

The refcount package

Heiko Oberdiek*
<heiko.oberdiek at gmail.com>

2016/05/16 v3.5

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>	7
2.4 <code>\refused</code>	7
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	11
3.1 Catcode checks for loading	11
3.2 Macro tests	12
3.3 Test with package <code>titleref</code>	15
4 Installation	17
4.1 Download	17
4.2 Bundle installation	17
4.3 Package installation	17
4.4 Refresh file name databases	18
4.5 Some details for the interested	18
5 Catalogue	18

*Please report any issues at <https://github.com/ho-tex/oberdiek/issues>

6 History	19
[1998/04/08 v1.0]	19
[2000/09/07 v2.0]	19
[2006/02/20 v3.0]	19
[2008/08/11 v3.1]	20
[2010/12/01 v3.2]	20
[2011/06/22 v3.3]	20
[2011/10/16 v3.4]	20
[2016/05/16 v3.5]	20
7 Index	20

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

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 expandable 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 {<reference>}
```

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 {<refname>} {<then>} {<else>}
\IfRefUndefinedBabel {<refname>} {<then>} {<else>}
```

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 [2016/05/16 v3.5 Data extraction from label 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}[2011/11/09]%
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@undefined
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 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

```

\rc@RobustDefZero 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   \def\rc@RobustDefZero#1#2{%
146     \rc@ifdefinable#1{%
147       \def#1{#2}%
148     }%
149   }%
150 }{%
151   \def\rc@RobustDefOne#1#2#3#4{%
152     \rc@ifdefinable#3{%
153       \DeclareRobustCommand#2#3[1]{#4}%
154     }%
155   }%
156   \def\rc@RobustDefZero#1#2{%
157     \rc@ifdefinable#1{%
158       \DeclareRobustCommand#1{#2}%
159     }%
160   }%
161 }%
162 }{%
163   \def\rc@RobustDefOne#1#2#3#4{%
164     \rc@ifdefinable#3{%
165       \protected#1\def#3##1{#4}%
166     }%
167   }%
168   \def\rc@RobustDefZero#1#2{%
169     \rc@ifdefinable#1{%
170       \protected\def#1{#2}%
171     }%
172   }%
173 }

```

\rc@newcommand

```

174 \ltx@ifundefined{newcommand}{%
175   \def\rc@newcommand*#1[#2]#3{% hash-ok
176     \rc@ifdefinable#1{%
177       \ifcase#2 %
178         \def#1{#3}%
179       \or
180         \def#1##1{#3}%
181       \or
182         \def#1##1##2{#3}%
183       \else
184         \rc@InternalError
185       \fi
186     }%

```

```

187 }%
188 }{%
189 \let\rc@newcommand\newcommand
190 }

```

2.3 \setrefcountdefault

\setrefcountdefault

```

191 \rc@RobustDefOne\long{ }\setrefcountdefault{%
192 \def\rc@default{#1}%
193 }
194 \setrefcountdefault{0}

```

2.4 \refused

\refused

```

195 \ltx@ifundefined{G@refunddefinedtrue}{%
196 \rc@RobustDefOne{ }{*}\refused{%
197 \begingroup
198 \csname @safe@activetrue\endcsname
199 \ltx@ifundefined{r@#1}{%
200 \protect\G@refunddefinedtrue
201 \rc@WarningUndefined{#1}%
202 }{%
203 \endgroup
204 }%
205 }{%
206 \rc@RobustDefOne{ }{*}\refused{%
207 \begingroup
208 \csname @safe@activetrue\endcsname
209 \ltx@ifundefined{r@#1}{%
210 \csname protect\expandafter\endcsname
211 \csname G@refunddefinedtrue\endcsname
212 \rc@WarningUndefined{#1}%
213 }{%
214 \endgroup
215 }%
216 }

```

\rc@WarningUndefined

```

217 \ltx@ifundefined{@latex@warning}{%
218 \def\rc@WarningUndefined#1{%
219 \ltx@ifundefined{thepage}{%
220 \def\thepage{\number\count0 }%
221 }{%
222 \@PackageWarning{refcount}{%
223 Reference `#1' on page \thepage\space undefined%
224 }%
225 }%
226 }{%
227 \def\rc@WarningUndefined#1{%
228 \@latex@warning{%
229 Reference `#1' on page \thepage\space undefined%
230 }%
231 }%
232 }

