x3924, 0x3925, 0x3926, 0x3927, 0x3928,
2000 0x3929, 0x392a, 0x392b, 0x392c, 0x392d, 0x392e, 0x392f, 0x3930,
2001 0x3931, 0x3932, 0x3933, 0x3934, 0x3935, 0x3936, 0x3937, 0x3938,
2002 0x3939, 0x393a, 0x393b, 0x393c, 0x393d, 0x393e, 0x393f, 0x3940,
2003 0x3941, 0x3942, 0x3943, 0x3944, 0x3945, 0x3946, 0x3947, 0x3948,
2004 0x3949, 0x394a, 0x394b, 0x394c, 0x394d, 0x394e, 0x394f, 0x3950,
2005 0x3951, 0x3952, 0x3953, 0x3954, 0x3955, 0x3956, 0x3957, 0x3958,
2006 0x3959, 0x395a, 0x395b, 0x395c, 0x395d, 0x395e, 0x395f, 0x3960,
2007 0x3961, 0x3962, 0x3963, 0x3964, 0x3965, 0x3966, 0x3967, 0x3968,
2008 0x3969, 0x396a, 0x396b, 0x396c, 0x396d, 0x396e, 0x396f, 0x3970,
2009 0x3971, 0x3972, 0x3973, 0x3974, 0x3975, 0x3976, 0x3977, 0x3978,
2010 0x3979, 0x397a, 0x397b, 0x397c, 0x397d, 0x397e, 0x397f, 0x3980,
2011 0x3a23, 0x3a24, 0x3a25, 0x3a26, 0x3a27, 0x3a28, 0x3a29, 0x3a2a,
2012 0x3a2b, 0x3a2c, 0x3a2d, 0x3a2e, 0x3a2f, 0x3a30, 0x3a31, 0x3a32,
2013 0x3a33, 0x3a34, 0x3a35, 0x3a36, 0x3a37, 0x3a38, 0x3a39, 0x3a3a,
2014 0x3a3b, 0x3a3c, 0x3a3d, 0x3a3e, 0x3a3f, 0x3a40, 0x3a41, 0x3a42,
2015 0x3a43, 0x3a44, 0x3a45, 0x3a46, 0x3a47, 0x3a48, 0x3a49, 0x3a4a,
2016 0x3a4b, 0x3a4c, 0x3a4d, 0x3a4e, 0x3a4f, 0x3a50, 0x3a51, 0x3a52,
2017 0x3a53, 0x3a54, 0x3a55, 0x3a56, 0x3a57, 0x3a58, 0x3a59, 0x3a5a,
2018 0x3a5b, 0x3a5c, 0x3a5d, 0x3a5e, 0x3a5f, 0x3a60, 0x3a61, 0x3a62,
2019 0x3a63, 0x3a64, 0x3a65, 0x3a66, 0x3a67, 0x3a68, 0x3a69, 0x3a6a,
2020 0x3a6b, 0x3a6c, 0x3a6d, 0x3a6e, 0x3a6f, 0x3a70, 0x3a71, 0x3a72,
2021 0x3a73, 0x3a74, 0x3a75, 0x3a76, 0x3a77, 0x3a78, 0x3a79, 0x3a7a,
2022 0x3a7b, 0x3a7c, 0x3a7d, 0x3a7e, 0x3a7f, 0x3a80, 0x3a81, 0x3a82,
2023 0x3b25, 0x3b26, 0x3b27, 0x3b28, 0x3b29, 0x3b2a, 0x3b2b, 0x3b2c,
2024 0x3b2d, 0x3b2e, 0x3b2f, 0x3b30, 0x3b31, 0x3b32, 0x3b33, 0x3b34,
2025 0x3b35, 0x3b36, 0x3b37, 0x3b38, 0x3b39, 0x3b3a, 0x3b3b, 0x3b3c,
2026 0x3b3d, 0x3b3e, 0x3b3f, 0x3b40, 0x3b41, 0x3b42, 0x3b43, 0x3b44,
2027 0x3b45, 0x3b46, 0x3b47, 0x3b48, 0x3b49, 0x3b4a, 0x3b4b, 0x3b4c,
2028 0x3b4d, 0x3b4e, 0x3b4f, 0x3b50, 0x3b51, 0x3b52, 0x3b53, 0x3b54,
2029 0x3b55, 0x3b56, 0x3b57, 0x3b58, 0x3b59, 0x3b5a, 0x3b5b, 0x3b5c,
2030 0x3b5d, 0x3b5e, 0x3b5f, 0x3b60, 0x3b61, 0x3b62, 0x3b63, 0x3b64,
2031 0x3b65, 0x3b66, 0x3b67, 0x3b68, 0x3b69, 0x3b6a, 0x3b6b, 0x3b6c,
2032 0x3b6d, 0x3b6e, 0x3b6f, 0x3b70, 0x3b71, 0x3b72, 0x3b73, 0x3b74,
2033 0x3b75, 0x3b76, 0x3b77, 0x3b78, 0x3b79, 0x3b7a, 0x3b7b, 0x3b7c,
2034 0x3b7d, 0x3b7e, 0x3b7f, 0x3b80, 0x3b81, 0x3b82, 0x3b83, 0x3b84,
2035 0x3c27, 0x3c28, 0x3c29, 0x3c2a, 0x3c2b, 0x3c2c, 0x3c2d, 0x3c2e,

2036 0x3c2f, 0x3c30, 0x3c31, 0x3c32, 0x3c33, 0x3c34, 0x3c35, 0x3c36,
2037 0x3c37, 0x3c38, 0x3c39, 0x3c3a, 0x3c3b, 0x3c3c, 0x3c3d, 0x3c3e,
2038 0x3c3f, 0x3c40, 0x3c41, 0x3c42, 0x3c43, 0x3c44, 0x3c45, 0x3c46,
2039 0x3c47, 0x3c48, 0x3c49, 0x3c4a, 0x3c4b, 0x3c4c, 0x3c4d, 0x3c4e,
2040 0x3c4f, 0x3c50, 0x3c51, 0x3c52, 0x3c53, 0x3c54, 0x3c55, 0x3c56,
2041 0x3c57, 0x3c58, 0x3c59, 0x3c5a, 0x3c5b, 0x3c5c, 0x3c5d, 0x3c5e,
2042 0x3c5f, 0x3c60, 0x3c61, 0x3c62, 0x3c63, 0x3c64, 0x3c65, 0x3c66,
2043 0x3c67, 0x3c68, 0x3c69, 0x3c6a, 0x3c6b, 0x3c6c, 0x3c6d, 0x3c6e,
2044 0x3c6f, 0x3c70, 0x3c71, 0x3c72, 0x3c73, 0x3c74, 0x3c75, 0x3c76,
2045 0x3c77, 0x3c78, 0x3c79, 0x3c7a, 0x3c7b, 0x3c7c, 0x3c7d, 0x3c7e,
2046 0x3d21, 0x3d22, 0x3d23, 0x3d24, 0x3d25, 0x3d26, 0x3d27, 0x3d28,
2047 0x3d29, 0x3d2a, 0x3d2b, 0x3d2c, 0x3d2d, 0x3d2e, 0x3d2f, 0x3d30,
2048 0x3d31, 0x3d32, 0x3d33, 0x3d34, 0x3d35, 0x3d36, 0x3d37, 0x3d38,
2049 0x3d39, 0x3d3a, 0x3d3b, 0x3d3c, 0x3d3d, 0x3d3e, 0x3d3f, 0x3d40,
2050 0x3d41, 0x3d42, 0x3d43, 0x3d44, 0x3d45, 0x3d46, 0x3d47, 0x3d48,
2051 0x3d49, 0x3d4a, 0x3d4b, 0x3d4c, 0x3d4d, 0x3d4e, 0x3d4f, 0x3d50,
2052 0x3d51, 0x3d52, 0x3d53, 0x3d54, 0x3d55, 0x3d56, 0x3d57, 0x3d58,
2053 0x3d59, 0x3d5a, 0x3d5b, 0x3d5c, 0x3d5d, 0x3d5e, 0x3d5f, 0x3d60,
2054 0x3d61, 0x3d62, 0x3d63, 0x3d64, 0x3d65, 0x3d66, 0x3d67, 0x3d68,
2055 0x3d69, 0x3d6a, 0x3d6b, 0x3d6c, 0x3d6d, 0x3d6e, 0x3d6f, 0x3d70,
2056 0x3d71, 0x3d72, 0x3d73, 0x3d74, 0x3d75, 0x3d76, 0x3d77, 0x3d78,
2057 0x3d79, 0x3d7a, 0x3d7b, 0x3d7c, 0x3d7d, 0x3d7e, 0x3e21, 0x3e22,
2058 0x3e23, 0x3e24, 0x3e25, 0x3e26, 0x3e27, 0x3e28, 0x3e29, 0x3e2a,
2059 0x3e2b, 0x3e2c, 0x3e2d, 0x3e2e, 0x3e2f, 0x3e30, 0x3e31, 0x3e32,
2060 0x3e33, 0x3e34, 0x3e35, 0x3e36, 0x3e37, 0x3e38, 0x3e39, 0x3e3a,
2061 0x3e3b, 0x3e3c, 0x3e3d, 0x3e3e, 0x3e3f, 0x3e40, 0x3e41, 0x3e42,
2062 0x3e43, 0x3e44, 0x3e45, 0x3e46, 0x3e47, 0x3e48, 0x3e49, 0x3e4a,
2063 0x3e4b, 0x3e4c, 0x3e4d, 0x3e4e, 0x3e4f, 0x3e50, 0x3e51, 0x3e52,
2064 0x3e53, 0x3e54, 0x3e55, 0x3e56, 0x3e57, 0x3e58, 0x3e59, 0x3e5a,
2065 0x3e5b, 0x3e5c, 0x3e5d, 0x3e5e, 0x3e5f, 0x3e60, 0x3e61, 0x3e62,
2066 0x3e63, 0x3e64, 0x3e65, 0x3e66, 0x3e67, 0x3e68, 0x3e69, 0x3e6a,
2067 0x3e6b, 0x3e6c, 0x3e6d, 0x3e6e, 0x3e6f, 0x3e70, 0x3e71, 0x3e72,
2068 0x3e73, 0x3e74, 0x3e75, 0x3e76, 0x3e77, 0x3e78, 0x3e79, 0x3e7a,
2069 0x3e7b, 0x3e7c, 0x3e7d, 0x3e7e, 0x3f21, 0x3f22, 0x3f23, 0x3f24,
2070 0x3f25, 0x3f26, 0x3f27, 0x3f28, 0x3f29, 0x3f2a, 0x3f2b, 0x3f2c,
2071 0x3f2d, 0x3f2e, 0x3f2f, 0x3f30, 0x3f31, 0x3f32, 0x3f33, 0x3f34,
2072 0x3f35, 0x3f36, 0x3f37, 0x3f38, 0x3f39, 0x3f3a, 0x3f3b, 0x3f3c,
2073 0x3f3d, 0x3f3e, 0x3f3f, 0x3f40, 0x3f41, 0x3f42, 0x3f43, 0x3f44,
2074 0x3f45, 0x3f46, 0x3f47, 0x3f48, 0x3f49, 0x3f4a, 0x3f4b, 0x3f4c,
2075 0x3f4d, 0x3f4e, 0x3f4f, 0x3f50, 0x3f51, 0x3f52, 0x3f53, 0x3f54,
2076 0x3f55, 0x3f56, 0x3f57, 0x3f58, 0x3f59, 0x3f5a, 0x3f5b, 0x3f5c,
2077 0x3f5d, 0x3f5e, 0x3f5f, 0x3f60, 0x3f61, 0x3f62, 0x3f63, 0x3f64,
2078 0x3f65, 0x3f66, 0x3f67, 0x3f68, 0x3f69, 0x3f6a, 0x3f6b, 0x3f6c,
2079 0x3f6d, 0x3f6e, 0x3f6f, 0x3f70, 0x3f71, 0x3f72, 0x3f73, 0x3f74,
2080 0x3f75, 0x3f76, 0x3f77, 0x3f78, 0x3f79, 0x3f7a, 0x3f7b, 0x3f7c,
2081 0x3f7d, 0x3f7e, 0x4021, 0x4022, 0x4023, 0x4024, 0x4025, 0x4026,
2082 0x4027, 0x4028, 0x4029, 0x402a, 0x402b, 0x402c, 0x402d, 0x402e,
2083 0x402f, 0x4030, 0x4031, 0x4032, 0x4033, 0x4034, 0x4035, 0x4036,
2084 0x4037, 0x4038, 0x4039, 0x403a, 0x403b, 0x403c, 0x403d, 0x403e,
2085 0x403f, 0x4040, 0x4041, 0x4042, 0x4043, 0x4044, 0x4045, 0x4046,
2086 0x4047, 0x4048, 0x4049, 0x404a, 0x404b, 0x404c, 0x404d, 0x404e,
2087 0x404f, 0x4050, 0x4051, 0x4052, 0x4053, 0x4054, 0x4055, 0x4056,
2088 0x4057, 0x4058, 0x4059, 0x405a, 0x405b, 0x405c, 0x405d, 0x405e,
2089 0x405f, 0x4060, 0x4061, 0x4062, 0x4063, 0x4064, 0x4065, 0x4066,
2090 0x4067, 0x4068, 0x4069, 0x406a, 0x406b, 0x406c, 0x406d, 0x406e,
2091 0x406f, 0x4070, 0x4071, 0x4072, 0x4073, 0x4074, 0x4075, 0x4076,
2092 0x4077, 0x4078, 0x4079, 0x407a, 0x407b, 0x407c, 0x407d, 0x407e,
2093 0x4121, 0x4122, 0x4123, 0x4124, 0x4125, 0x4126, 0x4127, 0x4128,
2094 0x4129, 0x412a, 0x412b, 0x412c, 0x412d, 0x412e, 0x412f, 0x4130,
2095 0x4131, 0x4132, 0x4133, 0x4134, 0x4135, 0x4136, 0x4137, 0x4138,
2096 0x4139, 0x413a, 0x413b, 0x413c, 0x413d, 0x413e, 0x413f, 0x4140,
2097 0x4141, 0x4142, 0x4143, 0x4144, 0x4145, 0x4146, 0x4147, 0x4148,
2098 0x4149, 0x414a, 0x414b, 0x414c, 0x414d, 0x414e, 0x414f, 0x4150,
2099 0x4151, 0x4152, 0x4153, 0x4154, 0x4155, 0x4156, 0x4157, 0x4158,
2100 0x4159, 0x415a, 0x415b, 0x415c, 0x415d, 0x415e, 0x415f, 0x4160,
2101 0x4161, 0x4162, 0x4163, 0x4164, 0x4165, 0x4166, 0x4167, 0x4168,
2102 0x4169, 0x416a, 0x416b, 0x416c, 0x416d, 0x416e, 0x416f, 0x4170,
2103 0x4171, 0x4172, 0x4173, 0x4174, 0x4175, 0x4176, 0x4177, 0x4178,
2104 0x4179, 0x417a, 0x417b, 0x417c, 0x417d, 0x417e, 0x4221, 0x4222,
2105 0x4223, 0x4224, 0x4225, 0x4226, 0x4227, 0x4228, 0x4229, 0x422a,
2106 0x422b, 0x422c, 0x422d, 0x422e, 0x422f, 0x4230, 0x4231, 0x4232,
2107 0x4233, 0x4234, 0x4235, 0x4236, 0x4237, 0x4238, 0x4239, 0x423a,
2108 0x423b, 0x423c, 0x423d, 0x423e, 0x423f, 0x4240, 0x4241, 0x4242,
2109 0x4243, 0x4244, 0x4245, 0x4246, 0x4247, 0x4248, 0x4249, 0x424a,
2110 0x424b, 0x424c, 0x424d, 0x424e, 0x424f, 0x4250, 0x4251, 0x4252,
2111 0x4253, 0x4254, 0x4255, 0x4256, 0x4257, 0x4258, 0x4259, 0x425a,
2112 0x425b, 0x425c, 0x425d, 0x425e, 0x425f, 0x4260, 0x4261, 0x4262,
2113 0x4263, 0x4264, 0x4265, 0x4266, 0x4267, 0x4268, 0x4269, 0x426a,
2114 0x426b, 0x426c, 0x426d, 0x426e, 0x426f, 0x4270, 0x4271, 0x4272,
2115 0x4273, 0x4274, 0x4275, 0x4276, 0x4277, 0x4278, 0x4279, 0x427a,
2116 0x427b, 0x427c, 0x427d, 0x427e, 0x4321, 0x4322, 0x4323, 0x4324,
2117 0x4325, 0x4326, 0x4327, 0x4328, 0x4329, 0x432a, 0x432b, 0x432c,
2118 0x432d, 0x432e, 0x432f, 0x4330, 0x4331, 0x4332, 0x4333, 0x4334,
2119 0x4335, 0x4336, 0x4337, 0x4338, 0x4339, 0x433a, 0x433b, 0x433c,
2120 0x433d, 0x433e, 0x433f, 0x4340, 0x4341, 0x4342, 0x4343, 0x4344,
2121 0x4345, 0x4346, 0x4347, 0x4348, 0x4349, 0x434a, 0x434b, 0x434c,
2122 0x434d, 0x434e, 0x434f, 0x4350, 0x4351, 0x4352, 0x4353, 0x4354,

2123 0x4355, 0x4356, 0x4357, 0x4358, 0x4359, 0x435a, 0x435b, 0x435c,
2124 0x435d, 0x435e, 0x435f, 0x4360, 0x4361, 0x4362, 0x4363, 0x4364,
2125 0x4365, 0x4366, 0x4367, 0x4368, 0x4369, 0x436a, 0x436b, 0x436c,
2126 0x436d, 0x436e, 0x436f, 0x4370, 0x4371, 0x4372, 0x4373, 0x4374,
2127 0x4375, 0x4376, 0x4377, 0x4378, 0x4379, 0x437a, 0x437b, 0x437c,
2128 0x437d, 0x437e, 0x4421, 0x4422, 0x4423, 0x4424, 0x4425, 0x4426,
2129 0x4427, 0x4428, 0x4429, 0x442a, 0x442b, 0x442c, 0x442d, 0x442e,
2130 0x442f, 0x4430, 0x4431, 0x4432, 0x4433, 0x4434, 0x4435, 0x4436,
2131 0x4437, 0x4438, 0x4439, 0x443a, 0x443b, 0x443c, 0x443d, 0x443e,
2132 0x443f, 0x4440, 0x4441, 0x4442, 0x4443, 0x4444, 0x4445, 0x4446,
2133 0x4447, 0x4448, 0x4449, 0x444a, 0x444b, 0x444c, 0x444d, 0x444e,
2134 0x444f, 0x4450, 0x4451, 0x4452, 0x4453, 0x4454, 0x4455, 0x4456,
2135 0x4457, 0x4458, 0x4459, 0x445a, 0x445b, 0x445c, 0x445d, 0x445e,
2136 0x445f, 0x4460, 0x4461, 0x4462, 0x4463, 0x4464, 0x4465, 0x4466,
2137 0x4467, 0x4468, 0x4469, 0x446a, 0x446b, 0x446c, 0x446d, 0x446e,
2138 0x446f, 0x4470, 0x4471, 0x4472, 0x4473, 0x4474, 0x4475, 0x4476,
2139 0x4477, 0x4478, 0x4479, 0x447a, 0x447b, 0x447c, 0x447d, 0x447e,
2140 0x4521, 0x4522, 0x4523, 0x4524, 0x4525, 0x4526, 0x4527, 0x4528,
2141 0x4529, 0x452a, 0x452b, 0x452c, 0x452d, 0x452e, 0x452f, 0x4530,
2142 0x4531, 0x4532, 0x4533, 0x4534, 0x4535, 0x4536, 0x4537, 0x4538,
2143 0x4539, 0x453a, 0x453b, 0x453c, 0x453d, 0x453e, 0x453f, 0x4540,
2144 0x4541, 0x4542, 0x4543, 0x4544, 0x4545, 0x4546, 0x4547, 0x4548,
2145 0x4549, 0x454a, 0x454b, 0x454c, 0x454d, 0x454e, 0x454f, 0x4550,
2146 0x4551, 0x4552, 0x4553, 0x4554, 0x4555, 0x4556, 0x4557, 0x4558,
2147 0x4559, 0x455a, 0x455b, 0x455c, 0x455d, 0x455e, 0x455f, 0x4560,
2148 0x4561, 0x4562, 0x4563, 0x4564, 0x4565, 0x4566, 0x4567, 0x4568,
2149 0x4569, 0x456a, 0x456b, 0x456c, 0x456d, 0x456e, 0x456f, 0x4570,
2150 0x4571, 0x4572, 0x4573, 0x4574, 0x4575, 0x4576, 0x4577, 0x4578,
2151 0x4579, 0x457a, 0x457b, 0x457c, 0x457d, 0x457e, 0x4621, 0x4622,
2152 0x4623, 0x4624, 0x4625, 0x4626, 0x4627, 0x4628, 0x4629, 0x462a,
2153 0x462b, 0x462c, 0x462d, 0x462e, 0x462f, 0x4630, 0x4631, 0x4632,
2154 0x4633, 0x4634, 0x4635, 0x4636, 0x4637, 0x4638, 0x4639, 0x463a,
2155 0x463b, 0x463c, 0x463d, 0x463e, 0x463f, 0x4640, 0x4641, 0x4642,
2156 0x4643, 0x4644, 0x4645, 0x4646, 0x4647, 0x4648, 0x4649, 0x464a,
2157 0x464b, 0x464c, 0x464d, 0x464e, 0x464f, 0x4650, 0x4651, 0x4652,
2158 0x4653, 0x4654, 0x4655, 0x4656, 0x4657, 0x4658, 0x4659, 0x465a,
2159 0x465b, 0x465c, 0x465d, 0x465e, 0x465f, 0x4660, 0x4661, 0x4662,
2160 0x4663, 0x4664, 0x4665, 0x4666, 0x4667, 0x4668, 0x4669, 0x466a,
2161 0x466b, 0x466c, 0x466d, 0x466e, 0x466f, 0x4670, 0x4671, 0x4672,
2162 0x4673, 0x4674, 0x4675, 0x4676, 0x4677, 0x4678, 0x4679, 0x467a,
2163 0x467b, 0x467c, 0x467d, 0x467e, 0x4721, 0x4722, 0x4723, 0x4724,
2164 0x4725, 0x4726, 0x4727, 0x4728, 0x4729, 0x472a, 0x472b, 0x472c,
2165 0x472d, 0x472e, 0x472f, 0x4730, 0x4731, 0x4732, 0x4733, 0x4734,
2166 0x4735, 0x4736, 0x4737, 0x4738, 0x4739, 0x473a, 0x473b, 0x473c,
2167 0x473d, 0x473e, 0x473f, 0x4740, 0x4741, 0x4742, 0x4743, 0x4744,
2168 0x4745, 0x4746, 0x4747, 0x4748, 0x4749, 0x474a, 0x474b, 0x474c,
2169 0x474d, 0x474e, 0x474f, 0x4750, 0x4751, 0x4752, 0x4753, 0x4754,
2170 0x4755, 0x4756, 0x4757, 0x4758, 0x4759, 0x475a, 0x475b, 0x475c,
2171 0x475d, 0x475e, 0x475f, 0x4760, 0x4761, 0x4762, 0x4763, 0x4764,
2172 0x4765, 0x4766, 0x4767, 0x4768, 0x4769, 0x476a, 0x476b, 0x476c,
2173 0x476d, 0x476e, 0x476f, 0x4770, 0x4771, 0x4772, 0x4773, 0x4774,
2174 0x4775, 0x4776, 0x4777, 0x4778, 0x4779, 0x477a, 0x477b, 0x477c,
2175 0x477d, 0x477e, 0x4821, 0x4822, 0x4823, 0x4824, 0x4825, 0x4826,
2176 0x4827, 0x4828, 0x4829, 0x482a, 0x482b, 0x482c, 0x482d, 0x482e,
2177 0x482f, 0x4830, 0x4831, 0x4832, 0x4833, 0x4834, 0x4835, 0x4836,
2178 0x4837, 0x4838, 0x4839, 0x483a, 0x483b, 0x483c, 0x483d, 0x483e,
2179 0x483f, 0x4840, 0x4841, 0x4842, 0x4843, 0x4844, 0x4845, 0x4846,
2180 0x4847, 0x4848, 0x4849, 0x484a, 0x484b, 0x484c, 0x484d, 0x484e,
2181 0x484f, 0x4850, 0x4851, 0x4852, 0x4853, 0x4854, 0x4855, 0x4856,
2182 0x4857, 0x4858, 0x4859, 0x485a, 0x485b, 0x485c, 0x485d, 0x485e,
2183 0x485f, 0x4860, 0x4861, 0x4862, 0x4863, 0x4864, 0x4865, 0x4866,
2184 0x4867, 0x4868, 0x4869, 0x486a, 0x486b, 0x486c, 0x486d, 0x486e,
2185 0x486f, 0x4870, 0x4871, 0x4872, 0x4873, 0x4874, 0x4875, 0x4876,
2186 0x4877, 0x4878, 0x4879, 0x487a, 0x487b, 0x487c, 0x487d, 0x487e,
2187 0x4b50, 0x4b56, 0x4b67, 0x4d4f, 0x4d68, 0x4e2d, 0x4f7b, 0x5022,
2188 0x5038, 0x5050, 0x505d, 0x5154, 0x5155, 0x5158, 0x515b, 0x515c,
2189 0x515d, 0x515e, 0x515f, 0x5160, 0x5162, 0x5163, 0x5164, 0x5165,
2190 0x5166, 0x5168, 0x5169, 0x516a, 0x516b, 0x516d, 0x516f, 0x5170,
2191 0x5172, 0x5176, 0x517a, 0x517c, 0x517d, 0x517e, 0x5222, 0x5223,
2192 0x5227, 0x5228, 0x5229, 0x522a, 0x522b, 0x522d, 0x5232, 0x523e,
2193 0x5242, 0x5243, 0x5244, 0x5246, 0x5247, 0x5248, 0x5249, 0x524a,
2194 0x524b, 0x524d, 0x524e, 0x524f, 0x5250, 0x5251, 0x5252, 0x5253,
2195 0x5254, 0x5255, 0x5256, 0x5257, 0x5259, 0x525a, 0x525e, 0x525f,
2196 0x5261, 0x5262, 0x5264, 0x5265, 0x5266, 0x5267, 0x5268, 0x5269,
2197 0x526a, 0x526b, 0x5270, 0x5271, 0x5272, 0x5273, 0x5274, 0x5275,
2198 0x5277, 0x5278, 0x5466, 0x547c, 0x5525, 0x5526, 0x552e, 0x5638,
2199 0x564d, 0x574b, 0x5764, 0x5b45, 0x5b64, 0x5c25, 0x5d25, 0x5d55,
2200 0x5d74, 0x5e7c, 0x5e7e, 0x5f33, 0x5f61, 0x5f68, 0x6071, 0x612d,
2201 0x616d, 0x6375, 0x6421, 0x6429, 0x652e, 0x6531, 0x6532, 0x6539,
2202 0x653b, 0x653c, 0x6544, 0x654e, 0x6550, 0x6552, 0x655e, 0x657a,
2203 0x657b, 0x657c, 0x657e, 0x6621, 0x6624, 0x6627, 0x662d, 0x662f,
2204 0x6630, 0x6631, 0x6633, 0x6637, 0x6638, 0x663c, 0x6644, 0x6646,
2205 0x6647, 0x664a, 0x6652, 0x665e, 0x6659, 0x665c, 0x665f, 0x6661,
2206 0x6664, 0x6665, 0x6666, 0x6668, 0x666a, 0x666b, 0x666c, 0x666f,
2207 0x6671, 0x6672, 0x6675, 0x6676, 0x6677, 0x6679, 0x6721, 0x6726,
2208 0x6729, 0x672a, 0x672c, 0x672d, 0x6730, 0x673f, 0x6741, 0x6746,
2209 0x6747, 0x674b, 0x674d, 0x674f, 0x6750, 0x6753, 0x675f, 0x6764,

```

2210 0x6766, 0x6777, 0x6867, 0x6868, 0x6870, 0x6871, 0x6877, 0x6879,
2211 0x687b, 0x687e, 0x6927, 0x692c, 0x694c, 0x6977, 0x6a41, 0x6a65,
2212 0x6a74, 0x6a77, 0x6a7c, 0x6a7e, 0x6b24, 0x6b27, 0x6b29, 0x6b2a,
2213 0x6b3a, 0x6b3b, 0x6b3d, 0x6b41, 0x6b42, 0x6b46, 0x6b47, 0x6b4c,
2214 0x6b4f, 0x6b50, 0x6b51, 0x6b52, 0x6b58, 0x6c26, 0x6c27, 0x6c2a,
2215 0x6c2f, 0x6c30, 0x6c31, 0x6c32, 0x6c35, 0x6c38, 0x6c3a, 0x6c40,
2216 0x6c41, 0x6c45, 0x6c46, 0x6c49, 0x6c4a, 0x6c55, 0x6c5d, 0x6c5e,
2217 0x6c61, 0x6c64, 0x6c67, 0x6c68, 0x6c77, 0x6c78, 0x6c7a, 0x6d21,
2218 0x6d22, 0x6d23, 0x6d6e, 0x6e5b, 0x723d, 0x727a, 0x7331, 0x7427,
2219 0x746e, 0x7674, 0x7676, 0x7738, 0x7748, 0x7753, 0x785b, 0x7870,
2220 0x7a21, 0x7a22, 0x7a66, 0x7c29, 0x2321, 0x2322, 0x2323, 0x2324,
2221 0x2325, 0x2326, 0x2327, 0x2328, 0x2329, 0x232a, 0x232b, 0x232c,
2222 0x232d, 0x232e, 0x232f, 0x2330, 0x2331, 0x2332, 0x2333, 0x2334,
2223 0x2335, 0x2336, 0x2337, 0x2338, 0x2339, 0x233a, 0x233b, 0x233c,
2224 0x233d, 0x233e, 0x233f, 0x2340, 0x2341, 0x2342, 0x2343, 0x2344,
2225 0x2345, 0x2346, 0x2347, 0x2348, 0x2349, 0x234a, 0x234b, 0x234c,
2226 0x234d, 0x234e, 0x234f, 0x2350, 0x2351, 0x2352, 0x2353, 0x2354,
2227 0x2355, 0x2356, 0x2357, 0x2358, 0x2359, 0x235a, 0x235b, 0x212c,
2228 0x235d, 0x235e, 0x235f, 0x2360, 0x2361, 0x2362, 0x2363, 0x2364,
2229 0x2365, 0x2366, 0x2367, 0x2368, 0x2369, 0x236a, 0x236b, 0x236c,
2230 0x236d, 0x236e, 0x236f, 0x2370, 0x2371, 0x2372, 0x2373, 0x2374,
2231 0x2375, 0x2376, 0x2377, 0x2378, 0x2379, 0x237a, 0x237b, 0x237c,
2232 0x237d, 0x2226, 0x214b, 0x214c, 0x217e, 0x237e, 0x214d, 0x235c,
2233 };
2234
2235 static const Summary16 ksc5601_uni2indx_page00[70] = {
2236 /* 0x0000 */
2237 { 0, 0x0000 }, { 0, 0x0000 }, { 0, 0x0000 }, { 0, 0x0000 },
2238 { 0, 0x0000 }, { 0, 0x0000 }, { 0, 0x0000 }, { 0, 0x0000 },
2239 { 0, 0x0000 }, { 0, 0x0000 }, { 0, 0x2592 }, { 6, 0xf7df },
2240 { 20, 0x0040 }, { 21, 0xc181 }, { 26, 0x0040 }, { 27, 0x4181 },
2241 /* 0x0100 */
2242 { 31, 0x0000 }, { 31, 0x0002 }, { 32, 0x00c0 }, { 34, 0x810e },
2243 { 39, 0x0e07 }, { 45, 0x000c }, { 47, 0x00c0 }, { 49, 0x0000 },
2244 { 49, 0x0000 }, { 49, 0x0000 }, { 49, 0x0000 }, { 49, 0x0000 },
2245 { 49, 0x0000 }, { 49, 0x0000 }, { 49, 0x0000 }, { 49, 0x0000 },
2246 /* 0x0200 */
2247 { 49, 0x0000 }, { 49, 0x0000 }, { 49, 0x0000 }, { 49, 0x0000 },
2248 { 49, 0x0000 }, { 49, 0x0000 }, { 49, 0x0000 }, { 49, 0x0000 },
2249 { 49, 0x0000 }, { 49, 0x0000 }, { 49, 0x0000 }, { 49, 0x0000 },
2250 { 49, 0x0080 }, { 50, 0x2f01 }, { 56, 0x0000 }, { 56, 0x0000 },
2251 /* 0x0300 */
2252 { 56, 0x0000 }, { 56, 0x0000 }, { 56, 0x0000 }, { 56, 0x0000 },
2253 { 56, 0x0000 }, { 56, 0x0000 }, { 56, 0x0000 }, { 56, 0x0000 },
2254 { 56, 0x0000 }, { 56, 0xffff }, { 71, 0x03fb }, { 80, 0xffff },
2255 { 95, 0x03fb }, { 104, 0x0000 }, { 104, 0x0000 }, { 104, 0x0000 },
2256 /* 0x0400 */
2257 { 104, 0x0002 }, { 105, 0xffff }, { 121, 0xffff }, { 137, 0xffff },
2258 { 153, 0xffff }, { 169, 0x0002 },
2259 };
2260 static const Summary16 ksc5601_uni2indx_page20[103] = {
2261 /* 0x2000 */
2262 { 170, 0x0000 }, { 170, 0x3320 }, { 175, 0x0063 }, { 179, 0x080d },
2263 { 183, 0x0000 }, { 183, 0x0000 }, { 183, 0x0000 }, { 183, 0x8010 },
2264 { 185, 0x001e }, { 189, 0x0000 }, { 189, 0x0000 }, { 189, 0x0000 },
2265 { 189, 0x0000 }, { 189, 0x0000 }, { 189, 0x0000 }, { 189, 0x0000 },
2266 /* 0x2100 */
2267 { 189, 0x0208 }, { 191, 0x0048 }, { 193, 0x0846 }, { 197, 0x0000 },
2268 { 197, 0x0000 }, { 197, 0x7818 }, { 203, 0x03ff }, { 213, 0x03ff },
2269 { 223, 0x0000 }, { 223, 0x03ff }, { 233, 0x0000 }, { 233, 0x0000 },
2270 { 233, 0x0000 }, { 233, 0x0014 }, { 235, 0x0000 }, { 235, 0x0000 },
2271 /* 0x2200 */
2272 { 235, 0x898d }, { 242, 0x6402 }, { 246, 0x5fa1 }, { 255, 0x3030 },
2273 { 259, 0x0000 }, { 259, 0x0004 }, { 260, 0x0c33 }, { 266, 0x0000 },
2274 { 266, 0x00cc }, { 270, 0x0200 }, { 271, 0x0020 }, { 272, 0x0000 },
2275 { 272, 0x0000 }, { 272, 0x0000 }, { 272, 0x0000 }, { 272, 0x0000 },
2276 /* 0x2300 */
2277 { 272, 0x0000 }, { 272, 0x0004 }, { 273, 0x0000 }, { 273, 0x0000 },
2278 { 273, 0x0000 }, { 273, 0x0000 }, { 273, 0x0000 }, { 273, 0x0000 },
2279 { 273, 0x0000 }, { 273, 0x0000 }, { 273, 0x0000 }, { 273, 0x0000 },
2280 { 273, 0x0000 }, { 273, 0x0000 }, { 273, 0x0000 }, { 273, 0x0000 },
2281 /* 0x2400 */
2282 { 273, 0x0000 }, { 273, 0x0000 }, { 273, 0x0000 }, { 273, 0x0000 },
2283 { 273, 0x0000 }, { 273, 0x0000 }, { 273, 0x7fff }, { 288, 0xffff },
2284 { 300, 0x0007 }, { 303, 0xf000 }, { 307, 0xffff }, { 323, 0x003f },
2285 { 329, 0x0000 }, { 329, 0xffff }, { 345, 0x03ff }, { 355, 0x0000 },
2286 /* 0x2500 */
2287 { 355, 0xf00f }, { 363, 0xffff }, { 379, 0xffff }, { 395, 0xffff },
2288 { 411, 0xffff }, { 423, 0x0000 }, { 423, 0x0000 }, { 423, 0x0000 },
2289 { 423, 0x0000 }, { 423, 0x0004 }, { 424, 0x03fb }, { 433, 0x30cc },
2290 { 439, 0xc9c3 }, { 447, 0x0003 }, { 449, 0x0000 }, { 449, 0x0000 },
2291 /* 0x2600 */
2292 { 449, 0xc060 }, { 453, 0x5000 }, { 455, 0x0000 }, { 455, 0x0000 },
2293 { 455, 0x0005 }, { 457, 0x0000 }, { 457, 0x37bb },
2294 };
2295 static const Summary16 ksc5601_uni2indx_page30[62] = {
2296 /* 0x3000 */

