The **Here Applies** LaTeX Package

December 11, 2022

Abstract

A LaTeX package for referencing groups of pages that share something in common.

1 Overview

Here Applies is a LaTeX package that allows to collect groups of labels and reference them altogether. It can be used for creating informal glossaries that cross-link concepts to their applications, or simply mentioning multiple pages that share something in common.

The package offers two commands: `\hereapplies` and `\whereapplies` (plus their "starred" versions `\hereapplies*` and `\whereapplies*`). In both cases an identifier is passed as argument – and this can be any string invented in the moment (`\hereapplies` additionally supports more than one identifier in the form of a comma-separated list).

Every time `\hereapplies` is invoked with known identifiers, the document is made aware that the place shares some kind of connection with other places in which the same identifiers were used. And so, every time the `\whereapplies` command is invoked with a known identifier, all the occurrences of the latter within the entire document will be printed in the form of a linkable page list (e.g. "pp. 1, 5, 8–9, 14–20…").

As `\hereapplies` is designed to be invoked in the middle of a chapter or a section and that location must be made linkable, the `\phantomsection` directive is invoked by default before a label is added. To avoid calling `\phantomsection`, the "starred" command `\hereapplies*` is available.

Finally, like `\whereapplies` resembles a pluralizable version of `\pageref`, its "starred" version `\whereapplies*` will resemble a pluralizable version of `\pageref*`.

If you use LyX, the package ships a LyX module as well (please check the `lyx-module` subdirectory).

2 Example usage

The following LaTeX manuscript

```
\documentclass{article}

\usepackage{hereapplies}

\begin{document}

\title{Some title}

\author{Some author}

\maketitle

This is concept one. To find this concept applied, please
see \whereapplies{conceptOne}.

This is concept two. To find this concept applied, please
see \whereapplies{conceptTwo}.\newpage

\hereapplies{conceptOne} This is page \thepage. As you can see,
``concept one'' applies here.\newpage

\hereapplies{conceptTwo} This is page \thepage. As you can see,
``concept two'' applies here.\newpage

\hereapplies{conceptOne, conceptTwo} This is page \thepage. As you
can see, both ``concept one'' and ``concept two'' apply here.\newpage

\hereapplies{conceptTwo} This is page \thepage. As you can see,
``concept two'' applies here.\newpage

\hereapplies[myref]{conceptOne} This is page \thepage. As you can
see, ``concept one'' applies here. This point in the document is
labeled \texttt{myref}.

\end{document}
```

will generate the `hereapplies-example.pdf` document attached.

3 A minimal tutorial

`\hereapplies` Syntax:

`\hereapplies [`⟨*label*⟩`] {`⟨*identifiers*⟩`}`
`\hereapplies* [`⟨*label*⟩`] {`⟨*identifiers*⟩`}`

The `\hereapplies` command notifies the document that one or more identifiers apply to a particular point and adds a label to it.

If the optional argument is passed the label created will be named accordingly, otherwise an opaque name will be chosen for it. This argument may contain only what is legal for `\pageref`.

The *identifiers* argument must be a comma-separated list of identifiers (leading and trailing spaces around each member will be ignored). Each of these strings will remain confined within the internal scope of the package and will not create conflicts with possible macros or labels of the same names.

After storing some internal values, `\hereapplies` will expand exactly to

```
\phantomsection\label{...}
```

Its "starred" version (`\hereapplies*`) will not invoke the `\phantomsection` directive.

`\whereapplies` Syntax:

`\whereapplies {`⟨*identifier*⟩`}`
`\whereapplies* {`⟨*identifier*⟩`}`

The `\whereapplies` command prints all the occurrences of an identifier, in the form "p. ..." or "pp. ..." (with page range support).

The *identifier* argument will remain confined within the internal scope of the package and will not create conflicts with possible commands or labels of the same name. Leading and trailing spaces around this string will be ignored.

If the same *identifier* is not passed to `\hereapplies` at least once throughout the document, `\whereapplies` will print "**??**".

The "starred" version of this command (`\whereapplies*`) will use `\pageref*` instead of `\pageref` for generating the page list.