```

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

233 \def\rc@set#1#2#3#4{%
234 \begingroup
235 \csname @safe@activestruel\endcsname
236 \refused{#4}%
237 \expandafter\rc@@set\csname r@#4\endcsname{#1}{#2}{#3}%
238 \endgroup
239 }

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

240 \def\rc@@set#1#2#3#4{%
241 \ifx#1\relax
242 #2{#4}{\rc@default}%
243 \else
244 #2{#4}{%
245 \expandafter#3#1\rc@default\rc@default\@nil
246 }%
247 \fi
248 }

```

2.5.2 User commands

```

\setcounterref
249 \rc@RobustDefZero\setcounterref{%
250 \rc@set\setcounter\ltx@car
251 }

```

```

\addtocounterref
252 \rc@RobustDefZero\addtocounterref{%
253 \rc@set\addtocounter\ltx@car
254 }

```

```

\setcounterpageref
255 \rc@RobustDefZero\setcounterpageref{%
256 \rc@set\setcounter\ltx@carsecond
257 }

```

```

\addtocounterpageref
258 \rc@RobustDefZero\addtocounterpageref{%
259 \rc@set\addtocounter\ltx@carsecond
260 }

```

2.6 Extracting references

```

\getrefnumber
261 \rc@newcommand*{\getrefnumber}[1]{%
262 \romannumeral
263 \ltx@ifundefined{r@#1}{%
264 \expandafter\ltx@zero
265 \rc@default
266 }{%
267 \expandafter\expandafter\expandafter\rc@extract@
268 \expandafter\expandafter\expandafter!%
269 \csname r@#1\expandafter\endcsname

```



```

270 \expandafter{\rc@default}\@nil
271 }%
272 }

\getpagerefnnumber
273 \rc@newcommand*{\getpagerefnnumber}[1]{%
274 \romannumeral
275 \ltx@ifundefined{r@#1}{%
276 \expandafter\ltx@zero
277 \rc@default
278 }{%
279 \expandafter\expandafter\expandafter\rc@extract@page
280 \expandafter\expandafter\expandafter!%
281 \csname r@#1\expandafter\expandafter\expandafter\endcsname
282 \expandafter\expandafter\expandafter{%
283 \expandafter\rc@default
284 \expandafter}\expandafter{\rc@default}\@nil
285 }%
286 }

\getrefbykeydefault
287 \rc@newcommand*{\getrefbykeydefault}[2]{%
288 \romannumeral
289 \expandafter\rc@getrefbykeydefault
290 \csname r@#1\expandafter\endcsname
291 \csname rc@extract@#2\endcsname
292 }

\rc@getrefbykeydefault      #1: \r@<...>
                           #2: \rc@extract@<...>
                           #3: default
293 \long\def\rc@getrefbykeydefault#1#2#3{%
294 \ifx#1\relax
295 % reference is undefined
296 \ltx@ReturnAfterElseFi{%
297 \ltx@zero
298 #3%
299 }%
300 \else
301 \ltx@ReturnAfterFi{%
302 \ifx#2\relax
303 % extract method is missing
304 \ltx@ReturnAfterElseFi{%
305 \ltx@zero
306 #3%
307 }%
308 \else
309 \ltx@ReturnAfterFi{%
310 \expandafter
311 \rc@generic#1{#3}{#3}{#3}{#3}\@nil#2{#3}%
312 }%
313 \fi
314 }%
315 \fi
316 }

\rc@generic                #1: first item in \r@<...>
                           #2: remaining items in \r@<...>
                           #3: \rc@extract@<...>
                           #4: default
317 \long\def\rc@generic#1#2\@nil#3#4{%

```

