

The refcount package

Heiko Oberdiek
<heiko.oberdiek at gmail.com>

2010/12/01 v3.2

Abstract

References are not numbers, however they often store numerical data such as section or page numbers. `\ref` or `\pageref` cannot be used for counter assignments or calculations because they are not expandable, generate warnings, or can even be links. The package provides expandable macros to extract the data from references. Packages `hyperref`, `nameref`, `titleref`, and `babel` are supported.

Contents

1	Usage	2
1.1	Setting counters	2
1.2	Expandable commands	2
1.3	Undefined references	3
1.3.1	Check for undefined references	3
1.4	Notes	3
2	Implementation	3
2.1	Loading packages	5
2.2	Defining commands	5
2.3	<code>\setrefcountdefault</code>	6
2.4	<code>\refused</code>	6
2.5	Setting counters by reference data	7
2.5.1	Generic setting	7
2.5.2	User commands	8
2.6	Extracting references	8
2.7	Macros for checking undefined references	10
3	Test	10
3.1	Catcode checks for loading	10
3.2	Macro tests	12
3.3	Test with package <code>titleref</code>	15
4	Installation	16
4.1	Download	16
4.2	Bundle installation	16
4.3	Package installation	16
4.4	Refresh file name databases	17
4.5	Some details for the interested	17
5	History	17
	[1998/04/08 v1.0]	17
	[2000/09/07 v2.0]	17
	[2006/02/20 v3.0]	17
	[2008/08/11 v3.1]	18
	[2010/12/01 v3.2]	18

1 Usage

1.1 Setting counters

The following commands are similar to L^AT_EX's `\setcounter` and `\addtocounter`, but they extract the number value from a reference:

```
\setcounterref, \addtocounterref
\setcounterpageref, \addtocounterpageref
```

They take two arguments:

```
\...counter...ref {\LaTeX counter} {\reference}
```

An undefined references produces the usual L^AT_EX warning and its value is assumed to be zero. Example:

```
\newcounter{ctrA}
\newcounter{ctrB}
\refstepcounter{ctrA}\label{ref:A}
\setcounterref{ctrB}{ref:A}
\addtocounterpageref{ctrB}{ref:A}
```

1.2 Expandable commands

These commands that can be used in expandible contexts (inside calculations, `\edef`, `\csname`, `\write`, ...):

```
\getrefnumber, \getpagerefnumber
```

They take one argument, the reference:

```
\get...refnumber {\reference}
```

The default for undefined references can be changed with macro `\setrefcountdefault`, for example this package calls:

```
\setrefcountdefault{0}
```

Since version 2.0 of this package there is a new command:

```
\getrefbykeydefault {\reference} {\key} {\default}
```

This generalized version allows the extraction of further properties of a reference than the two standard ones. Thus the following properties are supported, if they are available:

Key	Description	Package
<code>\empty</code>	same as <code>\ref</code>	L ^A T _E X
<code>page</code>	same as <code>\pageref</code>	L ^A T _E X
<code>title</code>	section and caption titles	<code>titleref</code>
<code>name</code>	section and caption titles	<code>nameref</code>
<code>anchor</code>	anchor name	<code>hyperref</code>
<code>url</code>	url/file	<code>hyperref/xr</code>

Since version 3.2 the expandable macros described before in this section are expandable in exact two expansion steps.

1.3 Undefined references

Because warnings and assignments cannot be used in expandible contexts, undefined references do not produce a warning, their values are assumed to be zero. Example:

```
\label{ref:here}% somewhere
\refused{ref:here}% see below
\ifodd\getpagerefnumber{ref:here}%
  reference is on an odd page
\else
  reference is on an even page
\fi
```

In case of undefined references the user usually want's to be informed. Also \LaTeX prints a warning at the end of the \LaTeX run. To notify \LaTeX and get a normal warning, just use

```
\refused {\langle reference \rangle}
```

outside the expanding context. Example, see above.

1.3.1 Check for undefined references

In version 3.2 macros were added, that test, whether references are defined.

```
\IfRefUndefinedExpandable {\langle refname \rangle} {\langle then \rangle} {\langle else \rangle}
\IfRefUndefinedBabel {\langle refname \rangle} {\langle then \rangle} {\langle else \rangle}
```

If the reference is not available and therefore undefined, then argument $\langle then \rangle$ is executed, otherwise argument $\langle else \rangle$ is called. Macro `\IfRefUndefinedExpandable` is expandable, but $\langle refname \rangle$ must not contain babel shorthand characters. Macro `\IfRefUndefinedBabel` supports shorthand characters of babel, but it is not expandable.

1.4 Notes

- The method of extracting the number in this package also works in cases, where the reference cannot be used directly, because a package such as `hyperref` has added extra stuff (hyper link), so that the reference cannot be used as number any more.
- If the reference does not contain a number, assignments to a counter will fail of course.

