The Here Applies \LaTeX Package

September 13, 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: \texttt{\hereapplies} and \texttt{\whereapplies} (plus their “starred” versions \texttt{\hereapplies*} and \texttt{\whereapplies*}). In both cases an identifier is passed as argument – and this can be any string invented in the moment (\texttt{\hereapplies} additionally supports more than one identifier in the form of a comma-separated list).

Every time \texttt{\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 \texttt{\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 \texttt{\hereapplies} is designed to be invoked in the middle of a chapter or a section and that location must be made linkable, the \texttt{\phantomsection} directive is invoked by default before a label is added. To avoid calling \texttt{\phantomsection}, the “starred” command \texttt{\hereapplies*} is available.

Finally, like \texttt{\whereapplies} resembles a pluralizable version of \texttt{\pageref}, its “starred” version \texttt{\whereapplies*} will resemble a pluralizable version of \texttt{\pageref*}.

If you use LyX, the package ships a LyX module as well (please check the \texttt{lyx-module} subdirectory).
2 Example usage

The following \LaTeX{} manuscript

\begin{verbatim}
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}
\end{verbatim}

will generate the \texttt{hereapplies-example.pdf} document attached.

3 A minimal tutorial

\hereapplies Syntax:

\begin{verbatim}
\hereapplies [(label)] \{identifiers\}
\hereapplies* [(label)] \{identifiers\}
\end{verbatim}