```

318 #3{#1\TR@TitleReference\@empty{#4}\@nil}{#1}#2\@nil
319 }

```

```
\rc@extract@
```

```

320 \long\def\rc@extract@#1#2#3\@nil{%
321 \ltx@zero
322 #2%
323 }

```

```
\rc@extract@page
```

```

324 \long\def\rc@extract@page#1#2#3#4\@nil{%
325 \ltx@zero
326 #3%
327 }

```

```
\rc@extract@name
```

```

328 \long\def\rc@extract@name#1#2#3#4#5\@nil{%
329 \ltx@zero
330 #4%
331 }

```

```
\rc@extract@anchor
```

```

332 \long\def\rc@extract@anchor#1#2#3#4#5#6\@nil{%
333 \ltx@zero
334 #5%
335 }

```

```
\rc@extract@url
```

```

336 \long\def\rc@extract@url#1#2#3#4#5#6#7\@nil{%
337 \ltx@zero
338 #6%
339 }

```

```
\rc@extract@title
```

```

340 \long\def\rc@extract@title#1#2\@nil{%
341 \rc@@extract@title#1%
342 }

```

```
\rc@@extract@title
```

```

343 \long\def\rc@@extract@title#1\TR@TitleReference#2#3#4\@nil{%
344 \ltx@zero
345 #3%
346 }

```

2.7 Macros for checking undefined references

```
\IfRefUndefinedExpandable
```

```

347 \rc@newcommand*{\IfRefUndefinedExpandable}[1]{%
348 \ltx@ifundefined{r@#1}\ltx@firstoftwo\ltx@secondoftwo
349 }

```

```
\IfRefUndefinedBabel
```

```

350 \rc@RobustDefOne{ }*\IfRefUndefinedBabel{%
351 \begingroup
352 \csname safe@actives@true\endcsname
353 \expandafter\expandafter\expandafter\endgroup
354 \expandafter\ifx\csname r@#1\endcsname\relax
355 \expandafter\ltx@firstoftwo
356 \else
357 \expandafter\ltx@secondoftwo
358 \fi
359 }

```

```
360 \rc@AtEnd%
361 \end{package}
```

3 Test

3.1 Catcode checks for loading

```
362 (*test1)
363 \catcode`\{=1 %
364 \catcode`\}=2 %
365 \catcode`\#=6 %
366 \catcode`\@=11 %
367 \expandafter\ifx\csname count@\endcsname\relax
368 \countdef\count@=255 %
369 \fi
370 \expandafter\ifx\csname @gobble\endcsname\relax
371 \long\def\@gobble#1{ }%
372 \fi
373 \expandafter\ifx\csname @firstofone\endcsname\relax
374 \long\def\@firstofone#1{#1}%
375 \fi
376 \expandafter\ifx\csname loop\endcsname\relax
377 \expandafter\@firstofone
378 \else
379 \expandafter\@gobble
380 \fi
381 {%
382 \def\loop#1\repeat{%
383 \def\body{#1}%
384 \iterate
385 }%
386 \def\iterate{%
387 \body
388 \let\next\iterate
389 \else
390 \let\next\relax
391 \fi
392 \next
393 }%
394 \let\repeat=\fi
395 }%
396 \def\RestoreCatcodes{
397 \count@=0 %
398 \loop
399 \edef\RestoreCatcodes{%
400 \RestoreCatcodes
401 \catcode\the\count@=\the\catcode\count@\relax
402 }%
403 \ifnum\count@<255 %
404 \advance\count@ 1 %
405 \repeat
406
407 \def\RangeCatcodeInvalid#1#2{%
408 \count@=#1\relax
409 \loop
410 \catcode\count@=15 %
411 \ifnum\count@<#2\relax
412 \advance\count@ 1 %
413 \repeat
414 }
415 \def\RangeCatcodeCheck#1#2#3{%
```