2 Implementation

```
1 (*package)
```

Reload check, especially if the package is not used with \LaTeX .

```
2 \begingroup\catcode61\catcode48\catcode32=10\relax%
3 \catcode13=5 % ^^M
4 \endlinechar=13 %
5 \catcode35=6 % #
6 \catcode39=12 % '
7 \catcode44=12 % ,
8 \catcode45=12 % -
9 \catcode46=12 % .
10 \catcode58=12 % :
11 \catcode64=11 % @
12 \catcode123=1 % {
```

```

13 \catcode125=2 % }
14 \expandafter\let\expandafter\x\csname ver@refcount.sty\endcsname
15 \ifx\x\relax % plain-TeX, first loading
16 \else
17 \def\empty{}%
18 \ifx\x\empty % LaTeX, first loading,
19 % variable is initialized, but \ProvidesPackage not yet seen
20 \else
21 \expandafter\ifx\csname PackageInfo\endcsname\relax
22 \def\x#1#2{%
23 \immediate\write-1{Package #1 Info: #2.}%
24 }%
25 \else
26 \def\x#1#2{\PackageInfo{#1}{#2, stopped}}%
27 \fi
28 \x{refcount}{The package is already loaded}%
29 \aftergroup\endinput
30 \fi
31 \fi
32 \endgroup%

```

Package identification:

```

33 \begingroup\catcode61\catcode48\catcode32=10\relax%
34 \catcode13=5 % ^^M
35 \endlinechar=13 %
36 \catcode35=6 % #
37 \catcode39=12 % '
38 \catcode40=12 % (
39 \catcode41=12 % )
40 \catcode44=12 % ,
41 \catcode45=12 % -
42 \catcode46=12 % .
43 \catcode47=12 % /
44 \catcode58=12 % :
45 \catcode64=11 % @
46 \catcode91=12 % [
47 \catcode93=12 % ]
48 \catcode123=1 % {
49 \catcode125=2 % }
50 \expandafter\ifx\csname ProvidesPackage\endcsname\relax
51 \def\x#1#2#3[#4]{\endgroup
52 \immediate\write-1{Package: #3 #4}%
53 \xdef#1{#4}%
54 }%
55 \else
56 \def\x#1#2[#3]{\endgroup
57 #2[#3]}%
58 \ifx#1@undefined
59 \xdef#1{#3}%
60 \fi
61 \ifx#1\relax
62 \xdef#1{#3}%
63 \fi
64 }%
65 \fi
66 \expandafter\x\csname ver@refcount.sty\endcsname
67 \ProvidesPackage{refcount}%
68 [2010/12/01 v3.2 Data extraction from references (HO)]%
69 \begingroup\catcode61\catcode48\catcode32=10\relax%
70 \catcode13=5 % ^^M
71 \endlinechar=13 %
72 \catcode123=1 % {
73 \catcode125=2 % }

```

```

74 \catcode64=11 % @
75 \def\x{\endgroup
76 \expandafter\edef\csname rc@AtEnd\endcsname{%
77 \endlinechar=\the\endlinechar\relax
78 \catcode13=\the\catcode13\relax
79 \catcode32=\the\catcode32\relax
80 \catcode35=\the\catcode35\relax
81 \catcode61=\the\catcode61\relax
82 \catcode64=\the\catcode64\relax
83 \catcode123=\the\catcode123\relax
84 \catcode125=\the\catcode125\relax
85 }%
86 }%
87 \x\catcode61\catcode48\catcode32=10\relax%
88 \catcode13=5 % ^^M
89 \endlinechar=13 %
90 \catcode35=6 % #
91 \catcode64=11 % @
92 \catcode123=1 % {
93 \catcode125=2 % }
94 \def\TMP@EnsureCode#1#2{%
95 \edef\rc@AtEnd{%
96 \rc@AtEnd
97 \catcode#1=\the\catcode#1\relax
98 }%
99 \catcode#1=#2\relax
100 }
101 \TMP@EnsureCode{33}{12}% !
102 \TMP@EnsureCode{39}{12}% '
103 \TMP@EnsureCode{42}{12}% *
104 \TMP@EnsureCode{45}{12}% -
105 \TMP@EnsureCode{46}{12}% .
106 \TMP@EnsureCode{47}{12}% /
107 \TMP@EnsureCode{91}{12}% [
108 \TMP@EnsureCode{93}{12}% ]
109 \TMP@EnsureCode{96}{12}% '
110 \edef\rc@AtEnd{\rc@AtEnd\noexpand\endinput}

```

2.1 Loading packages

```

111 \begingroup\expandafter\expandafter\expandafter\endgroup
112 \expandafter\ifx\csname RequirePackage\endcsname\relax
113 \input ltxcmds.sty\relax
114 \input infwarerr.sty\relax
115 \else
116 \RequirePackage{ltxcmds}[2010/04/08]%
117 \RequirePackage{infwarerr}[2010/04/08]%
118 \fi

```

2.2 Defining commands

\rc@IfDefinable

```

119 \ltx@ifundefined{ifdefinable}{%
120 \def\rc@IfDefinable#1{%
121 \ifx#1\ltx@ifundefined
122 \expandafter\ltx@firstofone
123 \else
124 \ifx#1\relax
125 \expandafter\expandafter\expandafter\ltx@firstofone
126 \else
127 \@PackageError{refcount}{%
128 Command \string#1 is already defined.\MessageBreak
129 It will not be redefined by this package%
130 }\@ehc

```

```

131      \expandafter\expandafter\expandafter\ltx@gobble
132      \fi
133      \fi
134    }%
135  }{%
136    \let\rc@ifdefinable\@ifdefinable
137  }

```

\rc@RobustDefOne

```

138 \ltx@ifundefined{protected}{%
139   \ltx@ifundefined{DeclareRobustCommand}{%
140     \def\rc@RobustDefOne#1#2#3#4{%
141       \rc@ifdefinable#3{%
142         #1\def#3##1{#4}%
143       }%
144     }%
145   }{%
146     \def\rc@RobustDefOne#1#2#3#4{%
147       \rc@ifdefinable#3{%
148         \DeclareRobustCommand#2#3[1]{#4}%
149       }%
150     }%
151   }%
152 }{%
153   \def\rc@RobustDefOne#1#2#3#4{%
154     \rc@ifdefinable#3{%
155       \protected#1\def#3##1{#4}%
156     }%
157   }%
158 }