4 Internationalization

Currently the localization of **Here Applies** is not automatic. It is possible however to control the strings generated by overwriting the four macros `\hapage`, `\hapages`, `\hadelimiter` and `\halastdelimiter`. For example, writing at the beginning of a document

```
1 % German translation of **Here Applies**
2 % English: "p.\ "
3 \gdef\hapage{S.\ }
4 % English: "pp.\ "
5 \gdef\hapages{S.\ }
6 % English: "\ and\ "
7 \gdef\halastdelimiter{\ und\ }
8 % English: ",\ " (exactly like in German --- leave it)
9 %\gdef\hadelimiter{,\ }
```

will translate "pp. 2, 4 and 6" into "S. 2, 4 und 6".

5 Get involved

If you wish to get involved, please do not hesitate to send merge requests or participate in the discussion. The package is also available on **CTAN** under `macros/latex/contrib/hereapplies/`. For any issue, please drop a message.

6 Free software

Here Applies is free software. You can redistribute it and/or modify it under the terms of the **AGPL** license version 3 or any later version. See `COPYING` for details.

Code appendix

```
% -*- Mode: latex; indent-tabs-mode: nil; c-basic-offset: 4; tab-width: 4 -*-
%
%
% hereapplies.sty
%
% A LaTeX package for referencing groups of pages that share something in
% common
%
% https://github.com/madmurphy/hereapplies.sty
%
% Version 1.0.1
%
% Copyright (C) 2022 madmurphy <madmurphy333@gmail.com>
%
% **Here Applies** is free software: you can redistribute it and/or modify it
% under the terms of the GNU Affero General Public License as published by the
% Free Software Foundation, either version 3 of the License, or (at your
% option) any later version.
%
% **Here Applies** is distributed in the hope that it will be useful, but
% WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
% FITNESS FOR A PARTICULAR PURPOSE. See the GNU Affero General Public License
% for more details.
%
% You should have received a copy of the GNU Affero General Public License
% along with this program. If not, see <http://www.gnu.org/licenses/>.
%
%
%
% Example usage:
%
%     \documentclass{article}
%
%     \usepackage{hereapplies}
%
%     \begin{document}
%
%     \title{Some title}
%
%     \author{Some author}
%
%     \maketitle
%
%     This is concept one. To find this concept applied, please
%     see \whereapplies{conceptOne}.
%
%     This is concept two. To find this concept applied, please
%     see \whereapplies{conceptTwo}.\newpage
%
%     \hereapplies{conceptOne} This is page \thepage. As you can see,
%     ``concept one'' applies here.\newpage
%
%     \hereapplies{conceptTwo} This is page \thepage. As you can see,
```

```
%     ``concept two'' applies here.\newpage
%
%     \hereapplies{conceptOne, conceptTwo} This is page \thepage. As you
%     can see, both ``concept one'' and ``concept two'' apply here.\newpage
%
%     \hereapplies{conceptTwo} This is page \thepage. As you can see,
%     ``concept two'' applies here.\newpage
%
%     \hereapplies[myref]{conceptOne} This is page \thepage. As you can
%     see, ``concept one'' applies here. This point in the document is
%     labeled \texttt{myref}.
%
%     \end{document}
%
%
\ProvidesPackage{hereapplies}[2022/12/11 Here Applies]
\RequirePackage{hyperref}
\RequirePackage{refcount}
%
%
%
%          TRANSLATABLE STRINGS
%          ====================
%
%
% The abbreviation of one single page
\providecommand*{\hapage}{p.\ }
% The abbreviation of two or more pages
\providecommand*{\hapages}{pp.\ }
% The delimiter between page numbers
\providecommand*{\hadelimiter}{,\ }
% The delimiter before the last page number
\providecommand*{\halastdelimiter}{\ and\ }
%
%
%
%          ABSTRACT UTILITIES
%          ==================
%
% These macros are not strictly related to this package, but are required.
%
%
% Macro: `\@ha@ifcomma text to check,\@then{if yes}{if no}`
% ********************************************************************************
%
% Check if a string contains a comma
%
% This macro is mainly for internal purposes (but nothing forbids invoking it
% directly). When invoked it checks whether a comma is present in `text to
% check`, then expands to `if yes` or `if no` accordingly.
%
% Please do not put curly brackets around the text to check. The comma at the
% end of the text is mandatory.
%
\long\gdef\@ha@ifcomma#1,#2\@then#3#4{%
```

```