```

416 \count@=#1\relax
417 \loop
418 \ifnum#3=\catcode\count@
419 \else
420 \errmessage{%
421 Character \the\count@\space
422 with wrong catcode \the\catcode\count@\space
423 instead of \number#3%
424 }%
425 \fi
426 \ifnum\count@<#2\relax
427 \advance\count@ 1 %
428 \repeat
429 }
430 \def\space{ }
431 \expandafter\ifx\csname LoadCommand\endcsname\relax
432 \def\LoadCommand{\input refcount.sty\relax}%
433 \fi
434 \def\Test{%
435 \RangeCatcodeInvalid{0}{47}%
436 \RangeCatcodeInvalid{58}{64}%
437 \RangeCatcodeInvalid{91}{96}%
438 \RangeCatcodeInvalid{123}{255}%
439 \catcode`\@=12 %
440 \catcode`\=0 %
441 \catcode`\%=14 %
442 \LoadCommand
443 \RangeCatcodeCheck{0}{36}{15}%
444 \RangeCatcodeCheck{37}{37}{14}%
445 \RangeCatcodeCheck{38}{47}{15}%
446 \RangeCatcodeCheck{48}{57}{12}%
447 \RangeCatcodeCheck{58}{63}{15}%
448 \RangeCatcodeCheck{64}{64}{12}%
449 \RangeCatcodeCheck{65}{90}{11}%
450 \RangeCatcodeCheck{91}{91}{15}%
451 \RangeCatcodeCheck{92}{92}{0}%
452 \RangeCatcodeCheck{93}{96}{15}%
453 \RangeCatcodeCheck{97}{122}{11}%
454 \RangeCatcodeCheck{123}{255}{15}%
455 \RestoreCatcodes
456 }
457 \Test
458 \csname @@end\endcsname
459 \end
460 </test1>

```

3.2 Macro tests

```

461 /*test2)
462 \errorcontextlines=10000 %
463 \showboxbreadth=10000 %
464 \showboxdepth=10000 %
465 \begingroup\expandafter\expandafter\expandafter\endgroup
466 \expandafter\ifx\csname RequirePackage\endcsname\relax
467 \input refcount.sty\relax
468 \else
469 \RequirePackage{refcount}[2016/05/16]%
470 \fi
471 \catcode`\@=11 %
472 \begingroup\expandafter\expandafter\expandafter\endgroup
473 \expandafter\ifx\csname @onelevel@sanitize\endcsname\relax
474 \begingroup\expandafter\expandafter\expandafter\endgroup
475 \expandafter\ifx\csname detokenize\endcsname\relax

```

```

476 \def\strip@prefix#1->{}%
477 \def\@onelevel@sanitize#1{%
478   \edef#1{%
479     \expandafter\strip@prefix\meaning#1%
480   }%
481 }%
482 \else
483 \def\@onelevel@sanitize#1{%
484   \edef#1{%
485     \detokenize\expandafter{#1}%
486   }%
487 }%
488 \fi
489 \fi
490 \def\msg#{\immediate\write16}
491 \def\empty{}
492 \def\space{ }

493 \def\r@foo{{\empty 1}{\empty 2}}
494 \long\def\test#1#2{%
495   \begingroup
496   \setbox0=\hbox{%
497     \def\TestTask{#1}%
498     \@onelevel@sanitize\TestTask
499     \msg{* \TestTask}%
500     \expandafter\expandafter\expandafter\def
501     \expandafter\expandafter\expandafter\TestResult
502     \expandafter\expandafter\expandafter{%
503       #1%
504     }%
505     \def\TestExpected{#2}%
506     \ifx\TestResult\TestExpected
507       \msg{ \space ok.}%
508     \else
509       \@onelevel@sanitize\TestResult
510       \@onelevel@sanitize\TestExpected
511       \msg{ \space Result: \space\space[TestResult]}%
512       \msg{ \space Expected: [\TestExpected]}%
513       \errmessage{Test failed!}%
514     \fi
515   }%
516   \ifdim\wd0=0pt %
517   \else
518     \showbox0 %
519   \fi
520 \endgroup
521 }
522 \test{\getrefnumber{foo}}{\empty 1}
523 \test{\getpagerefnumber{foo}}{\empty 2}
524 \test{\getrefbykeydefault{foo}{}\{\empty default}}{\empty 1}
525 \test{\getrefbykeydefault{foo}{page}{\empty default}}{\empty 2}
526 \test{\getrefbykeydefault{foo}{name}{\empty default}}{\empty default}
527 \test{\getrefbykeydefault{foo}{anchor}{\empty default}}{\empty default}
528 \test{\getrefbykeydefault{foo}{url}{\empty default}}{\empty default}
529 \test{\getrefbykeydefault{foo}{title}{\empty default}}{\empty default}
530 \msg{}
531 \def\r@foo{{}{}}{}{}{}{}{}{}{}{}
532 \def\Test#1#2\\{%
533   \test{#1{foo}#2}{}%
534 }
535 \def\TestGroup{%
536   \Test\getrefnumber\\%
537   \Test\getpagerefnumber\\%

```