```

\rc@newcommand

```

159 \ltx@ifundefined{newcommand}{%
160   \def\rc@newcommand*#1[#2]#3{% hash-ok
161     \rc@ifdefinable#1{%
162       \ifcase#2 %
163         \def#1{#3}%
164       \or
165         \def#1##1{#3}%
166       \or
167         \def#1##1##2{#3}%
168       \else
169         \rc@InternalError
170       \fi
171     }%
172   }%
173 }{%
174   \let\rc@newcommand\newcommand
175 }

```

2.3 \setrefcountdefault

\setrefcountdefault

```

176 \rc@RobustDefOne\long{}\setrefcountdefault{%
177   \def\rc@default{#1}%
178 }

179 \setrefcountdefault{0}

```

2.4 \refused

\refused

```

180 \ltx@ifundefined{G@refundefinedtrue}{%
181   \rc@RobustDefOne{}{*}\refused{%
182     \begingroup
183       \csname @safe@activetrue\endcsname
184       \ltx@ifundefined{r@#1}{%
185         \protect\G@refundefinedtrue
186         \rc@WarningUndefined{#1}%
187       }{}%
188     \endgroup
189   }%
190 }{%
191   \rc@RobustDefOne{}{*}\refused{%
192     \begingroup
193       \csname @safe@activetrue\endcsname
194       \ltx@ifundefined{r@#1}{%
195         \csname protect\expandafter\endcsname
196         \csname G@refundefinedtrue\endcsname
197         \rc@WarningUndefined{#1}%
198       }{}%
199     \endgroup
200   }%
201 }

\rc@WarningUndefined

202 \ltx@ifundefined{@latex@warning}{%
203   \def\rc@WarningUndefined#1{%
204     \ltx@ifundefined{thepage}{%
205       \def\thepage{\number\count0 }%
206     }{}%
207     \@PackageWarning{refcount}{%
208       Reference ‘#1’ on page \thepage\space undefined%
209     }%
210   }%
211 }{%
212   \def\rc@WarningUndefined#1{%
213     \@latex@warning{%
214       Reference ‘#1’ on page \thepage\space undefined%
215     }%
216   }%
217 }

```

2.5 Setting counters by reference data

2.5.1 Generic setting

`\rc@set` Generic command for `\{set,addto\}counter{page,}ref:`

```

#1: \setcounter, \addtocounter
#2: \ltx@car (for \ref), \ltx@cartwo (for \pageref)
#3: LATEX counter
#4: reference

```

```

218 \def\rc@set#1#2#3#4{%
219   \begingroup
220     \csname @safe@activetrue\endcsname
221     \rc@refused{#4}%
222     \expandafter\rc@set\csname r@#4\endcsname{#1}{#2}{#3}%
223   \endgroup
224 }

```

```

\rc@@set      #1: \r@<...>
               #2: \setcounter, \addtocounter
               #3: \ltx@car (for \ref), \ltx@cartwo (for \pageref)
               #4: LATEX counter

225 \def\rc@@set#1#2#3#4{%
226   \ifx#1\relax
227     #2{#4}{\rc@default}%
228   \else
229     #2{#4}{%
230       \expandafter#3#1\rc@default\rc@default\@nil
231     }%
232   \fi
233 }

```

2.5.2 User commands

```

\setcounterref

234 \rc@newcommand*\setcounterref}[0]{%
235   \rc@set\setcounter\ltx@car
236 }

```

```

\addtocounterref

237 \rc@newcommand*\addtocounterref}[0]{%
238   \rc@set\addtocounter\ltx@car
239 }

```

```

\setcounterpageref

240 \rc@newcommand*\setcounterpageref}[0]{%
241   \rc@set\setcounter\ltx@cartwo
242 }

```

```

\addtocounterpageref

243 \rc@newcommand*\addtocounterpageref}[0]{%
244   \rc@set\addtocounter\ltx@cartwo
245 }

```

2.6 Extracting references

```

\getrefnumber

246 \rc@newcommand*\getrefnumber}[1]{%
247   \romannumeral
248   \ltx@ifundefined{r@#1}{%
249     \expandafter\ltx@zero
250     \rc@default
251   }{%
252     \expandafter\expandafter\expandafter\rc@extract@
253     \expandafter\expandafter\expandafter!%
254     \csname r@#1\expandafter\endcsname
255     \expandafter{\rc@default}\@nil
256   }%
257 }

```

```

\getpagerefnumber

258 \rc@newcommand*\getpagerefnumber}[1]{%
259   \romannumeral
260   \ltx@ifundefined{r@#1}{%
261     \expandafter\ltx@zero
262     \rc@default
263   }{%
264     \expandafter\expandafter\expandafter\rc@extract@page

