

The **Here Applies** L^AT_EX Package

October 24, 2023

Abstract

A L^AT_EX package for referencing groups of pages that share something in common.

1 Overview

Here Applies is a L^AT_EX 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 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 L^AT_EX manuscript

```
1 \documentclass{article}
2
3 \usepackage{hereapplies}
4
5 \begin{document}
6
7 \title{Some title}
8
9 \author{Some author}
10
11 \maketitle
12
13 This is concept one. To find this concept applied, please
14 see \whereapplies{conceptOne}.
15
16 This is concept two. To find this concept applied, please
17 see \whereapplies{conceptTwo}.\newpage
18
19 \hereapplies{conceptOne} This is page \thepage. As you can see,
20 ``concept one'' applies here.\newpage
21
22 \hereapplies{conceptTwo} This is page \thepage. As you can see,
23 ``concept two'' applies here.\newpage
24
25 \hereapplies{conceptOne, conceptTwo} This is page \thepage. As you
26 can see, both ``concept one'' and ``concept two'' apply here.\newpage
27
28 \hereapplies{conceptTwo} This is page \thepage. As you can see,
29 ``concept two'' applies here.\newpage
30
31 \hereapplies[myref]{conceptOne} This is page \thepage. As you can
32 see, ``concept one'' applies here. This point in the document is
33 labeled \texttt{myref}.
34
35 \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

```
1 % -- Mode: 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/hereapplies.sty
10 %
11 % Version 1.0.2
12 %
13 % Copyright (C) 2022 madmurphy <madmurphy333@gmail.com>
14 %
15 % **Here Applies** is free software: you can redistribute it and/or modify it
16 % under the terms of the GNU Affero General Public License as published by the
17 % Free Software Foundation, either version 3 of the License, or (at your
18 % option) any later version.
19 %
20 % **Here Applies** is distributed in the hope that it will be useful, but
21 % WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
22 % FITNESS FOR A PARTICULAR PURPOSE. See the GNU Affero General Public License
23 % for more details.
24 %
25 % You should have received a copy of the GNU Affero General Public License
26 % along with this program. If not, see <http://www.gnu.org/licenses/>.
27 %
28 %
29 %
30 % Example usage:
31 %
32 %     \documentclass{article}
33 %
34 %     \usepackage{hereapplies}
35 %
36 %     \begin{document}
37 %
38 %     \title{Some title}
39 %
40 %     \author{Some author}
41 %
42 %     \maketitle
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 %     \hereapplies{conceptOne} This is page \thepage. As you can see,
51 %     `concept one' applies here.\newpage
52 %
53 %     \hereapplies{conceptTwo} This is page \thepage. As you can see,
```

```

54 %   ``concept two'' applies here. |newpage
55 %
56 %   \hereapplies{conceptOne, conceptTwo} This is page |thepage. As you
57 %   can see, both ``concept one'' and ``concept two'' apply here. |newpage
58 %
59 %   \hereapplies{conceptTwo} This is page |thepage. As you can see,
60 %   ``concept two'' applies here. |newpage
61 %
62 %   \hereapplies[myref]{conceptOne} This is page |thepage. As you can
63 %   see, ``concept one'' applies here. This point in the document is
64 %   labeled \texttt{myref}.
65 %
66 %   \end{document}
67 %
68 %
69 \ProvidesPackage{hereapplies}[2023/10/24 Here Applies]
70 \RequirePackage{hyperref}
71 \RequirePackage{refcount}
72 %
73 %
74 %
75 %   TRANSLATABLE STRINGS
76 %
77 %
78 %
79 % The abbreviation of one single page
80 \providecommand*\hapage{p.\ }
81 % The abbreviation of two or more pages
82 \providecommand*\hapages{pp.\ }
83 % The delimiter between page numbers
84 \providecommand*\hadelimiter{,\ }
85 % The delimiter before the last page number
86 \providecommand*\halastdelimiter{\ and\ }
87 %
88 %
89 %
90 %   ABSTRACT UTILITIES
91 %
92 %
93 % The following macros are not strictly related to this package, but the latter
94 % requires them.
95 %
96 %
97 % Macro: `|@ha@ifcomma text to check,|@then{if yes}{if no}`
98 % *****
99 %
100 % Check if a string contains a comma
101 %
102 % This macro is mainly for internal purposes (but nothing forbids invoking it
103 % directly). When invoked it checks whether a comma is present in `text to
104 % check`, then expands to `if yes` or `if no` accordingly.
105 %
106 % Please do not put curly brackets around the text to check. The comma at the
107 % end of the text is mandatory.
108 %

```