```

```

2297 { 468, 0xff0f }, { 480, 0x003b }, { 485, 0x0000 }, { 485, 0x0000 },
2298 { 485, 0xffff }, { 500, 0xffff }, { 516, 0xffff }, { 532, 0xffff },
2299 { 548, 0xffff }, { 564, 0x000f }, { 568, 0xffff }, { 583, 0xffff },
2300 { 599, 0xffff }, { 615, 0xffff }, { 631, 0xffff }, { 647, 0x007f },
2301 /* 0x3100 */
2302 { 654, 0x0000 }, { 654, 0x0000 }, { 654, 0x0000 }, { 654, 0xffff },
2303 { 669, 0xffff }, { 685, 0xffff }, { 701, 0xffff }, { 717, 0xffff },
2304 { 733, 0x7fff }, { 748, 0x0000 }, { 748, 0x0000 }, { 748, 0x0000 },
2305 { 748, 0x0000 }, { 748, 0x0000 }, { 748, 0x0000 }, { 748, 0x0000 },
2306 /* 0x3200 */
2307 { 748, 0xffff }, { 764, 0x1fff }, { 777, 0x0000 }, { 777, 0x0000 },
2308 { 777, 0x0000 }, { 777, 0x0000 }, { 777, 0xffff }, { 793, 0x8fff },
2309 { 806, 0x0000 }, { 806, 0x0000 }, { 806, 0x0000 }, { 806, 0x0000 },
2310 { 806, 0x0000 }, { 806, 0x0000 }, { 806, 0x0000 }, { 806, 0x0000 },
2311 /* 0x3300 */
2312 { 806, 0x0000 }, { 806, 0x0000 }, { 806, 0x0000 }, { 806, 0x0000 },
2313 { 806, 0x0000 }, { 806, 0x0000 }, { 806, 0x0000 }, { 806, 0x0000 },
2314 { 806, 0xff1f }, { 819, 0xffff }, { 835, 0xffff }, { 851, 0xffff },
2315 { 867, 0x87ff }, { 879, 0x3949 },
2316 };
2317 static const Summary16 ksc5601_uni2indx_page4e[1306] = {
2318 /* 0x4e00 */
2319 { 886, 0x2f8b }, { 895, 0x4372 }, { 902, 0x2000 }, { 903, 0x0b04 },
2320 { 907, 0xe82c }, { 914, 0xe340 }, { 920, 0x2800 }, { 922, 0x40c8 },
2321 { 926, 0x5944 }, { 932, 0x4937 }, { 940, 0x7976 }, { 950, 0x0440 },
2322 { 952, 0x2c93 }, { 959, 0xa3f0 }, { 967, 0x0038 }, { 970, 0x08c5 },
2323 /* 0x4f00 */
2324 { 975, 0xee02 }, { 982, 0x0003 }, { 984, 0x8000 }, { 985, 0x3550 },
2325 { 991, 0xe1c8 }, { 998, 0x1e23 }, { 1005, 0x8200 }, { 1007, 0xc449 },
2326 { 1013, 0xad5a }, { 1022, 0x2942 }, { 1027, 0xc000 }, { 1029, 0x8060 },
2327 { 1032, 0x461c }, { 1038, 0xa49a }, { 1045, 0xc003 }, { 1049, 0x052a },
2328 /* 0x5000 */
2329 { 1054, 0x2a44 }, { 1059, 0xd646 }, { 1067, 0x3dda }, { 1077, 0x0800 },
2330 { 1078, 0x8388 }, { 1083, 0x1420 }, { 1086, 0x0020 }, { 1087, 0x0170 },
2331 { 1091, 0x2021 }, { 1094, 0x0302 }, { 1097, 0x3000 }, { 1099, 0x40ac },
2332 { 1104, 0x8620 }, { 1108, 0x4462 }, { 1113, 0x20a0 }, { 1116, 0x8a00 },
2333 /* 0x5100 */
2334 { 1119, 0x0253 }, { 1124, 0x8004 }, { 1126, 0x0402 }, { 1128, 0x1484 },
2335 { 1132, 0x7bfb }, { 1145, 0x1004 }, { 1147, 0x7fa4 }, { 1157, 0x11e2 },
2336 { 1163, 0x2441 }, { 1167, 0x00a4 }, { 1170, 0x1421 }, { 1174, 0x20c0 },
2337 { 1177, 0x3a50 }, { 1183, 0x7000 }, { 1186, 0x0002 }, { 1187, 0x2743 },
2338 /* 0x5200 */
2339 { 1194, 0x45c9 }, { 1201, 0x2082 }, { 1204, 0x4630 }, { 1209, 0x0fc1 },
2340 { 1216, 0x3c88 }, { 1222, 0x2850 }, { 1226, 0x8602 }, { 1230, 0xa024 },
2341 { 1234, 0x2388 }, { 1239, 0x8806 }, { 1243, 0x0e19 }, { 1249, 0x4000 },
2342 { 1250, 0x22aa }, { 1256, 0xeb64 }, { 1265, 0x001c }, { 1268, 0xcd28 },
2343 /* 0x5300 */
2344 { 1275, 0xa120 }, { 1279, 0x02e1 }, { 1284, 0x840b }, { 1289, 0x8200 },
2345 { 1291, 0x279b }, { 1300, 0x549e }, { 1308, 0x8141 }, { 1312, 0xa0b3 },
2346 { 1319, 0x0010 }, { 1320, 0x8508 }, { 1324, 0x2061 }, { 1328, 0x0800 },
2347 { 1329, 0x2f08 }, { 1335, 0x08d0 }, { 1339, 0xbe3e }, { 1350, 0x010f },
2348 /* 0x5400 */
2349 { 1355, 0xf718 }, { 1364, 0xa803 }, { 1369, 0x0a41 }, { 1373, 0x5b08 },
2350 { 1379, 0x0504 }, { 1382, 0x0002 }, { 1383, 0x0500 }, { 1385, 0x382a },
2351 { 1391, 0x5041 }, { 1395, 0x0001 }, { 1396, 0x1910 }, { 1400, 0x2108 },
2352 { 1403, 0x0313 }, { 1408, 0x0000 }, { 1408, 0x6122 }, { 1413, 0x0404 },
2353 /* 0x5500 */
2354 { 1415, 0x40d0 }, { 1419, 0x1001 }, { 1421, 0x8000 }, { 1422, 0x4022 },
2355 { 1425, 0x8050 }, { 1428, 0x4048 }, { 1431, 0x0008 }, { 1432, 0x1000 },
2356 { 1433, 0x06d1 }, { 1439, 0x3700 }, { 1444, 0x5e80 }, { 1450, 0x0000 },
2357 { 1450, 0x00a0 }, { 1452, 0x9410 }, { 1456, 0x0018 }, { 1458, 0x6000 },
2358 /* 0x5600 */
2359 { 1460, 0x0240 }, { 1462, 0x0090 }, { 1464, 0x8000 }, { 1465, 0x0054 },
2360 { 1468, 0x0000 }, { 1468, 0x0008 }, { 1469, 0x0900 }, { 1471, 0x0010 },
2361 { 1472, 0x0040 }, { 1473, 0x0000 }, { 1473, 0x5020 }, { 1476, 0x1010 },
2362 { 1478, 0x2400 }, { 1480, 0x4c02 }, { 1484, 0x0001 }, { 1485, 0x0601 },
2363 /* 0x5700 */
2364 { 1488, 0x2918 }, { 1493, 0x814c }, { 1498, 0x2100 }, { 1500, 0x0801 },
2365 { 1502, 0x6485 }, { 1508, 0x0003 }, { 1510, 0x4452 }, { 1515, 0x1021 },
2366 { 1518, 0x0904 }, { 1521, 0x0008 }, { 1522, 0x000d }, { 1525, 0x0000 },
2367 { 1525, 0x4988 }, { 1530, 0x8000 }, { 1531, 0x0001 }, { 1532, 0x1691 },
2368 /* 0x5800 */
2369 { 1538, 0x0765 }, { 1545, 0x4000 }, { 1546, 0x8492 }, { 1551, 0x0433 },
2370 { 1556, 0x8c00 }, { 1559, 0x4592 }, { 1565, 0x0016 }, { 1568, 0x5220 },
2371 { 1572, 0x0228 }, { 1575, 0xd008 }, { 1579, 0x4300 }, { 1582, 0x4c08 },
2372 { 1586, 0x40a2 }, { 1590, 0xc32a }, { 1597, 0x9810 }, { 1601, 0x2e00 },
2373 /* 0x5900 */
2374 { 1605, 0x8000 }, { 1606, 0x1670 }, { 1612, 0x6e84 }, { 1619, 0x4082 },
2375 { 1622, 0xc390 }, { 1628, 0x04b3 }, { 1634, 0x7c85 }, { 1642, 0x2118 },
2376 { 1646, 0x041c }, { 1650, 0x02c8 }, { 1654, 0x1120 }, { 1657, 0x4a00 },
2377 { 1660, 0x0a48 }, { 1664, 0x361b }, { 1672, 0x5540 }, { 1677, 0x8900 },
2378 /* 0x5a00 */
2379 { 1680, 0x000a }, { 1682, 0x9902 }, { 1687, 0x0221 }, { 1690, 0x1040 },
2380 { 1692, 0x0242 }, { 1695, 0x0400 }, { 1696, 0x0044 }, { 1698, 0x0000 },
2381 { 1698, 0x0000 }, { 1698, 0x0c04 }, { 1701, 0x0010 }, { 1702, 0x0000 },
2382 { 1702, 0x1216 }, { 1707, 0x0000 }, { 1707, 0x0242 }, { 1710, 0x0000 },
2383 /* 0x5b00 */

```

```

2384 { 1710, 0x1a20 }, { 1714, 0x0040 }, { 1715, 0x0400 }, { 1716, 0x0000 },
2385 { 1716, 0x0009 }, { 1718, 0xb5b3 }, { 1728, 0x0a18 }, { 1732, 0x1523 },
2386 { 1738, 0x9ba0 }, { 1745, 0x1fe8 }, { 1754, 0x507c }, { 1761, 0x8379 },
2387 { 1769, 0x10fd }, { 1777, 0xc09d }, { 1784, 0xdbf6 }, { 1796, 0x0560 },
2388 /* 0x5c00 */
2389 { 1800, 0xef92 }, { 1810, 0x0242 }, { 1813, 0x0110 }, { 1815, 0xdf02 },
2390 { 1823, 0x6961 }, { 1830, 0x0822 }, { 1833, 0x9035 }, { 1839, 0x0202 },
2391 { 1841, 0x0000 }, { 1841, 0x0003 }, { 1843, 0x1a02 }, { 1847, 0x45aa },
2392 { 1854, 0x0001 }, { 1855, 0x0200 }, { 1856, 0x8101 }, { 1859, 0x2851 },
2393 /* 0x5d00 */
2394 { 1864, 0x6080 }, { 1867, 0x02d2 }, { 1872, 0x0280 }, { 1874, 0x0000 },
2395 { 1874, 0x1800 }, { 1876, 0x0001 }, { 1877, 0x9200 }, { 1880, 0x0000 },
2396 { 1880, 0x0880 }, { 1882, 0x2000 }, { 1883, 0x0405 }, { 1886, 0x3500 },
2397 { 1890, 0x2000 }, { 1891, 0x6044 }, { 1895, 0x49e6 }, { 1903, 0x609e },
2398 /* 0x5e00 */
2399 { 1910, 0x104c }, { 1914, 0x2a42 }, { 1919, 0x2820 }, { 1922, 0xa148 },
2400 { 1927, 0x10b1 }, { 1932, 0x8020 }, { 1934, 0x000e }, { 1937, 0x7b9c },
2401 { 1947, 0x8490 }, { 1951, 0x14a0 }, { 1955, 0x28c1 }, { 1960, 0x41e0 },
2402 { 1965, 0x0704 }, { 1969, 0x8c49 }, { 1975, 0x100d }, { 1979, 0x0cc8 },
2403 /* 0x5f00 */
2404 { 1984, 0x8412 }, { 1988, 0x89ba }, { 1996, 0x02c0 }, { 1999, 0x1422 },
2405 { 2003, 0x5500 }, { 2007, 0x0ac0 }, { 2011, 0x3ec4 }, { 2019, 0x9283 },
2406 { 2025, 0x1ca3 }, { 2032, 0x4387 }, { 2039, 0x4703 }, { 2045, 0x22a0 },
2407 { 2049, 0x3028 }, { 2053, 0x03c0 }, { 2057, 0x0801 }, { 2059, 0xa020 },
2408 /* 0x6000 */
2409 { 2062, 0x8000 }, { 2063, 0x3044 }, { 2067, 0x85a3 }, { 2074, 0x0000 },
2410 { 2074, 0x200e }, { 2078, 0x2225 }, { 2083, 0xb73c }, { 2093, 0x0001 },
2411 { 2094, 0x3220 }, { 2098, 0x8c50 }, { 2103, 0x0099 }, { 2107, 0x315d },
2412 { 2115, 0x00a0 }, { 2117, 0x9402 }, { 2121, 0x0003 }, { 2123, 0x0e4b },
2413 /* 0x6100 */
2414 { 2130, 0xe342 }, { 2137, 0x8c20 }, { 2141, 0x0080 }, { 2142, 0xd091 },
2415 { 2148, 0x1d94 }, { 2155, 0xa328 }, { 2161, 0x499c }, { 2168, 0x60c1 },
2416 { 2173, 0x4406 }, { 2177, 0x0713 }, { 2183, 0x5a90 }, { 2189, 0x4444 },
2417 { 2193, 0x0f88 }, { 2199, 0x0000 }, { 2199, 0x0040 }, { 2200, 0x95c4 },
2418 /* 0x6200 */
2419 { 2207, 0x7581 }, { 2214, 0x8447 }, { 2220, 0x4402 }, { 2223, 0xc053 },
2420 { 2229, 0x2b83 }, { 2236, 0x0108 }, { 2238, 0x4000 }, { 2239, 0x9242 },
2421 { 2244, 0x0611 }, { 2248, 0x09a6 }, { 2254, 0x0800 }, { 2255, 0x3222 },
2422 { 2260, 0xb384 }, { 2267, 0x1bdc }, { 2277, 0xf000 }, { 2281, 0xc08a },
2423 /* 0x6300 */
2424 { 2286, 0x0282 }, { 2289, 0x0002 }, { 2290, 0x8800 }, { 2292, 0x6c00 },
2425 { 2296, 0x9200 }, { 2299, 0x0021 }, { 2301, 0x4180 }, { 2304, 0x8c84 },
2426 { 2309, 0x1308 }, { 2313, 0x0944 }, { 2317, 0x07a7 }, { 2325, 0x0000 },
2427 { 2325, 0x8051 }, { 2329, 0x0c41 }, { 2333, 0x6002 }, { 2336, 0x00d0 },
2428 /* 0x6400 */
2429 { 2339, 0xa000 }, { 2341, 0x10d0 }, { 2345, 0x3004 }, { 2348, 0x4400 },
2430 { 2350, 0x0000 }, { 2350, 0x0100 }, { 2351, 0x8201 }, { 2354, 0x0700 },
2431 { 2357, 0x0100 }, { 2358, 0x440e }, { 2363, 0x6830 }, { 2368, 0x0805 },
2432 { 2371, 0x64b2 }, { 2378, 0x0514 }, { 2382, 0x10e6 }, { 2388, 0x4414 },
2433 /* 0x6500 */
2434 { 2392, 0x0011 }, { 2394, 0x2100 }, { 2396, 0x9c08 }, { 2401, 0xc0c0 },
2435 { 2408, 0xe120 }, { 2413, 0x40c2 }, { 2417, 0x304c }, { 2422, 0x41b4 },
2436 { 2428, 0x10ac }, { 2433, 0x9a83 }, { 2440, 0x98b2 }, { 2447, 0x3281 },
2437 { 2452, 0x9822 }, { 2457, 0x0084 }, { 2459, 0x3369 }, { 2467, 0xb0c12 },
2438 /* 0x6600 */
2439 { 2474, 0xd6c0 }, { 2481, 0xc03b }, { 2488, 0xa1a1 }, { 2494, 0x0c53 },
2440 { 2500, 0x8a1e }, { 2507, 0xea00 }, { 2512, 0xc0f0 }, { 2521, 0x05d8 },
2441 { 2527, 0x4390 }, { 2532, 0x21c3 }, { 2538, 0x4805 }, { 2542, 0x4a1c },
2442 { 2548, 0x02d0 }, { 2552, 0x3240 }, { 2556, 0x0041 }, { 2558, 0xd79d },
2443 /* 0x6700 */
2444 { 2569, 0x2b09 }, { 2575, 0xe8b0 }, { 2582, 0x7dc0 }, { 2590, 0x2452 },
2445 { 2595, 0xc240 }, { 2599, 0xd04b }, { 2606, 0xa000 }, { 2608, 0xc8ab },
2446 { 2616, 0x8a80 }, { 2620, 0x34a9 }, { 2627, 0x8000 }, { 2628, 0x41c9 },
2447 { 2634, 0x8010 }, { 2636, 0x241f }, { 2643, 0x9200 }, { 2646, 0x487b },
2448 /* 0x6800 */
2449 { 2654, 0x0000 }, { 2654, 0x00cc }, { 2658, 0x8406 }, { 2662, 0x3300 },
2450 { 2666, 0x410f }, { 2672, 0x001b }, { 2676, 0x2000 }, { 2677, 0x8040 },
2451 { 2679, 0x8022 }, { 2682, 0xa098 }, { 2687, 0xa186 }, { 2693, 0x006b },
2452 { 2698, 0x2a30 }, { 2703, 0x85a4 }, { 2709, 0x4181 }, { 2713, 0x0604 },
2453 /* 0x6900 */
2454 { 2716, 0x6021 }, { 2720, 0x0004 }, { 2721, 0x0080 }, { 2722, 0xa001 },
2455 { 2725, 0x0400 }, { 2726, 0x46b8 }, { 2733, 0xe90f }, { 2742, 0x03a0 },
2456 { 2746, 0x0000 }, { 2746, 0x1820 }, { 2749, 0x40a0 }, { 2752, 0x0810 },
2457 { 2754, 0x380a }, { 2759, 0x0001 }, { 2760, 0x0500 }, { 2762, 0xa800 },
2458 /* 0x6a00 */
2459 { 2765, 0x0404 }, { 2767, 0xc28a }, { 2773, 0x000a }, { 2775, 0x2720 },
2460 { 2780, 0x0910 }, { 2783, 0x830c }, { 2788, 0x0802 }, { 2790, 0x0000 },
2461 { 2790, 0x6211 }, { 2795, 0x1080 }, { 2797, 0x000c }, { 2799, 0x0808 },
2462 { 2801, 0x000c }, { 2803, 0x0c08 }, { 2806, 0x0000 }, { 2806, 0x0840 },
2463 /* 0x6b00 */
2464 { 2808, 0x1410 }, { 2811, 0x0044 }, { 2813, 0x000b }, { 2816, 0x6404 },
2465 { 2820, 0x50c0 }, { 2824, 0x8001 }, { 2826, 0x047e }, { 2833, 0x8984 },
2466 { 2838, 0x0658 }, { 2843, 0x4140 }, { 2846, 0xc000 }, { 2848, 0x94a4 },
2467 { 2854, 0xa862 }, { 2860, 0x09dc }, { 2867, 0x1800 }, { 2869, 0x0000 },
2468 /* 0x6c00 */
2469 { 2869, 0x8100 }, { 2871, 0x000a }, { 2873, 0x0008 }, { 2874, 0x4190 },
2470 { 2878, 0x4007 }, { 2882, 0xe4a1 }, { 2889, 0x2501 }, { 2893, 0x6445 },

```



```

2471 { 2899, 0x11ee }, { 2907, 0x0e7d }, { 2916, 0x4800 }, { 2918, 0xfb08 },
2472 { 2926, 0x1616 }, { 2932, 0x08a8 }, { 2936, 0xc92e }, { 2944, 0x0009 },
2473 /* 0x6d00 */
2474 { 2946, 0x1800 }, { 2948, 0x4a82 }, { 2953, 0x06a0 }, { 2957, 0x6b64 },
2475 { 2965, 0x0002 }, { 2966, 0x1600 }, { 2969, 0x5648 }, { 2975, 0x8390 },
2476 { 2980, 0x73a0 }, { 2987, 0x002a }, { 2990, 0x8000 }, { 2991, 0x0024 },
2477 { 2993, 0x88f9 }, { 3001, 0x4702 }, { 3006, 0x4d02 }, { 3011, 0x0faa },
2478 /* 0x6e00 */
2479 { 3019, 0x0000 }, { 3019, 0x8e80 }, { 3024, 0xb87b }, { 3034, 0x7554 },
2480 { 3042, 0x2418 }, { 3046, 0xd940 }, { 3052, 0xc880 }, { 3056, 0x040c },
2481 { 3059, 0x0000 }, { 3059, 0xb041 }, { 3064, 0x8c24 }, { 3069, 0x0442 },
2482 { 3072, 0x5a34 }, { 3079, 0x001a }, { 3082, 0x8000 }, { 3083, 0xc110 },
2483 /* 0x6f00 */
2484 { 3087, 0x8046 }, { 3091, 0x0032 }, { 3094, 0x180d }, { 3099, 0x8106 },
2485 { 3103, 0x0002 }, { 3104, 0xcd92 }, { 3112, 0x6014 }, { 3116, 0x7401 },
2486 { 3121, 0x6112 }, { 3126, 0x0091 }, { 3129, 0xc098 }, { 3134, 0x420a },
2487 { 3138, 0x040f }, { 3143, 0x8420 }, { 3146, 0x9a13 }, { 3153, 0x4002 },
2488 /* 0x7000 */
2489 { 3155, 0x8a62 }, { 3161, 0xfd22 }, { 3170, 0x8188 }, { 3174, 0x4080 },
2490 { 3176, 0x1000 }, { 3177, 0x2103 }, { 3181, 0x0808 }, { 3183, 0x3101 },
2491 { 3187, 0x4420 }, { 3190, 0x0704 }, { 3194, 0xb812 }, { 3200, 0x0388 },
2492 { 3204, 0x8900 }, { 3207, 0xa300 }, { 3211, 0x0000 }, { 3211, 0x2202 },
2493 /* 0x7100 */
2494 { 3214, 0x1210 }, { 3217, 0x4600 }, { 3220, 0x0042 }, { 3222, 0x0041 },
2495 { 3224, 0x5680 }, { 3229, 0x5241 }, { 3234, 0x52f0 }, { 3241, 0x2000 },
2496 { 3242, 0x8610 }, { 3246, 0x8214 }, { 3250, 0x1004 }, { 3252, 0x4602 },
2497 { 3256, 0x430a }, { 3261, 0x8035 }, { 3266, 0x60e0 }, { 3271, 0xd800 },
2498 /* 0x7200 */
2499 { 3275, 0x0041 }, { 3277, 0x0801 }, { 3279, 0x3400 }, { 3282, 0x6c65 },
2500 { 3290, 0x11c1 }, { 3295, 0xab04 }, { 3301, 0x0286 }, { 3305, 0x2204 },
2501 { 3308, 0x0003 }, { 3310, 0x0000 }, { 3310, 0x9084 }, { 3314, 0x0000 },
2502 { 3314, 0x4015 }, { 3318, 0x0281 }, { 3321, 0x0202 }, { 3323, 0x3300 },
2503 /* 0x7300 */
2504 { 3327, 0x0400 }, { 3328, 0x3840 }, { 3332, 0x0e20 }, { 3336, 0xc0c0 },
2505 { 3340, 0x0030 }, { 3342, 0x0085 }, { 3345, 0x0500 }, { 3347, 0x0d25 },
2506 { 3353, 0x4ad0 }, { 3359, 0x81d0 }, { 3364, 0x2280 }, { 3367, 0x020c },
2507 { 3370, 0xb605 }, { 3377, 0x6240 }, { 3381, 0x2679 }, { 3389, 0x6280 },
2508 /* 0x7400 */
2509 { 3393, 0x02ea }, { 3399, 0x0808 }, { 3401, 0xdd67 }, { 3412, 0x8579 },
2510 { 3420, 0x081b }, { 3425, 0xdea0 }, { 3433, 0x8735 }, { 3441, 0x4000 },
2511 { 3442, 0x0a8c }, { 3447, 0xd100 }, { 3451, 0x05aa }, { 3457, 0xa225 },
2512 { 3463, 0x8440 }, { 3466, 0x1510 }, { 3470, 0x404d }, { 3475, 0x0080 },
2513 /* 0x7500 */
2514 { 3476, 0x0012 }, { 3478, 0x8d22 }, { 3484, 0x1968 }, { 3490, 0x058f },
2515 { 3497, 0x9080 }, { 3500, 0x3a1a }, { 3507, 0x8464 }, { 3512, 0x8561 },
2516 { 3518, 0xcc00 }, { 3524, 0x2002 }, { 3526, 0x0820 }, { 3528, 0x732e },
2517 { 3537, 0x20a4 }, { 3541, 0x0b34 }, { 3547, 0x0004 }, { 3548, 0x1415 },
2518 /* 0x7600 */
2519 { 3553, 0x2001 }, { 3555, 0x8200 }, { 3557, 0x0057 }, { 3562, 0x0800 },
2520 { 3563, 0x5004 }, { 3566, 0x0044 }, { 3568, 0x1212 }, { 3572, 0x7905 },
2521 { 3579, 0x40d0 }, { 3583, 0x0009 }, { 3585, 0x4000 }, { 3586, 0x8400 },
2522 { 3588, 0x054c }, { 3593, 0xd844 }, { 3599, 0x409a }, { 3604, 0x5114 },
2523 /* 0x7700 */
2524 { 3609, 0x0b12 }, { 3614, 0x4000 }, { 3615, 0x0201 }, { 3617, 0x1580 },
2525 { 3621, 0x2001 }, { 3623, 0x0800 }, { 3624, 0x084a }, { 3628, 0xc200 },
2526 { 3631, 0x0800 }, { 3632, 0x4002 }, { 3634, 0x3020 }, { 3637, 0x9809 },
2527 { 3642, 0x0000 }, { 3642, 0x1880 }, { 3645, 0xe22c }, { 3652, 0x0008 },
2528 /* 0x7800 */
2529 { 3653, 0x0004 }, { 3654, 0x0004 }, { 3655, 0x10e0 }, { 3659, 0x0014 },
2530 { 3661, 0x8020 }, { 3663, 0x2000 }, { 3664, 0x9800 }, { 3667, 0x1000 },
2531 { 3668, 0x7082 }, { 3673, 0x0082 }, { 3675, 0x0288 }, { 3678, 0x1c00 },
2532 { 3681, 0x4c22 }, { 3686, 0x0001 }, { 3687, 0x9100 }, { 3690, 0x0820 },
2533 /* 0x7900 */
2534 { 3692, 0x4002 }, { 3694, 0x0040 }, { 3695, 0x1c00 }, { 3698, 0x4400 },
2535 { 3700, 0x0383 }, { 3705, 0x7cc1 }, { 3713, 0x2121 }, { 3717, 0x8400 },
2536 { 3719, 0xe002 }, { 3723, 0x0002 }, { 3724, 0x44c0 }, { 3728, 0xe20a },
2537 { 3734, 0x0e03 }, { 3739, 0x8126 }, { 3744, 0x02d0 }, { 3748, 0x0800 },
2538 /* 0x7a00 */
2539 { 3749, 0x2921 }, { 3754, 0x9690 }, { 3760, 0x4001 }, { 3762, 0xb8c2 },
2540 { 3769, 0x6241 }, { 3774, 0x0080 }, { 3775, 0x0a06 }, { 3779, 0xa651 },
2541 { 3786, 0x0112 }, { 3789, 0x812c }, { 3794, 0xc600 }, { 3798, 0x0400 },
2542 { 3799, 0x0cb0 }, { 3804, 0xa280 }, { 3808, 0xa429 }, { 3814, 0x8640 },
2543 /* 0x7b00 */
2544 { 3818, 0x8000 }, { 3819, 0x4a02 }, { 3823, 0x3041 }, { 3827, 0x0200 },
2545 { 3828, 0xba40 }, { 3834, 0x0057 }, { 3839, 0x5001 }, { 3842, 0x2020 },
2546 { 3844, 0x8880 }, { 3847, 0x24b0 }, { 3852, 0x2002 }, { 3854, 0x0112 },
2547 { 3857, 0x02d3 }, { 3863, 0x0004 }, { 3864, 0x0211 }, { 3867, 0x0000 },
2548 /* 0x7c00 */
2549 { 3867, 0x0080 }, { 3868, 0x4004 }, { 3870, 0x0c82 }, { 3874, 0xe000 },
2550 { 3877, 0x3008 }, { 3880, 0x0000 }, { 3880, 0x1011 }, { 3883, 0x0008 },
2551 { 3884, 0x0208 }, { 3886, 0x81a4 }, { 3891, 0x40a0 }, { 3894, 0x420e },
2552 { 3899, 0x0400 }, { 3900, 0xc040 }, { 3903, 0x0081 }, { 3905, 0x4800 },
2553 /* 0x7d00 */
2554 { 3907, 0x2df5 }, { 3917, 0x0f91 }, { 3924, 0xd807 }, { 3931, 0x0629 },
2555 { 3936, 0x007c }, { 3941, 0x4001 }, { 3943, 0x4546 }, { 3949, 0x824e },
2556 { 3955, 0xc000 }, { 3957, 0x1008 }, { 3959, 0x3005 }, { 3963, 0xed36 },
2557 { 3973, 0x0c80 }, { 3976, 0x6540 }, { 3981, 0x930b }, { 3988, 0x0810 },

```

```

2558 /* 0x7e00 */
2559 { 3990, 0x0600 }, { 3992, 0xe820 }, { 3997, 0xc80a }, { 4002, 0x6082 },
2560 { 4006, 0x00ca }, { 4010, 0x4034 }, { 4014, 0x2e02 }, { 4019, 0x1201 },
2561 { 4022, 0x9004 }, { 4025, 0x1948 }, { 4030, 0x0000 }, { 4030, 0x0000 },
2562 { 4030, 0x0000 }, { 4030, 0x0000 }, { 4030, 0x0000 }, { 4030, 0x0000 },
2563 /* 0x7f00 */
2564 { 4030, 0x0000 }, { 4030, 0x0000 }, { 4030, 0x0000 }, { 4030, 0x0540 },
2565 { 4033, 0x1000 }, { 4034, 0x0031 }, { 4037, 0x4c00 }, { 4040, 0x02a5 },
2566 { 4045, 0x5520 }, { 4050, 0x4410 }, { 4053, 0x0310 }, { 4056, 0x2304 },
2567 { 4060, 0x5422 }, { 4065, 0x8034 }, { 4069, 0x0a03 }, { 4073, 0x1201 },
2568 /* 0x8000 */
2569 { 4076, 0x126b }, { 4083, 0x01a1 }, { 4087, 0x2000 }, { 4088, 0xa048 },
2570 { 4092, 0x0448 }, { 4095, 0x4540 }, { 4099, 0x8000 }, { 4100, 0xe08d },
2571 { 4107, 0x1af0 }, { 4114, 0x2840 }, { 4117, 0x8626 }, { 4123, 0x0416 },
2572 { 4127, 0x5018 }, { 4131, 0x4c00 }, { 4134, 0x0032 }, { 4137, 0x2112 },
2573 /* 0x8100 */
2574 { 4141, 0x05e4 }, { 4147, 0x0d00 }, { 4150, 0x8a08 }, { 4154, 0x4200 },
2575 { 4156, 0x4800 }, { 4158, 0x0033 }, { 4162, 0x0860 }, { 4165, 0x8703 },
2576 { 4171, 0x8501 }, { 4175, 0x3400 }, { 4178, 0x0109 }, { 4181, 0xe428 },
2577 { 4187, 0x2045 }, { 4191, 0x8100 }, { 4193, 0x25a8 }, { 4199, 0x5c18 },
2578 /* 0x8200 */
2579 { 4205, 0x35a0 }, { 4211, 0xd804 }, { 4216, 0x1c02 }, { 4220, 0x02e0 },
2580 { 4224, 0x00a1 }, { 4227, 0x0200 }, { 4228, 0xc050 }, { 4232, 0x4146 },
2581 { 4237, 0x6800 }, { 4240, 0xa604 }, { 4245, 0xf260 }, { 4252, 0xbb8a },
2582 { 4261, 0x0000 }, { 4261, 0xc8b6 }, { 4269, 0x00e2 }, { 4273, 0x6002 },
2583 /* 0x8300 */
2584 { 4276, 0x023e }, { 4282, 0x0080 }, { 4283, 0x8900 }, { 4286, 0x0372 },
2585 { 4292, 0x8681 }, { 4297, 0x0006 }, { 4299, 0x0000 }, { 4299, 0x0888 },
2586 { 4302, 0x4600 }, { 4305, 0x4140 }, { 4308, 0x0e04 }, { 4312, 0x2000 },
2587 { 4313, 0x1622 }, { 4318, 0x1048 }, { 4321, 0x8a00 }, { 4324, 0x2217 },
2588 /* 0x8400 */
2589 { 4330, 0x7418 }, { 4336, 0x0000 }, { 4336, 0x1200 }, { 4338, 0x2102 },
2590 { 4341, 0x0200 }, { 4342, 0x0880 }, { 4344, 0x984a }, { 4350, 0x0420 },
2591 { 4352, 0x0000 }, { 4352, 0x1211 }, { 4356, 0x0002 }, { 4357, 0x9904 },
2592 { 4362, 0x2a55 }, { 4369, 0x0402 }, { 4371, 0x5000 }, { 4373, 0x1010 },
2593 /* 0x8500 */
2594 { 4375, 0x0000 }, { 4375, 0x459a }, { 4382, 0xb02a }, { 4388, 0xa000 },
2595 { 4390, 0x420a }, { 4394, 0x0208 }, { 4396, 0x2708 }, { 4401, 0x0000 },
2596 { 4401, 0x8090 }, { 4404, 0x0812 }, { 4407, 0x8740 }, { 4412, 0x0401 },
2597 { 4414, 0xe202 }, { 4419, 0x3020 }, { 4422, 0x0630 }, { 4426, 0x8c80 },
2598 /* 0x8600 */
2599 { 4430, 0x04c4 }, { 4434, 0x04c0 }, { 4437, 0x2000 }, { 4438, 0x8000 },
2600 { 4439, 0x4000 }, { 4440, 0xd831 }, { 4447, 0x0080 }, { 4448, 0x0200 },
2601 { 4449, 0x1400 }, { 4451, 0x0008 }, { 4452, 0x0218 }, { 4455, 0x0000 },
2602 { 4455, 0x0880 }, { 4457, 0x8a10 }, { 4461, 0x2010 }, { 4463, 0x4000 },
2603 /* 0x8700 */
2604 { 4464, 0x010d }, { 4468, 0x1500 }, { 4471, 0x0000 }, { 4471, 0x0000 },
2605 { 4471, 0x4000 }, { 4472, 0x80a0 }, { 4475, 0x0140 }, { 4477, 0x0150 },
2606 { 4480, 0x2004 }, { 4482, 0x8000 }, { 4483, 0x0004 }, { 4484, 0x0408 },
2607 { 4486, 0x0010 }, { 4487, 0x0000 }, { 4487, 0x9001 }, { 4490, 0x4a04 },
2608 /* 0x8800 */
2609 { 4494, 0x0020 }, { 4495, 0x8000 }, { 4496, 0x000c }, { 4498, 0x0842 },
2610 { 4501, 0x3041 }, { 4505, 0x2a8c }, { 4511, 0x090e }, { 4516, 0xc085 },
2611 { 4521, 0x2906 }, { 4526, 0x40c4 }, { 4530, 0x0800 }, { 4531, 0x0010 },
2612 { 4532, 0x8006 }, { 4535, 0xb230 }, { 4541, 0x0102 }, { 4543, 0x2138 },
2613 /* 0x8900 */
2614 { 4548, 0x0080 }, { 4549, 0x030d }, { 4554, 0x0420 }, { 4556, 0x0940 },
2615 { 4559, 0x0012 }, { 4561, 0x8000 }, { 4562, 0x0410 }, { 4564, 0x8004 },
2616 { 4566, 0x88ca }, { 4572, 0x0048 }, { 4574, 0x0602 }, { 4577, 0x2404 },
2617 { 4580, 0x0001 }, { 4581, 0x0004 }, { 4582, 0x0008 }, { 4583, 0x0110 },
2618 /* 0x8a00 */
2619 { 4585, 0x550d }, { 4592, 0xa9c8 }, { 4599, 0x2428 }, { 4603, 0x0c52 },
2620 { 4608, 0x0000 }, { 4608, 0x4831 }, { 4613, 0x624d }, { 4620, 0x022f },
2621 { 4626, 0x30a0 }, { 4630, 0x4128 }, { 4634, 0x057b }, { 4642, 0xd205 },
2622 { 4648, 0xa894 }, { 4654, 0x1844 }, { 4658, 0x6cc2 }, { 4665, 0x45c2 },
2623 /* 0x8b00 */
2624 { 4671, 0x4017 }, { 4676, 0x2ed1 }, { 4684, 0x1901 }, { 4688, 0x0208 },
2625 { 4690, 0xc202 }, { 4694, 0x1500 }, { 4697, 0x9040 }, { 4700, 0x2091 },
2626 { 4704, 0x0401 }, { 4706, 0x044d }, { 4711, 0x0000 }, { 4711, 0x0000 },
2627 { 4711, 0x0000 }, { 4711, 0x0000 }, { 4711, 0x0000 }, { 4711, 0x0000 },
2628 /* 0x8c00 */
2629 { 4711, 0x0000 }, { 4711, 0x0000 }, { 4711, 0x0000 }, { 4711, 0x8080 },
2630 { 4713, 0x1542 }, { 4718, 0x0420 }, { 4720, 0x0c02 }, { 4723, 0x0600 },
2631 { 4725, 0x1404 }, { 4728, 0x6000 }, { 4730, 0x9f87 }, { 4740, 0xb9d9 },
2632 { 4750, 0x059f }, { 4758, 0x540a }, { 4763, 0x245d }, { 4770, 0x3810 },
2633 /* 0x8d00 */
2634 { 4774, 0x25b0 }, { 4780, 0x0048 }, { 4782, 0x0000 }, { 4782, 0x0000 },
2635 { 4782, 0x0000 }, { 4782, 0x0000 }, { 4782, 0x0850 }, { 4785, 0x0099 },
2636 { 4789, 0x0420 }, { 4791, 0x0200 }, { 4792, 0x0108 }, { 4794, 0x4408 },
2637 { 4797, 0x9840 }, { 4801, 0x2800 }, { 4803, 0x810a }, { 4807, 0x0008 },
2638 /* 0x8e00 */
2639 { 4808, 0x8400 }, { 4810, 0x4001 }, { 4812, 0x0400 }, { 4813, 0x0021 },
2640 { 4815, 0x0794 }, { 4821, 0x8200 }, { 4823, 0x0001 }, { 4824, 0x0050 },
2641 { 4826, 0x2482 }, { 4830, 0x0000 }, { 4830, 0x1c00 }, { 4833, 0x0000 },
2642 { 4833, 0x3c01 }, { 4838, 0x8004 }, { 4840, 0x0800 }, { 4841, 0x4900 },
2643 /* 0x8f00 */
2644 { 4844, 0x0228 }, { 4847, 0xf83c }, { 4856, 0x86c0 }, { 4861, 0xcb08 },

```

```

2645 { 4867, 0x6230 }, { 4872, 0xa000 }, { 4874, 0x0004 }, { 4875, 0x0000 },
2646 { 4875, 0x0000 }, { 4875, 0x1800 }, { 4877, 0xa148 }, { 4882, 0x0007 },
2647 { 4885, 0x4024 }, { 4888, 0x0012 }, { 4890, 0x2c40 }, { 4894, 0x2285 },
2648 /* 0x9000 */
2649 { 4899, 0xa96f }, { 4909, 0xe6b3 }, { 4919, 0x400f }, { 4924, 0x5126 },
2650 { 4930, 0x6c86 }, { 4937, 0x723b }, { 4946, 0xe20b }, { 4953, 0xb5a4 },
2651 { 4961, 0x859f }, { 4970, 0x0222 }, { 4973, 0x854c }, { 4979, 0x0123 },
2652 { 4983, 0x0402 }, { 4985, 0x4000 }, { 4986, 0x2102 }, { 4989, 0x2020 },
2653 /* 0x9100 */
2654 { 4991, 0x0004 }, { 4992, 0x0224 }, { 4995, 0x2080 }, { 4997, 0x0004 },
2655 { 4998, 0x7e00 }, { 5004, 0x0004 }, { 5005, 0x1604 }, { 5009, 0x01a0 },
2656 { 5012, 0x2a80 }, { 5016, 0x1004 }, { 5018, 0xd800 }, { 5022, 0x0032 },
2657 { 5025, 0xfa81 }, { 5033, 0x3183 }, { 5039, 0x0488 }, { 5042, 0x0020 },
2658 /* 0x9200 */
2659 { 5043, 0x2000 }, { 5044, 0x4087 }, { 5049, 0x0000 }, { 5049, 0x8410 },
2660 { 5052, 0x0221 }, { 5055, 0x4880 }, { 5058, 0x0074 }, { 5062, 0x0000 },
2661 { 5062, 0x0029 }, { 5065, 0x114a }, { 5070, 0x0000 }, { 5070, 0x02c8 },
2662 { 5074, 0x9000 }, { 5076, 0x0004 }, { 5077, 0x0410 }, { 5079, 0x1100 },
2663 /* 0x9300 */
2664 { 5081, 0x0010 }, { 5082, 0xc501 }, { 5087, 0xc957 }, { 5096, 0x0000 },
2665 { 5096, 0x2d00 }, { 5100, 0x0810 }, { 5102, 0x4000 }, { 5103, 0x5020 },
2666 { 5106, 0x1000 }, { 5107, 0x0450 }, { 5110, 0x3088 }, { 5114, 0x0001 },
2667 { 5115, 0x0008 }, { 5116, 0x4002 }, { 5118, 0x0012 }, { 5120, 0x0040 },
2668 /* 0x9400 */
2669 { 5121, 0x0010 }, { 5122, 0x0100 }, { 5123, 0x0820 }, { 5125, 0x0120 },
2670 { 5127, 0x0010 }, { 5128, 0x0806 }, { 5131, 0x0000 }, { 5131, 0xa000 },
2671 { 5133, 0x0000 }, { 5133, 0x0000 }, { 5133, 0x0000 }, { 5133, 0x0000 },
2672 { 5133, 0x0000 }, { 5133, 0x0000 }, { 5133, 0x0000 }, { 5133, 0x0000 },
2673 /* 0x9500 */
2674 { 5133, 0x0000 }, { 5133, 0x0000 }, { 5133, 0x0000 }, { 5133, 0x0000 },
2675 { 5133, 0x0000 }, { 5133, 0x0000 }, { 5133, 0x0000 }, { 5133, 0x0080 },
2676 { 5134, 0x8a09 }, { 5139, 0x011e }, { 5144, 0x2138 }, { 5149, 0x1802 },
2677 { 5152, 0x0480 }, { 5154, 0x1070 }, { 5158, 0x0006 }, { 5160, 0x0000 },
2678 /* 0x9600 */
2679 { 5160, 0x0000 }, { 5160, 0x1000 }, { 5161, 0x4402 }, { 5164, 0x8804 },
2680 { 5167, 0x3815 }, { 5173, 0xf801 }, { 5179, 0x041c }, { 5183, 0x21e9 },
2681 { 5190, 0x6c60 }, { 5196, 0x1b30 }, { 5202, 0x0588 }, { 5206, 0x0882 },
2682 { 5209, 0x7af3 }, { 5220, 0x1a60 }, { 5225, 0x870c }, { 5231, 0x0ac5 },
2683 /* 0x9700 */
2684 { 5237, 0x00c1 }, { 5240, 0x524a }, { 5246, 0x0080 }, { 5247, 0x2205 },
2685 { 5251, 0x0114 }, { 5254, 0x5042 }, { 5258, 0x2206 }, { 5262, 0x0490 },
2686 { 5265, 0xa800 }, { 5268, 0x0000 }, { 5268, 0x2901 }, { 5272, 0x0000 },
2687 { 5272, 0x0840 }, { 5274, 0x1008 }, { 5276, 0x0000 }, { 5276, 0x8848 },
2688 /* 0x9800 */
2689 { 5280, 0x156f }, { 5289, 0x018f }, { 5295, 0x2000 }, { 5296, 0xb001 },
2690 { 5300, 0x7040 }, { 5304, 0x4510 }, { 5308, 0x88a0 }, { 5312, 0x0000 },
2691 { 5312, 0x0000 }, { 5312, 0x0000 }, { 5312, 0x8100 }, { 5314, 0x0002 },
2692 { 5315, 0x0090 }, { 5317, 0x9800 }, { 5320, 0xe006 }, { 5325, 0x7010 },
2693 /* 0x9900 */
2694 { 5329, 0x1608 }, { 5333, 0x4109 }, { 5337, 0x0101 }, { 5339, 0x0000 },
2695 { 5339, 0x3a20 }, { 5344, 0x0096 }, { 5348, 0x0000 }, { 5348, 0x0000 },
2696 { 5348, 0x0000 }, { 5348, 0x2240 }, { 5351, 0x7120 }, { 5356, 0x021a },
2697 { 5360, 0x0002 }, { 5361, 0xa227 }, { 5368, 0x2000 }, { 5369, 0x8002 },
2698 /* 0x9a00 */
2699 { 5371, 0xc102 }, { 5375, 0x0200 }, { 5376, 0x0800 }, { 5377, 0x00c1 },
2700 { 5380, 0x2029 }, { 5384, 0x8ca0 }, { 5389, 0x0624 }, { 5393, 0x0000 },
2701 { 5393, 0x0000 }, { 5393, 0x0000 }, { 5393, 0x0100 }, { 5394, 0x0100 },
2702 { 5395, 0x0000 }, { 5395, 0x0118 }, { 5398, 0x4020 }, { 5400, 0x0000 },
2703 /* 0x9b00 */
2704 { 5400, 0x0000 }, { 5400, 0x0400 }, { 5401, 0x0480 }, { 5403, 0x1002 },
2705 { 5405, 0x803e }, { 5411, 0x0410 }, { 5413, 0x8000 }, { 5414, 0x0000 },
2706 { 5414, 0x4000 }, { 5415, 0x8002 }, { 5417, 0x4800 }, { 5419, 0x0000 },
2707 { 5419, 0x0200 }, { 5420, 0x0040 }, { 5421, 0x0110 }, { 5423, 0x0000 },
2708 /* 0x9c00 */
2709 { 5423, 0x2000 }, { 5424, 0x0025 }, { 5427, 0x0020 }, { 5428, 0x0804 },
2710 { 5430, 0x0280 }, { 5432, 0x0080 }, { 5433, 0x0000 }, { 5433, 0x0000 },
2711 { 5433, 0x0000 }, { 5433, 0x0000 }, { 5433, 0x0000 }, { 5433, 0x0000 },
2712 { 5433, 0x0000 }, { 5433, 0x0000 }, { 5433, 0x02a0 }, { 5436, 0x0058 },
2713 /* 0x9d00 */
2714 { 5439, 0x0200 }, { 5440, 0x0800 }, { 5441, 0x0140 }, { 5443, 0x0800 },
2715 { 5444, 0x0000 }, { 5444, 0x2002 }, { 5446, 0x1003 }, { 5449, 0x0004 },
2716 { 5450, 0x0000 }, { 5450, 0x0000 }, { 5450, 0x8200 }, { 5452, 0x0010 },
2717 { 5453, 0x0010 }, { 5454, 0x0080 }, { 5455, 0x0000 }, { 5455, 0x0704 },
2718 /* 0x9e00 */
2719 { 5459, 0x0000 }, { 5459, 0x4400 }, { 5461, 0x0000 }, { 5461, 0x0000 },
2720 { 5461, 0x0000 }, { 5461, 0x0000 }, { 5461, 0x0000 }, { 5461, 0xa220 },
2721 { 5465, 0x0000 }, { 5465, 0xa08c }, { 5470, 0x0020 }, { 5471, 0x4830 },
2722 { 5475, 0x6008 }, { 5478, 0x5912 }, { 5484, 0x0100 }, { 5485, 0x0010 },
2723 /* 0x9f00 */
2724 { 5486, 0x4180 }, { 5489, 0x0008 }, { 5490, 0x0001 }, { 5491, 0x0800 },
2725 { 5492, 0x4c00 }, { 5495, 0x8004 }, { 5497, 0x1482 }, { 5501, 0x0080 },
2726 { 5502, 0x2000 }, { 5503, 0x1021 },
2727 };
2728 static const Summary16 ksc5601_uni2indx_pageac[698] = {
2729 /* 0xac00 */
2730 { 5506, 0x0793 }, { 5513, 0x3eff }, { 5526, 0xb011 }, { 5531, 0x1303 },
2731 { 5536, 0x2801 }, { 5539, 0x1110 }, { 5542, 0x0000 }, { 5542, 0x0593 },