```



```

265 \expandafter\expandafter\expandafter!%
266 \csname r@#1\expandafter\expandafter\expandafter\endcsname
267 \expandafter\expandafter\expandafter{%
268 \expandafter\rc@default
269 \expandafter}\expandafter{\rc@default}\@nil
270 }%
271 }

\getrefbykeydefault
272 \rc@newcommand*\getrefbykeydefault}[2]{%
273 \romannumeral
274 \expandafter\rc@getrefbykeydefault
275 \csname r@#1\expandafter\endcsname
276 \csname rc@extract@#2\endcsname
277 }

\rc@getrefbykeydefault #1: \r@<...>
#2: \rc@extract@<...>
#3: default
278 \long\def\rc@getrefbykeydefault#1#2#3{%
279 \ifx#1\relax
280 % reference is undefined
281 \ltx@ReturnAfterElseFi{%
282 \ltx@zero
283 #3%
284 }%
285 \else
286 \ltx@ReturnAfterFi{%
287 \ifx#2\relax
288 % extract method is missing
289 \ltx@ReturnAfterElseFi{%
290 \ltx@zero
291 #3%
292 }%
293 \else
294 \ltx@ReturnAfterFi{%
295 \expandafter
296 \rc@generic#1{#3}{#3}{#3}{#3}{#3}\@nil#2{#3}%
297 }%
298 \fi
299 }%
300 \fi
301 }

\rc@generic #1: first item in \r@<...>
#2: remaining items in \r@<...>
#3: \rc@extract@<...>
#4: default
302 \long\def\rc@generic#1#2\@nil#3#4{%
303 #3{#1\TR@TitleReference\@empty{#4}\@nil}{#1}#2\@nil
304 }

\rc@extract@
305 \long\def\rc@extract@#1#2#3\@nil{%
306 \ltx@zero
307 #2%
308 }

\rc@extract@page
309 \long\def\rc@extract@page#1#2#3#4\@nil{%
310 \ltx@zero

```

```

311 #3%
312 }

\rc@extract@name

313 \long\def\rc@extract@name#1#2#3#4#5\@nil{%
314 \ltx@zero
315 #4%
316 }

\rc@extract@anchor

317 \long\def\rc@extract@anchor#1#2#3#4#5#6\@nil{%
318 \ltx@zero
319 #5%
320 }

\rc@extract@url

321 \long\def\rc@extract@url#1#2#3#4#5#6#7\@nil{%
322 \ltx@zero
323 #6%
324 }

\rc@extract@title

325 \long\def\rc@extract@title#1#2\@nil{%
326 \rc@@extract@title#1%
327 }

\rc@@extract@title

328 \long\def\rc@@extract@title#1\TR@TitleReference#2#3#4\@nil{%
329 \ltx@zero
330 #3%
331 }

```

2.7 Macros for checking undefined references

```

\IfRefUndefinedExpandable

332 \rc@newcommand*{\IfRefUndefinedExpandable}[1]{%
333 \ltx@ifundefined{r@#1}\ltx@firstoftwo\ltx@secondoftwo
334 }

\IfRefUndefinedBabel

335 \rc@RobustDefOne{*}\IfRefUndefinedBabel{%
336 \begingroup
337 \csname safe@actives@true\endcsname
338 \expandafter\expandafter\expandafter\endgroup
339 \expandafter\ifx\csname r@#1\endcsname\relax
340 \expandafter\ltx@firstoftwo
341 \else
342 \expandafter\ltx@secondoftwo
343 \fi
344 }

345 \rc@AtEnd%
346 \</package>

```

3 Test

3.1 Catcode checks for loading

```

347 \<test1>

```

```

348 \catcode'\{=1 %
349 \catcode'\}=2 %
350 \catcode'\#=6 %
351 \catcode'\@=11 %
352 \expandafter\ifx\csname count@\endcsname\relax
353   \countdef\count@=255 %
354 \fi
355 \expandafter\ifx\csname @gobble\endcsname\relax
356   \long\def\@gobble#1{%
357 \fi
358 \expandafter\ifx\csname @firstofone\endcsname\relax
359   \long\def\@firstofone#1{#1}%
360 \fi
361 \expandafter\ifx\csname loop\endcsname\relax
362   \expandafter\@firstofone
363 \else
364   \expandafter\@gobble
365 \fi
366 {%
367   \def\loop#1\repeat{%
368     \def\body{#1}%
369     \iterate
370   }%
371   \def\iterate{%
372     \body
373     \let\next\iterate
374   \else
375     \let\next\relax
376   \fi
377   \next
378   }%
379   \let\repeat=\fi
380 }%
381 \def\RestoreCatcodes{}
382 \count@=0 %
383 \loop
384   \edef\RestoreCatcodes{%
385     \RestoreCatcodes
386     \catcode\the\count@=\the\catcode\count@\relax
387   }%
388 \ifnum\count@<255 %
389   \advance\count@ 1 %
390 \repeat
391
392 \def\RangeCatcodeInvalid#1#2{%
393   \count@=#1\relax
394   \loop
395     \catcode\count@=15 %
396   \ifnum\count@<#2\relax
397     \advance\count@ 1 %
398   \repeat
399 }
400 \def\RangeCatcodeCheck#1#2#3{%
401   \count@=#1\relax
402   \loop
403     \ifnum#3=\catcode\count@
404   \else
405     \errmessage{%
406       Character \the\count@\space
407       with wrong catcode \the\catcode\count@\space
408       instead of \number#3%
409     }%