```

109 \long\gdef\@ha@ifcomma#1,#2\@then#3#4{%
110   \if\relax\detokenize{#2}\relax#4\else#3\fi%
111 }
112 %
113 %
114 % Macro: ` \@ha@trim{text}`
115 % *****
116 %
117 % Trim leading and trailing spaces from a string
118 %
119 % This macro is mainly for internal purposes (but nothing forbids invoking it
120 % directly).
121 %
122 \begingroup
123 % Temporarily change the categories of `<` and `>`, for trimming safely
124 \catcode`\<=4\catcode`\>=3
125 % Helper macro
126 \long\gdef\@ha@trm< #1 < #2 < #3 < #4 < #5 < #6 < #7 < #8 < #9 >/{%
127   \ifcase\numexpr2#3#8\relax\or#2\or#7\or#5\or#1\fi%
128 }
129 % Usable macro
130 \long\gdef\ha@trim#1{%
131   \@ha@trm< #1 < #1 < - < + < ? < #1 < #1 < 0 < 2 < 1 < 3 < 2 < ! >/%
132 }
133 \endgroup
134 %
135 %
136 %
137 % PRIVATE ENVIRONMENT
138 %
139 %
140 % The following macros regulate the internal functioning of the package and
141 % should not be invoked directly.
142 %
143 %
144 % Assign a unique number to each unlabeled occurrence of an identifier
145 \newcounter{@ha@unlabeled@counter}
146 % Populate the .hax file when the document reaches the end
147 \AtEndDocument{%
148   % Do we have any content?
149   \ifdefined\@ha@commons@@haxcontent%
150     % We do - export it
151     \addtocontents{hax}{\@ha@commons@@haxcontent}%
152   \fi%
153 }
154 %
155 %
156 % Macro ` \@ha@makepagelist{hypermacro}{labels}`
157 % *****
158 %
159 % Generate the list of page numbers (with page range support)
160 %
161 % This macro is for internal purposes only. When invoked, it scans the
162 % comma-separated list of labels provided (`labels`), checks which labels refer
163 % to duplicate page numbers and which page numbers can be grouped together, and

```

```

164 % finally prints a list.
165 %
166 % The `hypermacro` argument is the macro (usually from the `hyperref` package)
167 % that will process the label name.
168 %
169 % The `labels` argument must be a comma-separated list of labels.
170 %
171 \gdef\@ha@makepagelist#1#2{%
172   \begingroup%
173   % Reset the current page number
174   \def\@ha@tmp@@curr{-1}%
175   % Reset the current range offset
176   \def\@ha@tmp@@prangeoffs{-1}%
177   % Ensure no comma before the first page number
178   \def\@ha@tmp@@psep{}%
179   % Ensure no text before the last number if it is also the first one
180   \def\@ha@tmp@@lastpsep{}%
181   % Iterate through the `labels` argument
182   \@for\@ha@tmp@@thislabel:=#2\do{%
183     % Store the page number associated with this label
184     \edef\@ha@tmp@@nextp{\getpagerefnumber{\@ha@tmp@@thislabel}}%
185     % Check that we are not on the same page as in the last iteration
186     \ifnum\@ha@tmp@@curr=\@ha@tmp@@nextp\else%
187       % This is not the same page as in the last iteration
188       % Is this the first page in which this identifier appears?
189       \ifnum\@ha@tmp@@curr>-1%
190         % We have already met pages in which this identifier appears
191         % Does this page follow immediately the previous page?
192         \ifnum\numexpr\@ha@tmp@@curr+1=\@ha@tmp@@nextp%
193           % This page follows immediately the previous page
194           % Are these the first two contiguous pages of the range?
195           \ifnum\@ha@tmp@@prangeoffs=-1%
196             % These are the first two contiguous pages of the range
197             % Store the first page number of the pair
198             \let\@ha@tmp@@prangeoffs\@ha@tmp@@curr%
199             % Store the first label of the pair
200             \let\@ha@tmp@@currangelbl\@ha@tmp@@currlbl%
201           \fi%
202         \else%
203           % This page is far from the previous label's page
204           % Was the previous page part of a contiguous range?
205           \ifnum\@ha@tmp@@prangeoffs=-1%
206             % The previous page was a standalone page
207             % Print "[, ]<p>"
208             {\@ha@tmp@@psep\cscname
209              #1\expandafter\endcscname%
210              \expandafter{\@ha@tmp@@currlbl}}%
211           \else%
212             % The previous page was part of a contiguous range
213             % Print "[, ]<p—q>"
214             {\@ha@tmp@@psep\cscname
215              #1\expandafter\endcscname%
216              \expandafter{\@ha@tmp@@currangelbl}—\cscname
217              #1\expandafter\endcscname%
218              \expandafter{\@ha@tmp@@currlbl}}%

```