```

```

2732 { 5548, 0x1e7b }, { 5558, 0xb011 }, { 5563, 0x9703 }, { 5570, 0x3b01 },
2733 { 5576, 0x1112 }, { 5580, 0x00a0 }, { 5582, 0x9593 }, { 5590, 0x306b },
2734 /* 0xad00 */
2735 { 5597, 0xb051 }, { 5603, 0x1102 }, { 5606, 0x3201 }, { 5610, 0x1130 },
2736 { 5614, 0x02b0 }, { 5618, 0x0111 }, { 5621, 0x300a }, { 5625, 0xb879 },
2737 { 5634, 0x1306 }, { 5639, 0x3001 }, { 5642, 0x0010 }, { 5643, 0x0080 },
2738 { 5644, 0x0113 }, { 5648, 0x100b }, { 5652, 0x0011 }, { 5654, 0x9300 },
2739 /* 0xae00 */
2740 { 5658, 0x2b03 }, { 5664, 0x0010 }, { 5665, 0x0000 }, { 5665, 0x0593 },
2741 { 5671, 0x746b }, { 5680, 0xb051 }, { 5686, 0x1323 }, { 5692, 0x3b01 },
2742 { 5698, 0x1030 }, { 5701, 0x0000 }, { 5701, 0x0000 }, { 5701, 0x7000 },
2743 { 5704, 0xb011 }, { 5709, 0x1303 }, { 5714, 0x2900 }, { 5717, 0x1110 },
2744 /* 0xaf00 */
2745 { 5720, 0x2180 }, { 5723, 0x0001 }, { 5724, 0x3000 }, { 5726, 0xb015 },
2746 { 5732, 0x030e }, { 5737, 0x3001 }, { 5740, 0x0030 }, { 5742, 0x0200 },
2747 { 5743, 0x0111 }, { 5746, 0x1023 }, { 5750, 0x0000 }, { 5750, 0x1300 },
2748 { 5753, 0x6b81 }, { 5760, 0x1010 }, { 5762, 0x0300 }, { 5764, 0x0113 },
2749 /* 0xb000 */
2750 { 5768, 0x1013 }, { 5772, 0x3011 }, { 5776, 0x0100 }, { 5777, 0x0000 },
2751 { 5777, 0x5530 }, { 5783, 0x22b8 }, { 5789, 0x0000 }, { 5789, 0x3000 },
2752 { 5791, 0xb011 }, { 5796, 0x9702 }, { 5802, 0xfb07 }, { 5812, 0x113a },
2753 { 5818, 0x03b0 }, { 5823, 0x0113 }, { 5827, 0x0021 }, { 5829, 0x0000 },
2754 /* 0xb100 */
2755 { 5829, 0x1b00 }, { 5833, 0x3b0d }, { 5841, 0x1138 }, { 5846, 0x03b0 },
2756 { 5851, 0x0113 }, { 5855, 0x1133 }, { 5861, 0x0001 }, { 5862, 0x1300 },
2757 { 5865, 0x2b05 }, { 5871, 0x111c }, { 5876, 0x0100 }, { 5877, 0x0000 },
2758 { 5877, 0x1000 }, { 5878, 0xb011 }, { 5883, 0x1300 }, { 5886, 0x2a01 },
2759 /* 0xb200 */
2760 { 5890, 0x1930 }, { 5895, 0x02b0 }, { 5899, 0x0001 }, { 5900, 0x1010 },
2761 { 5902, 0x0000 }, { 5902, 0x1100 }, { 5904, 0x0301 }, { 5907, 0x1030 },
2762 { 5910, 0x0230 }, { 5913, 0x0713 }, { 5919, 0x146b }, { 5926, 0x0011 },
2763 { 5928, 0x1300 }, { 5931, 0x2b05 }, { 5937, 0xf974 }, { 5947, 0x8fb8 },
2764 /* 0xb300 */
2765 { 5956, 0x0113 }, { 5960, 0x103b }, { 5966, 0x0000 }, { 5966, 0x0000 },
2766 { 5966, 0x0000 }, { 5966, 0xd970 }, { 5974, 0x4ab0 }, { 5980, 0x0113 },
2767 { 5984, 0x103b }, { 5990, 0x0011 }, { 5992, 0x1103 }, { 5996, 0x0000 },
2768 { 5996, 0x5930 }, { 6002, 0x2ab1 }, { 6009, 0x0111 }, { 6012, 0x1000 },
2769 /* 0xb400 */
2770 { 6013, 0x0000 }, { 6013, 0x1101 }, { 6016, 0x0b01 }, { 6020, 0x0010 },
2771 { 6021, 0x0000 }, { 6021, 0x0113 }, { 6025, 0x102b }, { 6030, 0x0000 },
2772 { 6030, 0x0101 }, { 6032, 0x2000 }, { 6033, 0x1110 }, { 6036, 0x02a0 },
2773 { 6039, 0x0111 }, { 6042, 0x3021 }, { 6046, 0xb059 }, { 6053, 0x0102 },
2774 /* 0xb500 */
2775 { 6055, 0x0000 }, { 6055, 0x1930 }, { 6060, 0x07b0 }, { 6066, 0x0113 },
2776 { 6070, 0x383b }, { 6078, 0xb011 }, { 6083, 0x0003 }, { 6085, 0x0000 },
2777 { 6085, 0x0000 }, { 6085, 0x0000 }, { 6085, 0x0d13 }, { 6091, 0x383b },
2778 { 6099, 0xb011 }, { 6104, 0x0103 }, { 6107, 0x1000 }, { 6108, 0x0000 },
2779 /* 0xb600 */
2780 { 6108, 0x0000 }, { 6108, 0x0113 }, { 6112, 0x1020 }, { 6114, 0x0010 },
2781 { 6115, 0x0100 }, { 6116, 0x0000 }, { 6116, 0x0110 }, { 6118, 0x0000 },
2782 { 6118, 0x0000 }, { 6118, 0x3000 }, { 6120, 0x1811 }, { 6124, 0x0002 },
2783 { 6125, 0x0000 }, { 6125, 0x0010 }, { 6126, 0x0000 }, { 6126, 0x0111 },
2784 /* 0xb700 */
2785 { 6129, 0x0023 }, { 6132, 0x0000 }, { 6132, 0x9300 }, { 6136, 0x0b01 },
2786 { 6140, 0x1110 }, { 6143, 0x0030 }, { 6145, 0x0111 }, { 6148, 0x302b },
2787 { 6154, 0xb011 }, { 6159, 0x13c7 }, { 6167, 0x3b01 }, { 6173, 0x0130 },
2788 { 6176, 0x0280 }, { 6178, 0x0000 }, { 6178, 0x3000 }, { 6180, 0xb011 },
2789 /* 0xb800 */
2790 { 6185, 0x1383 }, { 6191, 0x2b01 }, { 6196, 0x1130 }, { 6200, 0x03b0 },
2791 { 6205, 0x0011 }, { 6207, 0x300a }, { 6211, 0xb011 }, { 6216, 0x1102 },
2792 { 6219, 0x2000 }, { 6220, 0x0000 }, { 6220, 0x0100 }, { 6221, 0x0111 },
2793 { 6224, 0x102b }, { 6229, 0xa011 }, { 6233, 0x1302 }, { 6237, 0x2b01 },
2794 /* 0xb900 */
2795 { 6242, 0x0010 }, { 6243, 0x0100 }, { 6244, 0x0001 }, { 6245, 0x3000 },
2796 { 6247, 0x9011 }, { 6251, 0x1302 }, { 6255, 0x2b01 }, { 6260, 0x1130 },
2797 { 6264, 0x66b0 }, { 6271, 0x0000 }, { 6271, 0x3000 }, { 6273, 0xb011 },
2798 { 6278, 0xd302 }, { 6284, 0x6b07 }, { 6292, 0x113a }, { 6298, 0x07b0 },
2799 /* 0xba00 */
2800 { 6304, 0x0103 }, { 6307, 0x0020 }, { 6308, 0x0000 }, { 6308, 0x1300 },
2801 { 6311, 0x6b05 }, { 6318, 0x1138 }, { 6323, 0x03b0 }, { 6328, 0x0113 },
2802 { 6332, 0x10b8 }, { 6337, 0x0000 }, { 6337, 0x1b00 }, { 6341, 0x2b05 },
2803 { 6347, 0x0110 }, { 6349, 0x0300 }, { 6351, 0x0000 }, { 6351, 0x1000 },
2804 /* 0xbb00 */
2805 { 6352, 0xa011 }, { 6356, 0x1102 }, { 6359, 0xa0a1 }, { 6362, 0x7970 },
2806 { 6370, 0xa2b0 }, { 6376, 0x0111 }, { 6379, 0x100a }, { 6382, 0x0000 },
2807 { 6382, 0x1100 }, { 6384, 0x0001 }, { 6385, 0x1110 }, { 6388, 0x0090 },
2808 { 6390, 0x0111 }, { 6393, 0x0009 }, { 6395, 0x0000 }, { 6395, 0x9300 },
2809 /* 0xbc00 */
2810 { 6399, 0xbb05 }, { 6407, 0xf9f2 }, { 6418, 0x22b0 }, { 6423, 0x0113 },
2811 { 6427, 0x323b }, { 6435, 0x2001 }, { 6437, 0x0000 }, { 6437, 0x0000 },
2812 { 6437, 0x5930 }, { 6443, 0x06b0 }, { 6448, 0x0193 }, { 6453, 0x303b },
2813 { 6460, 0xa011 }, { 6464, 0x1123 }, { 6469, 0x0000 }, { 6469, 0x1170 },
2814 /* 0xbd00 */
2815 { 6474, 0x02b0 }, { 6478, 0x0011 }, { 6480, 0x1010 }, { 6482, 0x0000 },
2816 { 6482, 0x1301 }, { 6486, 0x0301 }, { 6489, 0x0110 }, { 6491, 0x0000 },
2817 { 6491, 0x0793 }, { 6498, 0x162b }, { 6505, 0x0010 }, { 6506, 0x0101 },
2818 { 6508, 0x0000 }, { 6508, 0x1130 }, { 6512, 0x0200 }, { 6513, 0x0111 },

```

```
2819 /* 0xbe00 */
2820 { 6516, 0x3029 }, { 6521, 0xb011 }, { 6526, 0x0000 }, { 6526, 0x0000 },
2821 { 6526, 0x5130 }, { 6531, 0x0eb0 }, { 6537, 0x0513 }, { 6542, 0x383b },
2822 { 6550, 0xb011 }, { 6555, 0x0303 }, { 6559, 0x0100 }, { 6560, 0x0000 },
2823 { 6560, 0x0000 }, { 6560, 0x0193 }, { 6565, 0x1039 }, { 6570, 0x0000 },
2824 /* 0xbf00 */
2825 { 6570, 0x0302 }, { 6573, 0x3b00 }, { 6578, 0x0000 }, { 6578, 0x0000 },
2826 { 6578, 0x0113 }, { 6582, 0x0023 }, { 6585, 0x0000 }, { 6585, 0x0000 },
2827 { 6585, 0x0000 }, { 6585, 0x0010 }, { 6586, 0x0000 }, { 6586, 0x0001 },
2828 { 6587, 0x3020 }, { 6590, 0x9011 }, { 6594, 0x0002 }, { 6595, 0x0000 },
2829 /* 0xc000 */
2830 { 6595, 0x0000 }, { 6595, 0x0000 }, { 6595, 0x0000 }, { 6595, 0x1000 },
2831 { 6596, 0x0000 }, { 6596, 0x1102 }, { 6599, 0x0301 }, { 6602, 0x0000 },
2832 { 6602, 0x0000 }, { 6602, 0x0113 }, { 6606, 0xb02b }, { 6613, 0xb079 },
2833 { 6621, 0x1323 }, { 6627, 0x3b01 }, { 6633, 0x1130 }, { 6637, 0x02b0 },
2834 /* 0xc100 */
2835 { 6641, 0x0111 }, { 6644, 0xf021 }, { 6650, 0xb0d9 }, { 6658, 0x1343 },
2836 { 6664, 0x3b01 }, { 6670, 0x1130 }, { 6674, 0x03b0 }, { 6679, 0x0111 },
2837 { 6682, 0x7020 }, { 6686, 0xb051 }, { 6692, 0x1322 }, { 6697, 0x2001 },
2838 { 6699, 0x1110 }, { 6702, 0x0190 }, { 6705, 0x0111 }, { 6708, 0x300b },
2839 /* 0xc200 */
2840 { 6713, 0xb011 }, { 6718, 0x9302 }, { 6723, 0xab01 }, { 6729, 0x0016 },
2841 { 6732, 0x0100 }, { 6733, 0x0113 }, { 6737, 0x3021 }, { 6741, 0xb011 },
2842 { 6746, 0x0302 }, { 6749, 0x2901 }, { 6753, 0x3130 }, { 6758, 0x02b0 },
2843 { 6762, 0x0000 }, { 6762, 0x3000 }, { 6764, 0xb819 }, { 6771, 0x1b42 },
2844 /* 0xc300 */
2845 { 6777, 0x3301 }, { 6782, 0x1138 }, { 6787, 0x0330 }, { 6791, 0x0000 },
2846 { 6791, 0x0020 }, { 6792, 0x0000 }, { 6792, 0x1300 }, { 6795, 0x3305 },
2847 { 6801, 0x1110 }, { 6804, 0x0000 }, { 6804, 0x0000 }, { 6804, 0x0000 },
2848 { 6804, 0x0001 }, { 6805, 0x9300 }, { 6809, 0x2305 }, { 6814, 0x0130 },
2849 /* 0xc400 */
2850 { 6817, 0x0100 }, { 6818, 0x0001 }, { 6819, 0x1010 }, { 6821, 0x3011 },
2851 { 6825, 0x0100 }, { 6826, 0x0000 }, { 6826, 0x1130 }, { 6830, 0x0230 },
2852 { 6833, 0x0001 }, { 6834, 0x1010 }, { 6836, 0x0000 }, { 6836, 0x1100 },
2853 { 6838, 0x0000 }, { 6838, 0x0000 }, { 6838, 0x0200 }, { 6839, 0x8513 },
2854 /* 0xc500 */
2855 { 6845, 0x1003 }, { 6848, 0x1011 }, { 6851, 0x1300 }, { 6854, 0x2b01 },
2856 { 6859, 0x7730 }, { 6867, 0x63b8 }, { 6875, 0x0113 }, { 6879, 0x303b },
2857 { 6886, 0xb091 }, { 6892, 0x11a2 }, { 6897, 0x0201 }, { 6899, 0x7b30 },
2858 { 6907, 0x57f0 }, { 6916, 0x0113 }, { 6920, 0x702b }, { 6927, 0xf0d1 },
2859 /* 0xc600 */
2860 { 6935, 0x11e3 }, { 6942, 0x1b01 }, { 6947, 0x7130 }, { 6953, 0x0ab9 },
2861 { 6960, 0x0113 }, { 6964, 0x303b }, { 6971, 0x9001 }, { 6974, 0x1302 },
2862 { 6978, 0x2b01 }, { 6983, 0x1130 }, { 6987, 0x02b0 }, { 6991, 0x0713 },
2863 { 6997, 0x302b }, { 7003, 0x3011 }, { 7007, 0x1303 }, { 7012, 0x2301 },
2864 /* 0xc700 */
2865 { 7016, 0x1130 }, { 7020, 0x02b0 }, { 7024, 0x0113 }, { 7028, 0x30ab },
2866 { 7035, 0xb411 }, { 7041, 0x11fe }, { 7050, 0x0901 }, { 7053, 0x7130 },
2867 { 7059, 0x47b8 }, { 7067, 0x05d3 }, { 7074, 0x307b }, { 7082, 0xb011 },
2868 { 7087, 0x5303 }, { 7093, 0x2101 }, { 7096, 0x1110 }, { 7099, 0x0000 },
2869 /* 0xc800 */
2870 { 7099, 0x0513 }, { 7104, 0x306b }, { 7111, 0xb011 }, { 7116, 0x1102 },
2871 { 7119, 0x3301 }, { 7124, 0x0010 }, { 7125, 0x0000 }, { 7125, 0x0513 },
2872 { 7130, 0x38eb }, { 7139, 0xa010 }, { 7142, 0x0102 }, { 7144, 0x3000 },
2873 { 7146, 0x1110 }, { 7149, 0x02b0 }, { 7153, 0x0013 }, { 7156, 0x3020 },
2874 /* 0xc900 */
2875 { 7159, 0xb071 }, { 7166, 0x0102 }, { 7168, 0x1000 }, { 7169, 0x0010 },
2876 { 7170, 0x0000 }, { 7170, 0x0113 }, { 7174, 0x100b }, { 7178, 0x1011 },
2877 { 7181, 0x1300 }, { 7184, 0x2b01 }, { 7189, 0x0000 }, { 7189, 0x0000 },
2878 { 7189, 0x0593 }, { 7195, 0x366b }, { 7204, 0xb095 }, { 7211, 0x1303 },
2879 /* 0xca00 */
2880 { 7216, 0x3b01 }, { 7222, 0x0110 }, { 7224, 0x0200 }, { 7225, 0x0000 },
2881 { 7225, 0x3000 }, { 7227, 0xb011 }, { 7232, 0x0103 }, { 7235, 0x2000 },
2882 { 7236, 0x0010 }, { 7237, 0x0100 }, { 7238, 0x0000 }, { 7238, 0x3000 },
2883 { 7240, 0xb011 }, { 7245, 0x030a }, { 7249, 0x1001 }, { 7251, 0x0010 },
2884 /* 0xcb00 */
2885 { 7252, 0x0100 }, { 7253, 0x0111 }, { 7256, 0x0003 }, { 7258, 0x0000 },
2886 { 7258, 0x1302 }, { 7262, 0x2301 }, { 7266, 0x0010 }, { 7267, 0x0300 },
2887 { 7269, 0x0000 }, { 7269, 0x1000 }, { 7270, 0x0000 }, { 7270, 0x0100 },
2888 { 7271, 0x0000 }, { 7271, 0x0010 }, { 7272, 0x0290 }, { 7275, 0x0000 },
2889 /* 0xcc00 */
2890 { 7275, 0x3000 }, { 7277, 0x3011 }, { 7281, 0x5386 }, { 7288, 0x7b01 },
2891 { 7295, 0x1130 }, { 7299, 0x03b0 }, { 7304, 0x0151 }, { 7308, 0x0021 },
2892 { 7310, 0x0000 }, { 7310, 0x1300 }, { 7313, 0x3b01 }, { 7319, 0x1130 },
2893 { 7323, 0x02b0 }, { 7327, 0x0011 }, { 7329, 0x1010 }, { 7331, 0x0001 },
2894 /* 0xcd00 */
2895 { 7332, 0x1302 }, { 7336, 0x2b01 }, { 7341, 0x1110 }, { 7344, 0x0200 },
2896 { 7345, 0x0000 }, { 7345, 0x1000 }, { 7346, 0xb011 }, { 7351, 0x0102 },
2897 { 7353, 0x0100 }, { 7354, 0x1130 }, { 7358, 0x02b0 }, { 7362, 0x0001 },
2898 { 7363, 0x1010 }, { 7365, 0x0001 }, { 7366, 0x1100 }, { 7368, 0x2b01 },
2899 /* 0xce00 */
2900 { 7373, 0x1110 }, { 7376, 0x0210 }, { 7378, 0x0113 }, { 7382, 0x002b },
2901 { 7386, 0x0000 }, { 7386, 0x9300 }, { 7390, 0x2b03 }, { 7396, 0x1130 },
2902 { 7400, 0x02b0 }, { 7404, 0x0113 }, { 7408, 0x303b }, { 7415, 0x0000 },
2903 { 7415, 0x0002 }, { 7416, 0x0000 }, { 7416, 0x1930 }, { 7421, 0x03b0 },
2904 /* 0xcf00 */
2905 { 7426, 0x0113 }, { 7430, 0x102b }, { 7435, 0xb011 }, { 7440, 0x0103 },
```

```

2906 { 7443, 0x0000 }, { 7443, 0x1130 }, { 7447, 0x02b0 }, { 7451, 0x0113 },
2907 { 7455, 0x1021 }, { 7458, 0x0000 }, { 7458, 0x0102 }, { 7460, 0x0001 },
2908 { 7461, 0x0010 }, { 7462, 0x0000 }, { 7462, 0x0113 }, { 7466, 0x102b },
2909 /* 0xd000 */
2910 { 7471, 0x0011 }, { 7473, 0x0102 }, { 7475, 0x2000 }, { 7476, 0x1130 },
2911 { 7480, 0x02b0 }, { 7484, 0x0111 }, { 7487, 0x3001 }, { 7490, 0x3011 },
2912 { 7494, 0x0002 }, { 7495, 0x0000 }, { 7495, 0x1130 }, { 7499, 0x02b0 },
2913 { 7503, 0x0313 }, { 7508, 0x303b }, { 7515, 0xb011 }, { 7520, 0x0103 },
2914 /* 0xd100 */
2915 { 7523, 0x2000 }, { 7524, 0x0000 }, { 7524, 0x0000 }, { 7524, 0x0513 },
2916 { 7529, 0x303b }, { 7536, 0xb011 }, { 7541, 0x1102 }, { 7544, 0x1000 },
2917 { 7545, 0x0110 }, { 7547, 0x0000 }, { 7547, 0x0113 }, { 7551, 0x142b },
2918 { 7557, 0x0001 }, { 7558, 0x0100 }, { 7559, 0x0000 }, { 7559, 0x0110 },
2919 /* 0xd200 */
2920 { 7561, 0x0280 }, { 7563, 0x0001 }, { 7564, 0x3000 }, { 7566, 0xb011 },
2921 { 7571, 0x0102 }, { 7573, 0x1000 }, { 7574, 0x0010 }, { 7575, 0x0000 },
2922 { 7575, 0x0113 }, { 7579, 0x1023 }, { 7583, 0x1011 }, { 7586, 0x9302 },
2923 { 7591, 0x0b05 }, { 7596, 0x1110 }, { 7599, 0x0030 }, { 7601, 0x0113 },
2924 /* 0xd300 */
2925 { 7605, 0x702b }, { 7612, 0xb051 }, { 7618, 0x1323 }, { 7624, 0x3b01 },
2926 { 7630, 0x0030 }, { 7632, 0x0000 }, { 7632, 0x0000 }, { 7632, 0x3000 },
2927 { 7634, 0xb011 }, { 7639, 0x1303 }, { 7644, 0x2b01 }, { 7649, 0x1110 },
2928 { 7652, 0x0330 }, { 7656, 0x0101 }, { 7658, 0x300a }, { 7662, 0xb011 },
2929 /* 0xd400 */
2930 { 7667, 0x0102 }, { 7669, 0x2000 }, { 7670, 0x0000 }, { 7670, 0x0000 },
2931 { 7670, 0x0011 }, { 7672, 0x1000 }, { 7673, 0xa011 }, { 7677, 0x9300 },
2932 { 7681, 0x2b05 }, { 7687, 0x0010 }, { 7688, 0x0200 }, { 7689, 0x0000 },
2933 { 7689, 0x1000 }, { 7690, 0x9011 }, { 7694, 0x1100 }, { 7696, 0x2901 },
2934 /* 0xd500 */
2935 { 7700, 0x1110 }, { 7703, 0x00b0 }, { 7706, 0x0000 }, { 7706, 0x3000 },
2936 { 7708, 0xb011 }, { 7713, 0x1302 }, { 7717, 0x2b21 }, { 7723, 0x1130 },
2937 { 7727, 0x03b0 }, { 7732, 0x0001 }, { 7733, 0x0020 }, { 7734, 0x0000 },
2938 { 7734, 0x1300 }, { 7737, 0x2b05 }, { 7743, 0x1130 }, { 7747, 0x02b0 },
2939 /* 0xd600 */
2940 { 7751, 0x0113 }, { 7755, 0x103b }, { 7761, 0x2011 }, { 7764, 0x1300 },
2941 { 7767, 0x2b21 }, { 7773, 0x1132 }, { 7778, 0x0280 }, { 7780, 0x0013 },
2942 { 7783, 0x3028 }, { 7787, 0xa011 }, { 7791, 0x1102 }, { 7794, 0xa01 },
2943 { 7797, 0x1130 }, { 7801, 0x0292 }, { 7805, 0x0111 }, { 7808, 0x3021 },
2944 /* 0xd700 */
2945 { 7812, 0x0011 }, { 7814, 0x1302 }, { 7818, 0x2b01 }, { 7823, 0x1130 },
2946 { 7827, 0x0290 }, { 7830, 0x03d3 }, { 7837, 0x122b }, { 7843, 0x3011 },
2947 { 7847, 0x1302 }, { 7851, 0x2b01 },
2948 };
2949 static const Summary16 ksc5601_uni2indx_pagef9[17] = {
2950 /* 0xf900 */
2951 { 7856, 0xffff }, { 7872, 0xffff }, { 7888, 0xffff }, { 7904, 0xffff },
2952 { 7920, 0xffff }, { 7936, 0xffff }, { 7952, 0xffff }, { 7968, 0xffff },
2953 { 7984, 0xffff }, { 8000, 0xffff }, { 8016, 0xffff }, { 8032, 0xffff },
2954 { 8048, 0xffff }, { 8064, 0xffff }, { 8080, 0xffff }, { 8096, 0xffff },
2955 /* 0xfa00 */
2956 { 8112, 0x0fff },
2957 };
2958 static const Summary16 ksc5601_uni2indx_pageff[15] = {
2959 /* 0xff00 */
2960 { 8124, 0xffff }, { 8139, 0xffff }, { 8155, 0xffff }, { 8171, 0xffff },
2961 { 8187, 0xffff }, { 8203, 0x7fff }, { 8218, 0x0000 }, { 8218, 0x0000 },
2962 { 8218, 0x0000 }, { 8218, 0x0000 }, { 8218, 0x0000 }, { 8218, 0x0000 },
2963 { 8218, 0x0000 }, { 8218, 0x0000 }, { 8218, 0x006f },
2964 };
2965
2966 static int
2967 ksc5601_wctomb (conv_t conv, unsigned char *r, ucs4_t wc, int n)
2968 {
2969     if (n >= 2) {
2970         const Summary16 *summary = NULL;
2971         if (wc < 0x0460)
2972             summary = &ksc5601_uni2indx_page00[(wc>>4)];
2973         else if (wc >= 0x2000 && wc < 0x2670)
2974             summary = &ksc5601_uni2indx_page20[(wc>>4)-0x200];
2975         else if (wc >= 0x3000 && wc < 0x33e0)
2976             summary = &ksc5601_uni2indx_page30[(wc>>4)-0x300];
2977         else if (wc >= 0x4e00 && wc < 0x9fa0)
2978             summary = &ksc5601_uni2indx_page4e[(wc>>4)-0x4e0];
2979         else if (wc >= 0xac00 && wc < 0xd7a0)
2980             summary = &ksc5601_uni2indx_pageac[(wc>>4)-0xac0];
2981         else if (wc >= 0xf900 && wc < 0xfa10)
2982             summary = &ksc5601_uni2indx_pagef9[(wc>>4)-0xf90];
2983         else if (wc >= 0xff00 && wc < 0xffff)
2984             summary = &ksc5601_uni2indx_pageff[(wc>>4)-0xff0];
2985         if (summary) {
2986             unsigned short used = summary->used;
2987             unsigned int i = wc & 0x0f;
2988             if (used & ((unsigned short) 1 << i)) {
2989                 unsigned short c;
2990                 /* Keep in 'used' only the bits 0..i-1. */
2991                 used &= ((unsigned short) 1 << i) - 1;
2992                 /* Add 'summary->indx' and the number of bits set in 'used'. */

```

```

2993     used = (used & 0x5555) + ((used & 0xaaaa) » 1);
2994     used = (used & 0x3333) + ((used & 0xcccc) » 2);
2995     used = (used & 0x0f0f) + ((used & 0xf0f0) » 4);
2996     used = (used & 0x00ff) + (used » 8);
2997     c = ksc5601_2charset[summary->indx + used];
2998     r[0] = (c » 8); r[1] = (c & 0xff);
2999     return 2;
3000 }
3001 }
3002     return RET_ILSEQ;
3003 }
3004     return RET_TOOSMALL;
3005 }
3006 #endif /* NEED_TOMB */

```

32.234 mulelao.h

```

1 /* $XFree86: xc/lib/X11/lcUniConv/mulelao.h,v 1.3 2000/11/29 17:40:35 dawes Exp $ */
2
3 /*
4  * MULELAO-1
5  */
6
7 static const unsigned short mulelao_2uni[96] = {
8     /* 0xa0 */
9     0x00a0, 0x0e81, 0x0e82, 0xffff, 0x0e84, 0xffff, 0xffff, 0x0e87,
10    0x0e88, 0xffff, 0x0e8a, 0xffff, 0xffff, 0x0e8d, 0xffff, 0xffff,
11    /* 0xb0 */
12    0xffff, 0xffff, 0xffff, 0xffff, 0x0e94, 0x0e95, 0x0e96, 0x0e97,
13    0xffff, 0x0e99, 0x0e9a, 0x0e9b, 0x0e9c, 0x0e9d, 0x0e9e, 0x0e9f,
14    /* 0xc0 */
15    0xffff, 0x0ea1, 0x0ea2, 0x0ea3, 0xffff, 0x0ea5, 0xffff, 0x0ea7,
16    0xffff, 0xffff, 0x0eaa, 0x0eab, 0xffff, 0x0ead, 0x0eae, 0x0eaf,
17    /* 0xd0 */
18    0x0eb0, 0x0eb1, 0x0eb2, 0x0eb3, 0x0eb4, 0x0eb5, 0x0eb6, 0x0eb7,
19    0x0eb8, 0x0eb9, 0xffff, 0x0ebb, 0x0ebc, 0x0ebd, 0xffff, 0xffff,
20    /* 0xe0 */
21    0x0ec0, 0x0ec1, 0x0ec2, 0x0ec3, 0x0ec4, 0xffff, 0x0ec6, 0xffff,
22    0x0ec8, 0x0ec9, 0x0eca, 0x0ecb, 0x0ecc, 0x0ecd, 0xffff, 0xffff,
23    /* 0xf0 */
24    0x0ed0, 0x0ed1, 0x0ed2, 0x0ed3, 0x0ed4, 0x0ed5, 0x0ed6, 0x0ed7,
25    0x0ed8, 0x0ed9, 0xffff, 0xffff, 0x0edc, 0x0edd, 0xffff, 0xffff,
26 };
27
28 static int
29 mulelao_mbtowc (conv_t conv, ucs4_t *pwc, const unsigned char *s, int n)
30 {
31     unsigned char c = *s;
32     if (c < 0xa0) {
33         *pwc = (ucs4_t) c;
34         return 1;
35     }
36     else {
37         unsigned short wc = mulelao_2uni[c-0xa0];
38         if (wc != 0xffff) {
39             *pwc = (ucs4_t) wc;
40             return 1;
41         }
42     }
43     return RET_ILSEQ;
44 }
45
46 static const unsigned char mulelao_page0e[96] = {
47     0x00, 0xa1, 0xa2, 0x00, 0xa4, 0x00, 0x00, 0xa7, /* 0x80-0x87 */
48     0xa8, 0x00, 0xaa, 0x00, 0x00, 0xad, 0x00, 0x00, /* 0x88-0x8f */
49     0x00, 0x00, 0x00, 0x00, 0xb4, 0xb5, 0xb6, 0xb7, /* 0x90-0x97 */
50     0x00, 0xb9, 0xba, 0xbb, 0xbc, 0xbd, 0xbe, 0xbf, /* 0x98-0x9f */
51     0x00, 0xc1, 0xc2, 0xc3, 0x00, 0xc5, 0x00, 0xc7, /* 0xa0-0xa7 */
52     0x00, 0x00, 0xca, 0xcb, 0x00, 0xcd, 0xce, 0xcf, /* 0xa8-0xaf */
53     0xd0, 0xd1, 0xd2, 0xd3, 0xd4, 0xd5, 0xd6, 0xd7, /* 0xb0-0xb7 */
54     0xd8, 0xd9, 0x00, 0xdb, 0xdc, 0xdd, 0x00, 0x00, /* 0xb8-0xbf */
55     0xe0, 0xe1, 0xe2, 0xe3, 0xe4, 0x00, 0xe6, 0x00, /* 0xc0-0xc7 */
56     0xe8, 0xe9, 0xea, 0xeb, 0xec, 0xed, 0x00, 0x00, /* 0xc8-0xcf */
57     0xf0, 0xf1, 0xf2, 0xf3, 0xf4, 0xf5, 0xf6, 0xf7, /* 0xd0-0xd7 */
58     0xf8, 0xf9, 0x00, 0x00, 0xfc, 0xfd, 0x00, 0x00, /* 0xd8-0xdf */
59 };
60
61 static int
62 mulelao_wctomb (conv_t conv, unsigned char *r, ucs4_t wc, int n)
63 {
64     unsigned char c = 0;
65     if (wc < 0x00a0) {
66         *r = wc;
67         return 1;
68     }

```

```

69 else if (wc == 0x00a0)
70     c = 0xa0;
71 else if (wc >= 0x0e80 && wc < 0x0ee0)
72     c = mulelao_page0e[wc-0x0e80];
73 if (c != 0) {
74     *r = c;
75     return 1;
76 }
77 return RET_ILSEQ;
78 }

```

32.235 tatar_cyr.h

```

1 /* $XFree86: xc/lib/X11/lcUniConv/tatar_cyr.h,v 1.3 2000/12/04 18:49:42 dawes Exp $ */
2
3 /*
4 * TATAR-CYR
5 */
6
7 static const unsigned short tatar_cyr_2uni[128] = {
8     /* 0x80 */
9     0x04d8, 0x0403, 0x201a, 0x0453, 0x201e, 0x2026, 0x2020, 0x2021,
10    0x20ac, 0x2030, 0x04e8, 0x2039, 0x04ae, 0x0496, 0x04a2, 0x04ba,
11    /* 0x90 */
12    0x04d9, 0x2018, 0x2019, 0x201c, 0x201d, 0x2022, 0x2013, 0x2014,
13    0x98, 0x2122, 0x04e9, 0x203a, 0x04af, 0x0497, 0x04a3, 0x04bb,
14    /* 0xa0 */
15    0x00a0, 0x040e, 0x045e, 0x0408, 0x00a4, 0x0490, 0x00a6, 0x00a7,
16    0x0401, 0x00a9, 0x0404, 0x00ab, 0x00ac, 0x00ad, 0x00ae, 0x0407,
17    /* 0xb0 */
18    0x00b0, 0x00b1, 0x0406, 0x0456, 0x0491, 0x00b5, 0x00b6, 0x00b7,
19    0x0451, 0x2116, 0x0454, 0x00bb, 0x0458, 0x0405, 0x0455, 0x0457,
20    /* 0xc0 */
21    0x0410, 0x0411, 0x0412, 0x0413, 0x0414, 0x0415, 0x0416, 0x0417,
22    0x0418, 0x0419, 0x041a, 0x041b, 0x041c, 0x041d, 0x041e, 0x041f,
23    /* 0xd0 */
24    0x0420, 0x0421, 0x0422, 0x0423, 0x0424, 0x0425, 0x0426, 0x0427,
25    0x0428, 0x0429, 0x042a, 0x042b, 0x042c, 0x042d, 0x042e, 0x042f,
26    /* 0xe0 */
27    0x0430, 0x0431, 0x0432, 0x0433, 0x0434, 0x0435, 0x0436, 0x0437,
28    0x0438, 0x0439, 0x043a, 0x043b, 0x043c, 0x043d, 0x043e, 0x043f,
29    /* 0xf0 */
30    0x0440, 0x0441, 0x0442, 0x0443, 0x0444, 0x0445, 0x0446, 0x0447,
31    0x0448, 0x0449, 0x044a, 0x044b, 0x044c, 0x044d, 0x044e, 0x044f,
32 };
33
34 static int
35 tatar_cyr_mbtowc (conv_t conv, ucs4_t *pwc, const unsigned char *s, int n)
36 {
37     unsigned char c = *s;
38     if (c < 0x80)
39         *pwc = (ucs4_t) c;
40     else
41         *pwc = (ucs4_t) tatar_cyr_2uni[c-0x80];
42     return 1;
43 }
44
45 static const unsigned char tatar_cyr_page00[32] = {
46     0xa0, 0x00, 0x00, 0x00, 0xa4, 0x00, 0xa6, 0xa7, /* 0xa0-0xa7 */
47     0x00, 0xa9, 0x00, 0xab, 0xac, 0xad, 0xae, 0x00, /* 0xa8-0xaf */
48     0xb0, 0xb1, 0x00, 0x00, 0x00, 0xb5, 0xb6, 0xb7, /* 0xb0-0xb7 */
49     0x00, 0x00, 0x00, 0xb8, 0x00, 0x00, 0x00, 0x00, /* 0xb8-0xbf */
50 };
51
52 static const unsigned char tatar_cyr_page04[240] = {
53     0x00, 0xa8, 0x00, 0x81, 0xaa, 0xbd, 0xb2, 0xaf, /* 0x00-0x07 */
54     0xa3, 0x00, 0x00, 0x00, 0x00, 0x00, 0xa1, 0x00, /* 0x08-0x0f */
55     0xc0, 0xc1, 0xc2, 0xc3, 0xc4, 0xc5, 0xc6, 0xc7, /* 0x10-0x17 */
56     0xc8, 0xc9, 0xca, 0xcb, 0xcc, 0xcd, 0xce, 0xcf, /* 0x18-0x1f */
57     0xd0, 0xd1, 0xd2, 0xd3, 0xd4, 0xd5, 0xd6, 0xd7, /* 0x20-0x27 */
58     0xd8, 0xd9, 0xda, 0xdb, 0xdc, 0xdd, 0xde, 0xdf, /* 0x28-0x2f */
59     0xe0, 0xe1, 0xe2, 0xe3, 0xe4, 0xe5, 0xe6, 0xe7, /* 0x30-0x37 */
60     0xe8, 0xe9, 0xea, 0xeb, 0xec, 0xed, 0xee, 0xef, /* 0x38-0x3f */
61     0xf0, 0xf1, 0xf2, 0xf3, 0xf4, 0xf5, 0xf6, 0xf7, /* 0x40-0x47 */
62     0xf8, 0xf9, 0xfa, 0xfb, 0xfc, 0xfd, 0xfe, 0xff, /* 0x48-0x4f */
63     0x00, 0xb8, 0x00, 0x83, 0xba, 0xbe, 0xb3, 0xbf, /* 0x50-0x57 */
64     0xbc, 0x00, 0x00, 0x00, 0x00, 0x00, 0xa2, 0x00, /* 0x58-0x5f */
65     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x60-0x67 */
66     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x68-0x6f */
67     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x70-0x77 */
68     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x78-0x7f */
69     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x80-0x87 */
70     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x88-0x8f */
71     0xa5, 0xb4, 0x00, 0x00, 0x00, 0x00, 0x8d, 0x9d, /* 0x90-0x97 */
72     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x98-0x9f */
73     0x00, 0x00, 0x8e, 0x9e, 0x00, 0x00, 0x00, 0x00, /* 0xa0-0xa7 */