```

```

410 \fi
411 \ifnum\count@<#2\relax
412 \advance\count@ 1 %
413 \repeat
414 }
415 \def\space{ }
416 \expandafter\ifx\csname LoadCommand\endcsname\relax
417 \def\LoadCommand{\input refcount.sty\relax}%
418 \fi
419 \def\Test{%
420 \RangeCatcodeInvalid{0}{47}%
421 \RangeCatcodeInvalid{58}{64}%
422 \RangeCatcodeInvalid{91}{96}%
423 \RangeCatcodeInvalid{123}{255}%
424 \catcode'\@=12 %
425 \catcode'\=0 %
426 \catcode'\%=14 %
427 \LoadCommand
428 \RangeCatcodeCheck{0}{36}{15}%
429 \RangeCatcodeCheck{37}{37}{14}%
430 \RangeCatcodeCheck{38}{47}{15}%
431 \RangeCatcodeCheck{48}{57}{12}%
432 \RangeCatcodeCheck{58}{63}{15}%
433 \RangeCatcodeCheck{64}{64}{12}%
434 \RangeCatcodeCheck{65}{90}{11}%
435 \RangeCatcodeCheck{91}{91}{15}%
436 \RangeCatcodeCheck{92}{92}{0}%
437 \RangeCatcodeCheck{93}{96}{15}%
438 \RangeCatcodeCheck{97}{122}{11}%
439 \RangeCatcodeCheck{123}{255}{15}%
440 \RestoreCatcodes
441 }
442 \Test
443 \csname @@end\endcsname
444 \end
445 </test1>

```

3.2 Macro tests

```

446 <*test2>
447 \errorcontextlines=10000 %
448 \showboxbreadth=10000 %
449 \showboxdepth=10000 %
450 \begingroup\expandafter\expandafter\expandafter\endgroup
451 \expandafter\ifx\csname RequirePackage\endcsname\relax
452 \input refcount.sty\relax
453 \else
454 \RequirePackage{refcount}[2010/12/01]%
455 \fi
456 \catcode'\@=11 %
457 \begingroup\expandafter\expandafter\expandafter\endgroup
458 \expandafter\ifx\csname @onelevel@sanitize\endcsname\relax
459 \begingroup\expandafter\expandafter\expandafter\endgroup
460 \expandafter\ifx\csname detokenize\endcsname\relax
461 \def\strip@prefix#1->{%
462 \def\@onelevel@sanitize#1{%
463 \edef#1{%
464 \expandafter\strip@prefix\meaning#1%
465 }%
466 }%
467 \else
468 \def\@onelevel@sanitize#1{%
469 \edef#1{%

```

```

470         \detokenize\expandafter{#1}%
471     }%
472 }%
473 \fi
474 \fi
475 \def\msg#{\immediate\write16}
476 \def\empty{}
477 \def\space{ }

478 \def\r@foo{{\empty 1}{\empty 2}}
479 \long\def\test#1#2{%
480     \begingroup
481         \setbox0=\hbox{%
482             \def\TestTask{#1}%
483             \@onelevel@sanitize\TestTask
484             \msg{* \TestTask}%
485             \expandafter\expandafter\expandafter\def
486             \expandafter\expandafter\expandafter\TestResult
487             \expandafter\expandafter\expandafter{%
488                 #1%
489             }%
490             \def\TestExpected{#2}%
491             \ifx\TestResult\TestExpected
492                 \msg{ \space ok.}%
493             \else
494                 \@onelevel@sanitize\TestResult
495                 \@onelevel@sanitize\TestExpected
496                 \msg{ \space Result: \space\space[\TestResult]}%
497                 \msg{ \space Expected: [\TestExpected]}%
498                 \errmessage{Test failed!}%
499             \fi
500         }%
501         \ifdim\wd0=Opt %
502         \else
503             \showbox0 %
504         \fi
505     \endgroup
506 }
507 \test{\getrefnumber{foo}}{\empty 1}
508 \test{\getpagerefnumber{foo}}{\empty 2}
509 \test{\getrefbykeydefault{foo}{}{\empty default}}{\empty 1}
510 \test{\getrefbykeydefault{foo}{page}{\empty default}}{\empty 2}
511 \test{\getrefbykeydefault{foo}{name}{\empty default}}{\empty default}
512 \test{\getrefbykeydefault{foo}{anchor}{\empty default}}{\empty default}
513 \test{\getrefbykeydefault{foo}{url}{\empty default}}{\empty default}
514 \test{\getrefbykeydefault{foo}{title}{\empty default}}{\empty default}
515 \msg{}
516 \def\r@foo{{}{}}{}{}{}{}{}{}{}{}{}{}
517 \def\Test#1#2\\{%
518     \test{#1{foo}#2}{}%
519 }
520 \def\TestGroup{%
521     \Test\getrefnumber\\%
522     \Test\getpagerefnumber\\%
523     \Test\getrefbykeydefault{}{}\\%
524     \Test\getrefbykeydefault{page}{}\\%
525     \Test\getrefbykeydefault{anchor}{}\\%
526     \Test\getrefbykeydefault{name}{}\\%
527     \Test\getrefbykeydefault{url}{}\\%
528 }
529 \TestGroup
530 \Test\getrefbykeydefault{title}{}\\%
531 \msg{}