```

219             % Reset the current range offset
220             \def\@ha@tmp@@prangeoffs{-1}%
221             \fi%
222             % Ensure a comma before the next page number
223             \let\@ha@tmp@@psep\hadelimiter%
224             % Ensure " and " before the last page number
225             \let\@ha@tmp@@lastpsep\halastdelimiter%
226             \fi%
227         \fi%
228         % Prepare the next page number
229         \let\@ha@tmp@@currp\@ha@tmp@@nextp%
230         % Prepare the next label
231         \let\@ha@tmp@@currlbl\@ha@tmp@@thislabel%
232     \fi%
233 }%
234 % Print the last page number
235 % Is there at least one page to print?
236 \ifnum\@ha@tmp@@currp>-1%
237     % There is at least one page to print
238     % Was the previous page part of a contiguous range?
239     \ifnum\@ha@tmp@@prangeoffs=-1%
240         % The previous page was a standalone page
241         % Print "[ and ]<p>"
242         {\@ha@tmp@@lastpsep\csname
243             #1\expandafter\endcsname%
244             \expandafter{\@ha@tmp@@currlbl}}%
245     \else%
246         % The previous page was part of a contiguous range
247         % Print "[ and ]<p—q>"
248         {\@ha@tmp@@lastpsep\csname
249             #1\expandafter\endcsname%
250             \expandafter{\@ha@tmp@@currangelbl}—\csname
251             #1\expandafter\endcsname%
252             \expandafter{\@ha@tmp@@currlbl}}}%
253     \fi%
254 \fi%
255 \endgroup%
256 }
257 %
258 %
259 % Macro ` \@ha@makeoutputstrings{identifier}{preamble}{labels}`
260 % *****
261 %
262 % Generate the output strings of ` \whereapplies ` and ` \whereapplies* `
263 %
264 % This macro is for internal purposes only. When invoked, it updates the two
265 % macros ` \@ha@prop@@soutput@... ` and ` \@ha@prop@@doutput@... `.
266 %
267 % The ` identifier ` argument remains confined within the internal scope of the
268 % package and does not create conflicts with possible macros or labels of the
269 % same name. Leading and trailing spaces around this string will not be
270 % ignored.
271 %
272 % The ` preamble ` argument is the text that will be expanded before the page
273 % list (usually "p." or "pp. ").

```

```

274 %
275 % The `labels` argument must be a comma-separated list of labels.
276 %
277 \gdef\@ha@makeoutputstrings#1#2#3{%
278   % Write "p./pp. \pageref..." to the output
279   \expandafter\gdef\csname @ha@prop@@doutput@#1\endcsname{%
280     % `T@pageref` is a synonym of `pageref`
281     #2\@ha@makepagelist{T@pageref}{#3}%
282   }%
283   % Write "p./pp. \pageref*..." to the starred output
284   \expandafter\gdef\csname @ha@prop@@soutput@#1\endcsname{%
285     % `@pagerefstar` is a synonym of `pageref*`
286     #2\@ha@makepagelist{@pagerefstar}{#3}%
287   }%
288   % Make the list of labels available to the API (via `get@hainfo`)
289   \expandafter\gdef\csname @ha@prop@@labels@#1\endcsname{#3}%
290 }
291 %
292 %
293 % Macro `@ha@newidentifier{identifier}`
294 % *****
295 %
296 % Initialize a new identifier
297 %
298 % This macro is for internal purposes only. When invoked, it sets up the helper
299 % macros, counters and auxiliary files needed for keeping track of an
300 % identifier. If the identifier was already initialized the macro will be no
301 % op.
302 %
303 % The `identifier` argument remains confined within the internal scope of the
304 % package and does not create conflicts with possible macros or labels of the
305 % same name. Leading and trailing spaces around this string will not be
306 % ignored.
307 %
308 \gdef\@ha@newidentifier#1{%
309   % Was this identifier already initialized?
310   \ifcsname @ha@iter@@preamble@#1\endcsname\else%
311     % The identifier was never initialized
312     % Was the .hax input already initialized during this run?
313     \ifdefined\@ha@commons@@haxcontent\else%
314       % The .hax input was never initialized
315       % Previous versions created unwanted whitespaces; I am thankful to
316       % David Carlisle for suggesting `|endlinechar=|m@ne`
317       {\endlinechar=|m@ne\@starttoc{hax}}%
318       % Initialize the content to export to the .hax file
319       \gdef\@ha@commons@@haxcontent{}%
320     \fi%
321     % Was a .hax file already exported during a previous run?
322     \ifcsname @ha@prop@@labels@#1\endcsname\else%
323       % This is the first run
324       % Set the output to "???" - to be updated by the .hax file
325       \expandafter\gdef\csname
326         @ha@prop@@doutput@#1\endcsname{\textbf{??}}%
327       % Set the starred output to "???" - to be updated by the .hax file
328       \expandafter\gdef\csname

```