```



```

73 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x8c, 0x9c, /* 0xa8-0xaf */
74 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xb0-0xb7 */
75 0x00, 0x00, 0x8f, 0x9f, 0x00, 0x00, 0x00, 0x00, /* 0xb8-0xbf */
76 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xc0-0xc7 */
77 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xc8-0xcf */
78 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xd0-0xd7 */
79 0x80, 0x90, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xd8-0xdf */
80 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xe0-0xe7 */
81 0x8a, 0x9a, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xe8-0xef */
82 };
83 static const unsigned char tatar_cyr_page20[48] = {
84 0x00, 0x00, 0x00, 0x96, 0x97, 0x00, 0x00, 0x00, /* 0x10-0x17 */
85 0x91, 0x92, 0x82, 0x00, 0x93, 0x94, 0x84, 0x00, /* 0x18-0x1f */
86 0x86, 0x87, 0x95, 0x00, 0x00, 0x00, 0x85, 0x00, /* 0x20-0x27 */
87 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x28-0x2f */
88 0x89, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x30-0x37 */
89 0x00, 0x8b, 0x9b, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x38-0x3f */
90 };
91 static const unsigned char tatar_cyr_page21[24] = {
92 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xb9, 0x00, /* 0x10-0x17 */
93 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x18-0x1f */
94 0x00, 0x00, 0x99, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x20-0x27 */
95 };
96 static const unsigned char tatar_cyr_page22[1] = {
97 0xb0, /* 0x16-0x16 */
98 };
99
100 static int
101 tatar_cyr_wctomb (conv_t conv, unsigned char *, ucs4_t wc, int n)
102 {
103     unsigned char c = 0;
104     if (wc < 0x0080) {
105         *r = wc;
106         return 1;
107     }
108     else if (wc >= 0x00a0 && wc < 0x00bc)
109         c = tatar_cyr_page00[wc-0x00a0];
110     else if (wc >= 0x0400 && wc < 0x04ef)
111         c = tatar_cyr_page04[wc-0x0400];
112     else if (wc >= 0x2010 && wc < 0x203b)
113         c = tatar_cyr_page20[wc-0x2010];
114     else if (wc == 0x20ac)
115         c = 0x88;
116     else if (wc >= 0x2110 && wc < 0x2123)
117         c = tatar_cyr_page21[wc-0x2110];
118     if (c != 0) {
119         *r = c;
120         return 1;
121     }
122     return RET_ILSEQ;
123 }

```

32.236 tcvn.h

```

1 /* $XFree86: xc/lib/X11/lcUniConv/tcvn.h,v 1.3 2000/11/29 17:40:35 dawes Exp $ */
2
3 /*
4 * TCVN-5712
5 */
6
7 static const unsigned short tcvn_2uni_1[32] = {
8     /* 0x00 */
9     0x0000, 0x00da, 0x1ee4, 0x0003, 0x1eea, 0x1eec, 0x1eee, 0x0007,
10     0x0008, 0x0009, 0x000a, 0x000b, 0x000c, 0x000d, 0x000e, 0x000f,
11     /* 0x10 */
12     0x0010, 0x1ee8, 0x1ef0, 0x1ef2, 0x1ef6, 0x1ef8, 0x00dd, 0x1ef4,
13     0x0018, 0x0019, 0x001a, 0x001b, 0x001c, 0x001d, 0x001e, 0x001f,
14 };
15 static const unsigned short tcvn_2uni_2[128] = {
16     /* 0x80 */
17     0x00c0, 0x1ea2, 0x00c3, 0x00c1, 0x1ea0, 0x1eb6, 0x1eac, 0x00c8,
18     0x1eba, 0x1ebc, 0x00c9, 0x1eb8, 0x1ec6, 0x00cc, 0x1ec8, 0x0128,
19     /* 0x90 */
20     0x00cd, 0x1eca, 0x00d2, 0x1ece, 0x00d5, 0x00d3, 0x1ecc, 0x1ed8,
21     0x1edc, 0x1ede, 0x1ee0, 0x1eda, 0x1ee2, 0x00d9, 0x1ee6, 0x0168,
22     /* 0xa0 */
23     0x00a0, 0x0102, 0x00c2, 0x00ca, 0x00d4, 0x01a0, 0x01af, 0x0110,
24     0x0103, 0x00e2, 0x00ea, 0x00f4, 0x01a1, 0x01b0, 0x0111, 0x1eb0,
25     /* 0xb0 */
26     0x0300, 0x0309, 0x0303, 0x0301, 0x0323, 0x00e0, 0x1ea3, 0x00e3,
27     0x00e1, 0x1ea1, 0x1eb2, 0x1eb1, 0x1eb3, 0x1eb5, 0x1eaf, 0x1eb4,
28     /* 0xc0 */
29     0x1eae, 0x1ea6, 0x1ea8, 0x1eaa, 0x1ea4, 0x1ec0, 0x1eb7, 0x1ea7,
30     0x1ea9, 0x1eab, 0x1ea5, 0x1ead, 0x00e8, 0x1ec2, 0x1ebb, 0x1ebd,
31     /* 0xd0 */

```

```

32 0x00e9, 0x1eb9, 0x1ec1, 0x1ec3, 0x1ec5, 0x1ebf, 0x1ec7, 0x00ec,
33 0x1ec9, 0x1ec4, 0x1ebe, 0x1ed2, 0x0129, 0x00ed, 0x1ecb, 0x00f2,
34 /* 0xe0 */
35 0x1ed4, 0x1ecf, 0x00f5, 0x00f3, 0x1ecd, 0x1ed3, 0x1ed5, 0x1ed7,
36 0x1ed1, 0x1ed9, 0x1edd, 0x1edf, 0x1ee1, 0x1edb, 0x1ee3, 0x00f9,
37 /* 0xf0 */
38 0x1ed6, 0x1ee7, 0x0169, 0x00fa, 0x1ee5, 0x1eeb, 0x1eed, 0x1eef,
39 0x1ee9, 0x1ef1, 0x1ef3, 0x1ef7, 0x1ef9, 0x00fd, 0x1ef5, 0x1ed0,
40 };
41
42 static int
43 tcvn_mbtowc (conv_t conv, ucs4_t *pwc, const unsigned char *s, int n)
44 {
45     unsigned char c = *s;
46     if (c < 0x20)
47         *pwc = (ucs4_t) tcvn_2uni_1[c];
48     else if (c < 0x80)
49         *pwc = (ucs4_t) c;
50     else
51         *pwc = (ucs4_t) tcvn_2uni_2[c-0x80];
52     return 1;
53 }
54
55 static const unsigned char tcvn_page00[96+184] = {
56 0xa0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xa0-0xa7 */
57 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xa8-0xaf */
58 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xb0-0xb7 */
59 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xb8-0xbf */
60 0x80, 0x83, 0xa2, 0x82, 0x00, 0x00, 0x00, 0x00, /* 0xc0-0xc7 */
61 0x87, 0x8a, 0xa3, 0x00, 0x8d, 0x90, 0x00, 0x00, /* 0xc8-0xcf */
62 0x00, 0x00, 0x92, 0x95, 0xa4, 0x94, 0x00, 0x00, /* 0xd0-0xd7 */
63 0x00, 0x9d, 0x01, 0x00, 0x00, 0x16, 0x00, 0x00, /* 0xd8-0xdf */
64 0xb5, 0xb8, 0xa9, 0xb7, 0x00, 0x00, 0x00, 0x00, /* 0xe0-0xe7 */
65 0xcc, 0xd0, 0xaa, 0x00, 0xd7, 0xdd, 0x00, 0x00, /* 0xe8-0xef */
66 0x00, 0x00, 0xdf, 0xe3, 0xab, 0xe2, 0x00, 0x00, /* 0xf0-0xf7 */
67 0x00, 0xef, 0xf3, 0x00, 0x00, 0xfd, 0x00, 0x00, /* 0xf8-0xff */
68 /* 0x0100 */
69 0x00, 0x00, 0xa1, 0xa8, 0x00, 0x00, 0x00, 0x00, /* 0x00-0x07 */
70 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x08-0x0f */
71 0xa7, 0xae, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x10-0x17 */
72 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x18-0x1f */
73 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x20-0x27 */
74 0x8f, 0xdc, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x28-0x2f */
75 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x30-0x37 */
76 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x38-0x3f */
77 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x40-0x47 */
78 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x48-0x4f */
79 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x50-0x57 */
80 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x58-0x5f */
81 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x60-0x67 */
82 0x9f, 0xf2, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x68-0x6f */
83 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x70-0x77 */
84 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x78-0x7f */
85 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x80-0x87 */
86 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x88-0x8f */
87 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x90-0x97 */
88 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x98-0x9f */
89 0xa5, 0xac, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xa0-0xa7 */
90 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xa6, /* 0xa8-0xaf */
91 0xad, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xb0-0xb7 */
92 };
93 static const unsigned char tcvn_page03[40] = {
94 0xb0, 0xb3, 0x00, 0xb2, 0x00, 0x00, 0x00, 0x00, /* 0x00-0x07 */
95 0x00, 0xb1, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x08-0x0f */
96 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x10-0x17 */
97 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x18-0x1f */
98 0x00, 0x00, 0x00, 0xb4, 0x00, 0x00, 0x00, 0x00, /* 0x20-0x27 */
99 };
100 static const unsigned char tcvn_page1e[96] = {
101 0x84, 0xb9, 0x81, 0xb6, 0xc4, 0xca, 0xc1, 0xc7, /* 0xa0-0xa7 */
102 0xc2, 0xc8, 0xc3, 0xc9, 0x86, 0xcb, 0xc0, 0xbe, /* 0xa8-0xaf */
103 0xaf, 0xbb, 0xba, 0xbc, 0xbf, 0xbd, 0x85, 0xc6, /* 0xb0-0xb7 */
104 0x8b, 0xd1, 0x88, 0xce, 0x89, 0xcf, 0xda, 0xd5, /* 0xb8-0xbf */
105 0xc5, 0xd2, 0xcd, 0xd3, 0xd9, 0xd4, 0x8c, 0xd6, /* 0xc0-0xc7 */
106 0x8e, 0xd8, 0x91, 0xde, 0x96, 0xe4, 0x93, 0xe1, /* 0xc8-0xcf */
107 0xff, 0xe8, 0xdb, 0xe5, 0xe0, 0xe6, 0xf0, 0xe7, /* 0xd0-0xd7 */
108 0x97, 0xe9, 0x9b, 0xed, 0x98, 0xea, 0x99, 0xeb, /* 0xd8-0xdf */
109 0x9a, 0xec, 0x9c, 0xee, 0x02, 0xf4, 0x9e, 0xf1, /* 0xe0-0xe7 */
110 0x11, 0xf8, 0x04, 0xf5, 0x05, 0xf6, 0x06, 0xf7, /* 0xe8-0xef */
111 0x12, 0xf9, 0x13, 0xfa, 0x17, 0xfe, 0x14, 0xfb, /* 0xf0-0xf7 */
112 0x15, 0xfc, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xf8-0xff */
113 };
114
115 static int
116 tcvn_wctomb (conv_t conv, unsigned char *r, ucs4_t wc, int n)
117 {
118     unsigned char c = 0;

```

```

119  if (wc < 0x0080 && (wc >= 0x0020 || (0x00fe0076 & (1 << wc)) == 0)) {
120      *r = wc;
121      return 1;
122  }
123  else if (wc >= 0x00a0 && wc < 0x01b8)
124      c = tcvn_page00[wc-0x00a0];
125  else if (wc >= 0x0300 && wc < 0x0328)
126      c = tcvn_page03[wc-0x0300];
127  else if (wc >= 0x1ea0 && wc < 0x1f00)
128      c = tcvn_pagele[wc-0x1ea0];
129  if (c != 0) {
130      *r = c;
131      return 1;
132  }
133  return RET_ILSEQ;
134 }

```

32.237 tis620.h

```

1 /* $XFree86: xc/lib/X11/lcUniConv/tis620.h,v 1.4 2001/02/09 00:02:54 dawes Exp $ */
2
3 /*
4  * TIS620-0
5  */
6
7 static const unsigned short tis620_2uni[96] = {
8     /* 0xa0 */
9     0xffff, 0x0e01, 0x0e02, 0x0e03, 0x0e04, 0x0e05, 0x0e06, 0x0e07,
10    0x0e08, 0x0e09, 0x0e0a, 0x0e0b, 0x0e0c, 0x0e0d, 0x0e0e, 0x0e0f,
11    /* 0xb0 */
12    0x0e10, 0x0e11, 0x0e12, 0x0e13, 0x0e14, 0x0e15, 0x0e16, 0x0e17,
13    0x0e18, 0x0e19, 0x0e1a, 0x0e1b, 0x0e1c, 0x0e1d, 0x0e1e, 0x0e1f,
14    /* 0xc0 */
15    0x0e20, 0x0e21, 0x0e22, 0x0e23, 0x0e24, 0x0e25, 0x0e26, 0x0e27,
16    0x0e28, 0x0e29, 0x0e2a, 0x0e2b, 0x0e2c, 0x0e2d, 0x0e2e, 0x0e2f,
17    /* 0xd0 */
18    0x0e30, 0x0e31, 0x0e32, 0x0e33, 0x0e34, 0x0e35, 0x0e36, 0x0e37,
19    0x0e38, 0x0e39, 0x0e3a, 0xffff, 0xffff, 0xffff, 0xffff, 0x0e3f,
20    /* 0xe0 */
21    0x0e40, 0x0e41, 0x0e42, 0x0e43, 0x0e44, 0x0e45, 0x0e46, 0x0e47,
22    0x0e48, 0x0e49, 0x0e4a, 0x0e4b, 0x0e4c, 0x0e4d, 0x0e4e, 0x0e4f,
23    /* 0xf0 */
24    0x0e50, 0x0e51, 0x0e52, 0x0e53, 0x0e54, 0x0e55, 0x0e56, 0x0e57,
25    0x0e58, 0x0e59, 0x0e5a, 0x0e5b, 0xffff, 0xffff, 0xffff, 0xffff,
26 };
27
28 static int
29 tis620_mbtowc (conv_t conv, ucs4_t *pwc, const unsigned char *s, int n)
30 {
31     unsigned char c = *s;
32     if (c < 0x80) {
33         *pwc = (ucs4_t) c;
34         return 1;
35     }
36     else if (c < 0xa0) {
37     }
38     else {
39         unsigned short wc = tis620_2uni[c-0xa0];
40         if (wc != 0xffff) {
41             *pwc = (ucs4_t) wc;
42             return 1;
43         }
44     }
45     return RET_ILSEQ;
46 }
47
48 static const unsigned char tis620_page0e[96] = {
49     0x00, 0xa1, 0xa2, 0xa3, 0xa4, 0xa5, 0xa6, 0xa7, /* 0x00-0x07 */
50     0xa8, 0xa9, 0xaa, 0xab, 0xac, 0xad, 0xae, 0xaf, /* 0x08-0x0f */
51     0xb0, 0xb1, 0xb2, 0xb3, 0xb4, 0xb5, 0xb6, 0xb7, /* 0x10-0x17 */
52     0xb8, 0xb9, 0xba, 0xbb, 0xbc, 0xbd, 0xbe, 0xbf, /* 0x18-0x1f */
53     0xc0, 0xc1, 0xc2, 0xc3, 0xc4, 0xc5, 0xc6, 0xc7, /* 0x20-0x27 */
54     0xc8, 0xc9, 0xca, 0xcb, 0xcc, 0xcd, 0xce, 0xcf, /* 0x28-0x2f */
55     0xd0, 0xd1, 0xd2, 0xd3, 0xd4, 0xd5, 0xd6, 0xd7, /* 0x30-0x37 */
56     0xd8, 0xd9, 0xda, 0x00, 0x00, 0x00, 0x00, 0xdf, /* 0x38-0x3f */
57     0xe0, 0xe1, 0xe2, 0xe3, 0xe4, 0xe5, 0xe6, 0xe7, /* 0x40-0x47 */
58     0xe8, 0xe9, 0xea, 0xeb, 0xec, 0xed, 0xee, 0xef, /* 0x48-0x4f */
59     0xf0, 0xf1, 0xf2, 0xf3, 0xf4, 0xf5, 0xf6, 0xf7, /* 0x50-0x57 */
60     0xf8, 0xf9, 0xfa, 0xfb, 0x00, 0x00, 0x00, 0x00, /* 0x58-0x5f */
61 };
62
63 static int
64 tis620_wctomb (conv_t conv, unsigned char *r, ucs4_t wc, int n)
65 {
66     unsigned char c = 0;

```

```

67  if (wc < 0x0080) {
68      *r = wc;
69      return 1;
70  }
71  else if (wc >= 0x0e00 && wc < 0x0e60)
72      c = tis620_page0e[wc-0x0e00];
73  if (c != 0) {
74      *r = c;
75      return 1;
76  }
77  return RET_ILSEQ;
78  }

```

32.238 ucs2be.h

```

1  /*
2  * UCS-2BE = UCS-2 big endian
3  */
4  /* $XFree86: xc/lib/X11/lcUniConv/ucs2be.h,v 1.1 2000/11/28 17:25:09 dawes Exp $ */
5
6  static int
7  ucs2be_mbtowc (conv_t conv, ucs4_t *pwc, const unsigned char *s, int n)
8  {
9      if (n >= 2) {
10         if (s[0] >= 0xd8 && s[0] < 0xe0) {
11             return RET_ILSEQ;
12         } else {
13             *pwc = (s[0] << 8) + s[1];
14             return 2;
15         }
16     }
17     return RET_TOOFEW(0);
18 }
19
20 static int
21 ucs2be_wctomb (conv_t conv, unsigned char *r, ucs4_t wc, int n)
22 {
23     if (wc < 0x10000 && !(wc >= 0xd800 && wc < 0xe000)) {
24         if (n >= 2) {
25             r[0] = (unsigned char) (wc >> 8);
26             r[1] = (unsigned char) wc;
27             return 2;
28         } else
29             return RET_TOOSMALL;
30     }
31     return RET_ILSEQ;
32 }

```

32.239 utf8.h

```

1  /* $XFree86: xc/lib/X11/lcUniConv/utf8.h,v 1.3 2000/11/28 18:50:07 dawes Exp $ */
2
3  /*
4  * UTF-8
5  */
6
7  /* Specification: RFC 2279 */
8
9  static int
10 utf8_mbtowc (conv_t conv, ucs4_t *pwc, const unsigned char *s, int n)
11 {
12     unsigned char c = s[0];
13
14     if (c < 0x80) {
15         *pwc = c;
16         return 1;
17     } else if (c < 0xc2) {
18         return RET_ILSEQ;
19     } else if (c < 0xe0) {
20         if (n < 2)
21             return RET_TOOFEW(0);
22         if (!(s[1] ^ 0x80) < 0x40)
23             return RET_ILSEQ;
24         *pwc = ((ucs4_t) (c & 0x1f) << 6)
25             | (ucs4_t) (s[1] ^ 0x80);
26         return 2;
27     } else if (c < 0xf0) {
28         if (n < 3)
29             return RET_TOOFEW(0);
30         if (!(s[1] ^ 0x80) < 0x40 && (s[2] ^ 0x80) < 0x40
31             && (c >= 0xe1 || s[1] >= 0xa0))
32             return RET_ILSEQ;

```

```

33     *pwc = ((ucs4_t) (c & 0x0f) << 12)
34           | ((ucs4_t) (s[1] ^ 0x80) << 6)
35           | (ucs4_t) (s[2] ^ 0x80);
36     return 3;
37 } else if (c < 0xf8) {
38     if (n < 4)
39         return RET_TOOFEW(0);
40     if (!(s[1] ^ 0x80) < 0x40 && (s[2] ^ 0x80) < 0x40
41         && (s[3] ^ 0x80) < 0x40
42         && (c >= 0xf1 || s[1] >= 0x90))
43         return RET_ILSEQ;
44     *pwc = ((ucs4_t) (c & 0x07) << 18)
45           | ((ucs4_t) (s[1] ^ 0x80) << 12)
46           | ((ucs4_t) (s[2] ^ 0x80) << 6)
47           | (ucs4_t) (s[3] ^ 0x80);
48     return 4;
49 } else if (c < 0xfc) {
50     if (n < 5)
51         return RET_TOOFEW(0);
52     if (!(s[1] ^ 0x80) < 0x40 && (s[2] ^ 0x80) < 0x40
53         && (s[3] ^ 0x80) < 0x40 && (s[4] ^ 0x80) < 0x40
54         && (c >= 0xf9 || s[1] >= 0x88))
55         return RET_ILSEQ;
56     *pwc = ((ucs4_t) (c & 0x03) << 24)
57           | ((ucs4_t) (s[1] ^ 0x80) << 18)
58           | ((ucs4_t) (s[2] ^ 0x80) << 12)
59           | ((ucs4_t) (s[3] ^ 0x80) << 6)
60           | (ucs4_t) (s[4] ^ 0x80);
61     return 5;
62 } else if (c < 0xfe) {
63     if (n < 6)
64         return RET_TOOFEW(0);
65     if (!(s[1] ^ 0x80) < 0x40 && (s[2] ^ 0x80) < 0x40
66         && (s[3] ^ 0x80) < 0x40 && (s[4] ^ 0x80) < 0x40
67         && (s[5] ^ 0x80) < 0x40
68         && (c >= 0xfd || s[1] >= 0x84))
69         return RET_ILSEQ;
70     *pwc = ((ucs4_t) (c & 0x01) << 30)
71           | ((ucs4_t) (s[1] ^ 0x80) << 24)
72           | ((ucs4_t) (s[2] ^ 0x80) << 18)
73           | ((ucs4_t) (s[3] ^ 0x80) << 12)
74           | ((ucs4_t) (s[4] ^ 0x80) << 6)
75           | (ucs4_t) (s[5] ^ 0x80);
76     return 6;
77 } else
78     return RET_ILSEQ;
79 }
80
81 static int
82 utf8_wctomb (conv_t conv, unsigned char *r, ucs4_t wc, int n) /* n == 0 is acceptable */
83 {
84     int count;
85     if (wc < 0x80)
86         count = 1;
87     else if (wc < 0x800)
88         count = 2;
89     else if (wc < 0x10000)
90         count = 3;
91     else if (wc < 0x200000)
92         count = 4;
93     else if (wc < 0x4000000)
94         count = 5;
95     else if (wc <= 0x7fffffff)
96         count = 6;
97     else
98         return RET_ILSEQ;
99     if (n < count)
100         return RET_TOOSMALL;
101     switch (count) { /* note: code falls through cases! */
102     case 6: r[5] = 0x80 | (wc & 0x3f); wc = wc >> 6; wc |= 0x4000000;
103     case 5: r[4] = 0x80 | (wc & 0x3f); wc = wc >> 6; wc |= 0x200000;
104     case 4: r[3] = 0x80 | (wc & 0x3f); wc = wc >> 6; wc |= 0x10000;
105     case 3: r[2] = 0x80 | (wc & 0x3f); wc = wc >> 6; wc |= 0x800;
106     case 2: r[1] = 0x80 | (wc & 0x3f); wc = wc >> 6; wc |= 0xc0;
107     case 1: r[0] = wc;
108     }
109     return count;
110 }

```

32.240 viscii.h

```

1 /* $XFree86: xc/lib/X11/lcUniConv/viscii.h,v 1.3 2000/11/29 17:40:35 dawes Exp $ */
2
3 /*
4 * VISCI1.1-1

```

```

5 */
6
7 /* Specification: RFC 1456 */
8
9 static const unsigned short viscii_2uni_1[32] = {
10     /* 0x00 */
11     0x0000, 0x0001, 0x1eb2, 0x0003, 0x0004, 0x1eb4, 0x1eaa, 0x0007,
12     0x0008, 0x0009, 0x000a, 0x000b, 0x000c, 0x000d, 0x000e, 0x000f,
13     /* 0x10 */
14     0x0010, 0x0011, 0x0012, 0x0013, 0x1ef6, 0x0015, 0x0016, 0x0017,
15     0x0018, 0x1ef8, 0x001a, 0x001b, 0x001c, 0x001d, 0x1ef4, 0x001f,
16 };
17 static const unsigned short viscii_2uni_2[128] = {
18     /* 0x80 */
19     0x1ea0, 0x1eae, 0x1eb0, 0x1eb6, 0x1ea4, 0x1ea6, 0x1ea8, 0x1eac,
20     0x1ebc, 0x1eb8, 0x1ebe, 0x1ec0, 0x1ec2, 0x1ec4, 0x1ec6, 0x1ed0,
21     /* 0x90 */
22     0x1ed2, 0x1ed4, 0x1ed6, 0x1ed8, 0x1ee2, 0x1eda, 0x1edc, 0x1ede,
23     0x1eca, 0x1ece, 0x1ecc, 0x1ec8, 0x1ee6, 0x0168, 0x1ee4, 0x1ef2,
24     /* 0xa0 */
25     0x00d5, 0x1eaf, 0x1eb1, 0x1eb7, 0x1ea5, 0x1ea7, 0x1ea9, 0x1ead,
26     0x1ebd, 0x1eb9, 0x1ebf, 0x1ec1, 0x1ec3, 0x1ec5, 0x1ec7, 0x1ed1,
27     /* 0xb0 */
28     0x1ed3, 0x1ed5, 0x1ed7, 0x1ee0, 0x01a0, 0x1ed9, 0x1edd, 0x1edf,
29     0x1ecb, 0x1ef0, 0x1ee8, 0x1eea, 0x1eec, 0x01a1, 0x1edb, 0x01af,
30     /* 0xc0 */
31     0x00c0, 0x00c1, 0x00c2, 0x00c3, 0x1ea2, 0x0102, 0x1eb3, 0x1eb5,
32     0x00c8, 0x00c9, 0x00ca, 0x1eba, 0x00cc, 0x00cd, 0x0128, 0x1ef3,
33     /* 0xd0 */
34     0x0110, 0x1ee9, 0x00d2, 0x00d3, 0x00d4, 0x1ea1, 0x1ef7, 0x1eeb,
35     0x1eed, 0x00d9, 0x00da, 0x1ef9, 0x1ef5, 0x00dd, 0x1eel, 0x01b0,
36     /* 0xe0 */
37     0x00e0, 0x00e1, 0x00e2, 0x00e3, 0x1ea3, 0x0103, 0x1eef, 0x1eab,
38     0x00e8, 0x00e9, 0x00ea, 0x1ebb, 0x00ec, 0x00ed, 0x0129, 0x1ec9,
39     /* 0xf0 */
40     0x0111, 0x1ef1, 0x00f2, 0x00f3, 0x00f4, 0x00f5, 0x1ecf, 0x1ecd,
41     0x1ee5, 0x00f9, 0x00fa, 0x0169, 0x1ee7, 0x00fd, 0x1ee3, 0x1eee,
42 };
43
44 static int
45 viscii_mbtowc (conv_t conv, ucs4_t *pwc, const unsigned char *s, int n)
46 {
47     unsigned char c = *s;
48     if (c < 0x20)
49         *pwc = (ucs4_t) viscii_2uni_1[c];
50     else if (c < 0x80)
51         *pwc = (ucs4_t) c;
52     else
53         *pwc = (ucs4_t) viscii_2uni_2[c-0x80];
54     return 1;
55 }
56
57 static const unsigned char viscii_page00[64+184] = {
58     0xc0, 0xc1, 0xc2, 0xc3, 0x00, 0x00, 0x00, 0x00, /* 0xc0-0xc7 */
59     0xc8, 0xc9, 0xca, 0x00, 0xcc, 0xcd, 0x00, 0x00, /* 0xc8-0xcf */
60     0x00, 0x00, 0xd2, 0xd3, 0xd4, 0xa0, 0x00, 0x00, /* 0xd0-0xd7 */
61     0x00, 0xd9, 0xda, 0x00, 0x00, 0xdd, 0x00, 0x00, /* 0xd8-0xdf */
62     0xe0, 0xe1, 0xe2, 0xe3, 0x00, 0x00, 0x00, 0x00, /* 0xe0-0xe7 */
63     0xe8, 0xe9, 0xea, 0x00, 0xec, 0xed, 0x00, 0x00, /* 0xe8-0xef */
64     0x00, 0x00, 0xf2, 0xf3, 0xf4, 0xf5, 0x00, 0x00, /* 0xf0-0xf7 */
65     0x00, 0xf9, 0xfa, 0x00, 0x00, 0xfd, 0x00, 0x00, /* 0xf8-0xff */
66     /* 0x0100 */
67     0x00, 0x00, 0xc5, 0xe5, 0x00, 0x00, 0x00, 0x00, /* 0x00-0x07 */
68     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x08-0x0f */
69     0xd0, 0xf0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x10-0x17 */
70     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x18-0x1f */
71     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x20-0x27 */
72     0xce, 0xee, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x28-0x2f */
73     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x30-0x37 */
74     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x38-0x3f */
75     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x40-0x47 */
76     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x48-0x4f */
77     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x50-0x57 */
78     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x58-0x5f */
79     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x60-0x67 */
80     0x9d, 0xfb, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x68-0x6f */
81     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x70-0x77 */
82     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x78-0x7f */
83     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x80-0x87 */
84     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x88-0x8f */
85     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x90-0x97 */
86     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0x98-0x9f */
87     0xb4, 0xbd, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xa0-0xa7 */
88     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xbf, /* 0xa8-0xaf */
89     0xdf, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xb0-0xb7 */
90 };
91 static const unsigned char viscii_pagele[96] = {

```

```
92 0x80, 0xd5, 0xc4, 0xe4, 0x84, 0xa4, 0x85, 0xa5, /* 0xa0-0xa7 */
93 0x86, 0xa6, 0x06, 0xe7, 0x87, 0xa7, 0x81, 0xa1, /* 0xa8-0xaf */
94 0x82, 0xa2, 0x02, 0xc6, 0x05, 0xc7, 0x83, 0xa3, /* 0xb0-0xb7 */
95 0x89, 0xa9, 0xcb, 0xeb, 0x88, 0xa8, 0x8a, 0xaa, /* 0xb8-0xbf */
96 0x8b, 0xab, 0x8c, 0xac, 0x8d, 0xad, 0x8e, 0xae, /* 0xc0-0xc7 */
97 0x9b, 0xef, 0x98, 0xb8, 0x9a, 0xf7, 0x99, 0xf6, /* 0xc8-0xcf */
98 0x8f, 0xaf, 0x90, 0xb0, 0x91, 0xb1, 0x92, 0xb2, /* 0xd0-0xd7 */
99 0x93, 0xb5, 0x95, 0xbe, 0x96, 0xb6, 0x97, 0xb7, /* 0xd8-0xdf */
100 0xb3, 0xde, 0x94, 0xfe, 0x9e, 0xf8, 0x9c, 0xfc, /* 0xe0-0xe7 */
101 0xba, 0xd1, 0xbb, 0xd7, 0xbc, 0xd8, 0xff, 0xe6, /* 0xe8-0xef */
102 0xb9, 0xf1, 0x9f, 0xcf, 0x1e, 0xdc, 0x14, 0xd6, /* 0xf0-0xf7 */
103 0x19, 0xdb, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /* 0xf8-0xff */
104 };
105
106 static int
107 viscii_wctomb (conv_t conv, unsigned char *r, ucs4_t wc, int n)
108 {
109     unsigned char c = 0;
110     if (wc < 0x0080 && (wc >= 0x0020 || (0x42100064 & (1 << wc)) == 0)) {
111         *r = wc;
112         return 1;
113     }
114     else if (wc >= 0x00c0 && wc < 0x01b8)
115         c = viscii_page0[wc-0x00c0];
116     else if (wc >= 0x1ea0 && wc < 0x1f00)
117         c = viscii_pagele[wc-0x1ea0];
118     if (c != 0) {
119         *r = c;
120         return 1;
121     }
122     return RET_ILSEQ;
123 }
```

32.241 Ximint.h

32.242 Xlibint.h

Index

- [_FL_DIAMOND_DOWN_BOX](#)
 - [Enumerations.H, 1002](#)
- [_FL_DIAMOND_UP_BOX](#)
 - [Enumerations.H, 1002](#)
- [_FL_EMBOSSSED_LABEL](#)
 - [Enumerations.H, 1007](#)
- [_FL_ENGRAVED_LABEL](#)
 - [Enumerations.H, 1007](#)
- [_FL_GLEAM_DOWN_BOX](#)
 - [Enumerations.H, 1002](#)
- [_FL_GLEAM_DOWN_FRAME](#)
 - [Enumerations.H, 1002](#)
- [_FL_GLEAM_ROUND_DOWN_BOX](#)
 - [Enumerations.H, 1002](#)
- [_FL_GLEAM_ROUND_UP_BOX](#)
 - [Enumerations.H, 1002](#)
- [_FL_GLEAM_THIN_DOWN_BOX](#)
 - [Enumerations.H, 1002](#)
- [_FL_GLEAM_THIN_UP_BOX](#)
 - [Enumerations.H, 1002](#)
- [_FL_GLEAM_UP_BOX](#)
 - [Enumerations.H, 1002](#)
- [_FL_GLEAM_UP_FRAME](#)
 - [Enumerations.H, 1002](#)
- [_FL_GTK_DOWN_BOX](#)
 - [Enumerations.H, 1002](#)
- [_FL_GTK_DOWN_FRAME](#)
 - [Enumerations.H, 1002](#)
- [_FL_GTK_ROUND_DOWN_BOX](#)
 - [Enumerations.H, 1002](#)
- [_FL_GTK_ROUND_UP_BOX](#)
 - [Enumerations.H, 1002](#)
- [_FL_GTK_THIN_DOWN_BOX](#)
 - [Enumerations.H, 1002](#)
- [_FL_GTK_THIN_DOWN_FRAME](#)
 - [Enumerations.H, 1002](#)
- [_FL_GTK_THIN_UP_BOX](#)
 - [Enumerations.H, 1002](#)
- [_FL_GTK_THIN_UP_FRAME](#)
 - [Enumerations.H, 1002](#)
- [_FL_GTK_UP_BOX](#)
 - [Enumerations.H, 1002](#)
- [_FL_GTK_UP_FRAME](#)
 - [Enumerations.H, 1002](#)
- [_FL_ICON_LABEL](#)
 - [Enumerations.H, 1007](#)
- [_FL_IMAGE_LABEL](#)
 - [Enumerations.H, 1007](#)
- [_FL_MULTI_LABEL](#)
 - [Enumerations.H, 1007](#)
- [_FL_OFLAT_BOX](#)
 - [Enumerations.H, 1002](#)
- [_FL_OSHADOW_BOX](#)
 - [Enumerations.H, 1002](#)
- [_FL_OVAL_BOX](#)
 - [Enumerations.H, 1002](#)
- [_FL_OVAL_FRAME](#)
 - [Enumerations.H, 1002](#)
- [_FL_PLASTIC_DOWN_BOX](#)
 - [Enumerations.H, 1002](#)
- [_FL_PLASTIC_DOWN_FRAME](#)
 - [Enumerations.H, 1002](#)
- [_FL_PLASTIC_ROUND_DOWN_BOX](#)
 - [Enumerations.H, 1002](#)
- [_FL_PLASTIC_ROUND_UP_BOX](#)
 - [Enumerations.H, 1002](#)
- [_FL_PLASTIC_THIN_DOWN_BOX](#)
 - [Enumerations.H, 1002](#)
- [_FL_PLASTIC_THIN_UP_BOX](#)
 - [Enumerations.H, 1002](#)
- [_FL_PLASTIC_UP_BOX](#)
 - [Enumerations.H, 1002](#)
- [_FL_PLASTIC_UP_FRAME](#)
 - [Enumerations.H, 1002](#)
- [_FL_RFLAT_BOX](#)
 - [Enumerations.H, 1002](#)
- [_FL_ROUNDED_BOX](#)
 - [Enumerations.H, 1002](#)
- [_FL_ROUNDED_FRAME](#)
 - [Enumerations.H, 1002](#)
- [_FL_ROUND_DOWN_BOX](#)
 - [Enumerations.H, 1002](#)
- [_FL_ROUND_UP_BOX](#)
 - [Enumerations.H, 1002](#)
- [_FL_RSHADOW_BOX](#)
 - [Enumerations.H, 1002](#)
- [_FL_SHADOW_BOX](#)
 - [Enumerations.H, 1002](#)
- [_FL_SHADOW_FRAME](#)
 - [Enumerations.H, 1002](#)
- [_FL_SHADOW_LABEL](#)
 - [Enumerations.H, 1007](#)
- [_remove](#)
 - [FI_Browser, 348](#)
 - [~FI_Device](#)
 - [FI_Device, 417](#)
 - [~FI_Double_Window](#)
 - [FI_Double_Window, 424](#)

- ~FI_Group
 - FI_Group, [503](#)
- ~FI_Help_View
 - FI_Help_View, [517](#)
- ~FI_Input_
 - FI_Input_, [537](#)
- ~FI_Native_File_Chooser
 - FI_Native_File_Chooser, [603](#)
- ~FI_Plugin_Manager
 - FI_Plugin_Manager, [624](#)
- ~FI_Preferences
 - FI_Preferences, [655](#)
- ~FI_Shared_Image
 - FI_Shared_Image, [706](#)
- ~FI_Table
 - FI_Table, [739](#)
- ~FI_Table_Row
 - FI_Table_Row, [749](#)
- ~FI_Text_Display
 - FI_Text_Display, [781](#)
- ~FI_Widget
 - FI_Widget, [928](#)
- ~FI_Window
 - FI_Window, [961](#)
- A0
 - FI_Paged_Device, [614](#)
- A4
 - FI_Paged_Device, [614](#)
- abi-version.h, [989](#)
- abi_check
 - FI, [324](#)
- abi_version
 - FI, [324](#)
- absolute_top_line_number
 - FI_Text_Display, [782](#)
- activate
 - FI_Tree_Item, [881](#)
 - FI_Widget, [928](#)
- active
 - FI_Widget, [928](#)
- active_r
 - FI_Widget, [929](#)
- add
 - FI_Browser, [348](#)
 - FI_Chart, [390](#)
 - FI_Check_Browser, [394](#)
 - FI_File_Icon, [435](#)
 - FI_Input_Choice, [555](#)
 - FI_Menu_, [566](#), [567](#)
 - FI_Menu_Item, [586](#)
 - FI_Shared_Image, [706](#)
 - FI_Sys_Menu_Bar, [723](#), [724](#)
 - FI_Tree, [844](#)
 - FI_Tree_Item, [881](#), [882](#)
 - FI_Tree_Item_Array, [898](#)
- add_check
 - FI, [324](#)
- add_clipboard_notify
 - Selection & Clipboard functions, [231](#)
- add_color
 - FI_File_Icon, [436](#)
- add_extra
 - FI_File_Chooser, [432](#)
- add_fd
 - FI, [324](#)
- add_handler
 - Events handling functions, [219](#)
- add_idle
 - FI, [325](#)
- add_key_binding
 - FI_Text_Editor, [811](#)
- add_modify_callback
 - FI_Text_Buffer, [762](#)
- add_system_handler
 - Events handling functions, [219](#)
- add_timeout
 - FI, [325](#)
- add_vertex
 - FI_File_Icon, [436](#)
- addPlugin
 - FI_Plugin_Manager, [624](#)
- address
 - FI_Text_Buffer, [762](#)
- align
 - FI_Widget, [929](#)
- ALWAYS_ON
 - FI_Browser_, [368](#)
- angle1
 - FI_Dial, [421](#)
- api_version
 - FI, [325](#)
- append
 - FI_Text_Buffer, [763](#)
- appendfile
 - FI_Text_Buffer, [763](#)
- arc
 - FI_Graphics_Driver, [477](#)
 - FI_PostScript_Graphics_Driver, [638](#)
- arg
 - FI, [325](#)
- args
 - FI, [326](#)
- argument
 - FI_Menu_Item, [586](#), [587](#)
 - FI_Widget, [929](#)
- armscii_8.h, [1311](#)
- array
 - FI_Group, [503](#)
- as_gl_window
 - FI_Gl_Window, [462](#)
 - FI_Widget, [930](#)
- as_group
 - FI_Group, [504](#)
 - FI_Widget, [930](#)
- as_window
 - FI_Widget, [930](#)