```

```

532 \def\r@foo{\par\par\par\par\par\par\par\par}
533 \long\def\Test#1#2\{
534   \test{#1{foo}#2}{\par}%
535 }
536 \TestGroup
537 \test{\getrefbykeydefault{title}{-}{-}}
538 \msg{}
539 \def\r@foo{{ }{ }{ }{ }{ }}
540 \def\Test#1#2\{
541   \test{#1{foo}#2}{ }%
542 }
543 \TestGroup
544 \msg{}
545 \long\def\TestDefault#1{
546   \begingroup
547     \setrefcountdefault{#1}%
548     \test{\getrefnumber{foo}}{#1}%
549     \test{\getpagerefnumber{foo}}{#1}%
550   \endgroup
551 }
552 \def\TestDefaultX{
553   \TestDefault}%
554   \TestDefault{\par}%
555   \TestDefault{ }%
556   \TestDefault{\space}%
557 }
558 \let\r@foo\@undefined
559 \TestDefaultX
560 \let\r@foo\relax
561 \TestDefaultX
562 \def\r@foo{
563   \TestDefaultX

564 \msg{}
565 \long\def\Test#1#2#3#4{
566   \begingroup
567     \def\TestTask{#1}%
568     \@onelevel@sanitize\TestTask
569     \msg{* [\TestTask]}%
570     \edef\TestResultA{\IfRefUndefinedExpandable{#1}{#2}{#3}}%
571     \IfRefUndefinedBabel{#1}{%
572       \def\TestResultB{#2}%
573     }{%
574       \def\TestResultB{#3}%
575     }%
576     \def\TestExpected{#4}%
577     \ifx\TestResultA\TestExpected
578       \msg{ \space ok.}%
579     \else
580       \begingroup
581         \@onelevel@sanitize\TestResultA
582         \@onelevel@sanitize\TestExpected
583         \msg{ \space Result: \space\space[\TestResultA]}%
584         \msg{ \space Expected: [\TestExpected]}%
585         \errmessage{Test failed!}%
586       \endgroup
587     \fi
588     \ifx\TestResultB\TestExpected
589       \msg{ \space ok.}%
590     \else
591       \begingroup
592         \@onelevel@sanitize\TestResultB
593         \@onelevel@sanitize\TestExpected

```

```

594      \msg{ \space Result: \space\space[\TestResultB]}%
595      \msg{ \space Expected: [\TestExpected]}%
596      \errmessage{Test failed!}%
597    \endgroup
598  \fi
599 \endgroup
600 }
601 \begingroup
602   \def\r@foo{{}}{}%
603   \let\r@bar\@undefined
604   \let\r@xyz\relax
605   \Test{foo}{true}{false}{false}%
606   \Test{bar}{true}{false}{true}%
607   \Test{xyz}{true}{false}{true}%
608 \endgroup
609 \csname @@end\endcsname\end
610 </test2>

```

3.3 Test with package titleref

```

611 <*test3>
612 \NeedsTeXFormat{LaTeX2e}
613 \documentclass{article}
614 \usepackage{refcount}[2010/12/01]
615 (test4)\usepackage{nameref}
616 \usepackage{titleref}
617 \begin{document}
618 \section{Hello World}
619 \label{sec:hello}
620 \section{\hbox{xy}}
621 \label{sec:foo}
622 %
623 \makeatletter
624 \@ifundefined{r@sec:hello}{%
625   \typeout{==> Compile twice!}%
626 }{%
627   \def\test#1#2{%
628     \begingroup
629       \def\TestTask{#1}%
630       \@onelevel@sanitize\TestTask
631       \typeout{* \TestTask}%
632       \expandafter\expandafter\expandafter\def
633       \expandafter\expandafter\expandafter\TestResult
634       \expandafter\expandafter\expandafter{%
635         #1%
636       }%
637       \def\TestExpected{#2}%
638       \ifx\TestResult\TestExpected
639         \typeout{ \space ok.}%
640       \else
641         \@onelevel@sanitize\TestResult
642         \@onelevel@sanitize\TestExpected
643         \typeout{ \space Result: \space\space[\TestResult]}%
644         \typeout{ \space Expected: [\TestExpected]}%
645         \errmessage{Test failed!}%
646       \fi
647     \endgroup
648   }%
649   \test{\getrefbykeydefault{sec:hello}{title}}{Hello World}%
650   \test{\getrefbykeydefault{sec:foo}{title}}{\hbox{xy}}%
651 \begingroup
652   \def\hbox#1{[#1]}% hash-ok
653   \test{\getrefbykeydefault{sec:foo}{title}}{\hbox{xy}}%

```

```

654 \endgroup
655 }
656 \makeatother
657 \end{document}
658 </test3>

```

4 Installation

4.1 Download

Package. This package is available on CTAN¹:

[CTAN:macros/latex/contrib/oberdiek/refcount.dtx](#) The source file.

[CTAN:macros/latex/contrib/oberdiek/refcount.pdf](#) Documentation.

Bundle. All the packages of the bundle ‘oberdiek’ are also available in a TDS compliant ZIP archive. There the packages are already unpacked and the documentation files are generated. The files and directories obey the TDS standard.

[CTAN:install/macros/latex/contrib/oberdiek.tds.zip](#)

TDS refers to the standard “A Directory Structure for T_EX Files” ([CTAN:tds/tds.pdf](#)). Directories with `texmf` in their name are usually organized this way.

