The refcount package

Heiko Oberdiek*
<heiko.oberdiek at googlemail.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 \setrefcountdefault . . . 7
- 2.4 \refused . . . 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 titleref . . . 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 LaTeX'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
⟨*empty*⟩	same as `\ref`	LaTeX
page	same as `\pageref`	LaTeX
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 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 {⟨*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 ⟨*then*⟩ is executed, otherwise argument ⟨*else*⟩ is called. Macro \IfRefUndefinedExpandable is expandable, but ⟨*refname*⟩ 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 % .
```

```
\catcode58=12 % :
\catcode64=11 % @
\catcode123=1 % {
\catcode125=2 % }
\expandafter\let\expandafter\x\csname ver@refcount.sty\endcsname
\ifx\x\relax % plain-TeX, first loading
\else
  \def\empty{}%
  \ifx\x\empty % LaTeX, first loading,
    % variable is initialized, but \ProvidesPackage not yet seen
  \else
    \expandafter\ifx\csname PackageInfo\endcsname\relax
      \def\x#1#2{%
        \immediate\write-1{Package #1 Info: #2.}%
      }%
    \else
      \def\x#1#2{\PackageInfo{#1}{#2, stopped}}%
    \fi
    \x{refcount}{The package is already loaded}%
    \aftergroup\endinput
  \fi
\fi
\endgroup%
```

Package identification:

```
\begingroup\catcode61\catcode48\catcode32=10\relax%
  \catcode13=5 % ^^M
  \endlinechar=13 %
  \catcode35=6 % #
  \catcode39=12 % '
  \catcode40=12 % (
  \catcode41=12 % )
  \catcode44=12 % ,
  \catcode45=12 % -
  \catcode46=12 % .
  \catcode47=12 % /
  \catcode58=12 % :
  \catcode64=11 % @
  \catcode91=12 % [
  \catcode93=12 % ]
  \catcode123=1 % {
  \catcode125=2 % }
  \expandafter\ifx\csname ProvidesPackage\endcsname\relax
    \def\x#1#2#3[#4]{\endgroup
      \immediate\write-1{Package: #3 #4}%
      \xdef#1{#4}%
    }%
  \else
    \def\x#1#2[#3]{\endgroup
      #2[{#3}]%
      \ifx#1\@undefined
        \xdef#1{#3}%
      \fi
      \ifx#1\relax
        \xdef#1{#3}%
      \fi
    }%
  \fi
\expandafter\x\csname ver@refcount.sty\endcsname
\ProvidesPackage{refcount}%
  [2016/05/16 v3.5 Data extraction from label references (HO)]%
\begingroup\catcode61\catcode48\catcode32=10\relax%
  \catcode13=5 % ^^M
```

```
\endlinechar=13 %
\catcode123=1 % {
\catcode125=2 % }
\catcode64=11 % @
\def\x{\endgroup
  \expandafter\edef\csname rc@AtEnd\endcsname{%
    \endlinechar=\the\endlinechar\relax
    \catcode13=\the\catcode13\relax
    \catcode32=\the\catcode32\relax
    \catcode35=\the\catcode35\relax
    \catcode61=\the\catcode61\relax
    \catcode64=\the\catcode64\relax
    \catcode123=\the\catcode123\relax
    \catcode125=\the\catcode125\relax
  }%
 }%
\x\catcode61\catcode48\catcode32=10\relax%
\catcode13=5 % ^^M
\endlinechar=13 %
\catcode35=6 % #
\catcode64=11 % @
\catcode123=1 % {
\catcode125=2 % }
\def\TMP@EnsureCode#1#2{%
  \edef\rc@AtEnd{%
    \rc@AtEnd
    \catcode#1=\the\catcode#1\relax
  }%
  \catcode#1=#2\relax
}
\TMP@EnsureCode{33}{12}% !
\TMP@EnsureCode{39}{12}% '
\TMP@EnsureCode{42}{12}% *
\TMP@EnsureCode{45}{12}% -
\TMP@EnsureCode{46}{12}% .
\TMP@EnsureCode{47}{12}% /
\TMP@EnsureCode{91}{12}% [
\TMP@EnsureCode{93}{12}% ]
\TMP@EnsureCode{96}{12}% `
\edef\rc@AtEnd{\rc@AtEnd\noexpand\endinput}
```

2.1 Loading packages