- FI_Window, 962
- ascii.h, 1312
- atclose
 - Windows handling functions, 216
- autosize
 - FI_Chart, 390
- awake
 - Multithreading support functions, 273
- b
 - FI_Color_Chooser, 408
- background
 - FI, 327
- background2
 - FI, 327
- bbox
 - FI_Browser_, 369
 - FI_Scroll, 695
- begin
 - FI_Group, 504
- begin_complex_polygon
 - FI_Graphics_Driver, 478
 - FI_PostScript_Graphics_Driver, 638
- begin_line
 - FI_Graphics_Driver, 478
 - FI_PostScript_Graphics_Driver, 638
- begin_loop
 - FI_Graphics_Driver, 478
 - FI_PostScript_Graphics_Driver, 638
- begin_points
 - FI_Graphics_Driver, 478
 - FI_PostScript_Graphics_Driver, 638
- begin_polygon
 - FI_Graphics_Driver, 478
 - FI_PostScript_Graphics_Driver, 639
- belowmouse
 - Events handling functions, 220
- big5.h, 1312
- big5_emacs.h, 1360
- BLOCK_CURSOR
 - FI_Text_Display, 781
- border
 - FI_Window, 962
- BOTH
 - FI_Browser_, 368
- BOTH_ALWAYS
 - FI_Browser_, 368
- bottomline
 - FI_Browser, 348
- bounds
 - FI_Chart, 391
 - FI_Slider, 714
- box
 - FI_Widget, 931
- box_color
 - FI, 327
- box_dh
 - FI, 328
- box_dw
 - FI, 328
- box_dx
 - FI, 328
- box_dy
 - FI, 328
- BROWSE_DIRECTORY
 - FI_Native_File_Chooser, 602
- BROWSE_FILE
 - FI_Native_File_Chooser, 602
- BROWSE_MULTI_DIRECTORY
 - FI_Native_File_Chooser, 602
- BROWSE_MULTI_FILE
 - FI_Native_File_Chooser, 602
- BROWSE_SAVE_DIRECTORY
 - FI_Native_File_Chooser, 603
- BROWSE_SAVE_FILE
 - FI_Native_File_Chooser, 603
- buffer
 - FI_Text_Display, 782
- buffer_modified_cb
 - FI_Text_Display, 782
- buffer_predelete_cb
 - FI_Text_Display, 783
- byte_at
 - FI_Text_Buffer, 763
- Cairo Support Functions and Classes, 277
 - cairo_autolink_context, 277
 - cairo_cc, 278
 - cairo_make_current, 278
- cairo_autolink_context
 - Cairo Support Functions and Classes, 277
- cairo_cc
 - Cairo Support Functions and Classes, 278
- cairo_make_current
 - Cairo Support Functions and Classes, 278
- calc_dimensions
 - FI_Tree, 845
- calc_item_height
 - FI_Tree_Item, 882
- calc_last_char
 - FI_Text_Display, 783
- calc_line_starts
 - FI_Text_Display, 783
- calc_tree
 - FI_Tree, 845
- callback
 - FI_Menu_Item, 587
 - FI_Table, 739
 - FI_Widget, 931, 932
- Callback function typedefs, 213
 - FI_Event_Dispatch, 214
- callback_col
 - FI_Table, 740
- callback_context
 - FI_Table, 740
- callback_item
 - FI_Tree, 845, 846
- callback_reason

- FI_Tree, [846](#)
- callback_row
 - FI_Table, [740](#)
- can_do
 - FI_GI_Window, [462](#)
- can_do_overlay
 - FI_GI_Window, [462](#)
- CARET_CURSOR
 - FI_Text_Display, [781](#)
- case.h, [1248](#)
- cc
 - FI_Cairo_State, [386](#)
- cgdebug.h, [1226](#)
- CHANGED
 - FI_Widget, [927](#)
- changed
 - FI_Widget, [933](#)
- char_at
 - FI_Text_Buffer, [763](#)
- check
 - FI, [328](#)
 - FI_Menu_Item, [588](#)
- checkbox
 - FI_Menu_Item, [588](#)
- checked
 - FI_Menu_Item, [588](#)
- child
 - FI_Group, [504](#)
 - FI_Table, [740](#)
 - FI_Tree_Item, [883](#)
- children
 - FI_Table, [740](#)
- circle
 - FI_Graphics_Driver, [478](#)
 - FI_PostScript_Graphics_Driver, [639](#)
- class_id
 - FI_Device, [418](#)
- class_name
 - FI_Copy_Surface, [412](#)
 - FI_Device, [418](#)
 - FI_Display_Device, [423](#)
 - FI_GDI_Graphics_Driver, [452](#)
 - FI_GDI_Printer_Graphics_Driver, [457](#)
 - FI_Graphics_Driver, [478](#)
 - FI_Image_Surface, [530](#)
 - FI_Paged_Device, [614](#)
 - FI_PostScript_File_Device, [630](#)
 - FI_PostScript_Graphics_Driver, [639](#)
 - FI_PostScript_Printer, [650](#)
 - FI_Printer, [667](#)
 - FI_Quartz_Graphics_Driver, [675](#)
 - FI_Surface_Device, [721](#)
 - FI_System_Printer, [728](#)
 - FI_Xlib_Graphics_Driver, [978](#)
- clear
 - FI_Browser, [349](#)
 - FI_Button, [383](#)
 - FI_Group, [504](#)
 - FI_Menu_, [569](#)
 - FI_Sys_Menu_Bar, [724](#)
 - FI_Table, [740](#)
 - FI_Table_Row, [749](#)
 - FI_Tree, [846](#)
 - FI_Tree_Item_Array, [898](#)
- clear_active
 - FI_Widget, [933](#)
- clear_border
 - FI_Window, [962](#)
- clear_changed
 - FI_Widget, [933](#)
- clear_children
 - FI_Tree, [846](#)
- clear_damage
 - FI_Widget, [933](#)
- clear_modal_states
 - FI_Window, [962](#)
- clear_output
 - FI_Widget, [934](#)
- clear_overlay
 - FI_Menu_Window, [594](#)
- clear_rect
 - FI_Text_Display, [783](#)
- clear_submenu
 - FI_Menu_, [569](#)
 - FI_Sys_Menu_Bar, [725](#)
- clear_visible
 - FI_Widget, [934](#)
- clear_visible_focus
 - FI_Widget, [934](#)
- clear_widget_pointer
 - Safe widget deletion support functions, [275](#)
- client_area
 - FI_Tabs, [755](#)
- clip_box
 - FI_Graphics_Driver, [479](#)
 - FI_PostScript_Graphics_Driver, [639](#)
- CLIP_CHILDREN
 - FI_Widget, [927](#)
- clip_children
 - FI_Group, [504](#), [505](#)
- clipboard_contains
 - Selection & Clipboard functions, [231](#)
- locale_printf
 - FI_PostScript_Graphics_Driver, [639](#)
- close
 - FI_Tree, [846](#), [847](#)
- closedeicon
 - FI_Tree_Prefs, [903](#)
- closeicon
 - FI_Tree, [847](#), [848](#)
 - FI_Tree_Prefs, [903](#)
- col_header
 - FI_Table, [741](#)
- col_resize
 - FI_Table, [741](#)
- col_resize_min

- FI_Table, 741
- col_to_x
 - FI_Text_Display, 784
- col_width
 - FI_Table, 741
- col_width_all
 - FI_Table, 741
- color
 - FI_GDI_Graphics_Driver, 452, 453
 - FI_Graphics_Driver, 479
 - FI_PostScript_Graphics_Driver, 640
 - FI_Quartz_Graphics_Driver, 675
 - FI_Tooltip, 831
 - FI_Widget, 934, 935
 - FI_Xlib_Graphics_Driver, 978, 979
- Color & Font functions, 238
 - fl_color, 240
 - fl_color_average, 241
 - fl_contrast, 241
 - fl_font, 241
 - fl_height, 242
 - fl_latin1_to_local, 242
 - fl_local_to_latin1, 242
 - fl_local_to_mac_roman, 243
 - fl_mac_roman_to_local, 243
 - fl_show_colormap, 243
 - fl_size, 244
 - fl_text_extents, 244, 245
 - fl_width, 245
 - fl_xpixel, 245
 - free_color, 246
 - get_color, 246
 - get_font, 246
 - get_font_name, 246
 - get_font_sizes, 247
 - set_color, 247
 - set_font, 247
 - set_fonts, 247
- color2
 - FI_Widget, 935
- color_average
 - FI_Image, 525
 - FI_Pixmap, 620
 - FI_RGB_Image, 687
 - FI_Shared_Image, 706
 - FI_Tiled_Image, 825
- column_char
 - FI_Browser, 349
- column_widths
 - FI_Browser, 349
- Common Dialogs classes and functions, 294
 - error, 305
 - fatal, 305
 - fl_alert, 295
 - fl_ask, 295
 - fl_beep, 296
 - fl_choice, 296
 - fl_choice_n, 297
 - fl_color_chooser, 298, 299
 - fl_dir_chooser, 300
 - fl_file_chooser, 301
 - fl_file_chooser_callback, 302
 - fl_file_chooser_ok_label, 302
 - fl_input, 302
 - fl_message, 302
 - fl_message_hotspot, 303
 - fl_message_icon, 303
 - fl_message_title, 303
 - fl_message_title_default, 304
 - fl_password, 304
 - warning, 305
- compare
 - FI_Shared_Image, 707
- compose
 - Events handling functions, 220
- compose_reset
 - Events handling functions, 220
- connectorstyle
 - FI_Tree, 848
- contains
 - FI_Widget, 935
- context
 - FI_GI_Window, 463
- CONTEXT_CELL
 - FI_Table, 738
- CONTEXT_COL_HEADER
 - FI_Table, 738
- CONTEXT_ENDPAGE
 - FI_Table, 738
- CONTEXT_NONE
 - FI_Table, 738
- CONTEXT_RC_RESIZE
 - FI_Table, 738
- CONTEXT_ROW_HEADER
 - FI_Table, 738
- CONTEXT_STARTPAGE
 - FI_Table, 738
- CONTEXT_TABLE
 - FI_Table, 738
- context_valid
 - FI_GI_Window, 463
- COPIED_LABEL
 - FI_Widget, 927
- COPIED_TOOLTIP
 - FI_Widget, 927
- copy
 - FI_Bitmap, 340
 - FI_Image, 525, 526
 - FI_Input_, 538
 - FI_Menu_, 570
 - FI_Pixmap, 620
 - FI_RGB_Image, 687
 - FI_Shared_Image, 707
 - FI_Text_Buffer, 764
 - FI_Tiled_Image, 825
 - Selection & Clipboard functions, 231

- copy_cuts
 - FI_Input_, 538
- copy_label
 - FI_Widget, 936
- copy_offscreen
 - Drawing functions, 253
 - FI_GDI_Graphics_Driver, 453
 - FI_Graphics_Driver, 479
 - FI_Xlib_Graphics_Driver, 979
- copy_tooltip
 - FI_Widget, 936
- count
 - FI_Image, 526
 - FI_Native_File_Chooser, 603
- count_displayed_characters
 - FI_Text_Buffer, 764
- count_lines
 - FI_Text_Buffer, 764
 - FI_Text_Display, 784
- cp1133.h, 1362
- cp1251.h, 1363
- cp1255.h, 1364
- cp1256.h, 1366
- cp936ext.h, 1368
- current
 - FI_Group, 505
 - FI_Tooltip, 831
 - FI_Window, 963
- current_
 - FI_Window, 974
- cursor
 - FI_Window, 963
- cursor_color
 - FI_Input_, 538
 - FI_Text_Display, 784, 785
 - FI_Value_Input, 912
- cursor_style
 - FI_Text_Display, 785
- curve
 - FI_Graphics_Driver, 479
 - FI_PostScript_Graphics_Driver, 640
- custom_application_menu_items
 - FI_Mac_App_Menu, 563
- cut
 - FI_Input_, 539
- d
 - FI_Image, 526
- damage
 - FI_Widget, 936, 937
- data
 - FI_Browser, 350
 - FI_Image, 526
- deactivate
 - FI_Menu_Item, 588
 - FI_Tree_Item, 883
 - FI_Widget, 937
- decorated_h
 - FI_Window, 963
- decorated_w
 - FI_Window, 964
- default_atclose
 - Windows handling functions, 215
- default_callback
 - FI_Widget, 937
- default_cursor
 - FI_Window, 964
- default_icon
 - FI_Window, 964
- default_icons
 - FI_Window, 964
- default_xclass
 - FI_Window, 965
- deimage
 - FI_Widget, 938
- delay
 - FI_Tooltip, 831
- delete_widget
 - Safe widget deletion support functions, 275
- deleted
 - FI_Widget_Tracker, 956
- deleteEntry
 - FI_Preferences, 656
- deleteGroup
 - FI_Preferences, 656
- deleting
 - FI_Browser_, 369
- deparent
 - FI_Tree_Item, 883
 - FI_Tree_Item_Array, 898
- depth
 - FI_Tree_Item, 883
- desaturate
 - FI_Image, 526
 - FI_Pixmap, 621
 - FI_RGB_Image, 687
 - FI_Shared_Image, 708
 - FI_Tiled_Image, 825
- descent
 - FI_GDI_Graphics_Driver, 453
 - FI_Graphics_Driver, 480
 - FI_PostScript_Graphics_Driver, 640
 - FI_Quartz_Graphics_Driver, 675
 - FI_Xlib_Graphics_Driver, 979
- deselect
 - FI_Browser_, 369
 - FI_Tree, 848
- deselect_all
 - FI_Tree, 849
 - FI_Tree_Item, 883
- DIM_CURSOR
 - FI_Text_Display, 781
- dingbats_.h, 1268
- direction
 - FI_Timer, 827
- directory
 - FI_Native_File_Chooser, 603

- dirent.h, 989
- disable
 - FI_Tooltip, 832
- disable_im
 - Events handling functions, 220
- display
 - FI, 329
 - FI_Browser, 350
 - FI_Browser_, 370
 - FI_Tree, 849
- display_insert
 - FI_Text_Display, 785
- displayed
 - FI_Browser, 350
 - FI_Browser_, 370
 - FI_Tree, 850
- dnd
 - Selection & Clipboard functions, 232
- dnd_text_ops
 - FI, 329
- do_callback
 - FI_Menu_Item, 588
 - FI_Widget, 938, 939
- do_widget_deletion
 - Safe widget deletion support functions, 275
- down_box
 - FI_Button, 383, 384
 - FI_File_Input, 440
 - FI_Menu_, 570
- draw
 - FI_Adjuster, 338
 - FI_Bitmap, 340
 - FI_Box, 343
 - FI_Browser_, 370
 - FI_Button, 384
 - FI_Cairo_Window, 387
 - FI_Chart, 391
 - FI_Choice, 398
 - FI_Clock_Output, 404
 - FI_Copy_Surface, 413
 - FI_Counter, 415
 - FI_Dial, 421
 - FI_File_Icon, 436
 - FI_File_Input, 440
 - FI_FormsBitmap, 446
 - FI_FormsPixmap, 447
 - FI_FormsText, 448
 - FI_Free, 450
 - FI_GDI_Graphics_Driver, 453, 454
 - FI_GDI_Printer_Graphics_Driver, 457, 458
 - FI_Gl_Window, 463
 - FI_Glut_Window, 470
 - FI_Graphics_Driver, 480, 481
 - FI_Group, 505
 - FI_Help_View, 517
 - FI_Image, 526
 - FI_Image_Surface, 530
 - FI_Input, 533
 - FI_Label, 559
 - FI_Light_Button, 561
 - FI_Menu_Bar, 578
 - FI_Menu_Button, 581
 - FI_Pack, 611
 - FI_Pixmap, 621
 - FI_Positioner, 628
 - FI_PostScript_Graphics_Driver, 640, 641
 - FI_Progress, 672
 - FI_Quartz_Graphics_Driver, 675, 676
 - FI_Return_Button, 684
 - FI_RGB_Image, 688
 - FI_Roller, 690
 - FI_Scroll, 695
 - FI_Scrollbar, 699
 - FI_Shared_Image, 708
 - FI_Slider, 715
 - FI_Sys_Menu_Bar, 725
 - FI_Table, 741
 - FI_Tabs, 755
 - FI_Text_Display, 785
 - FI_Tiled_Image, 825
 - FI_Timer, 827
 - FI_Tree, 850
 - FI_Tree_Item, 883
 - FI_Value_Input, 912
 - FI_Value_Output, 915
 - FI_Value_Slider, 919
 - FI_Widget, 939
 - FI_Window, 965
 - FI_Xlib_Graphics_Driver, 979, 980
- draw_box_active
 - FI, 329
- draw_cell
 - FI_Table, 741
- draw_child
 - FI_Group, 505
- draw_children
 - FI_Group, 505
- draw_cursor
 - FI_Text_Display, 786
- draw_decorated_window
 - FI_Copy_Surface, 413
 - FI_Image_Surface, 530
- draw_empty
 - FI_Image, 527
- draw_horizontal_connector
 - FI_Tree_Item, 884
- draw_image
 - FI_GDI_Graphics_Driver, 454, 455
 - FI_Graphics_Driver, 481
 - FI_PostScript_Graphics_Driver, 641, 642
 - FI_Quartz_Graphics_Driver, 676, 677
 - FI_Xlib_Graphics_Driver, 980, 981
- draw_image_mono
 - FI_GDI_Graphics_Driver, 455
 - FI_Graphics_Driver, 482
 - FI_PostScript_Graphics_Driver, 642

- FI_Quartz_Graphics_Driver, 677
 - FI_Xlib_Graphics_Driver, 981
- draw_item_content
 - FI_Tree_Item, 884
- draw_label
 - FI_Widget, 940
- draw_line_numbers
 - FI_Text_Display, 786
- draw_overlay
 - FI_Glut_Window, 470
 - FI_Overlay_Window, 609
- draw_range
 - FI_Text_Display, 786
- draw_scaled
 - FI_GDI_Printer_Graphics_Driver, 458
 - FI_Graphics_Driver, 482
 - FI_PostScript_Graphics_Driver, 642
 - FI_Quartz_Graphics_Driver, 677
- draw_string
 - FI_Text_Display, 786
- draw_text
 - FI_Text_Display, 787
- draw_vertical_connector
 - FI_Tree_Item, 885
- draw_vline
 - FI_Text_Display, 787
- drawbgcolor
 - FI_Tree_Item, 886
- drawfgcolor
 - FI_Tree_Item, 886
- Drawing functions, 248
 - copy_offscreen, 253
 - fl_add_symbol, 253
 - fl_arc, 253, 254
 - fl_begin_complex_polygon, 254
 - fl_begin_offscreen, 255
 - fl_begin_points, 255
 - fl_can_do_alpha_blending, 255
 - FL_CAP_FLAT, 253
 - FL_CAP_ROUND, 253
 - FL_CAP_SQUARE, 253
 - fl_circle, 255
 - fl_clip, 252
 - fl_clip_box, 255
 - fl_clip_region, 256
 - fl_copy_offscreen, 256
 - fl_create_offscreen, 256
 - fl_cursor, 257
 - fl_curve, 257
 - FL_DASH, 253
 - FL_DASHDOT, 253
 - FL_DASHDOTDOT, 253
 - fl_delete_offscreen, 257
 - FL_DOT, 253
 - fl_draw, 257, 258
 - fl_draw_box, 258
 - fl_draw_image, 259
 - fl_draw_image_mono, 260
 - fl_draw_pixmap, 261
 - fl_draw_symbol, 261
 - fl_expand_text, 262
 - fl_frame, 262
 - fl_frame2, 262
 - fl_gap, 263
 - FL_JOIN_BEVEL, 253
 - FL_JOIN_MITER, 253
 - FL_JOIN_ROUND, 253
 - fl_line_style, 263
 - fl_measure, 263
 - fl_measure_pixmap, 264
 - fl_mult_matrix, 264
 - fl_not_clipped, 265
 - fl_old_shortcut, 265
 - fl_pie, 266
 - fl_polygon, 266
 - fl_pop_clip, 267
 - fl_push_clip, 267
 - fl_push_matrix, 267
 - fl_read_image, 267
 - fl_rect, 268
 - fl_rectf, 268
 - fl_reset_spot, 268
 - fl_rotate, 268
 - fl_scale, 268, 269
 - fl_scroll, 269
 - fl_set_spot, 269
 - fl_set_status, 269
 - fl_shortcut_label, 270
 - FL_SOLID, 253
 - fl_transform_dx, 271
 - fl_transform_dy, 271
 - fl_transform_x, 271
 - fl_transform_y, 271
 - fl_transformed_vertex, 272
 - fl_translate, 272
 - fl_vertex, 272
- drawtext
 - FI_Input_, 540
- empty_vlines
 - FI_Text_Display, 787
- enable
 - FI_Tooltip, 832
- enable_im
 - Events handling functions, 221
- enabled
 - FI_Tooltip, 832
- end
 - FI_Group, 506
 - FI_Text_Selection, 818
- end_complex_polygon
 - FI_Graphics_Driver, 482
 - FI_PostScript_Graphics_Driver, 643
- end_job
 - FI_Paged_Device, 614
 - FI_PostScript_File_Device, 630
 - FI_Printer, 667

- FI_System_Printer, 728
- end_line
 - FI_Graphics_Driver, 483
 - FI_PostScript_Graphics_Driver, 643
- end_loop
 - FI_Graphics_Driver, 483
 - FI_PostScript_Graphics_Driver, 643
- end_page
 - FI_Paged_Device, 614
 - FI_PostScript_File_Device, 631
 - FI_Printer, 667
 - FI_System_Printer, 728
- end_points
 - FI_Graphics_Driver, 483
 - FI_PostScript_Graphics_Driver, 643
- end_polygon
 - FI_Graphics_Driver, 483
 - FI_PostScript_Graphics_Driver, 643
- enter_area
 - FI_Tooltip, 832
- entries
 - FI_Preferences, 656
- entry
 - FI_Preferences, 656
- entryExists
 - FI_Preferences, 657
- Enumerations.H, 989, 1009
 - _FL_DIAMOND_DOWN_BOX, 1002
 - _FL_DIAMOND_UP_BOX, 1002
 - _FL_EMBOSSED_LABEL, 1007
 - _FL_ENGRAVED_LABEL, 1007
 - _FL_GLEAM_DOWN_BOX, 1002
 - _FL_GLEAM_DOWN_FRAME, 1002
 - _FL_GLEAM_ROUND_DOWN_BOX, 1002
 - _FL_GLEAM_ROUND_UP_BOX, 1002
 - _FL_GLEAM_THIN_DOWN_BOX, 1002
 - _FL_GLEAM_THIN_UP_BOX, 1002
 - _FL_GLEAM_UP_BOX, 1002
 - _FL_GLEAM_UP_FRAME, 1002
 - _FL_GTK_DOWN_BOX, 1002
 - _FL_GTK_DOWN_FRAME, 1002
 - _FL_GTK_ROUND_DOWN_BOX, 1002
 - _FL_GTK_ROUND_UP_BOX, 1002
 - _FL_GTK_THIN_DOWN_BOX, 1002
 - _FL_GTK_THIN_DOWN_FRAME, 1002
 - _FL_GTK_THIN_UP_BOX, 1002
 - _FL_GTK_THIN_UP_FRAME, 1002
 - _FL_GTK_UP_BOX, 1002
 - _FL_GTK_UP_FRAME, 1002
 - _FL_ICON_LABEL, 1007
 - _FL_IMAGE_LABEL, 1007
 - _FL_MULTI_LABEL, 1007
 - _FL_OFLAT_BOX, 1002
 - _FL_OSHADOW_BOX, 1002
 - _FL_OVAL_BOX, 1002
 - _FL_OVAL_FRAME, 1002
 - _FL_PLASTIC_DOWN_BOX, 1002
 - _FL_PLASTIC_DOWN_FRAME, 1002
 - _FL_PLASTIC_ROUND_DOWN_BOX, 1002
 - _FL_PLASTIC_ROUND_UP_BOX, 1002
 - _FL_PLASTIC_THIN_DOWN_BOX, 1002
 - _FL_PLASTIC_THIN_UP_BOX, 1002
 - _FL_PLASTIC_UP_BOX, 1002
 - _FL_PLASTIC_UP_FRAME, 1002
 - _FL_RFLAT_BOX, 1002
 - _FL_ROUNDED_BOX, 1002
 - _FL_ROUNDED_FRAME, 1002
 - _FL_ROUND_DOWN_BOX, 1002
 - _FL_ROUND_UP_BOX, 1002
 - _FL_RSHADOW_BOX, 1002
 - _FL_SHADOW_BOX, 1002
 - _FL_SHADOW_FRAME, 1002
 - _FL_SHADOW_LABEL, 1007
 - FL_ABI_VERSION, 999
 - FL_ACTIVATE, 1005
 - FL_ALIGN_LEFT, 1008
 - FL_ALIGN_TOP, 1008
 - FL_API_VERSION, 1000
 - FL_BORDER_BOX, 1002
 - FL_BORDER_FRAME, 1002
 - fl_box, 1007
 - FI_Boxtype, 1001
 - FL_CLOSE, 1005
 - fl_color_cube, 1007
 - FI_Cursor, 1003
 - FL_CURSOR_ARROW, 1003
 - FL_CURSOR_CROSS, 1003
 - FL_CURSOR_DEFAULT, 1003
 - FL_CURSOR_E, 1003
 - FL_CURSOR_HAND, 1003
 - FL_CURSOR_HELP, 1003
 - FL_CURSOR_INSERT, 1003
 - FL_CURSOR_MOVE, 1003
 - FL_CURSOR_N, 1003
 - FL_CURSOR_NE, 1003
 - FL_CURSOR_NESW, 1003
 - FL_CURSOR_NONE, 1003
 - FL_CURSOR_NS, 1003
 - FL_CURSOR_NW, 1003
 - FL_CURSOR_NWSE, 1003
 - FL_CURSOR_S, 1003
 - FL_CURSOR_SE, 1003
 - FL_CURSOR_SW, 1003
 - FL_CURSOR_W, 1003
 - FL_CURSOR_WAIT, 1003
 - FL_CURSOR_WE, 1003
 - FI_Damage, 1003
 - FL_DAMAGE_ALL, 1003
 - FL_DAMAGE_CHILD, 1003
 - FL_DAMAGE_EXPOSE, 1003
 - FL_DAMAGE_OVERLAY, 1003
 - FL_DAMAGE_SCROLL, 1003
 - FL_DAMAGE_USER1, 1003
 - FL_DAMAGE_USER2, 1003
 - FL_DEACTIVATE, 1005
 - FL_DND_DRAG, 1006

- FL_DND_ENTER, [1006](#)
- FL_DND_LEAVE, [1006](#)
- FL_DND_RELEASE, [1006](#)
- fl_down, [1008](#)
- FL_DOWN_BOX, [1001](#)
- FL_DOWN_FRAME, [1001](#)
- FL_DRAG, [1004](#)
- FL_EMBOSSED_BOX, [1002](#)
- FL_EMBOSSED_FRAME, [1002](#)
- FL_ENGRAVED_BOX, [1001](#)
- FL_ENGRAVED_FRAME, [1002](#)
- FL_ENTER, [1004](#)
- Fl_Event, [1003](#)
- FL_EXCEPT, [1001](#)
- FL_FLAT_BOX, [1001](#)
- FL_FOCUS, [1004](#)
- Fl_Fontsize, [1001](#)
- fl_frame, [1008](#)
- FL_FREE_BOXTYPE, [1002](#)
- FL_FREE_LABELTYPE, [1007](#)
- FL_FULLSCREEN, [1006](#)
- fl_gray_ramp, [1008](#)
- FL_HIDE, [1006](#)
- FL_KEYBOARD, [1005](#)
- FL_KEYDOWN, [1005](#)
- FL_KEYUP, [1005](#)
- Fl_Labeltype, [1006](#)
- FL_LEAVE, [1004](#)
- FL_MAJOR_VERSION, [1000](#)
- FL_MINOR_VERSION, [1000](#)
- FL_MOUSEWHEEL, [1006](#)
- FL_MOVE, [1005](#)
- FL_NO_BOX, [1001](#)
- FL_NO_EVENT, [1004](#)
- FL_NO_LABEL, [1007](#)
- FL_NORMAL_LABEL, [1007](#)
- FL_NORMAL_SIZE, [1008](#)
- FL_PASTE, [1006](#)
- FL_PATCH_VERSION, [1000](#)
- FL_PUSH, [1004](#)
- FL_READ, [1001](#)
- FL_RELEASE, [1004](#)
- FL_SCREEN_CONFIGURATION_CHANGED, [1006](#)
- FL_SELECTIONCLEAR, [1006](#)
- FL_SHORTCUT, [1005](#)
- FL_SHOW, [1006](#)
- FL_THIN_DOWN_BOX, [1001](#)
- FL_THIN_DOWN_FRAME, [1001](#)
- FL_THIN_UP_BOX, [1001](#)
- FL_THIN_UP_FRAME, [1001](#)
- FL_UNFOCUS, [1005](#)
- FL_UP_BOX, [1001](#)
- FL_UP_FRAME, [1001](#)
- FL_VERSION, [1000](#)
- Fl_When, [1007](#)
- FL_WHEN_CHANGED, [1007](#)
- FL_WHEN_ENTER_KEY, [1007](#)
- FL_WHEN_ENTER_KEY_ALWAYS, [1007](#)
- FL_WHEN_ENTER_KEY_CHANGED, [1007](#)
- FL_WHEN_NEVER, [1007](#)
- FL_WHEN_NOT_CHANGED, [1007](#)
- FL_WHEN_RELEASE, [1007](#)
- FL_WHEN_RELEASE_ALWAYS, [1007](#)
- FL_WRITE, [1001](#)
- FL_ZOOM_GESTURE, [1006](#)
- errmsg
 - Fl_Native_File_Chooser, [603](#)
- error
 - Common Dialogs classes and functions, [305](#)
- errorcolor
 - Fl_File_Input, [441](#)
- ERRORS_TO_CP1252
 - Unicode and UTF-8 functions, [280](#)
- ERRORS_TO_ISO8859_1
 - Unicode and UTF-8 functions, [280](#)
- event
 - Events handling functions, [221](#)
- event_button
 - Events handling functions, [221](#)
- event_button1
 - Events handling functions, [221](#)
- event_button2
 - Events handling functions, [221](#)
- event_button3
 - Events handling functions, [221](#)
- event_buttons
 - Events handling functions, [222](#)
- event_clicks
 - Events handling functions, [222](#)
- event_clipboard
 - Events handling functions, [222](#)
- event_clipboard_type
 - Events handling functions, [222](#)
- event_dispatch
 - Events handling functions, [222](#)
- event_dx
 - Events handling functions, [223](#)
- event_dy
 - Events handling functions, [223](#)
- event_inside
 - Events handling functions, [223](#), [224](#)
- event_is_click
 - Events handling functions, [224](#)
- event_key
 - Events handling functions, [224](#), [225](#)
- event_length
 - Events handling functions, [225](#)
- event_original_key
 - Events handling functions, [225](#)
- event_state
 - Events handling functions, [225](#), [226](#)
- event_text
 - Events handling functions, [226](#)
- event_x_root
 - Events handling functions, [226](#)

- event_y_root
 - Events handling functions, 226
- Events handling functions, 216
 - add_handler, 219
 - add_system_handler, 219
 - belowmouse, 220
 - compose, 220
 - compose_reset, 220
 - disable_im, 220
 - enable_im, 221
 - event, 221
 - event_button, 221
 - event_button1, 221
 - event_button2, 221
 - event_button3, 221
 - event_buttons, 222
 - event_clicks, 222
 - event_clipboard, 222
 - event_clipboard_type, 222
 - event_dispatch, 222
 - event_dx, 223
 - event_dy, 223
 - event_inside, 223, 224
 - event_is_click, 224
 - event_key, 224, 225
 - event_length, 225
 - event_original_key, 225
 - event_state, 225, 226
 - event_text, 226
 - event_x_root, 226
 - event_y_root, 226
 - fl_eventnames, 229
 - fl_fontnames, 229
 - focus, 227
 - get_key, 227
 - get_mouse, 227
 - handle, 227
 - handle_, 228
 - pushed, 228
 - remove_handler, 229
 - remove_system_handler, 229
 - test_shortcut, 229
- exists
 - FI_Widget_Tracker, 956
- extend_range_for_styles
 - FI_Text_Display, 788
- extend_selection
 - FI_Tree, 850
- extend_selection_dir
 - FI_Tree, 851
- fail
 - FI_Image, 527
- fastarrow.h, 1229
- fatal
 - Common Dialogs classes and functions, 305
- File names and URI utility functions, 306
 - fl_decode_uri, 307
 - FI_File_Sort_F, 306
 - fl_filename_absolute, 307
 - fl_filename_expand, 307
 - fl_filename_ext, 309
 - fl_filename_free_list, 309
 - fl_filename_isdir, 309
 - fl_filename_list, 310
 - fl_filename_match, 310
 - fl_filename_name, 311
 - fl_filename_relative, 311
 - fl_filename_setext, 312
 - fl_open_uri, 312
- file_encoding_warning_message
 - FI_Text_Buffer, 772
- filename
 - FI_Native_File_Chooser, 603
- filename.H, 1016, 1017
- filetype
 - FI_File_Browser, 427
- filter
 - FI_File_Browser, 427
 - FI_File_Chooser, 433
 - FI_Native_File_Chooser, 604
- filter_value
 - FI_Native_File_Chooser, 604
- find
 - FI_File_Icon, 437
 - FI_Group, 506
 - FI_Help_View, 518
 - FI_Shared_Image, 708
- find_child
 - FI_Tree_Item, 886
- find_child_item
 - FI_Tree_Item, 886, 887
- find_clicked
 - FI_Tree, 851
 - FI_Tree_Item, 887
- find_index
 - FI_Menu_, 570, 571
- find_item
 - FI_Browser_, 370
 - FI_Menu_, 571, 572
 - FI_Tree, 852
 - FI_Tree_Item, 887
- find_line
 - FI_Browser, 351
- find_line_end
 - FI_Text_Display, 788
- find_shortcut
 - FI_Menu_Item, 589
- find_wrap_range
 - FI_Text_Display, 788
- find_x
 - FI_Text_Display, 789
- findchar_backward
 - FI_Text_Buffer, 764
- findchar_forward
 - FI_Text_Buffer, 765
- first

- FI_Tree, [852](#)
- first_selected_item
 - FI_Tree, [852](#)
- first_visible
 - FI_Tree, [853](#)
- first_visible_item
 - FI_Tree, [853](#)
- first_window
 - Windows handling functions, [215](#)
- Fl, [315](#)
 - abi_check, [324](#)
 - abi_version, [324](#)
 - add_check, [324](#)
 - add_fd, [324](#)
 - add_idle, [325](#)
 - add_timeout, [325](#)
 - api_version, [325](#)
 - arg, [325](#)
 - args, [326](#)
 - background, [327](#)
 - background2, [327](#)
 - box_color, [327](#)
 - box_dh, [328](#)
 - box_dw, [328](#)
 - box_dx, [328](#)
 - box_dy, [328](#)
 - check, [328](#)
 - display, [329](#)
 - dnd_text_ops, [329](#)
 - draw_box_active, [329](#)
 - Fl_Option, [323](#)
 - flush, [329](#)
 - get_system_colors, [329](#)
 - gl_visual, [330](#)
 - help, [336](#)
 - idle, [336](#)
 - is_scheme, [330](#)
 - option, [330](#), [331](#)
 - OPTION_ARROW_FOCUS, [323](#)
 - OPTION_DND_TEXT, [323](#)
 - OPTION_FNFC_USES_GTK, [324](#)
 - OPTION_LAST, [324](#)
 - OPTION_SHOW_TOOLTIPS, [323](#)
 - OPTION_VISIBLE_FOCUS, [323](#)
 - own_colormap, [331](#)
 - readqueue, [331](#)
 - ready, [332](#)
 - release, [332](#)
 - reload_scheme, [332](#)
 - remove_check, [332](#)
 - remove_timeout, [332](#)
 - repeat_timeout, [333](#)
 - run, [333](#)
 - scheme, [333](#)
 - scrollbar_size, [333](#), [334](#)
 - set_box_color, [334](#)
 - set_idle, [334](#)
 - use_high_res_GL, [334](#)
 - version, [335](#)
 - visible_focus, [335](#)
 - visual, [335](#)
 - wait, [335](#)
- Fl.H, [1019](#), [1020](#)
- FL_ABI_VERSION
 - Enumerations.H, [999](#)
- fl_access
 - Unicode and UTF-8 functions, [280](#)
- FL_ACTIVATE
 - Enumerations.H, [1005](#)
- fl_add_symbol
 - Drawing functions, [253](#)
- Fl_Adjuster, [337](#)
 - draw, [338](#)
 - Fl_Adjuster, [338](#)
 - handle, [338](#)
 - soft, [339](#)
 - value_damage, [339](#)
- Fl_Adjuster.H, [1026](#)
- fl_alert
 - Common Dialogs classes and functions, [295](#)
- FL_ALIGN_LEFT
 - Enumerations.H, [1008](#)
- FL_ALIGN_TOP
 - Enumerations.H, [1008](#)
- FL_API_VERSION
 - Enumerations.H, [1000](#)
- fl_arc
 - Drawing functions, [253](#), [254](#)
 - Fl_Graphics_Driver, [489](#)
- fl_arc.cxx, [1229](#)
- fl_arci.cxx, [1229](#)
- fl_ask
 - Common Dialogs classes and functions, [295](#)
- fl_ask.cxx, [1229](#)
- fl_ask.H, [1026](#), [1028](#)
 - Fl_Beep, [1028](#)
 - FL_BEEP_DEFAULT, [1028](#)
 - FL_BEEP_ERROR, [1028](#)
 - FL_BEEP_MESSAGE, [1028](#)
 - FL_BEEP_NOTIFICATION, [1028](#)
 - FL_BEEP_PASSWORD, [1028](#)
 - FL_BEEP_QUESTION, [1028](#)
- Fl_Beep
 - fl_ask.H, [1028](#)
- fl_beep
 - Common Dialogs classes and functions, [296](#)
- FL_BEEP_DEFAULT
 - fl_ask.H, [1028](#)
- FL_BEEP_ERROR
 - fl_ask.H, [1028](#)
- FL_BEEP_MESSAGE
 - fl_ask.H, [1028](#)
- FL_BEEP_NOTIFICATION
 - fl_ask.H, [1028](#)
- FL_BEEP_PASSWORD
 - fl_ask.H, [1028](#)

- FL_BEEP_QUESTION
 - fl_ask.H, 1028
- fl_begin_complex_polygon
 - Drawing functions, 254
 - Fl_Graphics_Driver, 490
- fl_begin_offscreen
 - Drawing functions, 255
- fl_begin_points
 - Drawing functions, 255
 - Fl_Graphics_Driver, 490
- Fl_Bitmap, 339
 - copy, 340
 - draw, 340
 - label, 341
 - uncache, 341
- Fl_Bitmap.H, 1029
- Fl_BMP_Image, 341
 - Fl_BMP_Image, 342
- Fl_BMP_Image.H, 1030
- FL_BORDER_BOX
 - Enumerations.H, 1002
- FL_BORDER_FRAME
 - Enumerations.H, 1002
- Fl_Box, 342
 - draw, 343
 - Fl_Box, 343
 - handle, 343
- fl_box
 - Enumerations.H, 1007
- Fl_Box.H, 1030
- Fl_Boxtype
 - Enumerations.H, 1001
- fl_boxtype.cxx, 1231
 - fl_internal_boxtype, 1232
 - fl_rectbound, 1232
- Fl_Browser, 344
 - _remove, 348
 - add, 348
 - bottomline, 348
 - clear, 349
 - column_char, 349
 - column_widths, 349
 - data, 350
 - display, 350
 - displayed, 350
 - find_line, 351
 - Fl_Browser, 347
 - format_char, 351, 352
 - full_height, 352
 - hide, 352
 - icon, 353
 - incr_height, 353
 - insert, 354
 - item_at, 354
 - item_draw, 355
 - item_first, 355
 - item_height, 355
 - item_last, 356
 - item_next, 356
 - item_prev, 356
 - item_select, 357
 - item_selected, 357
 - item_swap, 357
 - item_text, 358
 - item_width, 358
 - lineno, 358
 - lineposition, 359
 - load, 359
 - make_visible, 359
 - middleline, 360
 - move, 360
 - remove, 360
 - remove_icon, 361
 - select, 361
 - selected, 361
 - show, 361, 362
 - size, 362
 - swap, 362
 - text, 363
 - textsize, 363
 - topline, 363, 364
 - value, 364
 - visible, 364
- Fl_Browser.H, 1031
- Fl_Browser_, 365
 - ALWAYS_ON, 368
 - bbox, 369
 - BOTH, 368
 - BOTH_ALWAYS, 368
 - deleting, 369
 - deselect, 369
 - display, 370
 - displayed, 370
 - draw, 370
 - find_item, 370
 - Fl_Browser_, 369
 - full_height, 371
 - full_width, 371
 - handle, 371
 - has_scrollbar, 371
 - HORIZONTAL, 368
 - HORIZONTAL_ALWAYS, 368
 - hposition, 372
 - hscrollbar, 381
 - incr_height, 372
 - inserting, 372
 - item_at, 373
 - item_draw, 373
 - item_first, 373
 - item_height, 373
 - item_last, 374
 - item_next, 374
 - item_prev, 374
 - item_quick_height, 374
 - item_select, 375
 - item_selected, 375

