The Here Applies \LaTeX\ Package

September 3, 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, \hereapplies*{conceptTwo} This is page \thepage. As you can see, both \goes{conceptOne}, \goes{conceptTwo} apply here.\newpage
\hereapplies{conceptOne, conceptTwo} This is page \thepage. As you can see, both \goes{conceptOne} and \goes{conceptTwo} apply here.\newpage
\hereapplies[myref]{conceptOne} This is page \thepage. As you can see, \goes{conceptOne} 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

\begin{verbatim}
\hereapplies Syntax:

\hereapplies [{\label}] \{\identifiers\}
\hereapplies* [{\label}] \{\identifiers\}
\end{verbatim}

The \texttt{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 **\whereapplies** 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 \textbf{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 \%
2 % German translation of **Here Applies**
3 \%
4 % English: "p. \ |
5 \% **\gdef** \hapage{S. \ }
6 \%
7 % English: "pp. \ |
8 \% **\gdef** \hapages{S. \ }
9 \%
10 % English: " | and "
11 \% **\gdef** \halastdelimiter{ \ und \ }
12 \%
13 % English: ", | " (exactly like in German — leave it)
14 \%
15 % **\gdef** \hadelimiter{, | }

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.
Code appendix

1 \% --latex; indent-tabs-mode: nil; c-basic-offset: 4; tab-width: 4 --
2 \%
3 \%
4 % hereapplies.sty
5 \%
6 % A LaTeX package for referencing groups of pages that share something in
7 % common
8 %
9 % https://github.com/madmurphy/herapplies.sty
10 \%
11 % Version 0.8.0
12 \%
13 % Copyright (C) 2022 madmurphy <madmurphy333@gmail.com>
14 \%
15 \%
16 **Here Applies** is free software: you can redistribute it and/or modify it
17 \% under the terms of the GNU Affero General Public License as published by the
18 % Free Software Foundation, either version 3 of the License, or (at your
19 \% option) any later version.
20 %
21 **Here Applies** is distributed in the hope that it will be useful, but
22 % WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
23 \% FITNESS FOR A PARTICULAR PURPOSE. See the GNU Affero General Public License
24 \% for more details.
25 %
26 \%
27 \%
28 \%
29 \%
30 % Example usage:
31 \%
32 \% \begin{document}
33 \%
34 \%
35 \%
36 % \title{Some title}
37 \%
38 \%
39 \%
40 \%
41 \%
42 \%
43 \%
44 % This is concept one. To find this concept applied, please
45 % see \whereapplies{conceptOne}.
46 \%
47 % This is concept two. To find this concept applied, please
48 % see \whereapplies{conceptTwo}. \newpage
49 %
50 % \whereapplies{conceptOne} This is page \thepage. As you can see,
51 % "concept one" applies here. \newpage
52 %
53 % \whereapplies{conceptTwo} This is page \thepage. As you can see,
% "concept two" applies here. 
% Here applies \{conceptOne, conceptTwo\} This is page \thepage. As you can see, both "concept one" and "concept two" apply here. 
% "concept two" applies here. 
% Here applies \{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}[2022/09/03 Here Applies]
\RequirePackage{hyperref}
\RequirePackage{refcount}

\long\gdef\@ha@ifcomma#1,#2@then#3#4{
% 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 \ }