```
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname RequirePackage\endcsname\relax
  \input ltxcmds.sty\relax
  \input infwarerr.sty\relax
\else
  \RequirePackage{ltxcmds}[2011/11/09]%
  \RequirePackage{infwarerr}[2010/04/08]%
\fi
```

2.2 Defining commands

\rc@IfDefinable

```
\ltx@IfUndefined{@ifdefinable}{%
  \def\rc@IfDefinable#1{%
    \ifx#1\ltx@undefined
      \expandafter\ltx@firstofone
    \else
      \ifx#1\relax
        \expandafter\expandafter\expandafter\ltx@firstofone
      \else
        \@PackageError{refcount}{%
```

```
        Command \string#1 is already defined.\MessageBreak
        It will not redefined by this package%
      }\@ehc
      \expandafter\expandafter\expandafter\ltx@gobble
    \fi
  \fi
 }%
}{%
 \let\rc@IfDefinable\@ifdefinable
}
```

\rc@RobustDefOne
\rc@RobustDefZero

```
\ltx@IfUndefined{protected}{%
 \ltx@IfUndefined{DeclareRobustCommand}{%
  \def\rc@RobustDefOne#1#2#3#4{%
   \rc@IfDefinable#3{%
    #1\def#3##1{#4}%
   }%
  }%
  \def\rc@RobustDefZero#1#2{%
   \rc@IfDefinable#1{%
    \def#1{#2}%
   }%
  }%
 }{%
  \def\rc@RobustDefOne#1#2#3#4{%
   \rc@IfDefinable#3{%
    \DeclareRobustCommand#2#3[1]{#4}%
   }%
  }%
  \def\rc@RobustDefZero#1#2{%
   \rc@IfDefinable#1{%
    \DeclareRobustCommand#1{#2}%
   }%
  }%
 }%
}{%
 \def\rc@RobustDefOne#1#2#3#4{%
  \rc@IfDefinable#3{%
   \protected#1\def#3##1{#4}%
  }%
 }%
 \def\rc@RobustDefZero#1#2{%
  \rc@IfDefinable#1{%
   \protected\def#1{#2}%
  }%
 }%
}
```

\rc@newcommand

```
\ltx@IfUndefined{newcommand}{%
 \def\rc@newcommand*#1[#2]#3{% hash-ok
  \rc@IfDefinable#1{%
   \ifcase#2 %
    \def#1{#3}%
   \or
    \def#1##1{#3}%
   \or
    \def#1##1##2{#3}%
   \else
    \rc@InternalError
   \fi
  }%
```

```
  }%
}{%
  \let\rc@newcommand\newcommand
}
```

2.3 \setrefcountdefault

\setrefcountdefault

```
\rc@RobustDefOne\long{}\setrefcountdefault{%
  \def\rc@default{#1}%
}
\setrefcountdefault{0}
```

2.4 \refused

\refused

```
\ltx@IfUndefined{G@refundefinedtrue}{%
  \rc@RobustDefOne{}{*}\refused{%
    \begingroup
      \csname @safe@activestrue\endcsname
      \ltx@IfUndefined{r@#1}{%
        \protect\G@refundefinedtrue
        \rc@WarningUndefined{#1}%
      }{}%
    \endgroup
  }%
}{%
  \rc@RobustDefOne{}{*}\refused{%
    \begingroup
      \csname @safe@activestrue\endcsname
      \ltx@IfUndefined{r@#1}{%
        \csname protect\expandafter\endcsname
        \csname G@refundefinedtrue\endcsname
        \rc@WarningUndefined{#1}%
      }{}%
    \endgroup
  }%
}
```

\rc@WarningUndefined

```
\ltx@IfUndefined{@latex@warning}{%
  \def\rc@WarningUndefined#1{%
    \ltx@ifundefined{thepage}{%
      \def\thepage{\number\count0 }%
    }{}%
    \@PackageWarning{refcount}{%
      Reference `#1' on page \thepage\space undefined%
    }%
  }%
}{%
  \def\rc@WarningUndefined#1{%
    \@latex@warning{%
      Reference `#1' on page \thepage\space undefined%
    }%
  }%
}
```

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