- item_swap, 375
- item_text, 375
- item_width, 376
- leftedge, 376
- new_list, 376
- position, 376, 377
- redraw_line, 377
- redraw_lines, 377
- replacing, 377
- resize, 378
- scrollbar, 381
- scrollbar_left, 378
- scrollbar_right, 378
- scrollbar_size, 378
- scrollbar_width, 379
- select, 379
- select_only, 380
- selection, 380
- sort, 380
- swapping, 380
- textfont, 381
- VERTICAL, 368
- VERTICAL_ALWAYS, 368
- Fl_Browser_.H, 1033
- Fl_Button, 381
 - clear, 383
 - down_box, 383, 384
 - draw, 384
 - Fl_Button, 383
 - handle, 384
 - set, 385
 - shortcut, 385
 - value, 385
- Fl_Button.H, 1035
- Fl_Cairo.H, 1036
- Fl_Cairo_State, 386
 - cc, 386
- Fl_Cairo_Window, 387
 - draw, 387
 - set_draw_cb, 388
- Fl_Cairo_Window.H, 1037
- fl_can_do_alpha_blending
 - Drawing functions, 255
- FL_CAP_FLAT
 - Drawing functions, 253
- FL_CAP_ROUND
 - Drawing functions, 253
- FL_CAP_SQUARE
 - Drawing functions, 253
- Fl_Chart, 388
 - add, 390
 - autosize, 390
 - bounds, 391
 - draw, 391
 - Fl_Chart, 390
 - insert, 391
 - maxsize, 392
 - replace, 392
- Fl_Chart.H, 1037
- FL_CHART_ENTRY, 392
- Fl_Check_Browser, 393
 - add, 394
 - handle, 394
 - nchecked, 395
 - nitems, 395
 - remove, 395
 - set_checked, 395
- Fl_Check_Browser.H, 1038
- Fl_Check_Button, 395
 - Fl_Check_Button, 396
- Fl_Check_Button.H, 1040
- fl_chmod
 - Unicode and UTF-8 functions, 281
- Fl_Choice, 396
 - draw, 398
 - Fl_Choice, 398
 - handle, 398
 - value, 399
- fl_choice
 - Common Dialogs classes and functions, 296
- Fl_Choice.H, 1040
- fl_choice_n
 - Common Dialogs classes and functions, 297
- fl_circle
 - Drawing functions, 255
 - Fl_Graphics_Driver, 490
- fl_clip
 - Drawing functions, 252
- fl_clip_box
 - Drawing functions, 255
 - Fl_Graphics_Driver, 490
- fl_clip_region
 - Drawing functions, 256
 - Fl_Graphics_Driver, 491
- Fl_Clock, 400
 - Fl_Clock, 401
 - handle, 401
- Fl_Clock.H, 1041
- Fl_Clock_Output, 402
 - draw, 404
 - Fl_Clock_Output, 403
 - hour, 404
 - minute, 404
 - second, 404
 - value, 405
- FL_CLOSE
 - Enumerations.H, 1005
- fl_cmap.h, 1232
- fl_color
 - Color & Font functions, 240
 - Fl_Graphics_Driver, 491
- fl_color.cxx, 1236
- fl_color_average
 - Color & Font functions, 241
- Fl_Color_Chooser, 406
 - b, 408

- FL_Color_Chooser, [407](#)
 - g, [408](#)
 - hsv, [408](#)
 - hsv2rgb, [408](#)
 - hue, [408](#)
 - mode, [408](#), [409](#)
 - r, [409](#)
 - rgb, [409](#)
 - rgb2hsv, [409](#)
 - saturation, [411](#)
 - value, [411](#)
 - fl_color_chooser
 - Common Dialogs classes and functions, [298](#), [299](#)
 - FL_Color_Chooser.H, [1042](#)
 - fl_color_cube
 - Enumerations.H, [1007](#)
 - FL_compose.cxx, [1237](#)
 - fl_contrast
 - Color & Font functions, [241](#)
 - fl_copy_offscreen
 - Drawing functions, [256](#)
 - FL_Graphics_Driver, [491](#)
 - FL_Copy_Surface, [411](#)
 - class_name, [412](#)
 - draw, [413](#)
 - draw_decorated_window, [413](#)
 - FL_Copy_Surface, [412](#)
 - set_current, [413](#)
 - FL_Copy_Surface.H, [1043](#)
 - FL_Counter, [413](#)
 - draw, [415](#)
 - FL_Counter, [415](#)
 - handle, [415](#)
 - lstep, [416](#)
 - step, [416](#)
 - FL_Counter.H, [1045](#)
 - fl_create_offscreen
 - Drawing functions, [256](#)
 - FL_CString
 - fl_types.h, [1176](#)
 - FL_Cursor
 - Enumerations.H, [1003](#)
 - fl_cursor
 - Drawing functions, [257](#)
 - FL_CURSOR_ARROW
 - Enumerations.H, [1003](#)
 - FL_CURSOR_CROSS
 - Enumerations.H, [1003](#)
 - FL_CURSOR_DEFAULT
 - Enumerations.H, [1003](#)
 - FL_CURSOR_E
 - Enumerations.H, [1003](#)
 - FL_CURSOR_HAND
 - Enumerations.H, [1003](#)
 - FL_CURSOR_HELP
 - Enumerations.H, [1003](#)
 - FL_CURSOR_INSERT
 - Enumerations.H, [1003](#)
 - FL_CURSOR_MOVE
 - Enumerations.H, [1003](#)
 - FL_CURSOR_N
 - Enumerations.H, [1003](#)
 - FL_CURSOR_NE
 - Enumerations.H, [1003](#)
 - FL_CURSOR_NESW
 - Enumerations.H, [1003](#)
 - FL_CURSOR_NONE
 - Enumerations.H, [1003](#)
 - FL_CURSOR_NS
 - Enumerations.H, [1003](#)
 - FL_CURSOR_NW
 - Enumerations.H, [1003](#)
 - FL_CURSOR_NWSE
 - Enumerations.H, [1003](#)
 - FL_CURSOR_S
 - Enumerations.H, [1003](#)
 - FL_CURSOR_SE
 - Enumerations.H, [1003](#)
 - FL_CURSOR_SW
 - Enumerations.H, [1003](#)
 - FL_CURSOR_W
 - Enumerations.H, [1003](#)
 - FL_CURSOR_WAIT
 - Enumerations.H, [1003](#)
 - FL_CURSOR_WE
 - Enumerations.H, [1003](#)
- fl_curve
 - Drawing functions, [257](#)
 - FL_Graphics_Driver, [492](#)
 - fl_curve.cxx, [1237](#)
 - FL_Damage
 - Enumerations.H, [1003](#)
 - FL_DAMAGE_ALL
 - Enumerations.H, [1003](#)
 - FL_DAMAGE_CHILD
 - Enumerations.H, [1003](#)
 - FL_DAMAGE_EXPOSE
 - Enumerations.H, [1003](#)
 - FL_DAMAGE_OVERLAY
 - Enumerations.H, [1003](#)
 - FL_DAMAGE_SCROLL
 - Enumerations.H, [1003](#)
 - FL_DAMAGE_USER1
 - Enumerations.H, [1003](#)
 - FL_DAMAGE_USER2
 - Enumerations.H, [1003](#)
 - FL_DASH
 - Drawing functions, [253](#)
 - FL_DASHDOT
 - Drawing functions, [253](#)
 - FL_DASHDOTDOT
 - Drawing functions, [253](#)
 - FL_DEACTIVATE
 - Enumerations.H, [1005](#)
 - fl_decode_uri
 - File names and URI utility functions, [307](#)

- fl_delete_offscreen
 - Drawing functions, 257
- Fl_Device, 417
 - ~Fl_Device, 417
 - class_id, 418
 - class_name, 418
- Fl_Device.H, 1046, 1047
 - Fl_Draw_Image_Cb, 1047
- Fl_Device_Plugin, 418
 - print, 419
 - rectangle_capture, 419
- Fl_Dial, 419
 - angle1, 421
 - draw, 421
 - Fl_Dial, 421
 - handle, 421, 422
- Fl_Dial.H, 1052
- fl_dir_chooser
 - Common Dialogs classes and functions, 300
- Fl_Display_Device, 422
 - class_name, 423
- FL_DND_DRAG
 - Enumerations.H, 1006
- FL_DND_ENTER
 - Enumerations.H, 1006
- FL_DND_LEAVE
 - Enumerations.H, 1006
- FL_DND_RELEASE
 - Enumerations.H, 1006
- FL_DOT
 - Drawing functions, 253
- Fl_Double_Window, 423
 - ~Fl_Double_Window, 424
 - flush, 424
 - hide, 424
 - resize, 425
 - show, 425
- Fl_Double_Window.cxx, 1237
- Fl_Double_Window.H, 1053
- fl_down
 - Enumerations.H, 1008
- FL_DOWN_BOX
 - Enumerations.H, 1001
- FL_DOWN_FRAME
 - Enumerations.H, 1001
- FL_DRAG
 - Enumerations.H, 1004
- fl_draw
 - Drawing functions, 257, 258
 - Fl_Graphics_Driver, 492
- fl_draw.H, 1053, 1059
- fl_draw_box
 - Drawing functions, 258
- fl_draw_image
 - Drawing functions, 259
 - Fl_Graphics_Driver, 493
- Fl_Draw_Image_Cb
 - Fl_Device.H, 1047
- fl_draw_image_mono
 - Drawing functions, 260
 - Fl_Graphics_Driver, 494
- fl_draw_pixmap
 - Drawing functions, 261
- fl_draw_symbol
 - Drawing functions, 261
- FL_EMBOSSSED_BOX
 - Enumerations.H, 1002
- FL_EMBOSSSED_FRAME
 - Enumerations.H, 1002
- Fl_End, 425
- FL_ENGRAVED_BOX
 - Enumerations.H, 1001
- FL_ENGRAVED_FRAME
 - Enumerations.H, 1002
- FL_ENTER
 - Enumerations.H, 1004
- Fl_Event
 - Enumerations.H, 1003
- Fl_Event_Dispatch
 - Callback function typedefs, 214
- fl_eventnames
 - Events handling functions, 229
- FL_EXCEPT
 - Enumerations.H, 1001
- fl_expand_text
 - Drawing functions, 262
- Fl_Export.H, 1062
- Fl_File_Browser, 426
 - filetype, 427
 - filter, 427
 - Fl_File_Browser, 427
 - iconsize, 427, 428
 - load, 428
- Fl_File_Browser.H, 1062
- Fl_File_Chooser, 428
 - add_extra, 432
 - filter, 433
 - Fl_File_Chooser, 432
 - iconsize, 433
 - preview, 433
 - showHiddenButton, 433
 - value, 433
- fl_file_chooser
 - Common Dialogs classes and functions, 301
- Fl_File_Chooser.H, 1063
- fl_file_chooser_callback
 - Common Dialogs classes and functions, 302
- fl_file_chooser_ok_label
 - Common Dialogs classes and functions, 302
- Fl_File_Icon, 434
 - add, 435
 - add_color, 436
 - add_vertex, 436
 - draw, 436
 - find, 437
 - Fl_File_Icon, 435

- label, [437](#)
- labeltype, [437](#)
- load, [437](#)
- load_fti, [438](#)
- load_image, [438](#)
- load_system_icons, [438](#)
- next, [438](#)
- type, [438](#)
- Fl_File_Icon.H, [1065](#)
- Fl_File_Input, [439](#)
 - down_box, [440](#)
 - draw, [440](#)
 - errorcolor, [441](#)
 - Fl_File_Input, [440](#)
 - handle, [441](#)
 - value, [441](#)
- Fl_File_Input.H, [1067](#)
- Fl_File_Sort_F
 - File names and URI utility functions, [306](#)
- fl_filename_absolute
 - File names and URI utility functions, [307](#)
- fl_filename_expand
 - File names and URI utility functions, [307](#)
- fl_filename_ext
 - File names and URI utility functions, [309](#)
- fl_filename_free_list
 - File names and URI utility functions, [309](#)
- fl_filename_isdir
 - File names and URI utility functions, [309](#)
- fl_filename_list
 - File names and URI utility functions, [310](#)
- fl_filename_match
 - File names and URI utility functions, [310](#)
- fl_filename_name
 - File names and URI utility functions, [311](#)
- fl_filename_relative
 - File names and URI utility functions, [311](#)
- fl_filename_setext
 - File names and URI utility functions, [312](#)
- Fl_Fill_Dial, [442](#)
- Fl_Fill_Dial.H, [1068](#)
- Fl_Fill_Slider, [442](#)
- Fl_Fill_Slider.H, [1068](#)
- FL_FLAT_BOX
 - Enumerations.H, [1001](#)
- Fl_Float_Input, [443](#)
 - Fl_Float_Input, [443](#)
- Fl_Float_Input.H, [1069](#)
- Fl_FLTK_File_Chooser, [444](#)
- FL_FOCUS
 - Enumerations.H, [1004](#)
- fl_font
 - Color & Font functions, [241](#)
 - Fl_Graphics_Driver, [494](#)
- Fl_Font.H, [1238](#)
- Fl_Font_Descriptor, [445](#)
- Fl_Fontdesc, [445](#)
- fl_fontnames
 - Events handling functions, [229](#)
- Fl_Fontsize
 - Enumerations.H, [1001](#)
- fl_fopen
 - Unicode and UTF-8 functions, [281](#)
- Fl_FormsBitmap, [445](#)
 - draw, [446](#)
 - set, [446](#)
- Fl_FormsBitmap.H, [1069](#)
- Fl_FormsPixmap, [446](#)
 - draw, [447](#)
 - Fl_FormsPixmap, [447](#)
 - Pixmap, [448](#)
 - set, [448](#)
- Fl_FormsPixmap.H, [1070](#)
- Fl_FormsText, [448](#)
 - draw, [448](#)
- fl_frame
 - Drawing functions, [262](#)
 - Enumerations.H, [1008](#)
- fl_frame2
 - Drawing functions, [262](#)
- Fl_Free, [449](#)
 - draw, [450](#)
 - Fl_Free, [450](#)
 - handle, [450](#)
- Fl_Free.H, [1070](#)
- FL_FREE_BOXTYPE
 - Enumerations.H, [1002](#)
- FL_FREE_LABELTYPE
 - Enumerations.H, [1007](#)
- FL_FULLSCREEN
 - Enumerations.H, [1006](#)
- fl_gap
 - Drawing functions, [263](#)
 - Fl_Graphics_Driver, [495](#)
- Fl_GDI_Graphics_Driver, [451](#)
 - class_name, [452](#)
 - color, [452](#), [453](#)
 - copy_offscreen, [453](#)
 - descent, [453](#)
 - draw, [453](#), [454](#)
 - draw_image, [454](#), [455](#)
 - draw_image_mono, [455](#)
 - font, [455](#)
 - height, [455](#)
 - rtl_draw, [456](#)
 - text_extents, [456](#)
 - width, [456](#)
- Fl_GDI_Printer_Graphics_Driver, [456](#)
 - class_name, [457](#)
 - draw, [457](#), [458](#)
 - draw_scaled, [458](#)
- fl_getcwd
 - Unicode and UTF-8 functions, [282](#)
- fl_getenv
 - Unicode and UTF-8 functions, [282](#)
- Fl_GIF_Image, [458](#)

- FI_GIF_Image, 459
- FI_GIF_Image.H, 1071
- FI_Gl_Choice, 459
- FI_Gl_Choice.H, 1239
- FI_Gl_Window, 459
 - as_gl_window, 462
 - can_do, 462
 - can_do_overlay, 462
 - context, 463
 - context_valid, 463
 - draw, 463
 - FI_Gl_Window, 462
 - flush, 463
 - handle, 463
 - hide, 464
 - make_current, 464
 - make_overlay_current, 464
 - mode, 464
 - ortho, 465
 - pixel_h, 465
 - pixel_w, 465
 - pixels_per_unit, 466
 - redraw_overlay, 466
 - resize, 466
 - show, 466
 - swap_buffers, 467
 - valid, 467
- FI_Gl_Window.H, 1071
- FI_Glut_Bitmap_Font, 467
- FI_Glut_StrokeChar, 468
- FI_Glut_StrokeFont, 468
- FI_Glut_StrokeStrip, 468
- FI_Glut_StrokeVertex, 468
- FI_Glut_Window, 469
 - draw, 470
 - draw_overlay, 470
 - handle, 470
- FI_Graphics_Driver, 470
 - arc, 477
 - begin_complex_polygon, 478
 - begin_line, 478
 - begin_loop, 478
 - begin_points, 478
 - begin_polygon, 478
 - circle, 478
 - class_name, 478
 - clip_box, 479
 - color, 479
 - copy_offscreen, 479
 - curve, 479
 - descent, 480
 - draw, 480, 481
 - draw_image, 481
 - draw_image_mono, 482
 - draw_scaled, 482
 - end_complex_polygon, 482
 - end_line, 483
 - end_loop, 483
 - end_points, 483
 - end_polygon, 483
 - fl_arc, 489
 - fl_begin_complex_polygon, 490
 - fl_begin_points, 490
 - fl_circle, 490
 - fl_clip_box, 490
 - fl_clip_region, 491
 - fl_color, 491
 - fl_copy_offscreen, 491
 - fl_curve, 492
 - fl_draw, 492
 - fl_draw_image, 493
 - fl_draw_image_mono, 494
 - fl_font, 494
 - fl_gap, 495
 - fl_line_style, 495
 - fl_mult_matrix, 495
 - fl_not_clipped, 496
 - fl_pie, 496
 - fl_polygon, 496, 497
 - fl_pop_clip, 497
 - fl_push_clip, 497
 - fl_push_matrix, 497
 - fl_rect, 497
 - fl_rotate, 497
 - fl_scale, 498
 - fl_transform_dx, 498
 - fl_transform_dy, 498
 - fl_transform_x, 499
 - fl_transform_y, 499
 - fl_transformed_vertex, 499
 - fl_translate, 499
 - fl_vertex, 499
 - font, 483
 - gap, 483
 - height, 483
 - line, 483, 484
 - line_style, 484
 - loop, 484
 - not_clipped, 485
 - pie, 485
 - point, 485
 - polygon, 485
 - pop_clip, 486
 - push_clip, 486
 - push_no_clip, 486
 - rect, 486
 - rectf, 486
 - rtl_draw, 486
 - text_extents, 487
 - transformed_vertex, 487
 - vertex, 487
 - width, 487
 - xyline, 487, 488
 - yxline, 488
- FI_Graphics_Driver::matrix, 984
- fl_gray_ramp

- Enumerations.H, 1008
- Fl_Group, 500
 - ~Fl_Group, 503
 - array, 503
 - as_group, 504
 - begin, 504
 - child, 504
 - clear, 504
 - clip_children, 504, 505
 - current, 505
 - draw, 505
 - draw_child, 505
 - draw_children, 505
 - end, 506
 - find, 506
 - Fl_Group, 503
 - focus, 506
 - handle, 506
 - init_sizes, 507
 - insert, 507
 - remove, 507
 - resizable, 508
 - resize, 508
 - sizes, 509
 - update_child, 509
- Fl_Group.H, 1073
- Fl_GTK_File_Chooser, 510
- fl_height
 - Color & Font functions, 242
- Fl_Help_Block, 510
- Fl_Help_Dialog, 510
 - load, 511
 - show, 512
 - textsize, 512
 - value, 512
- Fl_Help_Dialog.H, 1074
- Fl_Help_Font_Stack, 512
- Fl_Help_Font_Style, 513
- Fl_Help_Link, 513
- Fl_Help_Target, 514
- Fl_Help_View, 514
 - ~Fl_Help_View, 517
 - draw, 517
 - find, 518
 - handle, 518
 - leftline, 518
 - link, 518
 - load, 518
 - resize, 519
 - scrollbar_size, 519
 - topline, 519, 520
 - value, 520
- Fl_Help_View.H, 1075
- FL_HIDE
 - Enumerations.H, 1006
- Fl_Hold_Browser, 520
 - Fl_Hold_Browser, 521
- Fl_Hold_Browser.H, 1078
- Fl_Hor_Fill_Slider, 521
- Fl_Hor_Fill_Slider.H, 1079
- Fl_Hor_Nice_Slider, 522
- Fl_Hor_Nice_Slider.H, 1079
- Fl_Hor_Slider, 522
- Fl_Hor_Slider.H, 1080
- Fl_Hor_Value_Slider, 523
- Fl_Hor_Value_Slider.H, 1080
- Fl_Image, 523
 - color_average, 525
 - copy, 525, 526
 - count, 526
 - d, 526
 - data, 526
 - desaturate, 526
 - draw, 526
 - draw_empty, 527
 - fail, 527
 - Fl_Image, 525
 - inactive, 527
 - label, 527
 - ld, 528
 - RGB_scaling, 528
 - uncache, 528
- Fl_Image.H, 1081
 - Fl_RGB_Scaling, 1081
 - FL_RGB_SCALING_BILINEAR, 1081
 - FL_RGB_SCALING_NEAREST, 1081
- Fl_Image_Surface, 528
 - class_name, 530
 - draw, 530
 - draw_decorated_window, 530
 - Fl_Image_Surface, 529
 - highres_image, 531
 - image, 531
 - set_current, 531
- Fl_Image_Surface.H, 1083
- Fl_Input, 531
 - draw, 533
 - Fl_Input, 533
 - handle, 534
- fl_input
 - Common Dialogs classes and functions, 302
- Fl_Input.H, 1084
- Fl_Input_, 534
 - ~Fl_Input_, 537
 - copy, 538
 - copy_cuts, 538
 - cursor_color, 538
 - cut, 539
 - drawtext, 540
 - Fl_Input_, 537
 - handle_mouse, 540
 - handletext, 540
 - index, 540
 - input_type, 541
 - insert, 541
 - line_end, 541

- line_start, [542](#)
- mark, [542](#)
- maximum_size, [542](#), [543](#)
- position, [543](#)
- readonly, [544](#)
- replace, [544](#)
- resize, [545](#)
- shortcut, [545](#), [546](#)
- size, [546](#)
- static_value, [546](#), [547](#)
- tab_nav, [547](#)
- textcolor, [549](#)
- textfont, [549](#)
- textsize, [550](#)
- undo, [550](#)
- up_down_position, [550](#)
- value, [550](#), [551](#)
- word_end, [552](#)
- word_start, [552](#)
- wrap, [552](#)
- Fl_Input.H, [1085](#)
- Fl_Input_Choice, [553](#)
 - add, [555](#)
 - Fl_Input_Choice, [555](#)
 - input, [555](#)
 - menubutton, [555](#)
 - resize, [555](#)
 - value, [556](#)
- Fl_Input_Choice.H, [1088](#)
- Fl_Int_Input, [556](#)
 - Fl_Int_Input, [557](#)
- Fl_Int_Input.H, [1090](#)
- fl_internal_boxtype
 - fl_boxtype.cxx, [1232](#)
- fl_intptr_t
 - Fl_Widget.H, [1185](#)
- FL_JOIN_BEVEL
 - Drawing functions, [253](#)
- FL_JOIN_MITER
 - Drawing functions, [253](#)
- FL_JOIN_ROUND
 - Drawing functions, [253](#)
- Fl_JPEG_Image, [557](#)
 - Fl_JPEG_Image, [557](#), [558](#)
- Fl_JPEG_Image.H, [1090](#)
- FL_KEYBOARD
 - Enumerations.H, [1005](#)
- FL_KEYDOWN
 - Enumerations.H, [1005](#)
- FL_KEYUP
 - Enumerations.H, [1005](#)
- Fl_Label, [558](#)
 - draw, [559](#)
 - measure, [559](#)
 - type, [559](#)
- Fl_Labeltype
 - Enumerations.H, [1006](#)
- fl_latin1_to_local
 - Color & Font functions, [242](#)
- FL_LEAVE
 - Enumerations.H, [1004](#)
- Fl_Light_Button, [560](#)
 - draw, [561](#)
 - Fl_Light_Button, [561](#)
 - handle, [561](#)
- Fl_Light_Button.H, [1091](#)
- Fl_Line_Dial, [562](#)
- Fl_Line_Dial.H, [1091](#)
- fl_line_style
 - Drawing functions, [263](#)
 - Fl_Graphics_Driver, [495](#)
- fl_line_style.cxx, [1240](#)
- fl_local_to_latin1
 - Color & Font functions, [242](#)
- fl_local_to_mac_roman
 - Color & Font functions, [243](#)
- Fl_Mac_App_Menu, [562](#)
 - custom_application_menu_items, [563](#)
 - print, [563](#)
- fl_mac_quit_early
 - Mac OS X-specific symbols, [293](#)
- fl_mac_roman_to_local
 - Color & Font functions, [243](#)
- fl_mac_set_about
 - Mac OS X-specific symbols, [293](#)
- FL_MAJOR_VERSION
 - Enumerations.H, [1000](#)
- fl_make_path
 - Unicode and UTF-8 functions, [282](#)
- fl_make_path_for_file
 - Unicode and UTF-8 functions, [282](#)
- fl_measure
 - Drawing functions, [263](#)
- fl_measure_pixmap
 - Drawing functions, [264](#)
- Fl_Menu.H, [1092](#)
- Fl_Menu_, [563](#)
 - add, [566](#), [567](#)
 - clear, [569](#)
 - clear_submenu, [569](#)
 - copy, [570](#)
 - down_box, [570](#)
 - find_index, [570](#), [571](#)
 - find_item, [571](#), [572](#)
 - Fl_Menu_, [566](#)
 - global, [572](#)
 - insert, [572](#)
 - item_pathname, [573](#)
 - menu, [574](#)
 - mode, [574](#)
 - mvalue, [574](#)
 - picked, [574](#)
 - remove, [575](#)
 - replace, [575](#)
 - size, [575](#)
 - test_shortcut, [575](#)

- text, [575](#)
- textcolor, [576](#)
- textfont, [576](#)
- textsize, [576](#)
- value, [576](#)
- FI_Menu_.H, [1092](#)
- FI_Menu_Bar, [577](#)
 - draw, [578](#)
 - FI_Menu_Bar, [578](#)
 - handle, [578](#)
- FI_Menu_Bar.H, [1093](#)
- FI_Menu_Button, [579](#)
 - draw, [581](#)
 - FI_Menu_Button, [581](#)
 - handle, [581](#)
 - popup, [582](#)
 - POPUP1, [581](#)
 - POPUP12, [581](#)
 - POPUP123, [581](#)
 - POPUP13, [581](#)
 - POPUP2, [581](#)
 - POPUP23, [581](#)
 - POPUP3, [581](#)
 - popup_buttons, [580](#)
- FI_Menu_Button.H, [1094](#)
- FL_MENU_DIVIDER
 - FI_Menu_Item.H, [1095](#)
- FL_MENU_HORIZONTAL
 - FI_Menu_Item.H, [1095](#)
- FL_MENU_INACTIVE
 - FI_Menu_Item.H, [1095](#)
- FL_MENU_INVISIBLE
 - FI_Menu_Item.H, [1095](#)
- FI_Menu_Item, [582](#)
 - add, [586](#)
 - argument, [586](#), [587](#)
 - callback, [587](#)
 - check, [588](#)
 - checkbox, [588](#)
 - checked, [588](#)
 - deactivate, [588](#)
 - do_callback, [588](#)
 - find_shortcut, [589](#)
 - insert, [589](#)
 - label, [589](#)
 - labelcolor, [590](#)
 - labelfont, [590](#)
 - labeltype, [590](#)
 - measure, [590](#)
 - next, [591](#)
 - popup, [591](#)
 - pulldown, [591](#)
 - radio, [591](#)
 - set, [592](#)
 - setonly, [592](#)
 - shortcut, [592](#)
 - size, [592](#)
 - submenu, [592](#)
 - test_shortcut, [592](#)
 - uncheck, [593](#)
 - value, [593](#)
- FI_Menu_Item.H, [1094](#), [1095](#)
 - FL_MENU_DIVIDER, [1095](#)
 - FL_MENU_HORIZONTAL, [1095](#)
 - FL_MENU_INACTIVE, [1095](#)
 - FL_MENU_INVISIBLE, [1095](#)
 - FL_MENU_RADIO, [1095](#)
 - FL_MENU_TOGGLE, [1095](#)
 - FL_MENU_VALUE, [1095](#)
 - FL_SUBMENU, [1095](#)
 - FL_SUBMENU_POINTER, [1095](#)
- FL_MENU_RADIO
 - FI_Menu_Item.H, [1095](#)
- FL_MENU_TOGGLE
 - FI_Menu_Item.H, [1095](#)
- FL_MENU_VALUE
 - FI_Menu_Item.H, [1095](#)
- FI_Menu_Window, [593](#)
 - clear_overlay, [594](#)
 - flush, [594](#)
 - hide, [594](#)
 - set_overlay, [594](#)
 - show, [594](#)
- FI_Menu_Window.H, [1098](#)
- fl_message
 - Common Dialogs classes and functions, [302](#)
- fl_message.H, [1098](#)
- fl_message_hotspot
 - Common Dialogs classes and functions, [303](#)
- fl_message_icon
 - Common Dialogs classes and functions, [303](#)
- fl_message_title
 - Common Dialogs classes and functions, [303](#)
- fl_message_title_default
 - Common Dialogs classes and functions, [304](#)
- FL_MINOR_VERSION
 - Enumerations.H, [1000](#)
- fl_mkdir
 - Unicode and UTF-8 functions, [282](#)
- FL_MOUSEWHEEL
 - Enumerations.H, [1006](#)
- FL_MOVE
 - Enumerations.H, [1005](#)
- fl_mult_matrix
 - Drawing functions, [264](#)
 - FI_Graphics_Driver, [495](#)
- FI_Multi_Browser, [595](#)
 - FI_Multi_Browser, [596](#)
- FI_Multi_Browser.H, [1099](#)
- FI_Multi_Label, [596](#)
 - labela, [597](#)
 - labelb, [597](#)
 - typea, [597](#)
 - typeb, [597](#)
- FI_Multi_Label.H, [1099](#)
- FI_Multiline_Input, [597](#)

- FI_Multiline_Input, 598
- FI_Multiline_Input.H, 1100
- FI_Multiline_Output, 598
 - FI_Multiline_Output, 599
- FI_Multiline_Output.H, 1100
- FI_Native_File_Chooser, 599
 - ~FI_Native_File_Chooser, 603
 - BROWSE_DIRECTORY, 602
 - BROWSE_FILE, 602
 - BROWSE_MULTI_DIRECTORY, 602
 - BROWSE_MULTI_FILE, 602
 - BROWSE_SAVE_DIRECTORY, 603
 - BROWSE_SAVE_FILE, 603
 - count, 603
 - directory, 603
 - errmsg, 603
 - filename, 603
 - filter, 604
 - filter_value, 604
 - FI_Native_File_Chooser, 603
 - NEW_FOLDER, 602
 - NO_OPTIONS, 602
 - Option, 602
 - options, 604
 - preset_file, 605
 - PREVIEW, 602
 - SAVEAS_CONFIRM, 602
 - show, 605
 - title, 605
 - Type, 602
 - USE_FILTER_EXT, 602
- FI_Native_File_Chooser.H, 1101
- FI_Nice_Slider, 605
- FI_Nice_Slider.H, 1104
- FL_NO_BOX
 - Enumerations.H, 1001
- FL_NO_EVENT
 - Enumerations.H, 1004
- FL_NO_LABEL
 - Enumerations.H, 1007
- fl_nonspacing
 - Unicode and UTF-8 functions, 283
- FL_NORMAL_LABEL
 - Enumerations.H, 1007
- FL_NORMAL_SIZE
 - Enumerations.H, 1008
- fl_not_clipped
 - Drawing functions, 265
 - FI_Graphics_Driver, 496
- FI_Object.H, 1105
- fl_old_shortcut
 - Drawing functions, 265
- fl_open
 - Unicode and UTF-8 functions, 283
- fl_open_callback
 - Mac OS X-specific symbols, 293
- fl_open_uri
 - File names and URI utility functions, 312
- FI_Option
 - FI, 323
- FI_Output, 606
 - FI_Output, 607
- FI_Output.H, 1105
- FI_Overlay_Window, 607
 - draw_overlay, 609
 - FI_Overlay_Window, 608
 - flush, 609
 - hide, 609
 - redraw_overlay, 609
 - resize, 609
 - show, 610
- FI_Overlay_Window.H, 1106
- FI_Pack, 610
 - draw, 611
 - FI_Pack, 611
- FI_Pack.H, 1106
- FI_Paged_Device, 612
 - A0, 614
 - A4, 614
 - class_name, 614
 - end_job, 614
 - end_page, 614
 - LANDSCAPE, 614
 - LETTER, 614
 - margins, 615
 - ORIENTATION, 614
 - origin, 615
 - Page_Format, 614
 - Page_Layout, 614
 - PORTRAIT, 614
 - print_widget, 616
 - print_window, 616
 - print_window_part, 616
 - printable_rect, 617
 - REVERSED, 614
 - rotate, 617
 - scale, 617
 - start_job, 617
 - start_page, 618
 - translate, 618
 - untranslate, 618
- FI_Paged_Device.cxx, 1241
- FI_Paged_Device.H, 1107
- FI_Paged_Device::page_format, 986
- fl_password
 - Common Dialogs classes and functions, 304
- FL_PASTE
 - Enumerations.H, 1006
- FL_PATCH_VERSION
 - Enumerations.H, 1000
- fl_pie
 - Drawing functions, 266
 - FI_Graphics_Driver, 496
- FI_Pixmap, 618
 - color_average, 620
 - copy, 620

- desaturate, 621
- draw, 621
- FI_Pixmap, 620
- label, 621
- uncache, 621
- FI_Pixmap.H, 1109
- FI_Plugin, 622
 - FI_Plugin, 622
- FI_Plugin.H, 1110
- FI_Plugin_Manager, 623
 - ~FI_Plugin_Manager, 624
 - addPlugin, 624
 - load, 624
 - removePlugin, 624
- FI_PNG_Image, 624
 - FI_PNG_Image, 625
- FI_PNG_Image.H, 1111
- FI_PNM_Image, 626
 - FI_PNM_Image, 626
- FI_PNM_Image.H, 1111
- fl_polygon
 - Drawing functions, 266
 - FI_Graphics_Driver, 496, 497
- fl_pop_clip
 - Drawing functions, 267
 - FI_Graphics_Driver, 497
- FI_Positioner, 626
 - draw, 628
 - FI_Positioner, 628
 - handle, 628
- FI_Positioner.H, 1112
- FI_PostScript.H, 1112, 1113
- FI_PostScript_File_Device, 629
 - class_name, 630
 - end_job, 630
 - end_page, 631
 - margins, 631
 - origin, 631
 - printable_rect, 632
 - rotate, 632
 - scale, 632
 - start_job, 633
 - start_page, 634
 - translate, 634
 - untranslate, 634
- FI_PostScript_Graphics_Driver, 634
 - arc, 638
 - begin_complex_polygon, 638
 - begin_line, 638
 - begin_loop, 638
 - begin_points, 638
 - begin_polygon, 639
 - circle, 639
 - class_name, 639
 - clip_box, 639
 - locale_printf, 639
 - color, 640
 - curve, 640
 - descent, 640
 - draw, 640, 641
 - draw_image, 641, 642
 - draw_image_mono, 642
 - draw_scaled, 642
 - end_complex_polygon, 643
 - end_line, 643
 - end_loop, 643
 - end_points, 643
 - end_polygon, 643
 - font, 643
 - gap, 643
 - height, 643
 - line, 644
 - line_style, 644
 - loop, 644
 - not_clipped, 645
 - pie, 645
 - point, 645
 - polygon, 645
 - pop_clip, 646
 - push_clip, 646
 - push_no_clip, 646
 - rect, 646
 - rectf, 646
 - rtl_draw, 646
 - text_extents, 647
 - transformed_vertex, 647
 - vertex, 647
 - width, 647
 - xyline, 647, 648
 - yxline, 648
- FI_PostScript_Printer, 649
 - class_name, 650
 - start_job, 650
- FI_Preferences, 650
 - ~FI_Preferences, 655
 - deleteEntry, 656
 - deleteGroup, 656
 - entries, 656
 - entry, 656
 - entryExists, 657
 - FI_Preferences, 653–655
 - flush, 657
 - get, 657–659
 - getUserdataPath, 660
 - group, 660
 - groupExists, 661
 - groups, 661
 - ID, 653
 - newUUID, 661
 - Root, 653
 - set, 661–663
 - size, 664
 - SYSTEM, 653
 - USER, 653
- FI_Preferences.H, 1116
- FI_Preferences::Entry, 315

- Fl_Preferences::Name, [984](#)
 - Name, [984](#), [985](#)
- Fl_Preferences::Node, [985](#)
- Fl_Preferences::RootNode, [986](#)
- Fl_Printer, [664](#)
 - class_name, [667](#)
 - end_job, [667](#)
 - end_page, [667](#)
 - margins, [667](#)
 - origin, [668](#)
 - print_widget, [668](#)
 - print_window_part, [669](#)
 - printable_rect, [669](#)
 - rotate, [669](#)
 - scale, [670](#)
 - set_current, [670](#)
 - start_job, [670](#)
 - start_page, [671](#)
 - translate, [671](#)
 - untranslate, [671](#)
- Fl_Printer.H, [1118](#), [1119](#)
- Fl_Progress, [671](#)
 - draw, [672](#)
 - Fl_Progress, [672](#)
 - maximum, [672](#)
 - minimum, [673](#)
 - value, [673](#)
- Fl_Progress.H, [1120](#)
- FL_PUSH
 - Enumerations.H, [1004](#)
- fl_push_clip
 - Drawing functions, [267](#)
 - Fl_Graphics_Driver, [497](#)
- fl_push_matrix
 - Drawing functions, [267](#)
 - Fl_Graphics_Driver, [497](#)
- Fl_Quartz_Graphics_Driver, [673](#)
 - class_name, [675](#)
 - color, [675](#)
 - descent, [675](#)
 - draw, [675](#), [676](#)
 - draw_image, [676](#), [677](#)
 - draw_image_mono, [677](#)
 - draw_scaled, [677](#)
 - font, [678](#)
 - height, [678](#)
 - rtl_draw, [678](#)
 - text_extents, [678](#)
 - width, [678](#)
- Fl_Radio_Button, [679](#)
 - Fl_Radio_Button, [679](#)
- Fl_Radio_Button.H, [1121](#)
- Fl_Radio_Light_Button, [680](#)
- Fl_Radio_Light_Button.H, [1122](#)
- Fl_Radio_Round_Button, [680](#)
 - Fl_Radio_Round_Button, [680](#)
- Fl_Radio_Round_Button.H, [1122](#)
- FL_READ
 - Enumerations.H, [1001](#)
- fl_read_image
 - Drawing functions, [267](#)
- fl_rect
 - Drawing functions, [268](#)
 - Fl_Graphics_Driver, [497](#)
- fl_rect.cxx, [1241](#)
- fl_rectbound
 - fl_boxtype.cxx, [1232](#)
- fl_rectf
 - Drawing functions, [268](#)
- fl_register_images
 - Fl_Shared_Image.H, [1129](#)
- FL_RELEASE
 - Enumerations.H, [1004](#)
- fl_rename
 - Unicode and UTF-8 functions, [283](#)
- Fl_Repeat_Button, [682](#)
 - Fl_Repeat_Button, [682](#)
 - handle, [682](#)
- Fl_Repeat_Button.H, [1123](#)
- FL_RESERVED_TYPE
 - Fl_Widget.H, [1185](#)
- fl_reset_spot
 - Drawing functions, [268](#)
- Fl_Return_Button, [683](#)
 - draw, [684](#)
 - Fl_Return_Button, [684](#)
 - handle, [684](#)
- Fl_Return_Button.H, [1123](#)
- Fl_RGB_Image, [685](#)
 - color_average, [687](#)
 - copy, [687](#)
 - desaturate, [687](#)
 - draw, [688](#)
 - Fl_RGB_Image, [686](#), [687](#)
 - label, [688](#)
 - max_size, [688](#)
 - uncache, [689](#)
- Fl_RGB_Image.H, [1124](#)
- Fl_RGB_Scaling
 - Fl_Image.H, [1081](#)
- FL_RGB_SCALING_BILINEAR
 - Fl_Image.H, [1081](#)
- FL_RGB_SCALING_NEAREST
 - Fl_Image.H, [1081](#)
- fl_rmdir
 - Unicode and UTF-8 functions, [284](#)
- Fl_Roller, [689](#)
 - draw, [690](#)
 - Fl_Roller, [690](#)
 - handle, [690](#)
- Fl_Roller.H, [1124](#)
- fl_rotate
 - Drawing functions, [268](#)
 - Fl_Graphics_Driver, [497](#)
- Fl_Round_Button, [691](#)
 - Fl_Round_Button, [691](#)