```

600     \errmessage{Test failed!}%
601   \endgroup
602 \fi
603 \ifx\TestResultB\TestExpected
604   \msg{ \space ok.}%
605 \else
606   \begingroup
607     \@onelevel@sanitize\TestResultB
608     \@onelevel@sanitize\TestExpected
609     \msg{ \space Result: \space\space[\TestResultB]}%
610     \msg{ \space Expected: [\TestExpected]}%
611     \errmessage{Test failed!}%
612   \endgroup
613 \fi
614 \endgroup
615 }
616 \begingroup
617 \def\r@foo{{}{}}%
618 \let\r@bar\@undefined
619 \let\r@xyz\relax
620 \Test{foo}{true}{false}{false}%
621 \Test{bar}{true}{false}{true}%
622 \Test{xyz}{true}{false}{true}%
623 \endgroup

624 \csname @@end\endcsname\end
625 /test2)

```

3.3 Test with package titleref

```

626 (*test3)
627 \NeedsTeXFormat{LaTeX2e}
628 \documentclass{article}
629 \usepackage{refcount}[2016/05/16]
630 (test4)\usepackage{nameref}
631 \usepackage{titleref}
632 \begin{document}
633 \section{Hello World}
634 \label{sec:hello}
635 \section{\hbox{xy}}
636 \label{sec:foo}
637 %
638 \makeatletter
639 \@ifundefined{r@sec:hello}{%
640   \typeout{==> Compile twice!}%
641 }{%
642   \def\test#1#2{%
643     \begingroup
644       \def\TestTask{#1}%
645       \@onelevel@sanitize\TestTask
646       \typeout{* \TestTask}%
647       \expandafter\expandafter\expandafter\def
648         \expandafter\expandafter\expandafter\TestResult
649         \expandafter\expandafter\expandafter{%
650           #1%
651         }%
652       \def\TestExpected{#2}%
653       \ifx\TestResult\TestExpected
654         \typeout{ \space ok.}%
655       \else
656         \@onelevel@sanitize\TestResult
657         \@onelevel@sanitize\TestExpected
658         \typeout{ \space Result: \space\space[\TestResult]}%
659         \typeout{ \space Expected: [\TestExpected]}%

```

```

660     \errmessage{Test failed!}%
661     \fi
662   \endgroup
663 }%
664 \test{\getrefbykeydefault{sec:hello}{title}{}}{Hello World}%
665 \test{\getrefbykeydefault{sec:foo}{title}{}}{\hbox{xy}}%
666 \begingroup
667   \def\hbox#1{[#1]}% hash-ok
668   \test{\getrefbykeydefault{sec:foo}{title}{}}{\hbox{xy}}%
669 \endgroup
670 }
671 \makeatother