\SECTION{TRANSLATABLE STRINGS}

\SECTION{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: \texttt{\textbackslash ha@trim\{}text\texttt{\}}  
\% ***********************************************  
\% 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 \texttt{<} and \texttt{>}, for trimming safely  
\%  
\% 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\{  
\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}  
\%  
\%  
\% Macro \texttt{\textbackslash ha@makepagelist\{}hypermacro\texttt{\}}\{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 \texttt{(\texttt{labels})}, checks which labels refer  
\% to duplicate page numbers and which page numbers can be grouped together, and  
\% finally prints a list.  
\%  
\% The \texttt{hypermacro} argument is the macro (usually from the \texttt{hyperref} package)  
\% that will process the label name.  
\%  
\% The \texttt{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\ha@tmp@@currp\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%
\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 "f", \langle p\rangle
\{\ha@tmp@@psep\csname #1\expandafter\endcsname\%
\expandafter\expandafter\\csname \ha@tmp@@currlbl\endcsname%
\else%
% The previous page was part of a contiguous range
% Print "f", \langle p\rangle
\{\ha@tmp@@psep\csname #1\expandafter\endcsname\%
\expandafter\expandafter\\csname \ha@tmp@@currrangelbl\ha@tmp@@currlbl\csname \expandafter\endcsname\%
\expandafter\expandafter\expandafter\\csname \ha@tmp@@currlbl\csname \expandafter\endcsname\%
\else%
% Reset the current range offset
\def\ha@tmp@@prangeoffs{-1}%
\fi%
% Ensure a comma before the next page number
\let\ha@tmp@@psep\halastdelimitert%
% Ensure " and " before the last page number
\let\ha@tmp@@lastpsep\halastdelimitert%
\fi%
% Prepare the next page number
\let\@ha@tmp@@currp\@ha@tmp@@nextp
% Prepare the next label
\let\@ha@tmp@@currbl\@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>'
\expandafter\endsname%
\expandafter{\@ha@tmp@@currlbl}
\else%
% The previous page was part of a contiguous range
% Print '<p−q>'
\expandafter\endsname%
\expandafter{\@ha@tmp@@currrangelbl}−−\expandafter\endsname%
\expandafter{\@ha@tmp@@currlbl}
\fi
\fi
% 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\endsname\@ha@prop@@doutput\@ha@tmp@@thislabel%
% '\T@pageref' is a synonym of '\pageref'
\expandafter\endsname\@ha@tmp@@thislabel{T@pageref}{#3}%
}%
% Write "p./ pp. |pageref|..." to the starred output
\edef\@ha@prop@@soutput@#1{\textbf{??}}
% \pagerefstar is a synonym of \pageref
\makepagelist{\@ha@prop@@soutput@#1}{\@ha@makepagelist{\pagerefstar}{\@ha@prop@@labels@#1}}

% Make the list of labels available to the API (via \get@hainfo)
\edef\@ha@prop@@labels@#1{\@ha@makepagelist{\@ha@prop@@labels@#1}}

% Initializer 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 \texttt{\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 **not** be
% ignored.
%
\@ha@newidentifier{identifier}{
    \ifcsname @ha@iter@@preamble@#1\endcsname
        \else
            \ifdef\@ha@commons@@haxcontent\else
                \edef\@ha@commons@@haxcontent{}\@starttoc{hax}
            \fi
            \AtEndDocument{
                \addtocontents{hax}{\@ha@commons@@haxcontent}
            }
    \fi

    \ifcsname @ha@prop@@labels@#1\endcsname
        \else
            \edef\@ha@prop@@labels@#1{\@ha@makepagelist{\@ha@prop@@labels@#1{\@ha@prop@@labels@#1}}}
    \fi

    \edef\@ha@prop@@doutput@#1{\textbf{??}}
    \edef\@ha@prop@@soutput@#1{\textbf{??}}
}

% Was this identifier already initialized?
\@ha@newidentifier{#1}{
    \ifcsname @ha@iter@@preamble@#1\endcsname\else
        \ifdef\@ha@commons@@haxcontent\else
            \edef\@ha@commons@@haxcontent{}\@starttoc{hax}
        \fi
        \AtEndDocument{
            \addtocontents{hax}{\@ha@commons@@haxcontent}
        }
    \fi

    \ifdef\@ha@commons@@haxcontent\else
        \edef\@ha@commons@@haxcontent{}\@starttoc{hax}
    \fi

    \ifdef\@ha@commons@@haxcontent\else
        \edef\@ha@commons@@haxcontent{}\@starttoc{hax}
    \fi