- FI_Round_Button.H, [1124](#)
- FI_Round_Clock, [692](#)
- FI_Round_Clock.H, [1125](#)
- fl_scale
 - Drawing functions, [268](#), [269](#)
 - FI_Graphics_Driver, [498](#)
- FL_SCREEN_CONFIGURATION_CHANGED
 - Enumerations.H, [1006](#)
- FI_Scroll, [692](#)
 - bbox, [695](#)
 - draw, [695](#)
 - FI_Scroll, [695](#)
 - handle, [695](#)
 - recalc_scrollbars, [696](#)
 - resize, [696](#)
 - scroll_to, [697](#)
 - scrollbar_size, [697](#)
 - xposition, [698](#)
 - yposition, [698](#)
- fl_scroll
 - Drawing functions, [269](#)
- FI_Scroll.H, [1125](#)
- FI_Scroll::FI_Region_LRTB, [681](#)
- FI_Scroll::FI_Region_XYWH, [681](#)
- FI_Scroll::FI_Scrollbar_Data, [701](#)
- FI_Scroll::ScrollInfo, [986](#)
- FI_Scrollbar, [698](#)
 - draw, [699](#)
 - FI_Scrollbar, [699](#)
 - handle, [700](#)
 - linesize, [700](#)
 - value, [700](#), [701](#)
- FI_Scrollbar.H, [1127](#)
- FI_Secret_Input, [702](#)
 - FI_Secret_Input, [702](#)
 - handle, [702](#)
- FI_Secret_Input.H, [1127](#)
- FI_Select_Browser, [703](#)
 - FI_Select_Browser, [704](#)
- FI_Select_Browser.H, [1128](#)
- FL_SELECTIONCLEAR
 - Enumerations.H, [1006](#)
- fl_set_spot
 - Drawing functions, [269](#)
- fl_set_status
 - Drawing functions, [269](#)
- FI_Shared_Image, [704](#)
 - ~FI_Shared_Image, [706](#)
 - add, [706](#)
 - color_average, [706](#)
 - compare, [707](#)
 - copy, [707](#)
 - desaturate, [708](#)
 - draw, [708](#)
 - find, [708](#)
 - FI_Shared_Image, [706](#)
 - get, [708](#), [709](#)
 - original, [709](#)
 - refcount, [709](#)
 - release, [710](#)
 - scale, [710](#)
 - scaling_algorithm, [710](#)
 - uncache, [710](#)
- FI_Shared_Image.H, [1128](#), [1129](#)
 - fl_register_images, [1129](#)
- FL_SHORTCUT
 - Enumerations.H, [1005](#)
- fl_shortcut_label
 - Drawing functions, [270](#)
- FL_SHOW
 - Enumerations.H, [1006](#)
- fl_show_colormap
 - Color & Font functions, [243](#)
- fl_show_colormap.H, [1130](#), [1131](#)
- fl_show_input.H, [1131](#)
- FI_Simple_Counter, [711](#)
- FI_Simple_Counter.H, [1131](#)
- FI_Single_Window, [711](#)
 - flush, [712](#)
 - show, [712](#)
- FI_Single_Window.H, [1132](#)
- fl_size
 - Color & Font functions, [244](#)
- FI_Slider, [713](#)
 - bounds, [714](#)
 - draw, [715](#)
 - FI_Slider, [714](#)
 - handle, [715](#)
 - scrollvalue, [715](#)
 - slider_size, [716](#)
- FI_Slider.H, [1132](#)
- FL_SOLID
 - Drawing functions, [253](#)
- FI_Spinner, [716](#)
 - FI_Spinner, [718](#)
 - handle, [718](#)
 - maximum, [719](#)
 - minimum, [719](#)
 - resize, [719](#)
 - step, [719](#)
 - type, [719](#)
 - value, [720](#)
- FI_Spinner.H, [1133](#)
- fl_stat
 - Unicode and UTF-8 functions, [284](#)
- FI_String
 - fl_types.h, [1176](#)
- FL_SUBMENU
 - FI_Menu_Item.H, [1095](#)
- FL_SUBMENU_POINTER
 - FI_Menu_Item.H, [1095](#)
- FI_Surface_Device, [720](#)
 - class_name, [721](#)
 - set_current, [721](#)
 - surface, [721](#)
- FI_Sys_Menu_Bar, [722](#)

- add, [723](#), [724](#)
- clear, [724](#)
- clear_submenu, [725](#)
- draw, [725](#)
- Fl_Sys_Menu_Bar, [723](#)
- insert, [725](#)
- menu, [726](#)
- mode, [726](#)
- remove, [726](#)
- replace, [726](#)
- Fl_Sys_Menu_Bar.H, [1136](#)
- fl_system
 - Unicode and UTF-8 functions, [285](#)
- Fl_System_Printer, [727](#)
 - class_name, [728](#)
 - end_job, [728](#)
 - end_page, [728](#)
 - margins, [728](#)
 - origin, [729](#)
 - printable_rect, [729](#)
 - rotate, [730](#)
 - scale, [730](#)
 - start_job, [730](#)
 - start_page, [731](#)
 - translate, [731](#)
 - untranslate, [731](#)
- Fl_Table, [731](#)
 - ~Fl_Table, [739](#)
 - callback, [739](#)
 - callback_col, [740](#)
 - callback_context, [740](#)
 - callback_row, [740](#)
 - child, [740](#)
 - children, [740](#)
 - clear, [740](#)
 - col_header, [741](#)
 - col_resize, [741](#)
 - col_resize_min, [741](#)
 - col_width, [741](#)
 - col_width_all, [741](#)
 - CONTEXT_CELL, [738](#)
 - CONTEXT_COL_HEADER, [738](#)
 - CONTEXT_ENDPAGE, [738](#)
 - CONTEXT_NONE, [738](#)
 - CONTEXT_RC_RESIZE, [738](#)
 - CONTEXT_ROW_HEADER, [738](#)
 - CONTEXT_STARTPAGE, [738](#)
 - CONTEXT_TABLE, [738](#)
 - draw, [741](#)
 - draw_cell, [741](#)
 - Fl_Table, [738](#)
 - get_selection, [743](#)
 - handle, [743](#)
 - is_interactive_resize, [744](#)
 - is_selected, [744](#)
 - resize, [744](#)
 - row_header, [744](#)
 - row_height, [744](#)
 - row_height_all, [744](#)
 - row_resize, [745](#)
 - row_resize_min, [745](#)
 - rows, [745](#)
 - scrollbar_size, [745](#)
 - set_selection, [746](#)
 - tab_cell_nav, [746](#)
 - table_box, [747](#)
 - TableContext, [738](#)
 - top_row, [747](#)
 - visible_cells, [747](#)
 - when, [747](#)
- Fl_Table.H, [1137](#)
- Fl_Table_Row, [748](#)
 - ~Fl_Table_Row, [749](#)
 - clear, [749](#)
 - Fl_Table_Row, [749](#)
 - handle, [749](#)
 - row_selected, [750](#)
 - rows, [750](#)
 - select_all_rows, [750](#)
 - select_row, [750](#)
 - type, [750](#)
- Fl_Table_Row.H, [1143](#)
- Fl_Tabs, [751](#)
 - client_area, [755](#)
 - draw, [755](#)
 - Fl_Tabs, [754](#)
 - handle, [755](#)
 - push, [756](#)
 - value, [756](#)
 - which, [757](#)
- Fl_Tabs.H, [1145](#)
- Fl_Text_Buffer, [757](#)
 - add_modify_callback, [762](#)
 - address, [762](#)
 - append, [763](#)
 - appendfile, [763](#)
 - byte_at, [763](#)
 - char_at, [763](#)
 - copy, [764](#)
 - count_displayed_characters, [764](#)
 - count_lines, [764](#)
 - file_encoding_warning_message, [772](#)
 - findchar_backward, [764](#)
 - findchar_forward, [765](#)
 - Fl_Text_Buffer, [762](#)
 - highlight, [765](#)
 - highlight_text, [765](#)
 - insert, [765](#)
 - insert_, [766](#)
 - insertfile, [766](#)
 - length, [766](#)
 - line_end, [766](#)
 - line_start, [767](#)
 - line_text, [767](#)
 - loadfile, [767](#)
 - mTabDist, [772](#)

- next_char, [767](#)
- outputfile, [768](#)
- prev_char, [768](#)
- remove, [768](#)
- remove_, [768](#)
- replace, [769](#)
- rewind_lines, [769](#)
- savefile, [769](#)
- search_backward, [769](#)
- search_forward, [770](#)
- secondary_selection_text, [770](#)
- selection_text, [770](#)
- skip_displayed_characters, [770](#)
- tab_distance, [771](#)
- text, [771](#)
- text_range, [771](#)
- transcoding_warning_action, [772](#)
- word_end, [772](#)
- word_start, [772](#)
- FI_Text_Buffer.H, [1146](#)
- FI_Text_Display, [773](#)
 - ~FI_Text_Display, [781](#)
 - absolute_top_line_number, [782](#)
 - BLOCK_CURSOR, [781](#)
 - buffer, [782](#)
 - buffer_modified_cb, [782](#)
 - buffer_predelete_cb, [783](#)
 - calc_last_char, [783](#)
 - calc_line_starts, [783](#)
 - CARET_CURSOR, [781](#)
 - clear_rect, [783](#)
 - col_to_x, [784](#)
 - count_lines, [784](#)
 - cursor_color, [784](#), [785](#)
 - cursor_style, [785](#)
 - DIM_CURSOR, [781](#)
 - display_insert, [785](#)
 - draw, [785](#)
 - draw_cursor, [786](#)
 - draw_line_numbers, [786](#)
 - draw_range, [786](#)
 - draw_string, [786](#)
 - draw_text, [787](#)
 - draw_vline, [787](#)
 - empty_vlines, [787](#)
 - extend_range_for_styles, [788](#)
 - find_line_end, [788](#)
 - find_wrap_range, [788](#)
 - find_x, [789](#)
 - FI_Text_Display, [781](#)
 - get_absolute_top_line_number, [789](#)
 - handle, [789](#)
 - handle_vline, [790](#)
 - HEAVY_CURSOR, [781](#)
 - highlight_data, [790](#)
 - in_selection, [791](#)
 - insert, [791](#)
 - insert_position, [791](#), [792](#)
 - line_end, [792](#)
 - line_start, [792](#)
 - linenumber_align, [793](#)
 - linenumber_bgcolor, [793](#)
 - linenumber_fgcolor, [793](#)
 - linenumber_font, [793](#)
 - linenumber_format, [793](#)
 - linenumber_size, [794](#)
 - linenumber_width, [794](#)
 - longest_vline, [794](#)
 - maintain_absolute_top_line_number, [794](#)
 - maintaining_absolute_top_line_number, [795](#)
 - measure_deleted_lines, [795](#)
 - measure_proportional_character, [795](#)
 - measure_vline, [796](#)
 - move_down, [796](#)
 - move_left, [796](#)
 - move_right, [796](#)
 - move_up, [796](#)
 - NORMAL_CURSOR, [781](#)
 - offset_line_starts, [796](#)
 - overstrike, [797](#)
 - position_style, [797](#)
 - position_to_line, [797](#)
 - position_to_linecol, [798](#)
 - position_to_xy, [798](#)
 - redisplay_range, [799](#)
 - reset_absolute_top_line_number, [799](#)
 - resize, [799](#)
 - rewind_lines, [799](#)
 - scroll, [800](#)
 - scroll_, [800](#)
 - scroll_timer_cb, [800](#)
 - scrollbar_align, [800](#), [801](#)
 - scrollbar_width, [801](#)
 - shortcut, [801](#)
 - show_cursor, [802](#)
 - show_insert_position, [802](#)
 - SIMPLE_CURSOR, [781](#)
 - skip_lines, [802](#)
 - string_width, [802](#)
 - textcolor, [803](#)
 - textfont, [803](#)
 - textsize, [803](#), [804](#)
 - update_h_scrollbar, [804](#)
 - update_line_starts, [804](#)
 - update_v_scrollbar, [804](#)
 - vline_length, [804](#)
 - word_end, [805](#)
 - word_start, [805](#)
 - WRAP_AT_BOUNDS, [781](#)
 - WRAP_AT_COLUMN, [781](#)
 - WRAP_AT_PIXEL, [781](#)
 - wrap_mode, [805](#)
 - WRAP_NONE, [781](#)
 - wrap_uses_character, [806](#)
 - wrapped_column, [806](#)
 - wrapped_line_counter, [807](#)

- wrapped_row, 807
 - x_to_col, 808
 - xy_to_position, 808
 - xy_to_rowcol, 808
- Fl_Text_Display.H, 1149
- Fl_Text_Display::Style_Table_Entry, 988
- Fl_Text_Editor, 809
 - add_key_binding, 811
 - global_key_bindings, 817
 - handle, 812
 - insert_mode, 812
 - kf_backspace, 812
 - kf_c_s_move, 812
 - kf_copy, 812
 - kf_ctrl_move, 812
 - kf_cut, 813
 - kf_default, 813
 - kf_delete, 813
 - kf_down, 813
 - kf_end, 813
 - kf_enter, 813
 - kf_home, 814
 - kf_ignore, 814
 - kf_insert, 814
 - kf_left, 814
 - kf_m_s_move, 814
 - kf_meta_move, 814
 - kf_move, 814
 - kf_page_down, 815
 - kf_page_up, 815
 - kf_paste, 815
 - kf_right, 815
 - kf_select_all, 815
 - kf_shift_move, 815
 - kf_undo, 816
 - kf_up, 816
 - remove_key_binding, 816
 - tab_nav, 816
- Fl_Text_Editor.H, 1153
- Fl_Text_Editor::Key_Binding, 983
- fl_text_extents
 - Color & Font functions, 244, 245
- Fl_Text_Selection, 817
 - end, 818
 - position, 818
 - selected, 819
 - set, 819
 - start, 819
 - update, 819
- FL_THIN_DOWN_BOX
 - Enumerations.H, 1001
- FL_THIN_DOWN_FRAME
 - Enumerations.H, 1001
- FL_THIN_UP_BOX
 - Enumerations.H, 1001
- FL_THIN_UP_FRAME
 - Enumerations.H, 1001
- Fl_Tile, 821
 - Fl_Tile, 822
 - handle, 823
 - position, 823
 - resize, 823
- Fl_Tile.H, 1155
- Fl_Tiled_Image, 824
 - color_average, 825
 - copy, 825
 - desaturate, 825
 - draw, 825
 - Fl_Tiled_Image, 824
- Fl_Tiled_Image.H, 1155
- Fl_Timer, 826
 - direction, 827
 - draw, 827
 - Fl_Timer, 827
 - handle, 828
 - suspended, 828
- Fl_Timer.H, 1156
- Fl_Toggle_Button, 828
 - Fl_Toggle_Button, 829
- Fl_Toggle_Button.H, 1157
- Fl_Toggle_Light_Button.H, 1157
- Fl_Toggle_Round_Button.H, 1158
- Fl_Tooltip, 829
 - color, 831
 - current, 831
 - delay, 831
 - disable, 832
 - enable, 832
 - enabled, 832
 - enter_area, 832
 - font, 832
 - hoverdelay, 832, 833
 - margin_height, 833
 - margin_width, 833
 - size, 833
 - textcolor, 833
 - wrap_width, 834
- Fl_Tooltip.H, 1158
- fl_transform_dx
 - Drawing functions, 271
 - Fl_Graphics_Driver, 498
- fl_transform_dy
 - Drawing functions, 271
 - Fl_Graphics_Driver, 498
- fl_transform_x
 - Drawing functions, 271
 - Fl_Graphics_Driver, 499
- fl_transform_y
 - Drawing functions, 271
 - Fl_Graphics_Driver, 499
- fl_transformed_vertex
 - Drawing functions, 272
 - Fl_Graphics_Driver, 499
- fl_translate
 - Drawing functions, 272
 - Fl_Graphics_Driver, 499

- Fl_Tree, 834
 - add, 844
 - calc_dimensions, 845
 - calc_tree, 845
 - callback_item, 845, 846
 - callback_reason, 846
 - clear, 846
 - clear_children, 846
 - close, 846, 847
 - closeicon, 847, 848
 - connectorstyle, 848
 - deselect, 848
 - deselect_all, 849
 - display, 849
 - displayed, 850
 - draw, 850
 - extend_selection, 850
 - extend_selection_dir, 851
 - find_clicked, 851
 - find_item, 852
 - first, 852
 - first_selected_item, 852
 - first_visible, 853
 - first_visible_item, 853
 - get_selected_items, 853
 - handle, 854
 - hposition, 854
 - insert, 854
 - insert_above, 855
 - is_close, 855, 857
 - is_hscroll_visible, 857
 - is_open, 857, 858
 - is_scrollbar, 858
 - is_selected, 858, 859
 - is_vscroll_visible, 859
 - item_clicked, 859
 - item_draw_mode, 859, 860
 - item_labelbgcolor, 860
 - item_labelbgcolor, 860
 - item_labelfont, 860
 - item_labelsize, 861
 - item_pathname, 861
 - item_reselect_mode, 861
 - last, 862
 - last_selected_item, 862
 - last_visible, 862
 - last_visible_item, 862
 - load, 863
 - next, 863
 - next_item, 863
 - next_selected_item, 864
 - next_visible_item, 865
 - open, 865, 866
 - open_toggle, 866
 - openicon, 867
 - prev, 867
 - recalc_tree, 868
 - remove, 868
 - resize, 868
 - root, 868
 - root_label, 868
 - scrollbar_size, 869
 - select, 869, 870
 - select_all, 870
 - select_only, 871
 - select_toggle, 871
 - selectbox, 872
 - selectmode, 872
 - set_item_focus, 872
 - show_item, 872, 873
 - show_item_bottom, 873
 - show_item_middle, 873
 - show_item_top, 873
 - show_self, 874
 - showcollapse, 874
 - showroot, 874
 - sortorder, 875
 - usericon, 875
 - vposition, 875
- Fl_Tree.H, 1159, 1160
 - Fl_Tree_Reason, 1160
 - FL_TREE_REASON_CLOSED, 1160
 - FL_TREE_REASON_DESELECTED, 1160
 - FL_TREE_REASON_DRAGGED, 1160
 - FL_TREE_REASON_NONE, 1160
 - FL_TREE_REASON_OPENED, 1160
 - FL_TREE_REASON_RESELECTED, 1160
 - FL_TREE_REASON_SELECTED, 1160
- Fl_Tree_Connector
 - Fl_Tree_Prefs.H, 1171
- FL_TREE_CONNECTOR_DOTTED
 - Fl_Tree_Prefs.H, 1171
- FL_TREE_CONNECTOR_NONE
 - Fl_Tree_Prefs.H, 1171
- FL_TREE_CONNECTOR_SOLID
 - Fl_Tree_Prefs.H, 1171
- Fl_Tree_Item, 876
 - activate, 881
 - add, 881, 882
 - calc_item_height, 882
 - child, 883
 - deactivate, 883
 - deparent, 883
 - depth, 883
 - deselect_all, 883
 - draw, 883
 - draw_horizontal_connector, 884
 - draw_item_content, 884
 - draw_vertical_connector, 885
 - drawbgcolor, 886
 - drawfgcolor, 886
 - find_child, 886
 - find_child_item, 886, 887
 - find_clicked, 887
 - find_item, 887
 - Fl_Tree_Item, 881

- hide_widgets, 888
- insert, 888
- insert_above, 888
- label, 888
- label_h, 888
- label_w, 888
- label_x, 889
- label_y, 889
- labelbgcolor, 889
- move, 889, 890
- move_above, 890
- move_below, 890
- move_into, 890
- next, 891
- next_displayed, 891
- next_sibling, 891
- next_visible, 891
- parent, 891
- prefs, 892
- prev, 892
- prev_displayed, 892
- prev_sibling, 892
- prev_visible, 892
- recalc_tree, 893
- remove_child, 893
- reparent, 893
- replace, 893
- replace_child, 894
- select, 894
- select_all, 894
- show_self, 895
- show_widgets, 895
- swap_children, 895
- tree, 895, 896
- update_prev_next, 896
- userdeicon, 896
- usericon, 896
- visible_r, 897
- FI_Tree_Item.H, 1164
- FI_Tree_Item_Array, 897
 - add, 898
 - clear, 898
 - deparent, 898
 - FI_Tree_Item_Array, 898
 - insert, 899
 - manage_item_destroy, 899
 - move, 899
 - remove, 899
 - reparent, 900
 - replace, 900
- FI_Tree_Item_Array.H, 1168, 1169
- FL_TREE_ITEM_DRAW_DEFAULT
 - FI_Tree_Prefs.H, 1171
- FL_TREE_ITEM_DRAW_LABEL_AND_WIDGET
 - FI_Tree_Prefs.H, 1171
- FI_Tree_Item_Draw_Mode
 - FI_Tree_Prefs.H, 1171
- FL_TREE_ITEM_HEIGHT_FROM_WIDGET
 - FI_Tree_Prefs.H, 1171
- FI_Tree_Item_Resize_Mode
 - FI_Tree_Prefs.H, 1171
- FI_Tree_Prefs, 900
 - closedeicon, 903
 - closeicon, 903
 - item_draw_mode, 903
 - item_labelbgcolor, 903, 904
 - marginbottom, 904
 - opendeicon, 904
 - openicon, 904
 - selectmode, 904
 - showcollapse, 904
 - showroot, 905
 - sortorder, 905
 - userdeicon, 905
- FI_Tree_Prefs.H, 1170, 1172
 - FI_Tree_Connector, 1171
 - FL_TREE_CONNECTOR_DOTTED, 1171
 - FL_TREE_CONNECTOR_NONE, 1171
 - FL_TREE_CONNECTOR_SOLID, 1171
 - FL_TREE_ITEM_DRAW_DEFAULT, 1171
 - FL_TREE_ITEM_DRAW_LABEL_AND_WIDGET, 1171
 - FI_Tree_Item_Draw_Mode, 1171
 - FL_TREE_ITEM_HEIGHT_FROM_WIDGET, 1171
 - FI_Tree_Item_Resize_Mode, 1171
 - FI_Tree_Select, 1171
 - FL_TREE_SELECT_MULTI, 1171
 - FL_TREE_SELECT_NONE, 1171
 - FL_TREE_SELECT_SINGLE, 1171
 - FL_TREE_SELECT_SINGLE_DRAGGABLE, 1171
 - FL_TREE_SELECTABLE_ALWAYS, 1171
 - FL_TREE_SELECTABLE_ONCE, 1171
 - FI_Tree_Sort, 1172
 - FL_TREE_SORT_ASCENDING, 1172
 - FL_TREE_SORT_DESCENDING, 1172
 - FL_TREE_SORT_NONE, 1172
- FI_Tree_Reason
 - FI_Tree.H, 1160
- FL_TREE_REASON_CLOSED
 - FI_Tree.H, 1160
- FL_TREE_REASON_DESELECTED
 - FI_Tree.H, 1160
- FL_TREE_REASON_DRAGGED
 - FI_Tree.H, 1160
- FL_TREE_REASON_NONE
 - FI_Tree.H, 1160
- FL_TREE_REASON_OPENED
 - FI_Tree.H, 1160
- FL_TREE_REASON_RESELECTED
 - FI_Tree.H, 1160
- FL_TREE_REASON_SELECTED
 - FI_Tree.H, 1160
- FI_Tree_Select
 - FI_Tree_Prefs.H, 1171
- FL_TREE_SELECT_MULTI

- FI_Tree_Prefs.H, [1171](#)
- FL_TREE_SELECT_NONE
 - FI_Tree_Prefs.H, [1171](#)
- FL_TREE_SELECT_SINGLE
 - FI_Tree_Prefs.H, [1171](#)
- FL_TREE_SELECT_SINGLE_DRAGGABLE
 - FI_Tree_Prefs.H, [1171](#)
- FL_TREE_SELECTABLE_ALWAYS
 - FI_Tree_Prefs.H, [1171](#)
- FL_TREE_SELECTABLE_ONCE
 - FI_Tree_Prefs.H, [1171](#)
- FI_Tree_Sort
 - FI_Tree_Prefs.H, [1172](#)
- FL_TREE_SORT_ASCENDING
 - FI_Tree_Prefs.H, [1172](#)
- FL_TREE_SORT_DESCENDING
 - FI_Tree_Prefs.H, [1172](#)
- FL_TREE_SORT_NONE
 - FI_Tree_Prefs.H, [1172](#)
- fl_types.h, [1176](#)
 - FI_CString, [1176](#)
 - FI_String, [1176](#)
- fl_ucs_to_Utf16
 - Unicode and UTF-8 functions, [285](#)
- FL_UNFOCUS
 - Enumerations.H, [1005](#)
- fl_unlink
 - Unicode and UTF-8 functions, [285](#)
- FL_UP_BOX
 - Enumerations.H, [1001](#)
- FL_UP_FRAME
 - Enumerations.H, [1001](#)
- fl_utf8.h, [1177](#), [1179](#)
- fl_utf8back
 - Unicode and UTF-8 functions, [286](#)
- fl_utf8bytes
 - Unicode and UTF-8 functions, [286](#)
- fl_utf8decode
 - Unicode and UTF-8 functions, [286](#)
- fl_utf8encode
 - Unicode and UTF-8 functions, [286](#)
- fl_utf8from_mb
 - Unicode and UTF-8 functions, [287](#)
- fl_utf8froma
 - Unicode and UTF-8 functions, [287](#)
- fl_utf8fromwc
 - Unicode and UTF-8 functions, [287](#)
- fl_utf8fwd
 - Unicode and UTF-8 functions, [288](#)
- fl_utf8len
 - Unicode and UTF-8 functions, [288](#)
- fl_utf8len1
 - Unicode and UTF-8 functions, [288](#)
- fl_utf8locale
 - Unicode and UTF-8 functions, [288](#)
- fl_utf8test
 - Unicode and UTF-8 functions, [288](#)
- fl_utf8to_mb
 - Unicode and UTF-8 functions, [289](#)
- fl_utf8toa
 - Unicode and UTF-8 functions, [289](#)
- fl_utf8toUtf16
 - Unicode and UTF-8 functions, [289](#)
- fl_utf8towc
 - Unicode and UTF-8 functions, [290](#)
- fl_utf_strcasecmp
 - Unicode and UTF-8 functions, [290](#)
- fl_utf_strncasecmp
 - Unicode and UTF-8 functions, [290](#)
- fl_utf_tolower
 - Unicode and UTF-8 functions, [291](#)
- fl_utf_toupper
 - Unicode and UTF-8 functions, [291](#)
- FI_Valuator, [905](#)
 - FI_Valuator, [908](#)
 - format, [908](#)
 - increment, [908](#)
 - maximum, [908](#)
 - minimum, [908](#), [909](#)
 - precision, [909](#)
 - range, [909](#)
 - round, [909](#)
 - step, [909](#)
 - value, [909](#)
 - value_damage, [910](#)
- FI_Valuator.H, [1182](#)
- FI_Value_Input, [910](#)
 - cursor_color, [912](#)
 - draw, [912](#)
 - FI_Value_Input, [911](#)
 - handle, [912](#)
 - resize, [912](#)
 - shortcut, [913](#)
 - soft, [913](#)
 - textcolor, [913](#)
 - textfont, [913](#), [914](#)
 - textsize, [914](#)
- FI_Value_Input.H, [1183](#)
- FI_Value_Output, [914](#)
 - draw, [915](#)
 - FI_Value_Output, [915](#)
 - handle, [916](#)
 - soft, [916](#)
 - textcolor, [916](#), [917](#)
 - textfont, [917](#)
 - textsize, [917](#)
- FI_Value_Output.H, [1183](#)
- FI_Value_Slider, [917](#)
 - draw, [919](#)
 - FI_Value_Slider, [918](#)
 - handle, [919](#)
 - textcolor, [919](#)
 - textfont, [919](#), [920](#)
 - textsize, [920](#)
- FI_Value_Slider.H, [1184](#)
- FL_VERSION

- Enumerations.H, 1000
- fl_vertex
 - Drawing functions, 272
 - Fl_Graphics_Driver, 499
- fl_vertex.cxx, 1241
- fl_wcwidth
 - Unicode and UTF-8 functions, 291
- fl_wcwidth_
 - Unicode and UTF-8 functions, 292
- Fl_When
 - Enumerations.H, 1007
- FL_WHEN_CHANGED
 - Enumerations.H, 1007
- FL_WHEN_ENTER_KEY
 - Enumerations.H, 1007
- FL_WHEN_ENTER_KEY_ALWAYS
 - Enumerations.H, 1007
- FL_WHEN_ENTER_KEY_CHANGED
 - Enumerations.H, 1007
- FL_WHEN_NEVER
 - Enumerations.H, 1007
- FL_WHEN_NOT_CHANGED
 - Enumerations.H, 1007
- FL_WHEN_RELEASE
 - Enumerations.H, 1007
- FL_WHEN_RELEASE_ALWAYS
 - Enumerations.H, 1007
- Fl_Widget, 920
 - ~Fl_Widget, 928
 - activate, 928
 - active, 928
 - active_r, 929
 - align, 929
 - argument, 929
 - as_gl_window, 930
 - as_group, 930
 - as_window, 930
 - box, 931
 - callback, 931, 932
 - CHANGED, 927
 - changed, 933
 - clear_active, 933
 - clear_changed, 933
 - clear_damage, 933
 - clear_output, 934
 - clear_visible, 934
 - clear_visible_focus, 934
 - CLIP_CHILDREN, 927
 - color, 934, 935
 - color2, 935
 - contains, 935
 - COPIED_LABEL, 927
 - COPIED_TOOLTIP, 927
 - copy_label, 936
 - copy_tooltip, 936
 - damage, 936, 937
 - deactivate, 937
 - default_callback, 937
 - deimage, 938
 - do_callback, 938, 939
 - draw, 939
 - draw_label, 940
 - Fl_Widget, 927
 - FORCE_POSITION, 927
 - FULLSCREEN, 927
 - GROUP_RELATIVE, 927
 - h, 940
 - handle, 940
 - hide, 941
 - image, 941, 942
 - INACTIVE, 927
 - inside, 942
 - INVISIBLE, 927
 - is_label_copied, 942
 - label, 942, 943
 - label_shortcut, 943
 - labelcolor, 943, 944
 - labelfont, 944
 - labelsize, 944, 945
 - labeltype, 945
 - MAC_USE_ACCENTS_MENU, 927
 - measure_label, 945
 - MENU_WINDOW, 927
 - MODAL, 927
 - NO_OVERLAY, 927
 - NOBORDER, 927
 - NON_MODAL, 927
 - OUTPUT, 927
 - output, 946
 - OVERRIDE, 927
 - parent, 946
 - position, 946
 - redraw, 947
 - redraw_label, 947
 - resize, 947
 - selection_color, 947, 948
 - set_active, 948
 - set_changed, 948
 - set_output, 948
 - set_visible, 948
 - set_visible_focus, 948
 - SHORTCUT_LABEL, 927
 - show, 949
 - size, 949
 - take_focus, 949
 - takeevents, 949
 - test_shortcut, 950
 - tooltip, 950, 951
 - TOOLTIP_WINDOW, 927
 - top_window, 951
 - top_window_offset, 951
 - type, 952
 - user_data, 952
 - USERFLAG1, 927
 - USERFLAG2, 927
 - USERFLAG3, 927

- visible, [952](#)
- VISIBLE_FOCUS, [927](#)
- visible_focus, [953](#)
- visible_r, [953](#)
- w, [953](#)
- when, [954](#)
- window, [954](#)
- x, [955](#)
- y, [955](#)
- Fl_Widget.H, [1185](#), [1186](#)
 - fl_intptr_t, [1185](#)
 - FL_RESERVED_TYPE, [1185](#)
- Fl_Widget_Tracker, [956](#)
 - deleted, [956](#)
 - exists, [956](#)
 - widget, [957](#)
- fl_width
 - Color & Font functions, [245](#)
- Fl_Window, [957](#)
 - ~Fl_Window, [961](#)
 - as_window, [962](#)
 - border, [962](#)
 - clear_border, [962](#)
 - clear_modal_states, [962](#)
 - current, [963](#)
 - current_, [974](#)
 - cursor, [963](#)
 - decorated_h, [963](#)
 - decorated_w, [964](#)
 - default_cursor, [964](#)
 - default_icon, [964](#)
 - default_icons, [964](#)
 - default_xclass, [965](#)
 - draw, [965](#)
 - Fl_Window, [961](#)
 - flush, [966](#)
 - force_position, [966](#)
 - free_icons, [966](#)
 - free_position, [966](#)
 - fullscreen, [967](#)
 - fullscreen_screens, [967](#)
 - handle, [967](#)
 - hide, [968](#)
 - hotspot, [968](#)
 - icon, [968](#), [969](#)
 - iconize, [969](#)
 - icons, [969](#)
 - make_current, [969](#)
 - modal, [970](#)
 - resize, [970](#)
 - set_menu_window, [970](#)
 - set_modal, [970](#)
 - set_non_modal, [970](#)
 - set_tooltip_window, [970](#)
 - shape, [971](#)
 - show, [972](#)
 - shown, [973](#)
 - size_range, [973](#)
 - wait_for_expose, [973](#)
 - xclass, [974](#)
- Fl_Window.H, [1190](#)
- Fl_Window::shape_data_type, [987](#)
- Fl_Wizard, [975](#)
 - Fl_Wizard, [976](#)
 - next, [976](#)
- Fl_Wizard.H, [1193](#)
- FL_WRITE
 - Enumerations.H, [1001](#)
- Fl_XBM_Image, [976](#)
 - Fl_XBM_Image, [976](#)
- Fl_XBM_Image.H, [1194](#)
- Fl_XColor, [977](#)
- Fl_XColor.H, [1242](#)
- Fl_Xlib_Graphics_Driver, [977](#)
 - class_name, [978](#)
 - color, [978](#), [979](#)
 - copy_offscreen, [979](#)
 - descent, [979](#)
 - draw, [979](#), [980](#)
 - draw_image, [980](#), [981](#)
 - draw_image_mono, [981](#)
 - font, [981](#)
 - height, [981](#)
 - rtl_draw, [982](#)
 - text_extents, [982](#)
 - width, [982](#)
- fl_xpixel
 - Color & Font functions, [245](#)
- Fl_XPM_Image, [983](#)
 - Fl_XPM_Image, [983](#)
- Fl_XPM_Image.H, [1195](#)
- FL_ZOOM_GESTURE
 - Enumerations.H, [1006](#)
- flstring.h, [1242](#)
- flush
 - Fl, [329](#)
 - Fl_Double_Window, [424](#)
 - Fl_Gl_Window, [463](#)
 - Fl_Menu_Window, [594](#)
 - Fl_Overlay_Window, [609](#)
 - Fl_Preferences, [657](#)
 - Fl_Single_Window, [712](#)
 - Fl_Window, [966](#)
- focus
 - Events handling functions, [227](#)
 - Fl_Group, [506](#)
- font
 - Fl_GDI_Graphics_Driver, [455](#)
 - Fl_Graphics_Driver, [483](#)
 - Fl_PostScript_Graphics_Driver, [643](#)
 - Fl_Quartz_Graphics_Driver, [678](#)
 - Fl_Tooltip, [832](#)
 - Fl_Xlib_Graphics_Driver, [981](#)
- FORCE_POSITION
 - Fl_Widget, [927](#)
- force_position

- FI_Window, 966
- format
 - FI_Valuator, 908
- format_char
 - FI_Browser, 351, 352
- forms.H, 1195
- free_color
 - Color & Font functions, 246
- free_icons
 - FI_Window, 966
- free_position
 - FI_Window, 966
- freeglut_teapot_data.h, 1243
- full_height
 - FI_Browser, 352
 - FI_Browser_, 371
- full_width
 - FI_Browser_, 371
- FULLSCREEN
 - FI_Widget, 927
- fullscreen
 - FI_Window, 967
- fullscreen_screens
 - FI_Window, 967
- g
 - FI_Color_Chooser, 408
- gap
 - FI_Graphics_Driver, 483
 - FI_PostScript_Graphics_Driver, 643
- gb2312.h, 1439
- georgian_academy.h, 1469
- georgian_ps.h, 1470
- get
 - FI_Preferences, 657–659
 - FI_Shared_Image, 708, 709
- get_absolute_top_line_number
 - FI_Text_Display, 789
- get_color
 - Color & Font functions, 246
- get_font
 - Color & Font functions, 246
- get_font_name
 - Color & Font functions, 246
- get_font_sizes
 - Color & Font functions, 247
- get_key
 - Events handling functions, 227
- get_mouse
 - Events handling functions, 227
- get_selected_items
 - FI_Tree, 853
- get_selection
 - FI_Table, 743
- get_system_colors
 - FI, 329
- getUserdataPath
 - FI_Preferences, 660
- gl.h, 1205, 1208
 - gl_color, 1206
 - gl_draw, 1206, 1207
 - gl_rect, 1208
 - gl_rectf, 1208
- gl2opengl.h, 1209
- gl_color
 - gl.h, 1206
- gl_draw
 - gl.h, 1206, 1207
- gl_draw.H, 1209
- gl_rect
 - gl.h, 1208
- gl_rectf
 - gl.h, 1208
- gl_texture_pile_height
 - Mac OS X-specific symbols, 293
- gl_visual
 - FI, 330
- global
 - FI_Menu_, 572
- global_key_bindings
 - FI_Text_Editor, 817
- glu.h, 1210
- glut.H, 1211
- grab
 - Windows handling functions, 215
- group
 - FI_Preferences, 660
- GROUP_RELATIVE
 - FI_Widget, 927
- groupExists
 - FI_Preferences, 661
- groups
 - FI_Preferences, 661
- h
 - FI_Widget, 940
- handle
 - Events handling functions, 227
 - FI_Adjuster, 338
 - FI_Box, 343
 - FI_Browser_, 371
 - FI_Button, 384
 - FI_Check_Browser, 394
 - FI_Choice, 398
 - FI_Clock, 401
 - FI_Counter, 415
 - FI_Dial, 421, 422
 - FI_File_Input, 441
 - FI_Free, 450
 - FI_Gl_Window, 463
 - FI_Glut_Window, 470
 - FI_Group, 506
 - FI_Help_View, 518
 - FI_Input, 534
 - FI_Light_Button, 561
 - FI_Menu_Bar, 578
 - FI_Menu_Button, 581
 - FI_Positioner, 628

- FI_Repeat_Button, 682
- FI_Return_Button, 684
- FI_Roller, 690
- FI_Scroll, 695
- FI_Scrollbar, 700
- FI_Secret_Input, 702
- FI_Slider, 715
- FI_Spinner, 718
- FI_Table, 743
- FI_Table_Row, 749
- FI_Tabs, 755
- FI_Text_Display, 789
- FI_Text_Editor, 812
- FI_Tile, 823
- FI_Timer, 828
- FI_Tree, 854
- FI_Value_Input, 912
- FI_Value_Output, 916
- FI_Value_Slider, 919
- FI_Widget, 940
- FI_Window, 967
- handle_
 - Events handling functions, 228
- handle_mouse
 - FI_Input_, 540
- handle_vline
 - FI_Text_Display, 790
- handletext
 - FI_Input_, 540
- has_scrollbar
 - FI_Browser_, 371
- HEAVY_CURSOR
 - FI_Text_Display, 781
- height
 - FI_GDI_Graphics_Driver, 455
 - FI_Graphics_Driver, 483
 - FI_PostScript_Graphics_Driver, 643
 - FI_Quartz_Graphics_Driver, 678
 - FI_Xlib_Graphics_Driver, 981
- help
 - FI, 336
- hide
 - FI_Browser, 352
 - FI_Double_Window, 424
 - FI_GI_Window, 464
 - FI_Menu_Window, 594
 - FI_Overlay_Window, 609
 - FI_Widget, 941
 - FI_Window, 968
- hide_widgets
 - FI_Tree_Item, 888
- highlight
 - FI_Text_Buffer, 765
- highlight_data
 - FI_Text_Display, 790
- highlight_text
 - FI_Text_Buffer, 765
- highres_image
 - FI_Image_Surface, 531
- HORIZONTAL
 - FI_Browser_, 368
- HORIZONTAL_ALWAYS
 - FI_Browser_, 368
- hotspot
 - FI_Window, 968
- hour
 - FI_Clock_Output, 404
- hoverdelay
 - FI_Tooltip, 832, 833
- hposition
 - FI_Browser_, 372
 - FI_Tree, 854
- hscrollbar
 - FI_Browser_, 381
- hsv
 - FI_Color_Chooser, 408
- hsv2rgb
 - FI_Color_Chooser, 408
- hue
 - FI_Color_Chooser, 408
- icon
 - FI_Browser, 353
 - FI_Window, 968, 969
- iconize
 - FI_Window, 969
- icons
 - FI_Window, 969
- iconsize
 - FI_File_Browser, 427, 428
 - FI_File_Chooser, 433
- ID
 - FI_Preferences, 653
- idle
 - FI, 336
- image
 - FI_Image_Surface, 531
 - FI_Widget, 941, 942
- in_selection
 - FI_Text_Display, 791
- INACTIVE
 - FI_Widget, 927
- inactive
 - FI_Image, 527
- incr_height
 - FI_Browser, 353
 - FI_Browser_, 372
- increment
 - FI_Valuator, 908
- index
 - FI_Input_, 540
- init_sizes
 - FI_Group, 507
- input
 - FI_Input_Choice, 555
- input_type
 - FI_Input_, 541

- insert
 - FI_Browser, [354](#)
 - FI_Chart, [391](#)
 - FI_Group, [507](#)
 - FI_Input_, [541](#)
 - FI_Menu_, [572](#)
 - FI_Menu_Item, [589](#)
 - FI_Sys_Menu_Bar, [725](#)
 - FI_Text_Buffer, [765](#)
 - FI_Text_Display, [791](#)
 - FI_Tree, [854](#)
 - FI_Tree_Item, [888](#)
 - FI_Tree_Item_Array, [899](#)
- insert_
 - FI_Text_Buffer, [766](#)
- insert_above
 - FI_Tree, [855](#)
 - FI_Tree_Item, [888](#)
- insert_mode
 - FI_Text_Editor, [812](#)
- insert_position
 - FI_Text_Display, [791](#), [792](#)
- insertfile
 - FI_Text_Buffer, [766](#)
- inserting
 - FI_Browser_, [372](#)
- inside
 - FI_Widget, [942](#)
- INVISIBLE
 - FI_Widget, [927](#)
- is_close
 - FI_Tree, [855](#), [857](#)
- is_hscroll_visible
 - FI_Tree, [857](#)
- is_interactive_resize
 - FI_Table, [744](#)
- is_label_copied
 - FI_Widget, [942](#)
- is_open
 - FI_Tree, [857](#), [858](#)
- is_scheme
 - FI, [330](#)
- is_scrollbar
 - FI_Tree, [858](#)
- is_selected
 - FI_Table, [744](#)
 - FI_Tree, [858](#), [859](#)
- is_vscroll_visible
 - FI_Tree, [859](#)
- iso8859_1.h, [1471](#)
- iso8859_10.h, [1472](#)
- iso8859_11.h, [1473](#)
- iso8859_13.h, [1474](#)
- iso8859_14.h, [1475](#)
- iso8859_15.h, [1476](#)
- iso8859_16.h, [1477](#)
- iso8859_2.h, [1478](#)
- iso8859_3.h, [1479](#)
- iso8859_4.h, [1481](#)
- iso8859_5.h, [1482](#)
- iso8859_6.h, [1483](#)
- iso8859_7.h, [1484](#)
- iso8859_8.h, [1485](#)
- iso8859_9.h, [1486](#)
- iso8859_9e.h, [1487](#)
- item_at
 - FI_Browser, [354](#)
 - FI_Browser_, [373](#)
- item_clicked
 - FI_Tree, [859](#)
- item_draw
 - FI_Browser, [355](#)
 - FI_Browser_, [373](#)
- item_draw_mode
 - FI_Tree, [859](#), [860](#)
 - FI_Tree_Prefs, [903](#)
- item_first
 - FI_Browser, [355](#)
 - FI_Browser_, [373](#)
- item_height
 - FI_Browser, [355](#)
 - FI_Browser_, [373](#)
- item_labelbgcolor
 - FI_Tree, [860](#)
 - FI_Tree_Prefs, [903](#), [904](#)
- item_labelfgcolor
 - FI_Tree, [860](#)
- item_labelfont
 - FI_Tree, [860](#)
- item_labelsize
 - FI_Tree, [861](#)
- item_last
 - FI_Browser, [356](#)
 - FI_Browser_, [374](#)
- item_next
 - FI_Browser, [356](#)
 - FI_Browser_, [374](#)
- item_pathname
 - FI_Menu_, [573](#)
 - FI_Tree, [861](#)
- item_prev
 - FI_Browser, [356](#)
 - FI_Browser_, [374](#)
- item_quick_height
 - FI_Browser_, [374](#)
- item_reselect_mode
 - FI_Tree, [861](#)
- item_select
 - FI_Browser, [357](#)
 - FI_Browser_, [375](#)
- item_selected
 - FI_Browser, [357](#)
 - FI_Browser_, [375](#)
- item_swap
 - FI_Browser, [357](#)
 - FI_Browser_, [375](#)