672 \end{document}
673 </test3>

674 (*test5)
675 \NeedsTeXFormat{LaTeX2e}
676 \documentclass{book}
677 \usepackage{refcount}[2016/05/16]
678 \usepackage{zref-runs}
679 \newcounter{test}
680 \begin{document}
681 \ifnum\zruns>1 %
682   \makeatletter
683   \def\Test#1#2#3{%
684     \begingroup
685       \setcounter{test}{10}%
686       \sbox0{%
687         #1{test}{#2}%
688         \ifnum#3=\value{test}%
689           \else
690             \PackageError{test}{\string#1{#2} <> #3 (\the\value{test})}%
691           \fi
692       }%
693       \ifdim\wd0=0pt %
694         \else
695           \PackageError{test}{Non-empty box}\@ehc
696         \fi
697     \endgroup
698   }%
699   \makeatother
700   \Test\setcounterpageref{ch:two}{1}%
701   \Test\setcounterpageref{ch:three}{3}%
702   \Test\setcounterpageref{ch:four}{5}%
703   \Test\setcounterpageref{ch:five}{7}%
704   \Test\setcounterpageref{ch:six}{9}%
705   \Test\setcounterpageref{ch:seven}{13}%
706   \Test\addtocounterpageref{ch:two}{11}%
707   \Test\addtocounterpageref{ch:three}{13}%
708   \Test\addtocounterpageref{ch:four}{15}%
709   \Test\addtocounterpageref{ch:five}{17}%
710   \Test\addtocounterpageref{ch:six}{19}%
711   \Test\addtocounterpageref{ch:seven}{23}%
712   \Test\setcounterref{ch:two}{1}%
713   \Test\setcounterref{ch:three}{2}%
714   \Test\setcounterref{ch:four}{11}%
715   \Test\addtocounterref{ch:two}{11}%
716   \Test\addtocounterref{ch:three}{12}%
717   \Test\addtocounterref{ch:four}{21}%
718 \fi
719 \frontmatter
720 \chapter{Chapter one}\label{ch:one}
721 \cleardoublepage

```



```

722 \mainmatter
723 \chapter{Chapter two}\label{ch:two}
724 \cleardoublepage
725 \chapter{Chapter three}\label{ch:three}
726 \cleardoublepage
727 \setcounter{chapter}{10}
728 \chapter{Chapter four}\label{ch:four}
729 \cleardoublepage
730 \appendix
731 \chapter{Chapter five}\label{ch:five}
732 \cleardoublepage
733 \chapter{Chapter six}\label{ch:six}
734 \cleardoublepage
735 \null
736 \cleardoublepage
737 \chapter{Chapter seven}\label{ch:seven}
738 \end{document}
739 </test5>

```

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
```

¹<http://ctan.org/pkg/refcount>

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
test/refcount-test5.tex → doc/latex/oberdiek/test/refcount-test5.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`.

4.4 Refresh file name databases

If your `TEX` distribution (`teTEX`, `mikTEX`, ...) relies on file name databases, you must refresh these. For example, `teTEX` users run `texhash` or `mktextlsr`.

4.5 Some details for the interested

Unpacking with L^AT_EX. The `.dtx` chooses its action depending on the format:

plain T_EX: Run `docstrip` and extract the files.

L^AT_EX: Generate the documentation.

If you insist on using L^AT_EX for `docstrip` (really, `docstrip` does not need L^AT_EX), 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 `pdfLATEX`:

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

5 Catalogue

The following XML file can be used as source for the [T_EX Catalogue](#). The elements `caption` and `description` are imported from the original XML file from the Catalogue. The name of the XML file in the Catalogue is `refcount.xml`.

```
740 (*catalogue)
741 <?xml version='1.0' encoding='us-ascii'?>
742 <!DOCTYPE entry SYSTEM 'catalogue.dtd'>
743 <entry datestamp='$Date$' modifier='$Author$' id='refcount'>
744 <name>refcount</name>
```

```

745 <caption>Counter operations with label references.</caption>
746 <authorref id='auth:oberdiek' />
747 <copyright owner='Heiko Oberdiek' year='1998,2000,2006,2008,2010,2011' />
748 <license type='lppl1.3' />
749 <version number='3.5' />
750 <description>
751   Provides commands <tt>\setcounterref</tt> and
752   <tt>\addtocounterref</tt> which use the section (or whatever)
753   number from the reference as the value to put into the counter, as
754   in:
755
756   <pre>
757   ...\label{sec:foo}
758   ...
759   \setcounterref{foonum}{sec:foo}
760   </pre>
761   Commands <tt>\setcounterpageref</tt> and
762   <tt>\addtocounterpageref</tt> do the corresponding thing with the
763   page reference of the label.
764   <p/>
765   No <tt>.ins</tt> file is distributed; process the
766   <tt>.dtx</tt> with plain TeX to create one.
767   <p/>
768   The package is part of the <xref refid='oberdiek'>oberdiek</xref>
769   bundle.
770 </description>
771 <documentation details='Package documentation'
772   href='ctan:/macros/latex/contrib/oberdiek/refcount.pdf' />
773 <ctan file='true' path='/macros/latex/contrib/oberdiek/refcount.dtx' />
774 <miktex location='oberdiek' />
775 <texlive location='oberdiek' />
776 <install path='/macros/latex/contrib/oberdiek/oberdiek.tds.zip' />
777 </entry>
778 </catalogue>