    \if\relax\detokenize{#2}\relax#4\else#3\fi%
}
%
%
% Macro: `\ha@trim{text}`
% ******************************************************************************
%
% Trim leading and trailing spaces from a string
%
% This macro is mainly for internal purposes (but nothing forbids invoking it
% directly).
%
\begingroup
% Temporarily change the categories of `<` and `>`, for trimming safely
\catcode`\<=4\catcode`\>=3
% Helper macro
\long\gdef\@ha@trm< #1 >< #2 >< #3 >< #4 >< #5 >< #6 >< #7 >< #8 >< #9 >/{%
    \ifcase\numexpr2#3#8\relax\or#2\or#7\or#5\or#1\fi%
}
% Usable macro
\long\gdef\ha@trim#1{%
    \@ha@trm< #1 >< #1>< - >< + >< ? ><#1 ><#1>< 0 >< 2 >< 1 >< 3 >< 2 >< ! >/%
}
\endgroup
%
%
%
%          PRIVATE ENVIRONMENT
%          ===================
%
% These macros regulate the internal functioning of the package and should not
% be invoked directly.
%
%
% Assign a unique number to each unlabeled occurrence of an identifier
\newcounter{@ha@unlabeled@counter}
% Populate the .hax file when the document reaches the end
\AtEndDocument{%
    % Do we have any content?
    \ifdefined\@ha@commons@@haxcontent%
        % We do - export it
        \addtocontents{hax}{\@ha@commons@@haxcontent}%
    \fi%
}
%
%
% Macro `\@ha@makepagelist{hypermacro}{labels}`
% ******************************************************************************
%
% Generate the list of page numbers (with page range support)
%
% This macro is for internal purposes only. When invoked, it scans the
% comma-separated list of labels provided (`labels`), checks which labels refer
% to duplicate page numbers and which page numbers can be grouped together, and
% finally prints a list.
```

```
%
% The `hypermacro` argument is the macro (usually from the `hyperref` package)
% that will process the label name.
%
% The `labels` argument must be a comma-separated list of labels.
%
\gdef\@ha@makepagelist#1#2{%
    \begingroup%
    % Reset the current page number
    \def\@ha@tmp@@currp{-1}%
    % Reset the current range offset
    \def\@ha@tmp@@prangeoffs{-1}%
    % Ensure no comma before the first page number
    \def\@ha@tmp@@psep{}%
    % Ensure no text before the last number if it is also the first one
    \def\@ha@tmp@@lastpsep{}%
    % Iterate through the `labels` argument
    \@for\@ha@tmp@@thislabel:=#2\do{%
        % Store the page number associated with this label
        \edef\@ha@tmp@@nextp{\getpagerefnumber{\@ha@tmp@@thislabel}}%
        % Check that we are not on the same page as in the last iteration
        \ifnum\@ha@tmp@@currp=\@ha@tmp@@nextp\else%
            % This is not the same page as in the last iteration
            % Is this the first page in which this identifier appears?
            \ifnum\@ha@tmp@@currp>-1%
                % We have already met pages in which this identifiers appears
                % Does this page follow immediately the previous page?
                \ifnum\numexpr\@ha@tmp@@currp+1=\@ha@tmp@@nextp%
                    % This page follows immediately the previous page
                    % Are these the first two contiguous pages of the range?
                    \ifnum\@ha@tmp@@prangeoffs=-1%
                        % These are the first two contiguous pages of the range
                        % Store the first page number of the pair
                        \let\@ha@tmp@@prangeoffs\@ha@tmp@@currp%
                        % Store the first label of the pair
                        \let\@ha@tmp@@currrangelbl\@ha@tmp@@currlbl%
                    \fi%
                \else%
                    % This page is far from the previous label's page
                    % Was the previous page part of a contiguous range?
                    \ifnum\@ha@tmp@@prangeoffs=-1%
                        % The previous page was a standalone page
                        % Print "[, ]<p>"
                        {\@ha@tmp@@psep\csname
                            #1\expandafter\endcsname%
                            \expandafter{\@ha@tmp@@currlbl}}%
                    \else%
                        % The previous page was part of a contiguous range
                        % Print "[, ]<p--q>"
                        {\@ha@tmp@@psep\csname
                            #1\expandafter\endcsname%
                            \expandafter{\@ha@tmp@@currrangelbl}--\csname
