

The **Here Applies** L^AT_EX Package

September 13, 2022

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 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 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 1.0.0
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}[2022/09/13 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 % These macros are not strictly related to this package, but are required.
94 %
95 %
96 % Macro: ` \@ha@ifcomma text to check, \@then{if yes}{if no}`
97 % *****
98 %
99 % Check if a string contains a comma
100 %
101 % This macro is mainly for internal purposes (but nothing forbids invoking it
102 % directly). When invoked it checks whether a comma is present in `text to
103 % check`, then expands to `if yes` or `if no` accordingly.
104 %
105 % Please do not put curly brackets around the text to check. The comma at the
106 % end of the text is mandatory.
107 %
108 \long\gdef\@ha@ifcomma#1,#2\@then#3#4{%

```

```

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

```

```

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

```



```

219         % Prepare the next page number
220         \let \@ha@tmp@@currp \@ha@tmp@@nextp%
221         % Prepare the next label
222         \let \@ha@tmp@@currlbl \@ha@tmp@@thislabel%
223     \fi%
224 }%
225 % Print the last page number
226 % Is there at least one page to print?
227 \ifnum \@ha@tmp@@currp>-1%
228     % There is at least one page to print
229     % Was the previous page part of a contiguous range?
230     \ifnum \@ha@tmp@@prangeoffs=-1%
231         % The previous page was a standalone page
232         % Print "| and |<p>"
233         {\@ha@tmp@@lastpsep\csname
234             #1\expandafter\endcsname%
235             \expandafter{\@ha@tmp@@currlbl}}%
236     \else%
237         % The previous page was part of a contiguous range
238         % Print "| and |p—q>"
239         {\@ha@tmp@@lastpsep\csname
240             #1\expandafter\endcsname%
241             \expandafter{\@ha@tmp@@currangelbl}---\csname
242             #1\expandafter\endcsname%
243             \expandafter{\@ha@tmp@@currlbl}}%
244     \fi%
245 \fi%
246 \endgroup%
247 }
248 %
249 %
250 % Macro ` \@ha@makeoutputstrings{identifier}{preamble}{labels}`
251 % *****
252 %
253 % Generate the output strings of ` \whereapplies ` and ` \whereapplies* `
254 %
255 % This macro is for internal purposes only. When invoked, it updates the two
256 % macros ` \@ha@prop@@soutput@... ` and ` \@ha@prop@@doutput@... `.
257 %
258 % The ` identifier ` argument remains confined within the internal scope of the
259 % package and does not create conflicts with possible macros or labels of the
260 % same name. Leading and trailing spaces around this string will not be
261 % ignored.
262 %
263 % The ` preamble ` argument is the text that will be expanded before the page
264 % list (usually "p." or "pp.").
265 %
266 % The ` labels ` argument must be a comma-separated list of labels.
267 %
268 \gdef \@ha@makeoutputstrings#1#2#3{%
269     % Write "p./pp. \pageref..." to the output
270     \expandafter\gdef\csname @ha@prop@@doutput@#1\endcsname{%
271         % ` T@pageref ` is a synonym of ` \pageref `
272         #2\@ha@makepagelist{T@pageref}{#3}%
273     }%

```

```

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

```

```

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

```

```

384           % Set up the list with this label as value
385           \expandafter\gdef\csname
386             @ha@iter@@labels@\@ha@tmp@@id\endcsname{#1}%
387       \fi%
388     }%
389     % Clean the environment
390     \let\@ha@tmp@@id\undefined%
391   }
392   %
393   %
394   % Macro: '\starred@hereapplies[label]{identifiers}'
395   % *****
396   %
397   % Identical to '\hereapplies*'
398   %
399   % This macro is mainly for internal purposes (but nothing forbids invoking it
400   % directly). See the documentation of '\hereapplies' for more information.
401   %
402   \newcommand*{\starred@hereapplies}[2][]{%
403     % Check whether the macro has been called with one or two arguments
404     \if\relax\detokenize{#1}\relax%
405       % The macro has been called with only one argument
406       % Assign a unique number to the unnamed occurrence
407       \stepcounter{@ha@unlabeled@counter}%
408       % Create an opaque label
409       \edef\@ha@tmp@@lbl{hereapplies:unnamed\the@ha@unlabeled@counter}%
410     \else%
411       % The macro has been called with two arguments
412       % Expand the first argument for checking properly
413       \edef\@ha@tmp@@lbl{#1}%
414       % Make sure that there are no commas in the 'label' argument
415       \expandafter\@ha@ifcomma\@ha@tmp@@lbl,\@then{%
416         \PackageError{hereapplies}{Comma detected in "\@ha@tmp@@lbl"}{%
417           It is possible to assign only one single label.%
418         }%
419       }{}%
420     \fi%
421     % Call '\starred@nochecks@hereapplies'
422     \expandafter\starred@nochecks@hereapplies\expandafter{\@ha@tmp@@lbl}{#2}%
423     % Clean the environment
424     \let\@ha@tmp@@lbl\undefined%
425     % Ignore the spaces that might follow
426     \ignorespaces%
427   }
428   %
429   %
430   % Macro: '\get@hainfo[property]{identifier}'
431   % *****
432   %
433   % Get the value of an identifier's property
434   %
435   % This macro is mainly for internal purposes (but nothing forbids invoking it
436   % directly). If the identifier was never initialized the macro will initialize
437   % it.
438   %

```

```

439 % Possible values for the `property` argument are: `doutput`, `labels` and
440 % `soutput`. When omitted it defaults to `labels`.
441 %
442 % The `identifier` argument remains confined within the internal scope of the
443 % package and does not create conflicts with possible macros or labels of the
444 % same name. Leading and trailing spaces around this string will be ignored.
445 %
446 \newcommand*{\get@hainfo}[2][labels]{%
447     % Trim leading and trailing spaces from the identifier
448     \edef\@ha@tmp@id{\ha@trim{#2}}%
449     % Make sure that there are no commas
450     \expandafter\@ha@ifcomma\@ha@tmp@id,\@then{%
451         \PackageError{hereapplies}{Comma detected in "\@ha@tmp@id"}{%
452             It is possible to query only one single identifier at a time.%
453         }%
454     }{}%
455     % Make sure that the identifier is initialized
456     \expandafter\@ha@newidentifier\expandafter{\@ha@tmp@id}%
457     % Print the identifier's property
458     \csname @ha@prop@@#1@\@ha@tmp@id\endcsname%
459     % Clean the environment
460     \let\@ha@tmp@id\undefined%
461 }
462 %
463 %
464 %
465 % USER ENVIRONMENT
466 %
467 %
468 % These macros are directly available to the user.
469 %
470 %
471 % Macro: `\hereapplies[label]{identifiers}`
472 % *****
473 %
474 % Notify the document that one or more identifiers apply to a particular point
475 % and add a label to it
476 %
477 % If the optional argument is passed the label created will be named
478 % accordingly, otherwise an opaque name will be chosen. This argument may
479 % contain only what is legal for `\pageref'.
480 %
481 % The `identifiers` argument must be a comma-separated list of identifiers
482 % (leading and trailing spaces around each member will be ignored). Each of
483 % these strings will remain confined within the internal scope of the package
484 % and will not create conflicts with possible macros or labels of the same
485 % names.
486 %
487 % The starred version of this command (\hereapplies*) will not invoke the
488 % `\phantomsection` directive.
489 %
490 \newcommand*{\hereapplies}{%
491     % Check if a star is present in the invocation of the command
492     \@ifstar{\starred@hereapplies}{\phantomsection\starred@hereapplies}%
493 }

```

```

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

```