4.2 Bundle installation

Unpacking. Unpack the `oberdiek.tds.zip` in the TDS tree (also known as `texmf` tree) of your choice. Example (linux):

```
unzip oberdiek.tds.zip -d ~/texmf
```

Script installation. Check the directory `TDS:scripts/oberdiek/` for scripts that need further installation steps. Package `attachfile2` comes with the Perl script `pdfatfi.pl` that should be installed in such a way that it can be called as `pdfatfi`. Example (linux):

```

chmod +x scripts/oberdiek/pdfatfi.pl
cp scripts/oberdiek/pdfatfi.pl /usr/local/bin/

```

4.3 Package installation

Unpacking. The `.dtx` file is a self-extracting `docstrip` archive. The files are extracted by running the `.dtx` through plain T_EX:

```
tex refcount.dtx
```

TDS. Now the different files must be moved into the different directories in your installation TDS tree (also known as `texmf` tree):

```

refcount.sty      → tex/latex/oberdiek/refcount.sty
refcount.pdf      → doc/latex/oberdiek/refcount.pdf
test/refcount-test1.tex → doc/latex/oberdiek/test/refcount-test1.tex
test/refcount-test2.tex → doc/latex/oberdiek/test/refcount-test2.tex
test/refcount-test3.tex → doc/latex/oberdiek/test/refcount-test3.tex
test/refcount-test4.tex → doc/latex/oberdiek/test/refcount-test4.tex
refcount.dtx      → source/latex/oberdiek/refcount.dtx

```

If you have a `docstrip.cfg` that configures and enables `docstrip`’s TDS installing feature, then some files can already be in the right place, see the documentation of `docstrip`.

¹<http://ftp.ctan.org/tex-archive/>

4.4 Refresh file name databases

If your $\text{T}_{\text{E}}\text{X}$ distribution (te $\text{T}_{\text{E}}\text{X}$, mik $\text{T}_{\text{E}}\text{X}$, ...) relies on file name databases, you must refresh these. For example, te $\text{T}_{\text{E}}\text{X}$ users run `texhash` or `mktextlsr`.

4.5 Some details for the interested

Attached source. The PDF documentation on CTAN also includes the `.dtx` source file. It can be extracted by AcrobatReader 6 or higher. Another option is `pdftk`, e.g. unpack the file into the current directory:

```
pdftk refcount.pdf unpack_files output .
```

Unpacking with $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$. The `.dtx` chooses its action depending on the format:

plain $\text{T}_{\text{E}}\text{X}$: Run `docstrip` and extract the files.

$\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$: Generate the documentation.

If you insist on using $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ for `docstrip` (really, `docstrip` does not need $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$), then inform the autodetect routine about your intention:

```
latex \let\install=y\input{refcount.dtx}
```

Do not forget to quote the argument according to the demands of your shell.

Generating the documentation. You can use both the `.dtx` or the `.drv` to generate the documentation. The process can be configured by the configuration file `ltxdoc.cfg`. For instance, put this line into this file, if you want to have A4 as paper format:

```
\PassOptionsToClass{a4paper}{article}
```

An example follows how to generate the documentation with pdf $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$:

```
pdflatex refcount.dtx
makeindex -s gind.ist refcount.idx
pdflatex refcount.dtx
makeindex -s gind.ist refcount.idx
pdflatex refcount.dtx
```

5 History

[1998/04/08 v1.0]

- First public release, written as answer in the newsgroup `comp.text.tex`:
“Re: Adding a `\ref` to a counter?”²

[2000/09/07 v2.0]

- Documentation added.
- LPPL 1.2
- Package rewritten, new commands added.

[2006/02/20 v3.0]

- Support for `hyperref` and `nameref` improved.
- Support for `titleref` and `babel`’s shorthands added.
- New: `\refused`, `\getrefbykeydefault`

²Url: <http://groups.google.com/group/comp.text.tex/msg/c3f2a135ef5ee528>

[2008/08/11 v3.1]

- Code is not changed.
- URLs updated.

[2010/12/01 v3.2]

- `\IfRefUndefinedExpandable` and `\IfRefUndefinedBabel` added.
- `\getrefnumber`, `\getpagerefnumber`, `\getrefbykeydefault` are expandable in exact two expansion steps.
- Non-expandable macros are made robust.
- Test files added.

6 Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; plain numbers refer to the code lines where the entry is used.

Symbols	
<code>\#</code>	350
<code>\%</code>	426
<code>\@</code>	351, 424, 456
<code>\@PackageError</code>	127
<code>\@PackageWarning</code>	207
<code>\@ehc</code>	130
<code>\@empty</code>	303
<code>\@firstofone</code>	359, 362
<code>\@gobble</code>	356, 364
<code>\@ifdefinable</code>	136
<code>\@ifundefined</code>	624
<code>\@latex@warning</code>	213
<code>\@nil</code> ..	230, 255, 269, 296, 302, 303, 305, 309, 313, 317, 321, 325, 328
<code>\@onelevel@sanitize</code>	462, 468, 483, 494, 495, 568, 581, 582, 592, 593, 630, 641, 642
<code>\@undefined</code>	58, 558, 603
<code>\@</code>	425, 517, 521, 522, 523, 524, 525, 526, 527, 530, 533, 540
<code>\{</code>	348
<code>\}</code>	349
A	
<code>\addtocounter</code>	238, 244
<code>\addtocounterpageref</code>	243
<code>\addtocounterref</code>	237
<code>\advance</code>	389, 397, 412
<code>\aftergroup</code>	29
B	
<code>\begin</code>	617
<code>\body</code>	368, 372
C	
<code>\catcode</code> ..	2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13, 33, 34, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49,
<code>\count</code>	69, 70, 72, 73, 74, 78, 79, 80, 81, 82, 83, 84, 87, 88, 90, 91, 92, 93, 97, 99, 348, 349, 350, 351, 386, 395, 403, 407, 424, 425, 426, 456
<code>\count@</code>	205
<code>\count@</code>	353, 382, 386, 388, 389, 393, 395, 396, 397, 401, 403, 406, 407, 411, 412
<code>\countdef</code>	353
<code>\csname</code>	14, 21, 50, 66, 76, 112, 183, 193, 195, 196, 220, 222, 254, 266, 275, 276, 337, 339, 352, 355, 358, 361, 416, 443, 451, 458, 460, 609
D	
<code>\DeclareRobustCommand</code>	148
<code>\detokenize</code>	470
<code>\documentclass</code>	613
E	
<code>\empty</code>	17, 18, 476, 478, 507, 508, 509, 510, 511, 512, 513, 514
<code>\end</code>	444, 609, 657
<code>\endcsname</code>	14, 21, 50, 66, 76, 112, 183, 193, 195, 196, 220, 222, 254, 266, 275, 276, 337, 339, 352, 355, 358, 361, 416, 443, 451, 458, 460, 609
<code>\endinput</code>	29, 110
<code>\endlinechar</code>	4, 35, 71, 77, 89
<code>\errmessage</code> ...	405, 498, 585, 596, 645
<code>\errorcontextlines</code>	447
G	
<code>\G@refundefinedtrue</code>	185
<code>\getpagerefnumber</code> ..	258, 508, 522, 549
<code>\getrefbykeydefault</code>	272, 509, 510, 511,