                            #1\expandafter\endcsname%
                            \expandafter{\@ha@tmp@@currlbl}}%
                        % Reset the current range offset
```

```
                        \def\@ha@tmp@@prangeoffs{-1}%
                    \fi%
                    % Ensure a comma before the next page number
                    \let\@ha@tmp@@psep\hadelimiter%
                    % Ensure " and " before the last page number
                    \let\@ha@tmp@@lastpsep\halastdelimiter%
                \fi%
            \fi%
            % Prepare the next page number
            \let\@ha@tmp@@currp\@ha@tmp@@nextp%
            % Prepare the next label
            \let\@ha@tmp@@currlbl\@ha@tmp@@thislabel%
        \fi%
    }%
    % Print the last page number
    % Is there at least one page to print?
    \ifnum\@ha@tmp@@currp>-1%
        % There is at least one page to print
        % Was the previous page part of a contiguous range?
        \ifnum\@ha@tmp@@prangeoffs=-1%
            % The previous page was a standalone page
            % Print "[ and ]<p>"
            {\@ha@tmp@@lastpsep\csname
                #1\expandafter\endcsname%
                \expandafter{\@ha@tmp@@currlbl}}%
        \else%
            % The previous page was part of a contiguous range
            % Print "[ and ]<p--q>"
            {\@ha@tmp@@lastpsep\csname
                #1\expandafter\endcsname%
                \expandafter{\@ha@tmp@@currrangelbl}--\csname
                #1\expandafter\endcsname%
                \expandafter{\@ha@tmp@@currlbl}}%
        \fi%
    \fi%
    \endgroup%
}
%
%
% Macro `\@ha@makeoutputstrings{identifier}{preamble}{labels}`
% ******************************************************************************
%
% Generate the output strings of `\whereapplies` and `\whereapplies*`
%
% This macro is for internal purposes only. When invoked, it updates the two
% macros `@ha@prop@@soutput@...` and `@ha@prop@@doutput@...`.
%
% The `identifier` argument remains confined within the internal scope of the
% package and does not create conflicts with possible macros or labels of the
% same name. Leading and trailing spaces around this string will **not** be
% ignored.
%
% The `preamble` argument is the text that will be expanded before the page
% list (usually "p." or "pp.").
%
```

```
% The `labels` argument must be a comma-separated list of labels.
%
\gdef\@ha@makeoutputstrings#1#2#3{%
    % Write "p./pp. \pageref..." to the output
    \expandafter\gdef\csname @ha@prop@@doutput@#1\endcsname{%
        % `\T@pageref` is a synonym of `\pageref`
        #2\@ha@makepagelist{T@pageref}{#3}%
    }%
    % Write "p./pp. \pageref*..." to the starred output
    \expandafter\gdef\csname @ha@prop@@soutput@#1\endcsname{%
        % `\@pagerefstar` is a synonym of `\pageref*`
        #2\@ha@makepagelist{@pagerefstar}{#3}%
    }%
    % Make the list of labels available to the API (via `\get@hainfo`)
    \expandafter\gdef\csname @ha@prop@@labels@#1\endcsname{#3}%
}
%
%
% Macro `\@ha@newidentifier{identifier}`
% ******************************************************************************
%
% Initialize a new identifier
%
% This macro is for internal purposes only. When invoked, it sets up the helper
% macros, counters and auxiliary files needed for keeping track of an
% identifier. If the identifier was already initialized the macro will be no
% op.
%
% The `identifier` argument remains confined within the internal scope of the
% package and does not create conflicts with possible macros or labels of the
% same name. Leading and trailing spaces around this string will **not** be
% ignored.
%
\gdef\@ha@newidentifier#1{%
    % Was this identifier already initialized?
    \ifcsname @ha@iter@@preamble@#1\endcsname\else%
        % The identifier was never initialized
        % Was the .hax input already initialized during this run?
        \ifdefined\@ha@commons@@haxcontent\else%
            % The .hax input was never initialized
            % Previous versions created unwanted whitespaces; I am thankful to
            % David Carlisle for suggesting `\endlinechar=\m@ne`
            {\endlinechar=\m@ne\@starttoc{hax}}%
            % Initialize the content to export to the .hax file
            \gdef\@ha@commons@@haxcontent{}%
        \fi%
        % Was a .hax file already exported during a previous run?