```

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

```

```

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

```

```

439 %
440 % This macro is mainly for internal purposes (but nothing forbids invoking it
441 % directly). If the identifier was never initialized the macro will initialize
442 % it.
443 %
444 % Possible values for the `property` argument are: `doutput`, `labels` and
445 % `soutput`. When omitted it defaults to `labels`.
446 %
447 % The `identifier` argument remains confined within the internal scope of the
448 % package and does not create conflicts with possible macros or labels of the
449 % same name. Leading and trailing spaces around this string will be ignored.
450 %
451 \newcommand*{\get@hainfo}[2][labels]{%
452   % Trim leading and trailing spaces from the identifier
453   \edef\@ha@tmp@id{\ha@trim{#2}}%
454   % Make sure that there are no commas
455   \expandafter\@ha@ifcomma\@ha@tmp@id,\@then{%
456     \PackageError{hereapplies}{Comma detected in "\@ha@tmp@id"}{%
457       It is possible to query only one single identifier at a time.%
458     }%
459   }{}%
460   % Make sure that the identifier is initialized
461   \expandafter\@ha@newidentifier\expandafter{\@ha@tmp@id}%
462   % Print the identifier's property
463   \csname @ha@prop@@#1@\@ha@tmp@id\endcsname%
464   % Clean the environment
465   \let\@ha@tmp@id\undefined%
466 }
467 %
468 %
469 %
470 % USER ENVIRONMENT
471 %
472 %
473 % The following macros are available to the user.
474 %
475 %
476 % Macro: `|hereapplies[label]{identifiers}`
477 % *****
478 %
479 % Notify the document that one or more identifiers apply to a particular point
480 % and add a label to it
481 %
482 % If the optional argument is passed the label created will be named
483 % accordingly, otherwise an opaque name will be chosen. This argument may
484 % contain only what is legal for `|pageref`.
485 %
486 % The `identifiers` argument must be a comma-separated list of identifiers
487 % (leading and trailing spaces around each member will be ignored). Each of
488 % these strings will remain confined within the internal scope of the package
489 % and will not create conflicts with possible macros or labels of the same
490 % names.
491 %
492 % The starred version of this command (`|hereapplies*`) will not invoke the
493 % `|phantomsection` directive.

```

```

494 %
495 \newcommand*{\hereapplies}{%
496     % Check if a star is present in the invocation of the command
497     \@ifstar{\starred@hereapplies}{\phantomsection\starred@hereapplies}%
498 }
499 %
500 %
501 % Macro: \whereapplies{identifier}
502 % *****
503 %
504 % Print all occurrences of an identifier in the form "p. ..." or "pp. ..."
505 % (with page range support)
506 %
507 % The identifier argument remains confined within the internal scope of the
508 % package and does not create conflicts with possible macros or labels of the
509 % same name. Leading and trailing spaces around this string will be ignored.
510 %
511 % If the same identifier is not passed to \hereapplies at least once
512 % throughout the document, \whereapplies will print "??".
513 %
514 % The starred version of this command (\whereapplies*) will use \pageref*
515 % instead of \pageref for generating the page list.
516 %
517 \newcommand*{\whereapplies}{%
518     % Check if a star is present in the invocation of the command
519     \@ifstar{\get@hainfo[soutput]}{\get@hainfo[doutput]}%
520 }%
521 % EOF

```