```

6 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.

[2011/06/22 v3.3]

- Bug fix: `\rc@refused` is undefined for `\setcounterpageref` and similar macros. (Bug found by Marc van Dongen.)

[2011/10/16 v3.4]

- Bug fix: `\setcounterpageref` and `\addtocounterpageref` fixed. (Bug found by Staz.)
- Macros `\(setaddto)counter(page)ref` are made robust.

[2016/05/16 v3.5]

- Documentation updates.

7 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>	365	<code>\}</code>	364
<code>\%</code>	441	A	
<code>\@</code>	366, 439, 471	<code>\addtocounter</code>	253, 259
<code>\@PackageError</code>	127	<code>\addtocounterpageref</code>	<u>258</u> ,
<code>\@PackageWarning</code>	222	706, 707, 708, 709, 710, 711, 762	
<code>\@ehc</code>	130, 695	<code>\addtocounterref</code> <u>252</u> , 715, 716, 717, 752	
<code>\@empty</code>	318	<code>\advance</code>	404, 412, 427
<code>\@firstofone</code>	374, 377	<code>\aftergroup</code>	29
<code>\@gobble</code>	371, 379	<code>\appendix</code>	730
<code>\@ifdefinable</code>	136	B	
<code>\@ifundefined</code>	639	<code>\begin</code>	632, 680
<code>\@latex@warning</code>	228	<code>\body</code>	383, 387
<code>\@nil</code> .. 245, 270, 284, 311, 317, 318,		C	
320, 324, 328, 332, 336, 340, 343		<code>\catcode</code> .. 2, 3, 5, 6, 7, 8, 9, 10, 11,	
<code>\@onelevel@sanitize</code>		12, 13, 33, 34, 36, 37, 38, 39, 40,	
477, 483, 498, 509, 510, 583,		41, 42, 43, 44, 45, 46, 47, 48, 49,	
596, 597, 607, 608, 645, 656, 657		69, 70, 72, 73, 74, 78, 79, 80, 81,	
<code>\@undefined</code>	58, 573, 618	82, 83, 84, 87, 88, 90, 91, 92, 93,	
<code>\@</code>	440, 532, 536, 537, 538,	97, 99, 363, 364, 365, 366, 401,	
539, 540, 541, 542, 545, 548, 555		410, 418, 422, 439, 440, 441, 471	
<code>\{</code>	363		