        \ifcsname @ha@prop@@labels@#1\endcsname\else%
            % This is the first run
            % Set the output to "??" - to be updated by the .hax file
            \expandafter\gdef\csname
                @ha@prop@@doutput@#1\endcsname{\textbf{??}}%
            % Set the starred output to "??" - to be updated by the .hax file
            \expandafter\gdef\csname
                @ha@prop@@soutput@#1\endcsname{\textbf{??}}%
```

```
329             % Set the list of labels to an empty value
330             \expandafter\gdef\csname @ha@prop@@labels@#1\endcsname{}%
331         \fi%
332         % Use "p." for the preamble when there is only one occurrence
333         \global\expandafter\let\csname @ha@iter@@preamble@#1\endcsname\hapage%
334         % Generate the output strings
335         \g@addto@macro\@ha@commons@@haxcontent{%
336             % Make sure that there are occurrences
337             \ifcsname @ha@iter@@labels@#1\endcsname%
338                 % There are occurrences
339                 % Generate the output strings
340                 \protect\@ha@makeoutputstrings{#1}{\csname
341                     @ha@iter@@preamble@#1\endcsname}{\csname
342                     @ha@iter@@labels@#1\endcsname}%
343             \fi%
344         }%
345     \fi%
346 }
347 %
348 %
349 %
350 %           LIBRARY ENVIRONMENT
351 %           ===================
352 %
353 % These macros are not directly available to the user, but are callable by
354 % other packages, if needed.
355 %
356 %
357 % Macro: `\starred@nochecks@hereapplies{label}{identifiers}`
358 % ****************************************************************************
359 %
360 % Similar to `\hereapplies*`, but without checks and with two mandatory
361 % arguments
362 %
363 % This macro is mainly for internal purposes (but nothing forbids invoking it
364 % directly). Here the two arguments are both mandatory and there will be no
365 % checks that first argument does not contain a comma. See the documentation of
366 % `\hereapplies` for more information.
367 %
368 \newcommand*{\starred@nochecks@hereapplies}[2]{%
369     % Assign a label to this occurrence
370     \label{#1}%
371     % Iterate through the comma-separated list `identifiers`
372     \@for\@ha@tmp@@litem:=#2\do{%
373         % Remove trailing and leading spaces
374         \edef\@ha@tmp@@id{\expandafter\ha@trim\expandafter{\@ha@tmp@@litem}}%
375         % Make sure that the identifier is initialized
376         \expandafter\@ha@newidentifier\expandafter{\@ha@tmp@@id}%
377         % Is this the first time this identifier is mentioned?
378         \ifcsname @ha@iter@@labels@\@ha@tmp@@id\endcsname%
379             % This is *not* the first time this identifier is mentioned
380             % Add this label to the list
381             \expandafter\g@addto@macro\csname
382                 @ha@iter@@labels@\@ha@tmp@@id\endcsname{,#1}%
383             % Use "pp." for the preamble when there are multiple occurrences
```

```
384                 \global\expandafter\let\csname
385                     @ha@iter@@preamble@\@ha@tmp@@id\endcsname\hapages%
386             \else%
387                 % This is the first time this identifier is mentioned
388                 % Set up the list with this label as value
389                 \expandafter\gdef\csname
390                     @ha@iter@@labels@\@ha@tmp@@id\endcsname{#1}%
391             \fi%
392         }%
393         % Clean the environment
394         \let\@ha@tmp@@id\undefined%
395 }
396 %
397 %
398 % Macro: `\starred@hereapplies[label]{identifiers}`
399 % ********************************************************************************
400 %
401 % Identical to `\hereapplies*`
402 %
403 % This macro is mainly for internal purposes (but nothing forbids invoking it
404 % directly). See the documentation of `\hereapplies` for more information.
405 %
406 \newcommand*{\starred@hereapplies}[2][]{%
407     % Check whether the macro has been called with one or two arguments
408     \if\relax\detokenize{#1}\relax%
409         % The macro has been called with only one argument
410         % Assign a unique number to the unnamed occurrence
411         \stepcounter{@ha@unlabeled@counter}%
412         % Create an opaque label
413         \edef\@ha@tmp@@lbl{hereapplies:unnamed\the@ha@unlabeled@counter}%