- item_text
 - FI_Browser, 358
 - FI_Browser_, 375
- item_width
 - FI_Browser, 358
 - FI_Browser_, 376
- jsx0201.h, 1488
- jsx0208.h, 1489
- jsx0212.h, 1516
- kf_backspace
 - FI_Text_Editor, 812
- kf_c_s_move
 - FI_Text_Editor, 812
- kf_copy
 - FI_Text_Editor, 812
- kf_ctrl_move
 - FI_Text_Editor, 812
- kf_cut
 - FI_Text_Editor, 813
- kf_default
 - FI_Text_Editor, 813
- kf_delete
 - FI_Text_Editor, 813
- kf_down
 - FI_Text_Editor, 813
- kf_end
 - FI_Text_Editor, 813
- kf_enter
 - FI_Text_Editor, 813
- kf_home
 - FI_Text_Editor, 814
- kf_ignore
 - FI_Text_Editor, 814
- kf_insert
 - FI_Text_Editor, 814
- kf_left
 - FI_Text_Editor, 814
- kf_m_s_move
 - FI_Text_Editor, 814
- kf_meta_move
 - FI_Text_Editor, 814
- kf_move
 - FI_Text_Editor, 814
- kf_page_down
 - FI_Text_Editor, 815
- kf_page_up
 - FI_Text_Editor, 815
- kf_paste
 - FI_Text_Editor, 815
- kf_right
 - FI_Text_Editor, 815
- kf_select_all
 - FI_Text_Editor, 815
- kf_shift_move
 - FI_Text_Editor, 815
- kf_undo
 - FI_Text_Editor, 816
- kf_up
 - FI_Text_Editor, 816
- koi8_c.h, 1541
- koi8_r.h, 1543
- koi8_u.h, 1544
- ksc5601.h, 1546
- label
 - FI_Bitmap, 341
 - FI_File_Icon, 437
 - FI_Image, 527
 - FI_Menu_Item, 589
 - FI_Pixmap, 621
 - FI_RGB_Image, 688
 - FI_Tree_Item, 888
 - FI_Widget, 942, 943
- label_h
 - FI_Tree_Item, 888
- label_shortcut
 - FI_Widget, 943
- label_w
 - FI_Tree_Item, 888
- label_x
 - FI_Tree_Item, 889
- label_y
 - FI_Tree_Item, 889
- labela
 - FI_Multi_Label, 597
- labelb
 - FI_Multi_Label, 597
- labelbgcolor
 - FI_Tree_Item, 889
- labelcolor
 - FI_Menu_Item, 590
 - FI_Widget, 943, 944
- labelfont
 - FI_Menu_Item, 590
 - FI_Widget, 944
- labelsize
 - FI_Widget, 944, 945
- labeltype
 - FI_File_Icon, 437
 - FI_Menu_Item, 590
 - FI_Widget, 945
- LANDSCAPE
 - FI_Paged_Device, 614
- last
 - FI_Tree, 862
- last_selected_item
 - FI_Tree, 862
- last_visible
 - FI_Tree, 862
- last_visible_item
 - FI_Tree, 862
- ld
 - FI_Image, 528
- leftedge
 - FI_Browser_, 376
- leftline

- FI_Help_View, 518
- length
 - FI_Text_Buffer, 766
- LETTER
 - FI_Paged_Device, 614
- line
 - FI_Graphics_Driver, 483, 484
 - FI_PostScript_Graphics_Driver, 644
- line_end
 - FI_Input_, 541
 - FI_Text_Buffer, 766
 - FI_Text_Display, 792
- line_start
 - FI_Input_, 542
 - FI_Text_Buffer, 767
 - FI_Text_Display, 792
- line_style
 - FI_Graphics_Driver, 484
 - FI_PostScript_Graphics_Driver, 644
- line_text
 - FI_Text_Buffer, 767
- lineno
 - FI_Browser, 358
- linenumber_align
 - FI_Text_Display, 793
- linenumber_bgcolor
 - FI_Text_Display, 793
- linenumber_fgcolor
 - FI_Text_Display, 793
- linenumber_font
 - FI_Text_Display, 793
- linenumber_format
 - FI_Text_Display, 793
- linenumber_size
 - FI_Text_Display, 794
- linenumber_width
 - FI_Text_Display, 794
- lineposition
 - FI_Browser, 359
- linesize
 - FI_Scrollbar, 700
- link
 - FI_Help_View, 518
- load
 - FI_Browser, 359
 - FI_File_Browser, 428
 - FI_File_Icon, 437
 - FI_Help_Dialog, 511
 - FI_Help_View, 518
 - FI_Plugin_Manager, 624
 - FI_Tree, 863
- load_fti
 - FI_File_Icon, 438
- load_image
 - FI_File_Icon, 438
- load_system_icons
 - FI_File_Icon, 438
- loadfile
 - FI_Text_Buffer, 767
- lock
 - Multithreading support functions, 273
- longest_vline
 - FI_Text_Display, 794
- loop
 - FI_Graphics_Driver, 484
 - FI_PostScript_Graphics_Driver, 644
- lstep
 - FI_Counter, 416
- Mac OS X-specific symbols, 292
 - fl_mac_quit_early, 293
 - fl_mac_set_about, 293
 - fl_open_callback, 293
 - gl_texture_pile_height, 293
- mac.H, 1216, 1217
- MAC_USE_ACCENTS_MENU
 - FI_Widget, 927
- maintain_absolute_top_line_number
 - FI_Text_Display, 794
- maintaining_absolute_top_line_number
 - FI_Text_Display, 795
- make_current
 - FI_Gl_Window, 464
 - FI_Window, 969
- make_overlay_current
 - FI_Gl_Window, 464
- make_visible
 - FI_Browser, 359
- manage_item_destroy
 - FI_Tree_Item_Array, 899
- margin_height
 - FI_Tooltip, 833
- margin_width
 - FI_Tooltip, 833
- marginbottom
 - FI_Tree_Prefs, 904
- margins
 - FI_Paged_Device, 615
 - FI_PostScript_File_Device, 631
 - FI_Printer, 667
 - FI_System_Printer, 728
- mark
 - FI_Input_, 542
- math.h, 1220
- max_size
 - FI_RGB_Image, 688
- maximum
 - FI_Progress, 672
 - FI_Valuator, 908
- maximum_size
 - FI_Input_, 542, 543
- maxinum
 - FI_Spinner, 719
- maxsize
 - FI_Chart, 392
- measure
 - FI_Label, 559

- FI_Menu_Item, 590
- measure_deleted_lines
 - FI_Text_Display, 795
- measure_label
 - FI_Widget, 945
- measure_proportional_character
 - FI_Text_Display, 795
- measure_vline
 - FI_Text_Display, 796
- mediumarrow.h, 1245
- menu
 - FI_Menu_, 574
 - FI_Sys_Menu_Bar, 726
- MENU_WINDOW
 - FI_Widget, 927
- menubutton
 - FI_Input_Choice, 555
- middleline
 - FI_Browser, 360
- minimum
 - FI_Progress, 673
 - FI_Valuator, 908, 909
- mininum
 - FI_Spinner, 719
- minute
 - FI_Clock_Output, 404
- MODAL
 - FI_Widget, 927
- modal
 - FI_Window, 970
 - Windows handling functions, 215
- mode
 - FI_Color_Chooser, 408, 409
 - FI_Gl_Window, 464
 - FI_Menu_, 574
 - FI_Sys_Menu_Bar, 726
- move
 - FI_Browser, 360
 - FI_Tree_Item, 889, 890
 - FI_Tree_Item_Array, 899
- move_above
 - FI_Tree_Item, 890
- move_below
 - FI_Tree_Item, 890
- move_down
 - FI_Text_Display, 796
- move_into
 - FI_Tree_Item, 890
- move_left
 - FI_Text_Display, 796
- move_right
 - FI_Text_Display, 796
- move_up
 - FI_Text_Display, 796
- mTabDist
 - FI_Text_Buffer, 772
- mulelao.h, 1581
- Multithreading support functions, 272
 - awake, 273
 - lock, 273
 - thread_message, 274
 - unlock, 274
- mvalue
 - FI_Menu_, 574
- Name
 - FI_Preferences::Name, 984, 985
- names.h, 1221
- nchecked
 - FI_Check_Browser, 395
- NEW_FOLDER
 - FI_Native_File_Chooser, 602
- new_list
 - FI_Browser_, 376
- newUUID
 - FI_Preferences, 661
- next
 - FI_File_Icon, 438
 - FI_Menu_Item, 591
 - FI_Tree, 863
 - FI_Tree_Item, 891
 - FI_Wizard, 976
- next_char
 - FI_Text_Buffer, 767
- next_displayed
 - FI_Tree_Item, 891
- next_item
 - FI_Tree, 863
- next_selected_item
 - FI_Tree, 864
- next_sibling
 - FI_Tree_Item, 891
- next_visible
 - FI_Tree_Item, 891
- next_visible_item
 - FI_Tree, 865
- next_window
 - Windows handling functions, 216
- nitems
 - FI_Check_Browser, 395
- NO_OPTIONS
 - FI_Native_File_Chooser, 602
- NO_OVERLAY
 - FI_Widget, 927
- NOBORDER
 - FI_Widget, 927
- NON_MODAL
 - FI_Widget, 927
- NORMAL_CURSOR
 - FI_Text_Display, 781
- not_clipped
 - FI_Graphics_Driver, 485
 - FI_PostScript_Graphics_Driver, 645
- offset_line_starts
 - FI_Text_Display, 796
- open

- FI_Tree, [865](#), [866](#)
- open_toggle
 - FI_Tree, [866](#)
- opendeicon
 - FI_Tree_Prefs, [904](#)
- openicon
 - FI_Tree, [867](#)
 - FI_Tree_Prefs, [904](#)
- Option
 - FI_Native_File_Chooser, [602](#)
- option
 - FI, [330](#), [331](#)
- OPTION_ARROW_FOCUS
 - FI, [323](#)
- OPTION_DND_TEXT
 - FI, [323](#)
- OPTION_FNFC_USES_GTK
 - FI, [324](#)
- OPTION_LAST
 - FI, [324](#)
- OPTION_SHOW_TOOLTIPS
 - FI, [323](#)
- OPTION_VISIBLE_FOCUS
 - FI, [323](#)
- options
 - FI_Native_File_Chooser, [604](#)
- ORIENTATION
 - FI_Paged_Device, [614](#)
- origin
 - FI_Paged_Device, [615](#)
 - FI_PostScript_File_Device, [631](#)
 - FI_Printer, [668](#)
 - FI_System_Printer, [729](#)
- original
 - FI_Shared_Image, [709](#)
- ortho
 - FI_Gl_Window, [465](#)
- OUTPUT
 - FI_Widget, [927](#)
- output
 - FI_Widget, [946](#)
- outputfile
 - FI_Text_Buffer, [768](#)
- OVERRIDE
 - FI_Widget, [927](#)
- overstrike
 - FI_Text_Display, [797](#)
- own_colormap
 - FI, [331](#)
- Page_Format
 - FI_Paged_Device, [614](#)
- Page_Layout
 - FI_Paged_Device, [614](#)
- parent
 - FI_Tree_Item, [891](#)
 - FI_Widget, [946](#)
- paste
 - Selection & Clipboard functions, [232](#)
- picked
 - FI_Menu_, [574](#)
- pie
 - FI_Graphics_Driver, [485](#)
 - FI_PostScript_Graphics_Driver, [645](#)
- pixel_h
 - FI_Gl_Window, [465](#)
- pixel_w
 - FI_Gl_Window, [465](#)
- pixels_per_unit
 - FI_Gl_Window, [466](#)
- Pixmap
 - FI_FormsPixmap, [448](#)
- platform.H, [1222](#)
- point
 - FI_Graphics_Driver, [485](#)
 - FI_PostScript_Graphics_Driver, [645](#)
- polygon
 - FI_Graphics_Driver, [485](#)
 - FI_PostScript_Graphics_Driver, [645](#)
- pop_clip
 - FI_Graphics_Driver, [486](#)
 - FI_PostScript_Graphics_Driver, [646](#)
- popup
 - FI_Menu_Button, [582](#)
 - FI_Menu_Item, [591](#)
- POPUP1
 - FI_Menu_Button, [581](#)
- POPUP12
 - FI_Menu_Button, [581](#)
- POPUP123
 - FI_Menu_Button, [581](#)
- POPUP13
 - FI_Menu_Button, [581](#)
- POPUP2
 - FI_Menu_Button, [581](#)
- POPUP23
 - FI_Menu_Button, [581](#)
- POPUP3
 - FI_Menu_Button, [581](#)
- popup_buttons
 - FI_Menu_Button, [580](#)
- PORTRAIT
 - FI_Paged_Device, [614](#)
- position
 - FI_Browser_, [376](#), [377](#)
 - FI_Input_, [543](#)
 - FI_Text_Selection, [818](#)
 - FI_Tile, [823](#)
 - FI_Widget, [946](#)
- position_style
 - FI_Text_Display, [797](#)
- position_to_line
 - FI_Text_Display, [797](#)
- position_to_linecol
 - FI_Text_Display, [798](#)
- position_to_xy
 - FI_Text_Display, [798](#)

- precision
 - FI_Valuator, [909](#)
- prefs
 - FI_Tree_Item, [892](#)
- preset_file
 - FI_Native_File_Chooser, [605](#)
- prev
 - FI_Tree, [867](#)
 - FI_Tree_Item, [892](#)
- prev_char
 - FI_Text_Buffer, [768](#)
- prev_displayed
 - FI_Tree_Item, [892](#)
- prev_sibling
 - FI_Tree_Item, [892](#)
- prev_visible
 - FI_Tree_Item, [892](#)
- PREVIEW
 - FI_Native_File_Chooser, [602](#)
- preview
 - FI_File_Chooser, [433](#)
- print
 - FI_Device_Plugin, [419](#)
 - FI_Mac_App_Menu, [563](#)
- print_panel.h, [1245](#)
- print_widget
 - FI_Paged_Device, [616](#)
 - FI_Printer, [668](#)
- print_window
 - FI_Paged_Device, [616](#)
- print_window_part
 - FI_Paged_Device, [616](#)
 - FI_Printer, [669](#)
- printable_rect
 - FI_Paged_Device, [617](#)
 - FI_PostScript_File_Device, [632](#)
 - FI_Printer, [669](#)
 - FI_System_Printer, [729](#)
- pulldown
 - FI_Menu_Item, [591](#)
- push
 - FI_Tabs, [756](#)
- push_clip
 - FI_Graphics_Driver, [486](#)
 - FI_PostScript_Graphics_Driver, [646](#)
- push_no_clip
 - FI_Graphics_Driver, [486](#)
 - FI_PostScript_Graphics_Driver, [646](#)
- pushed
 - Events handling functions, [228](#)
- r
 - FI_Color_Chooser, [409](#)
- radio
 - FI_Menu_Item, [591](#)
- range
 - FI_Valuator, [909](#)
- readonly
 - FI_Input_, [544](#)
- readqueue
 - FI, [331](#)
- ready
 - FI, [332](#)
- recalc_scrollbars
 - FI_Scroll, [696](#)
- recalc_tree
 - FI_Tree, [868](#)
 - FI_Tree_Item, [893](#)
- rect
 - FI_Graphics_Driver, [486](#)
 - FI_PostScript_Graphics_Driver, [646](#)
- rectangle_capture
 - FI_Device_Plugin, [419](#)
- rectf
 - FI_Graphics_Driver, [486](#)
 - FI_PostScript_Graphics_Driver, [646](#)
- redisplay_range
 - FI_Text_Display, [799](#)
- redraw
 - FI_Widget, [947](#)
- redraw_label
 - FI_Widget, [947](#)
- redraw_line
 - FI_Browser_, [377](#)
- redraw_lines
 - FI_Browser_, [377](#)
- redraw_overlay
 - FI_GI_Window, [466](#)
 - FI_Overlay_Window, [609](#)
- refcount
 - FI_Shared_Image, [709](#)
- release
 - FI, [332](#)
 - FI_Shared_Image, [710](#)
- release_widget_pointer
 - Safe widget deletion support functions, [276](#)
- reload_scheme
 - FI, [332](#)
- remove
 - FI_Browser, [360](#)
 - FI_Check_Browser, [395](#)
 - FI_Group, [507](#)
 - FI_Menu_, [575](#)
 - FI_Sys_Menu_Bar, [726](#)
 - FI_Text_Buffer, [768](#)
 - FI_Tree, [868](#)
 - FI_Tree_Item_Array, [899](#)
- remove_
 - FI_Text_Buffer, [768](#)
- remove_check
 - FI, [332](#)
- remove_child
 - FI_Tree_Item, [893](#)
- remove_handler
 - Events handling functions, [229](#)
- remove_icon
 - FI_Browser, [361](#)

- remove_key_binding
 - FI_Text_Editor, [816](#)
- remove_system_handler
 - Events handling functions, [229](#)
- remove_timeout
 - FI, [332](#)
- removePlugin
 - FI_Plugin_Manager, [624](#)
- reparent
 - FI_Tree_Item, [893](#)
 - FI_Tree_Item_Array, [900](#)
- repeat_timeout
 - FI, [333](#)
- replace
 - FI_Chart, [392](#)
 - FI_Input_, [544](#)
 - FI_Menu_, [575](#)
 - FI_Sys_Menu_Bar, [726](#)
 - FI_Text_Buffer, [769](#)
 - FI_Tree_Item, [893](#)
 - FI_Tree_Item_Array, [900](#)
- replace_child
 - FI_Tree_Item, [894](#)
- replacing
 - FI_Browser_, [377](#)
- reset_absolute_top_line_number
 - FI_Text_Display, [799](#)
- resizable
 - FI_Group, [508](#)
- resize
 - FI_Browser_, [378](#)
 - FI_Double_Window, [425](#)
 - FI_Gl_Window, [466](#)
 - FI_Group, [508](#)
 - FI_Help_View, [519](#)
 - FI_Input_, [545](#)
 - FI_Input_Choice, [555](#)
 - FI_Overlay_Window, [609](#)
 - FI_Scroll, [696](#)
 - FI_Spinner, [719](#)
 - FI_Table, [744](#)
 - FI_Text_Display, [799](#)
 - FI_Tile, [823](#)
 - FI_Tree, [868](#)
 - FI_Value_Input, [912](#)
 - FI_Widget, [947](#)
 - FI_Window, [970](#)
- REVERSED
 - FI_Paged_Device, [614](#)
- rewind_lines
 - FI_Text_Buffer, [769](#)
 - FI_Text_Display, [799](#)
- rgb
 - FI_Color_Chooser, [409](#)
- rgb2hsv
 - FI_Color_Chooser, [409](#)
- RGB_scaling
 - FI_Image, [528](#)
- Root
 - FI_Preferences, [653](#)
- root
 - FI_Tree, [868](#)
- root_label
 - FI_Tree, [868](#)
- rotate
 - FI_Paged_Device, [617](#)
 - FI_PostScript_File_Device, [632](#)
 - FI_Printer, [669](#)
 - FI_System_Printer, [730](#)
- round
 - FI_Valuator, [909](#)
- row_header
 - FI_Table, [744](#)
- row_height
 - FI_Table, [744](#)
- row_height_all
 - FI_Table, [744](#)
- row_resize
 - FI_Table, [745](#)
- row_resize_min
 - FI_Table, [745](#)
- row_selected
 - FI_Table_Row, [750](#)
- rows
 - FI_Table, [745](#)
 - FI_Table_Row, [750](#)
- rtl_draw
 - FI_GDI_Graphics_Driver, [456](#)
 - FI_Graphics_Driver, [486](#)
 - FI_PostScript_Graphics_Driver, [646](#)
 - FI_Quartz_Graphics_Driver, [678](#)
 - FI_Xlib_Graphics_Driver, [982](#)
- run
 - FI, [333](#)
- Safe widget deletion support functions, [274](#)
 - clear_widget_pointer, [275](#)
 - delete_widget, [275](#)
 - do_widget_deletion, [275](#)
 - release_widget_pointer, [276](#)
 - watch_widget_pointer, [276](#)
- saturation
 - FI_Color_Chooser, [411](#)
- SAVEAS_CONFIRM
 - FI_Native_File_Chooser, [602](#)
- savefile
 - FI_Text_Buffer, [769](#)
- scale
 - FI_Paged_Device, [617](#)
 - FI_PostScript_File_Device, [632](#)
 - FI_Printer, [670](#)
 - FI_Shared_Image, [710](#)
 - FI_System_Printer, [730](#)
- scaling_algorithm
 - FI_Shared_Image, [710](#)
- scheme
 - FI, [333](#)

- Screen functions, [233](#)
 - [screen_dpi](#), [234](#)
 - [screen_num](#), [234](#), [236](#)
 - [screen_work_area](#), [236](#)
 - [screen_xywh](#), [237](#), [238](#)
- [screen_dpi](#)
 - Screen functions, [234](#)
- [screen_num](#)
 - Screen functions, [234](#), [236](#)
- [screen_work_area](#)
 - Screen functions, [236](#)
- [screen_xywh](#)
 - Screen functions, [237](#), [238](#)
- [scroll](#)
 - [FI_Text_Display](#), [800](#)
- [scroll_](#)
 - [FI_Text_Display](#), [800](#)
- [scroll_timer_cb](#)
 - [FI_Text_Display](#), [800](#)
- [scroll_to](#)
 - [FI_Scroll](#), [697](#)
- [scrollbar](#)
 - [FI_Browser_](#), [381](#)
- [scrollbar_align](#)
 - [FI_Text_Display](#), [800](#), [801](#)
- [scrollbar_left](#)
 - [FI_Browser_](#), [378](#)
- [scrollbar_right](#)
 - [FI_Browser_](#), [378](#)
- [scrollbar_size](#)
 - [FI](#), [333](#), [334](#)
 - [FI_Browser_](#), [378](#)
 - [FI_Help_View](#), [519](#)
 - [FI_Scroll](#), [697](#)
 - [FI_Table](#), [745](#)
 - [FI_Tree](#), [869](#)
- [scrollbar_width](#)
 - [FI_Browser_](#), [379](#)
 - [FI_Text_Display](#), [801](#)
- [scrollvalue](#)
 - [FI_Slider](#), [715](#)
- [search_backward](#)
 - [FI_Text_Buffer](#), [769](#)
- [search_forward](#)
 - [FI_Text_Buffer](#), [770](#)
- [second](#)
 - [FI_Clock_Output](#), [404](#)
- [secondary_selection_text](#)
 - [FI_Text_Buffer](#), [770](#)
- [select](#)
 - [FI_Browser](#), [361](#)
 - [FI_Browser_](#), [379](#)
 - [FI_Tree](#), [869](#), [870](#)
 - [FI_Tree_Item](#), [894](#)
- [select_all](#)
 - [FI_Tree](#), [870](#)
 - [FI_Tree_Item](#), [894](#)
- [select_all_rows](#)
 - [FI_Table_Row](#), [750](#)
- [select_only](#)
 - [FI_Browser_](#), [380](#)
 - [FI_Tree](#), [871](#)
- [select_row](#)
 - [FI_Table_Row](#), [750](#)
- [select_toggle](#)
 - [FI_Tree](#), [871](#)
- [selectbox](#)
 - [FI_Tree](#), [872](#)
- [selected](#)
 - [FI_Browser](#), [361](#)
 - [FI_Text_Selection](#), [819](#)
- [selection](#)
 - [FI_Browser_](#), [380](#)
 - Selection & Clipboard functions, [233](#)
- Selection & Clipboard functions, [230](#)
 - [add_clipboard_notify](#), [231](#)
 - [clipboard_contains](#), [231](#)
 - [copy](#), [231](#)
 - [dnd](#), [232](#)
 - [paste](#), [232](#)
 - [selection](#), [233](#)
 - [selection_owner](#), [233](#)
- [selection_color](#)
 - [FI_Widget](#), [947](#), [948](#)
- [selection_owner](#)
 - Selection & Clipboard functions, [233](#)
- [selection_text](#)
 - [FI_Text_Buffer](#), [770](#)
- [selectmode](#)
 - [FI_Tree](#), [872](#)
 - [FI_Tree_Prefs](#), [904](#)
- [set](#)
 - [FI_Button](#), [385](#)
 - [FI_FormsBitmap](#), [446](#)
 - [FI_FormsPixmap](#), [448](#)
 - [FI_Menu_Item](#), [592](#)
 - [FI_Preferences](#), [661–663](#)
 - [FI_Text_Selection](#), [819](#)
- [set_active](#)
 - [FI_Widget](#), [948](#)
- [set_atclose](#)
 - Windows handling functions, [216](#)
- [set_box_color](#)
 - [FI](#), [334](#)
- [set_changed](#)
 - [FI_Widget](#), [948](#)
- [set_checked](#)
 - [FI_Check_Browser](#), [395](#)
- [set_color](#)
 - Color & Font functions, [247](#)
- [set_current](#)
 - [FI_Copy_Surface](#), [413](#)
 - [FI_Image_Surface](#), [531](#)
 - [FI_Printer](#), [670](#)
 - [FI_Surface_Device](#), [721](#)
- [set_draw_cb](#)

- FI_Cairo_Window, 388
- set_font
 - Color & Font functions, 247
- set_fonts
 - Color & Font functions, 247
- set_idle
 - FI, 334
- set_item_focus
 - FI_Tree, 872
- set_menu_window
 - FI_Window, 970
- set_modal
 - FI_Window, 970
- set_non_modal
 - FI_Window, 970
- set_output
 - FI_Widget, 948
- set_overlay
 - FI_Menu_Window, 594
- set_selection
 - FI_Table, 746
- set_tooltip_window
 - FI_Window, 970
- set_visible
 - FI_Widget, 948
- set_visible_focus
 - FI_Widget, 948
- setonly
 - FI_Menu_Item, 592
- shape
 - FI_Window, 971
- shortcut
 - FI_Button, 385
 - FI_Input_, 545, 546
 - FI_Menu_Item, 592
 - FI_Text_Display, 801
 - FI_Value_Input, 913
- SHORTCUT_LABEL
 - FI_Widget, 927
- show
 - FI_Browser, 361, 362
 - FI_Double_Window, 425
 - FI_GI_Window, 466
 - FI_Help_Dialog, 512
 - FI_Menu_Window, 594
 - FI_Native_File_Chooser, 605
 - FI_Overlay_Window, 610
 - FI_Single_Window, 712
 - FI_Widget, 949
 - FI_Window, 972
- show_cursor
 - FI_Text_Display, 802
- show_insert_position
 - FI_Text_Display, 802
- show_item
 - FI_Tree, 872, 873
- show_item_bottom
 - FI_Tree, 873
- show_item_middle
 - FI_Tree, 873
- show_item_top
 - FI_Tree, 873
- show_self
 - FI_Tree, 874
 - FI_Tree_Item, 895
- show_widgets
 - FI_Tree_Item, 895
- showcollapse
 - FI_Tree, 874
 - FI_Tree_Prefs, 904
- showHiddenButton
 - FI_File_Chooser, 433
- shown
 - FI_Window, 973
- showroot
 - FI_Tree, 874
 - FI_Tree_Prefs, 905
- SIMPLE_CURSOR
 - FI_Text_Display, 781
- size
 - FI_Browser, 362
 - FI_Input_, 546
 - FI_Menu_, 575
 - FI_Menu_Item, 592
 - FI_Preferences, 664
 - FI_Tooltip, 833
 - FI_Widget, 949
- size_range
 - FI_Window, 973
- sizes
 - FI_Group, 509
- skip_displayed_characters
 - FI_Text_Buffer, 770
- skip_lines
 - FI_Text_Display, 802
- slider_size
 - FI_Slider, 716
- slowarrow.h, 1246
- soft
 - FI_Adjuster, 339
 - FI_Value_Input, 913
 - FI_Value_Output, 916
- sort
 - FI_Browser_, 380
- sortorder
 - FI_Tree, 875
 - FI_Tree_Prefs, 905
- spacing.h, 1275
- start
 - FI_Text_Selection, 819
- start_job
 - FI_Paged_Device, 617
 - FI_PostScript_File_Device, 633
 - FI_PostScript_Printer, 650
 - FI_Printer, 670
 - FI_System_Printer, 730

- start_page
 - FI_Paged_Device, 618
 - FI_PostScript_File_Device, 634
 - FI_Printer, 671
 - FI_System_Printer, 731
- static_value
 - FI_Input_, 546, 547
- step
 - FI_Counter, 416
 - FI_Spinner, 719
 - FI_Valuator, 909
- STRICT_RFC3629
 - Unicode and UTF-8 functions, 280
- string_width
 - FI_Text_Display, 802
- submenu
 - FI_Menu_Item, 592
- surface
 - FI_Surface_Device, 721
- suspended
 - FI_Timer, 828
- swap
 - FI_Browser, 362
- swap_buffers
 - FI_Gl_Window, 467
- swap_children
 - FI_Tree_Item, 895
- swapping
 - FI_Browser_, 380
- symbol_.h, 1298
- SYSTEM
 - FI_Preferences, 653
- tab_cell_nav
 - FI_Table, 746
- tab_distance
 - FI_Text_Buffer, 771
- tab_nav
 - FI_Input_, 547
 - FI_Text_Editor, 816
- table_box
 - FI_Table, 747
- TableContext
 - FI_Table, 738
- take_focus
 - FI_Widget, 949
- takesevents
 - FI_Widget, 949
- tatar_cyr.h, 1582
- tcvn.h, 1583
- test_shortcut
 - Events handling functions, 229
 - FI_Menu_, 575
 - FI_Menu_Item, 592
 - FI_Widget, 950
- text
 - FI_Browser, 363
 - FI_Menu_, 575
 - FI_Text_Buffer, 771
- text_extents
 - FI_GDI_Graphics_Driver, 456
 - FI_Graphics_Driver, 487
 - FI_PostScript_Graphics_Driver, 647
 - FI_Quartz_Graphics_Driver, 678
 - FI_Xlib_Graphics_Driver, 982
- text_range
 - FI_Text_Buffer, 771
- textcolor
 - FI_Input_, 549
 - FI_Menu_, 576
 - FI_Text_Display, 803
 - FI_Tooltip, 833
 - FI_Value_Input, 913
 - FI_Value_Output, 916, 917
 - FI_Value_Slider, 919
- textfont
 - FI_Browser_, 381
 - FI_Input_, 549
 - FI_Menu_, 576
 - FI_Text_Display, 803
 - FI_Value_Input, 913, 914
 - FI_Value_Output, 917
 - FI_Value_Slider, 919, 920
- textsize
 - FI_Browser, 363
 - FI_Help_Dialog, 512
 - FI_Input_, 550
 - FI_Menu_, 576
 - FI_Text_Display, 803, 804
 - FI_Value_Input, 914
 - FI_Value_Output, 917
 - FI_Value_Slider, 920
- thread_message
 - Multithreading support functions, 274
- tis620.h, 1585
- title
 - FI_Native_File_Chooser, 605
- tooltip
 - FI_Widget, 950, 951
- TOOLTIP_WINDOW
 - FI_Widget, 927
- top_row
 - FI_Table, 747
- top_window
 - FI_Widget, 951
- top_window_offset
 - FI_Widget, 951
- topline
 - FI_Browser, 363, 364
 - FI_Help_View, 519, 520
- transcoding_warning_action
 - FI_Text_Buffer, 772
- transformed_vertex
 - FI_Graphics_Driver, 487
 - FI_PostScript_Graphics_Driver, 647
- translate
 - FI_Paged_Device, 618

- FI_PostScript_File_Device, [634](#)
 - FI_Printer, [671](#)
 - FI_System_Printer, [731](#)
- tree
 - FI_Tree_Item, [895](#), [896](#)
- Type
 - FI_Native_File_Chooser, [602](#)
- type
 - FI_File_Icon, [438](#)
 - FI_Label, [559](#)
 - FI_Spinner, [719](#)
 - FI_Table_Row, [750](#)
 - FI_Widget, [952](#)
- typea
 - FI_Multi_Label, [597](#)
- typeb
 - FI_Multi_Label, [597](#)
- ucs2be.h, [1586](#)
- uncache
 - FI_Bitmap, [341](#)
 - FI_Image, [528](#)
 - FI_Pixmap, [621](#)
 - FI_RGB_Image, [689](#)
 - FI_Shared_Image, [710](#)
- uncheck
 - FI_Menu_Item, [593](#)
- undo
 - FI_Input_, [550](#)
- Unicode and UTF-8 functions, [278](#)
 - ERRORS_TO_CP1252, [280](#)
 - ERRORS_TO_ISO8859_1, [280](#)
 - fl_access, [280](#)
 - fl_chmod, [281](#)
 - fl_fopen, [281](#)
 - fl_getcwd, [282](#)
 - fl_getenv, [282](#)
 - fl_make_path, [282](#)
 - fl_make_path_for_file, [282](#)
 - fl_mkdir, [282](#)
 - fl_nonspacing, [283](#)
 - fl_open, [283](#)
 - fl_rename, [283](#)
 - fl_rmdir, [284](#)
 - fl_stat, [284](#)
 - fl_system, [285](#)
 - fl_ucs_to_Utf16, [285](#)
 - fl_unlink, [285](#)
 - fl_utf8back, [286](#)
 - fl_utf8bytes, [286](#)
 - fl_utf8decode, [286](#)
 - fl_utf8encode, [286](#)
 - fl_utf8from_mb, [287](#)
 - fl_utf8froma, [287](#)
 - fl_utf8fromwc, [287](#)
 - fl_utf8fwd, [288](#)
 - fl_utf8len, [288](#)
 - fl_utf8len1, [288](#)
 - fl_utf8locale, [288](#)
 - fl_utf8test, [288](#)
 - fl_utf8to_mb, [289](#)
 - fl_utf8toa, [289](#)
 - fl_utf8toUtf16, [289](#)
 - fl_utf8towc, [290](#)
 - fl_utf_strcasecmp, [290](#)
 - fl_utf_strncasecmp, [290](#)
 - fl_utf_tolower, [291](#)
 - fl_utf_toupper, [291](#)
 - fl_wcwidth, [291](#)
 - fl_wcwidth_, [292](#)
 - STRICT_RFC3629, [280](#)
- unlock
 - Multithreading support functions, [274](#)
- untranslate
 - FI_Paged_Device, [618](#)
 - FI_PostScript_File_Device, [634](#)
 - FI_Printer, [671](#)
 - FI_System_Printer, [731](#)
- up_down_position
 - FI_Input_, [550](#)
- update
 - FI_Text_Selection, [819](#)
- update_child
 - FI_Group, [509](#)
- update_h_scrollbar
 - FI_Text_Display, [804](#)
- update_line_starts
 - FI_Text_Display, [804](#)
- update_prev_next
 - FI_Tree_Item, [896](#)
- update_v_scrollbar
 - FI_Text_Display, [804](#)
- USE_FILTER_EXT
 - FI_Native_File_Chooser, [602](#)
- use_high_res_GL
 - FI, [334](#)
- USER
 - FI_Preferences, [653](#)
- user_data
 - FI_Widget, [952](#)
- userdeicon
 - FI_Tree_Item, [896](#)
 - FI_Tree_Prefs, [905](#)
- USERFLAG1
 - FI_Widget, [927](#)
- USERFLAG2
 - FI_Widget, [927](#)
- USERFLAG3
 - FI_Widget, [927](#)
- usericon
 - FI_Tree, [875](#)
 - FI_Tree_Item, [896](#)
- utf8.h, [1586](#)
- valid
 - FI_GI_Window, [467](#)
- value
 - FI_Browser, [364](#)

- FI_Button, [385](#)
- FI_Choice, [399](#)
- FI_Clock_Output, [405](#)
- FI_Color_Chooser, [411](#)
- FI_File_Chooser, [433](#)
- FI_File_Input, [441](#)
- FI_Help_Dialog, [512](#)
- FI_Help_View, [520](#)
- FI_Input_, [550](#), [551](#)
- FI_Input_Choice, [556](#)
- FI_Menu_, [576](#)
- FI_Menu_Item, [593](#)
- FI_Progress, [673](#)
- FI_Scrollbar, [700](#), [701](#)
- FI_Spinner, [720](#)
- FI_Tabs, [756](#)
- FI_Valuator, [909](#)
- value_damage
 - FI_Adjuster, [339](#)
 - FI_Valuator, [910](#)
- version
 - FI, [335](#)
- vertex
 - FI_Graphics_Driver, [487](#)
 - FI_PostScript_Graphics_Driver, [647](#)
- VERTICAL
 - FI_Browser_, [368](#)
- VERTICAL_ALWAYS
 - FI_Browser_, [368](#)
- viscii.h, [1587](#)
- visible
 - FI_Browser, [364](#)
 - FI_Widget, [952](#)
- visible_cells
 - FI_Table, [747](#)
- VISIBLE_FOCUS
 - FI_Widget, [927](#)
- visible_focus
 - FI, [335](#)
 - FI_Widget, [953](#)
- visible_r
 - FI_Tree_Item, [897](#)
 - FI_Widget, [953](#)
- visual
 - FI, [335](#)
- vline_length
 - FI_Text_Display, [804](#)
- vposition
 - FI_Tree, [875](#)
- w
 - FI_Widget, [953](#)
- wait
 - FI, [335](#)
- wait_for_expose
 - FI_Window, [973](#)
- warning
 - Common Dialogs classes and functions, [305](#)
- watch_widget_pointer
 - Safe widget deletion support functions, [276](#)
- when
 - FI_Table, [747](#)
 - FI_Widget, [954](#)
- which
 - FI_Tabs, [757](#)
- widget
 - FI_Widget_Tracker, [957](#)
- width
 - FI_GDI_Graphics_Driver, [456](#)
 - FI_Graphics_Driver, [487](#)
 - FI_PostScript_Graphics_Driver, [647](#)
 - FI_Quartz_Graphics_Driver, [678](#)
 - FI_Xlib_Graphics_Driver, [982](#)
- win32.H, [1222](#)
- window
 - FI_Widget, [954](#)
- Windows handling functions, [214](#)
 - atclose, [216](#)
 - default_atclose, [215](#)
 - first_window, [215](#)
 - grab, [215](#)
 - modal, [215](#)
 - next_window, [216](#)
 - set_atclose, [216](#)
- word_end
 - FI_Input_, [552](#)
 - FI_Text_Buffer, [772](#)
 - FI_Text_Display, [805](#)
- word_start
 - FI_Input_, [552](#)
 - FI_Text_Buffer, [772](#)
 - FI_Text_Display, [805](#)
- wrap
 - FI_Input_, [552](#)
- WRAP_AT_BOUNDS
 - FI_Text_Display, [781](#)
- WRAP_AT_COLUMN
 - FI_Text_Display, [781](#)
- WRAP_AT_PIXEL
 - FI_Text_Display, [781](#)
- wrap_mode
 - FI_Text_Display, [805](#)
- WRAP_NONE
 - FI_Text_Display, [781](#)
- wrap_uses_character
 - FI_Text_Display, [806](#)
- wrap_width
 - FI_Tooltip, [834](#)
- wrapped_column
 - FI_Text_Display, [806](#)
- wrapped_line_counter
 - FI_Text_Display, [807](#)
- wrapped_row
 - FI_Text_Display, [807](#)
- x
 - FI_Widget, [955](#)
- x.H, [1224](#)

x_to_col
 FI_Text_Display, [808](#)
xclass
 FI_Window, [974](#)
Ximint.h, [1589](#)
Xlibint.h, [1589](#)
xposition
 FI_Scroll, [698](#)
Xutf8.h, [1246](#)
xy_to_position
 FI_Text_Display, [808](#)
xy_to_rowcol
 FI_Text_Display, [808](#)
xyline
 FI_Graphics_Driver, [487](#), [488](#)
 FI_PostScript_Graphics_Driver, [647](#), [648](#)

y
 FI_Widget, [955](#)
yposition
 FI_Scroll, [698](#)
yxline
 FI_Graphics_Driver, [488](#)
 FI_PostScript_Graphics_Driver, [648](#)