<code>\chapter</code>	720, 723, 725, 728, 731, 733, 737
<code>\cleardoublepage</code>	721, 724, 726, 729, 732, 734, 736
<code>\count</code>	220
<code>\count@</code>	368, 397, 401, 403, 404, 408, 410, 411, 412, 416, 418, 421, 422, 426, 427
<code>\countdef</code>	368
<code>\csname</code>	14, 21, 50, 66, 76, 112, 198, 208, 210, 211, 235, 237, 269, 281, 290, 291, 352, 354, 367, 370, 373, 376, 431, 458, 466, 473, 475, 624
D	
<code>\DeclareRobustCommand</code>	153, 158
<code>\detokenize</code>	485
<code>\documentclass</code>	628, 676
E	
<code>\empty</code>	17, 18, 491, 493, 522, 523, 524, 525, 526, 527, 528, 529
<code>\end</code>	459, 624, 672, 738
<code>\endcsname</code>	14, 21, 50, 66, 76, 112, 198, 208, 210, 211, 235, 237, 269, 281, 290, 291, 352, 354, 367, 370, 373, 376, 431, 458, 466, 473, 475, 624
<code>\endinput</code>	29, 110
<code>\newlinechar</code>	4, 35, 71, 77, 89
<code>\errmessage</code>	420, 513, 600, 611, 660
<code>\errorcontextlines</code>	462
F	
<code>\frontmatter</code>	719
G	
<code>\G@refundefinedtrue</code>	200
<code>\getpagerefnumber</code>	273, 523, 537, 564
<code>\getrefbykeydefault</code>	287, 524, 525, 526, 527, 528, 529, 538, 539, 540, 541, 542, 545, 552, 664, 665, 668
<code>\getrefnumber</code>	261, 522, 536, 563
H	
<code>\hbox</code>	496, 635, 665, 667, 668
I	
<code>\ifcase</code>	177
<code>\ifdim</code>	516, 693
<code>\ifnum</code>	403, 411, 418, 426, 681, 688
<code>\IfRefUndefinedBabel</code>	350, 586
<code>\IfRefUndefinedExpandable</code>	3, 347, 585
<code>\ifx</code>	15, 18, 21, 50, 58, 61, 112, 121, 124, 241, 294, 302, 354, 367, 370, 373, 376, 431, 466, 473, 475, 506, 592, 603, 653
<code>\immediate</code>	23, 52, 490
<code>\input</code>	113, 114, 432, 467
<code>\iterate</code>	384, 386, 388
L	
<code>\label</code>	634, 636, 720, 723, 725, 728, 731, 733, 737, 757
<code>\LoadCommand</code>	432, 442
<code>\loop</code>	382, 398, 409, 417
<code>\ltx@car</code>	250, 253
<code>\ltx@carsecond</code>	256, 259
<code>\ltx@firstofone</code>	122, 125
<code>\ltx@firstoftwo</code>	348, 355
<code>\ltx@gobble</code>	131
<code>\ltx@ifundefined</code>	119, 138, 139, 174, 195, 199, 209, 217
<code>\ltx@ifundefined</code>	219, 263, 275, 348
<code>\ltx@ReturnAfterElseFi</code>	296, 304
<code>\ltx@ReturnAfterFi</code>	301, 309
<code>\ltx@secondoftwo</code>	348, 357
<code>\ltx@undefined</code>	121
<code>\ltx@zero</code>	264, 276, 297, 305, 321, 325, 329, 333, 337, 344
M	
<code>\mainmatter</code>	722
<code>\makeatletter</code>	638, 682
<code>\makeatother</code>	671, 699
<code>\meaning</code>	479
<code>\MessageBreak</code>	128
<code>\msg</code>	490, 499, 507, 511, 512, 530, 546, 553, 559, 579, 584, 593, 598, 599, 604, 609, 610
N	
<code>\NeedsTeXFormat</code>	627, 675
<code>\newcommand</code>	189
<code>\newcounter</code>	679
<code>\next</code>	388, 390, 392
<code>\null</code>	735
<code>\number</code>	220, 423
P	
<code>\PackageError</code>	690, 695
<code>\PackageInfo</code>	26
<code>\par</code>	547, 549, 569
<code>\protect</code>	200
<code>\protected</code>	165, 170
<code>\ProvidesPackage</code>	19, 67
R	
<code>\r@bar</code>	618
<code>\r@foo</code>	493, 531, 547, 554, 573, 575, 577, 617
<code>\r@xyz</code>	619
<code>\RangeCatcodeCheck</code>	415, 443, 444, 445, 446, 447, 448, 449, 450, 451, 452, 453, 454
<code>\RangeCatcodeInvalid</code>	407, 435, 436, 437, 438
<code>\rc@@extract@title</code>	341, 343
<code>\rc@@set</code>	237, 240
<code>\rc@AtEnd</code>	95, 96, 110, 360
<code>\rc@default</code>	192, 242, 245, 265, 270, 277, 283, 284
<code>\rc@extract@</code>	267, 320
<code>\rc@extract@anchor</code>	332
<code>\rc@extract@name</code>	328
<code>\rc@extract@page</code>	279, 324
<code>\rc@extract@title</code>	340