414     \else%
415         % The macro has been called with two arguments
416         % Expand the first argument for checking properly
417         \edef\@ha@tmp@@lbl{#1}%
418         % Make sure that there are no commas in the `label` argument
419         \expandafter\@ha@ifcomma\@ha@tmp@@lbl,\@then{%
420             \PackageError{hereapplies}{Comma detected in "\@ha@tmp@@lbl"}{%
421                 It is possible to assign only one single label.%
422             }%
423         }{}%
424     \fi%
425     % Call `\starred@nochecks@hereapplies`
426     \expandafter\starred@nochecks@hereapplies\expandafter{\@ha@tmp@@lbl}{#2}%
427     % Clean the environment
428     \let\@ha@tmp@@lbl\undefined%
429     % Ignore the spaces that might follow
430     \ignorespaces%
431 }
432 %
433 %
434 % Macro: `\get@hainfo[property]{identifier}`
435 % ********************************************************************************
436 %
437 % Get the value of an identifier's property
438 %
```

```
% This macro is mainly for internal purposes (but nothing forbids invoking it
% directly). If the identifier was never initialized the macro will initialize
% it.
%
% Possible values for the `property` argument are: `doutput`, `labels` and
% `soutput`. When omitted it defaults to `labels`.
%
% The `identifier` argument remains confined within the internal scope of the
% package and does not create conflicts with possible macros or labels of the
% same name. Leading and trailing spaces around this string will be ignored.
%
\newcommand*{\get@hainfo}[2][labels]{%
    % Trim leading and trailing spaces from the identifier
    \edef\@ha@tmp@@id{\ha@trim{#2}}%
    % Make sure that there are no commas
    \expandafter\@ha@ifcomma\@ha@tmp@@id,\@then{%
        \PackageError{hereapplies}{Comma detected in "\@ha@tmp@@id"}{%
            It is possible to query only one single identifier at a time.%
        }%
    }{}%
    % Make sure that the identifier is initialized
    \expandafter\@ha@newidentifier\expandafter{\@ha@tmp@@id}%
    % Print the identifier's property
    \csname @ha@prop@@#1@\@ha@tmp@@id\endcsname%
    % Clean the environment
    \let\@ha@tmp@@id\undefined%
}
%
%
%
%          USER ENVIRONMENT
%          ================
%
% These macros are directly available to the user.
%
%
% Macro: `\hereapplies[label]{identifiers}`
% ************************************************************************
%
% Notify the document that one or more identifiers apply to a particular point
% and add a label to it
%
% If the optional argument is passed the label created will be named
% accordingly, otherwise an opaque name will be chosen. This argument may
% contain only what is legal for `\pageref`.
%
% The `identifiers` argument must be a comma-separated list of identifiers
% (leading and trailing spaces around each member will be ignored). Each of
% these strings will remain confined within the internal scope of the package
% and will not create conflicts with possible macros or labels of the same
% names.
%
% The starred version of this command (`\hereapplies*`) will not invoke the
% `\phantomsection` directive.
%
```

```
494 \newcommand*{\hereapplies}{%
495     % Check if a star is present in the invocation of the command
496     \@ifstar{\starred@hereapplies}{\phantomsection\starred@hereapplies}%
497 }
498 %
499 %
500 % Macro: `\whereapplies{identifier}`
501 % ************************************************************************
502 %
503 % Print all the occurrences of an identifier in the form "p. ..." or "pp. ..."
504 % (with page range support)
505 %
506 % The `identifier` argument remains confined within the internal scope of the
507 % package and does not create conflicts with possible macros or labels of the
508 % same name. Leading and trailing spaces around this string will be ignored.
509 %
510 % If the same `identifier` is not passed to `\hereapplies` at least once
511 % throughout the document, `\whereapplies` will print "??".
512 %
513 % The starred version of this command (`\whereapplies*`) will use `\pageref*`
514 % instead of `\pageref` for generating the page list.
515 %
516 \newcommand*{\whereapplies}{%
517     % Check if a star is present in the invocation of the command
518     \@ifstar{\get@hainfo[soutput]}{\get@hainfo[doutput]}%
519 }%
520 % EOF
```