	512, 513, 514, 523, 524, 525, 526, 527, 530, 537, 649, 650, 653		
\getrefnumber	246, 507, 521, 548	
H			
\hbox	481, 620, 650, 652, 653	
I			
\ifcase	162	
\ifdim	501	
\ifnum	388, 396, 403, 411	
\IfRefUndefinedBabel	335, 571	
\IfRefUndefinedExpandable	3, 332, 570		
\ifx	15, 18, 21, 50, 58, 61, 112, 121, 124, 226, 279, 287, 339, 352, 355, 358, 361, 416, 451, 458, 460, 491, 577, 588, 638	
\immediate	23, 52, 475	
\input	113, 114, 417, 452	
\iterate	369, 371, 373	
L			
\label	619, 621	
\LoadCommand	417, 427	
\loop	367, 383, 394, 402	
\ltx@car	235, 238	
\ltx@cartwo	241, 244	
\ltx@firstofone	122, 125	
\ltx@firstoftwo	333, 340	
\ltx@gobble	131	
\ltx@ifundefined	119, 138, 139, 159, 180, 184, 194, 202	
\ltx@ifundefined	..	204, 248, 260, 333	
\ltx@ReturnAfterElseFi	281, 289	
\ltx@ReturnAfterFi	286, 294	
\ltx@secondoftwo	333, 342	
\ltx@undefined	121	
\ltx@zero	249, 261, 282, 290, 306, 310, 314, 318, 322, 329	
M			
\makeatletter	623	
\makeatother	656	
\meaning	464	
\MessageBreak	128	
\msg	475, 484, 492, 496, 497, 515, 531, 538, 544, 564, 569, 578, 583, 584, 589, 594, 595	
N			
\NeedsTeXFormat	612	
\newcommand	174	
\next	373, 375, 377	
\number	205, 408	
P			
\PackageInfo	26	
\par	532, 534, 554	
\protect	185	
\protected	155	
\ProvidesPackage	19, 67	
R			
\r@bar	603	
\r@foo	478, 516, 532, 539, 558, 560, 562, 602	
\r@xyz	604	
\RangeCatcodeCheck	400, 428, 429, 430, 431, 432, 433, 434, 435, 436, 437, 438, 439	
\RangeCatcodeInvalid	392, 420, 421, 422, 423	
\rc@@extract@title	326, 328	
\rc@@set	222, 225	
\rc@AtEnd	95, 96, 110, 345	
\rc@default	177, 227, 230, 250, 255, 262, 268, 269	
\rc@extract@	252, 305	
\rc@extract@anchor	317	
\rc@extract@name	313	
\rc@extract@page	264, 309	
\rc@extract@title	325	
\rc@extract@url	321	
\rc@generic	296, 302	
\rc@getrefbykeydefault	274, 278	
\rc@ifdefinable	119, 141, 147, 154, 161		
\rc@InternalError	169	
\rc@newcommand	159, 234, 237, 240, 243, 246, 258, 272, 332	
\rc@refused	221	
\rc@RobustDefOne	138, 176, 181, 191, 335		
\rc@set	218, 235, 238, 241, 244	
\rc@WarningUndefined	..	186, 197, 202	
\refused	180	
\repeat	367, 379, 390, 398, 413	
\RequirePackage	116, 117, 454	
\RestoreCatcodes	..	381, 384, 385, 440	
\romannumeral	247, 259, 273	
S			
\section	618, 620	
\setbox	481	
\setcounter	235, 241	
\setcounterpageref	240	
\setcounterref	234	
\setrefcountdefault	... 176, 179, 547		
\showbox	503	
\showboxbreadth	448	
\showboxdepth	449	
\space	. 208, 214, 406, 407, 415, 477, 492, 496, 497, 556, 578, 583, 584, 589, 594, 595, 639, 643, 644		
\strip@prefix	461, 464	
T			
\Test	419, 442, 517, 521, 522, 523, 524, 525, 526, 527, 530, 533, 540, 565, 605, 606, 607	
\test	.. 479, 507, 508, 509, 510, 511, 512, 513, 514, 518, 534, 537, 541, 548, 549, 627, 649, 650, 653		
\TestDefault	.. 545, 553, 554, 555, 556		
\TestDefaultX 552, 559, 561, 563		
\TestExpected 490, 491, 495, 497, 576, 577, 582, 584, 588, 593, 595, 637, 638, 642, 644		
\TestGroup	520, 529, 536, 543	