    \ifdef\@ha@commons@@haxcontent\else
        \edef\@ha@commons@@haxcontent{}\@starttoc{hax}
    \fi

    \ifdef\@ha@commons@@haxcontent\else
        \edef\@ha@commons@@haxcontent{}\@starttoc{hax}
    \fi

    \ifdef\@ha@commons@@haxcontent\else
        \edef\@ha@commons@@haxcontent{}\@starttoc{hax}
    \fi

    \ifdef\@ha@commons@@haxcontent\else
        \edef\@ha@commons@@haxcontent{}\@starttoc{hax}
    \fi

    % This is the first run
    \ifdef\@ha@prop@@labels@#1\else
        \edef\@ha@prop@@labels@#1{\@ha@makepagelist{\@ha@prop@@labels@#1{\@ha@prop@@labels@#1}}}
    \fi

    \ifdef\@ha@prop@@labels@#1\else
        \edef\@ha@prop@@labels@#1{\@ha@makepagelist{\@ha@prop@@labels@#1{\@ha@prop@@labels@#1}}}
    \fi

    \ifdef\@ha@prop@@labels@#1\else
        \edef\@ha@prop@@labels@#1{\@ha@makepagelist{\@ha@prop@@labels@#1{\@ha@prop@@labels@#1}}}
    \fi

    \ifdef\@ha@prop@@labels@#1\else
        \edef\@ha@prop@@labels@#1{\@ha@makepagelist{\@ha@prop@@labels@#1{\@ha@prop@@labels@#1}}}
    \fi
}

% Use "p." for the preamble when there is only one occurrence
\begin{verbatim}
\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\endcsname}
  {\csname @ha@iter@@labels@#1\endcsname}
\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

\newcommand\*{\@ha@iter@@litem:=#2}{%
% Assign a label to this occurrence
\label{#1}%
% Iterate through the comma-separated list 'identifiers'
\@for \@ha@tmp@@id:=#2:do[%
% Remove trailing and leading spaces
\edef \@ha@tmp@@litem{\expandafter\ha@trim\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
  \global\expandafter\let\csname
    \@ha@iter@@preamble@\@ha@tmp@@id\endsname\hapage%
\fi%
}
\end{verbatim}
% Set up the list with this label as value
\expandafter\gdef\csname @ha@iter@@labels@\@ha@tmp@@id\endsname{#1}%
\fi%
% Clean the environment
\let\@ha@tmp@@id\ undefined%
}
%
%
%
% Macro: `\ starred@hereapplies{label}{identifiers}'
% ********************************************************************
%
% Identical to `\hereapplies'*
%
% This macro is mainly for internal purposes (but nothing forbids invoking it
directly). See the documentation of `\hereapplies' for more information.
%
\newcommand*{\ starred@hereapplies }[2][]{{}
  \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 `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 `\ starred@nochecks@hereapplies'*
\expandafter\starred@nochecks@hereapplies \expandafter\starred@nochecks@hereapplies\expandafter\starred@nochecks@hereapplies\expandafter\expandafter\{\@ha@tmp@@lbl\}{#2}%
% Clean the environment
\let\@ha@tmp@@lbl\ undefined%
}
%
%
%
% Macro: `\ get@hainfo{property}{identifier}'
% ********************************************************************
%
% Get the value of an identifier 's property
%
% 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]|%}
  \edef@ha@tmp@@id{\ha@trim{#2}}%
  \edef@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.%
    }%
  }{}
  \edef@ha@newidentifier\expandafter{\@ha@tmp@@id}%
  \csname @ha@prop@@#1 @\@ha@tmp@@id\endcsname%
  \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.

```
\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 instead of \pageref for generating the page list.

\newcommand*{\whereapplies}{%}
  \@ifstar{\get@hainfo[soutput]}{\get@hainfo[doutput]}%}

% EOF