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 identifiers 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\hadelimiter{| \ und\ }  
8 % English: ",\ " (exactly like in German — leave it)  
9 \gdef\halastdelimiter{,\ }  
```

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

3
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.
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.

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:

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}.

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

\{conceptTwo\} This is page \thepage. As you can see,

\{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{here applies}[2022/09/13 Here Applies]
\RequirePackage{hyperref}
\RequirePackage{refcount}

\textbf{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 \ }

\textbf{ABSTRACT UTILITIES}

These macros are not strictly related to this package, but are required.
\providecommand{\ha@ifcomma}{text to check,}{@then{if yes}{if no}}

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 \texttt{text to check}, then expands to \texttt{if yes} or \texttt{if no} accordingly.

Please do not put curly brackets around the \texttt{text to check}. The comma at the end of the \texttt{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@trim< #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@trim< #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}
%
%
% 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=−1%
% 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%
% 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 ", 
\expandafter\csname #1\expandafter\endcsname\fi%
\expandafter\csname #1\expandafter\endcsname\fi%
% The previous page was part of a contiguous range
% Print ", 
\expandafter\csname #1\expandafter\endcsname\else%
\expandafter\csname #1\expandafter\endcsname\fi%
\expandafter\csname #1\expandafter\endcsname\fi%
% 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%
\let\ha@tmp@@currp\ha@tmp@@nextp

% Prepare the next label
\let\ha@tmp@@currlbl\ha@tmp@@thislabel

% 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 "p" and "<p>"
    \{\ha@tmp@@lastpsep\csname #1\expandafter\endcsname%
      \expandafter{\ha@tmp@@currlbl}
    \}
  \else
    \% The previous page was part of a contiguous range
    % Print "p" and "<p--q>"
    \{\ha@tmp@@lastpsep\csname #1\expandafter\endcsname%
      \expandafter{\ha@tmp@@currrangelbl}
      \csname #1\expandafter\endcsname%
      \expandafter{\ha@tmp@@currlbl}
    \}
  \fi
\fi%

% Macro \ha@makeoutputstrings{identifier}{preamble}{labels}
% Generates the output strings of \whereappplies and \whereappplies*
% 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}%
  }%
\expandafter\gdef\csname @ha@prop@@soutput@#1\endcsname{%
\% Make the list of labels available to the API (via \texttt{\get@hainfo})
\expandafter\gdef\csname @ha@prop@@labels@#1\endcsname{}\%
}

\% Macro \texttt{\@ha@newidentifier\{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 \texttt{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 \texttt{not} be ignored.
\\
\expandafter\gdef\csname @ha@newidentifier#1\endcsname{%
\% Was this identifier already initialized?
\ifcsname @ha@iter@@preamble@#1\endcsname\else%
\% The identifier was never initialized
% Was the \texttt{.hax} input already initialized during this run?
\ifdef\@ha@commons@@haxcontent\else%
\% The \texttt{.hax} input was never initialized
\% Previous versions created unwanted whitespaces; I am thankful to David Carlisle for suggesting \texttt{\endlinechar=m@ne}
\{\endlinechar=m@ne\}@starttoc{hax}\%
\% Initialize the content to export to the \texttt{.hax} file
\expandafter\gdef\csname @ha@commons@@haxcontent\endcsname{}
\% Export the content when the document reaches the end
\AtEndDocument{\%
\adddtocontents{hax}{\@ha@commons@@haxcontent}\%
}\%}
\\
\fi%
% Was a \texttt{.hax} file already exported during a previous run?
\ifcsname @ha@prop@@labels@#1\endcsname\else%
\% This is the first run
% Set the output to \texttt{??} – to be updated by the \texttt{.hax} file
\expandafter\expandafter\expandafter\gdef\csname \@ha@prop@@output@#1\endcsname{%
\% Set the starred output to \texttt{??} – to be updated by the \texttt{.hax} file
\expandafter\expandafter\expandafter\gdef\csname \@ha@prop@@output@#1\endcsname{%
\% Set the list of labels to an empty value
\expandafter\expandafter\expandafter\gdef\csname \@ha@prop@@output@#1\endcsname{%
\fim}
% Use \texttt{\textbf{p.}} for the preamble when there is only one occurrence
\global\expandafter\let\csname @ha@iter@@preamble@#1\endcsname\hapage%
% Generate the output strings
\g@addto@macro\@ha@commons@@haxcontent{%
% Make sure that there are occurrences
\ifsname \@ha@iter@@labels@#1\endsname%
% There are occurrences
% Generate the output strings
\protect \@ha@makeoutputstrings{#1}{\csname
@ha@iter@@preamble@#1\endsname}{\csname
@ha@iter@@labels@#1\endsname}%
\fi%
}
\fi%
%
% LIBRARY ENVIRONMENT
%
% These macros are not directly available to the user, but are callable by
% other packages, if needed.
%
% Macro: \`\starred@nochecks@hereapplies \{ label \} \{ identifiers \}`
% *********************************************************
%
% Similar to `\hereapplies \*`, but without checks and with two mandatory
% arguments
%
% This macro is mainly for internal purposes (but nothing forbids invoking it
% directly). Here the two arguments are both mandatory and there will be no
% checks that first argument does not contain a comma. See the documentation of
% `\hereapplies` for more information.
%
% `\newcommand\*\{\starred@nochecks@hereapplies\}[2]{% 
% Assign a label to this occurrence
% \label{#1}%
% % Iterate through the comma-separated list \texttt{`identifiers`}
% \@for @ha@tmp@@litem:=#2:do{% 
% % Remove trailing and leading spaces
% \edef@ha@tmp@@id{\expandafter\ha@trim\expandafter\expandafter\expandafter{\@ha@tmp@@litem}}%
% % Make sure that the identifier is initialized
% \expandafter\@ha@newidentifier\expandafter{\@ha@tmp@@id}%
% % Is this the first time this identifier is mentioned?
% \ifsname \@ha@iter@@labels@@ha@tmp@@id\endsname%
% % This is not the first time this identifier is mentioned
% % Add this label to the list
% \expandafter\g@addto@macro\csname
% \@ha@iter@@labels@@ha@tmp@@id\endsname{,#1}%
% % Use "pp." for the preamble when there are multiple occurrences
% \global\expandafter\let\csname
% \@ha@iter@@preamble@@ha@tmp@@id\endsname\hapages%
% \else%
% % This is the first time this identifier is mentioned
%}}%
\begin{verbatim}
% Set up the list with this label as value
\expandafter\gdef\csname @ha@iter@@labels@\endcsname{\@ha@tmp@@id}
\fi
% Clean the environment
\let@ha@tmp@@id\undefined
}
%
% Macro: \texttt{\starred@hereapplies\{label\}{identifiers}}
%*****************************************************************************
%
% \texttt{\starred@hereapplies} is mainly for internal purposes (but nothing forbids invoking it
directly). See the documentation of \texttt{\starred@hereapplies} for more information.
%
\newcommand*{\starred@hereapplies}{2}{%
% Check whether the macro has been called with one or two arguments
\if\relax\detokenize{#1}\relax
% The macro has been called with only one argument
% Assign a unique number to the unnamed occurrence
\stepcounter{@ha@unlabeled@counter}
% Create an opaque label
\edef@ha@tmp@@lbl{\hereapplies:unnamed\the@ha@unlabeled@counter}
\else
% The macro has been called with two arguments
% Expand the first argument for checking properly
\edef@ha@tmp@@lbl{#1}
% Make sure that there are no commas in the \texttt{\texttt{label}} argument
\expandafter\ifcomma@ha@tmp@@lbl,\@then{%
\PackageError{\hereapplies}{Comma detected in "@ha@tmp@@lbl"}{%
   It is possible to assign only one single label.}%
}\fi%
}{% %
% Call \texttt{\starred@nochecks@hereapplies}
\expandafter\starred@nochecks@hereapplies\expandafter{\ha@tmp@@lbl}{#2}{%
% Clean the environment
\let@ha@tmp@@lbl\undefined
% Ignore the spaces that might follow
\ignorespaces%
}\}
%
% Macro: \texttt{\get@hainfo\{property\}{identifier}}
%*****************************************************************************
%
% This macro is mainly for internal purposes (but nothing forbids invoking it
% directly). If the identifier was never initialized the macro will initialize
% it.
%
\end{verbatim}
% 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
\@ha@ifcomma\@ha\tmp@@id, \@then{%
\PackageError{\hereapplies}{Comma detected in "@ha@tmp0@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.
% % Check if a star is present in the invocation of the command
\newcommand*{\hereapplies}{%
% \@ifstar{\starred@hereapplies}{\phantomsection\starred@hereapplies}%
}%}
% Macro: `\whereapplies\{identifier\}'
% ******************************************************************************
% Print all the occurrences of an identifier in the form "p. ..." or "pp. ..."
% (with page range support)
% 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.
% If the same 'identifier' is not passed to `\whereapplies' at least once
% throughout the document, `\whereapplies' will print "??".
% The starred version of this command (`\whereapplies\*') will use `\pageref\*' for generating the page list.
\newcommand*{\whereapplies}{\%}
\ifstar{\get@hainfo\[soutput]\}{\get@hainfo\[doutput]\}%
% EOF