```
\def\rc@set#1#2#3#4{%
  \begingroup
    \csname @safe@activestrue\endcsname
    \refused{#4}%
    \expandafter\rc@@set\csname r@#4\endcsname{#1}{#2}{#3}%
  \endgroup
}
```

\rc@@set

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

```
\def\rc@@set#1#2#3#4{%
  \ifx#1\relax
    #2{#4}{\rc@default}%
  \else
    #2{#4}{%
      \expandafter#3#1\rc@default\rc@default\@nil
    }%
  \fi
}
```

2.5.2 User commands

\setcounterref

```
\rc@RobustDefZero\setcounterref{%
  \rc@set\setcounter\ltx@car
}
```

\addtocounterref

```
\rc@RobustDefZero\addtocounterref{%
  \rc@set\addtocounter\ltx@car
}
```

\setcounterpageref

```
\rc@RobustDefZero\setcounterpageref{%
  \rc@set\setcounter\ltx@carsecond
}
```

\addtocounterpageref

```
\rc@RobustDefZero\addtocounterpageref{%
  \rc@set\addtocounter\ltx@carsecond
}
```

2.6 Extracting references

\getrefnumber

```
\rc@newcommand*{\getrefnumber}[1]{%
  \romannumeral
  \ltx@ifundefined{r@#1}{%
    \expandafter\ltx@zero
    \rc@default
  }{%
    \expandafter\expandafter\expandafter\rc@extract@
    \expandafter\expandafter\expandafter!%
    \csname r@#1\expandafter\endcsname
```

```
    \expandafter{\rc@default}\@nil
  }%
}
```

\getpagerefnumber

```
\rc@newcommand*{\getpagerefnumber}[1]{%
  \romannumeral
  \ltx@ifundefined{r@#1}{%
    \expandafter\ltx@zero
    \rc@default
  }{%
    \expandafter\expandafter\expandafter\rc@extract@page
    \expandafter\expandafter\expandafter!%
    \csname r@#1\expandafter\expandafter\expandafter\endcsname
    \expandafter\expandafter\expandafter{%
      \expandafter\rc@default
    \expandafter}\expandafter{\rc@default}\@nil
  }%
}
```

\getrefbykeydefault

```
\rc@newcommand*{\getrefbykeydefault}[2]{%
  \romannumeral
  \expandafter\rc@getrefbykeydefault
    \csname r@#1\expandafter\endcsname
    \csname rc@extract@#2\endcsname
}
```

\rc@getrefbykeydefault

#1: \r@<...>
#2: \rc@extract@<...>
#3: default

```
\long\def\rc@getrefbykeydefault#1#2#3{%
  \ifx#1\relax
    % reference is undefined
    \ltx@ReturnAfterElseFi{%
      \ltx@zero
      #3%
    }%
  \else
    \ltx@ReturnAfterFi{%
      \ifx#2\relax
        % extract method is missing
        \ltx@ReturnAfterElseFi{%
          \ltx@zero
          #3%
        }%
      \else
        \ltx@ReturnAfterFi{%
          \expandafter
          \rc@generic#1{#3}{#3}{#3}{#3}{#3}{#3}\@nil#2{#3}%
        }%
      \fi
    }%
  \fi
}
```

\rc@generic

#1: first item in \r@<...>
#2: remaining items in \r@<...>
#3: \rc@extract@<...>
#4: default

```
\long\def\rc@generic#1#2\@nil#3#4{%
```

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

\rc@extract@

```
\long\def\rc@extract@#1#2#3\@nil{%
  \ltx@zero
  #2%
}
```

\rc@extract@page

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

\rc@extract@name

```
\long\def\rc@extract@name#1#2#3#4#5\@nil{%
  \ltx@zero
  #4%
}
```

\rc@extract@anchor

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

\rc@extract@url

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

\rc@extract@title

```
\long\def\rc@extract@title#1#2\@nil{%
  \rc@@extract@title#1%
}
```

\rc@@extract@title

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

2.7 Macros for checking undefined references

\IfRefUndefinedExpandable

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

\IfRefUndefinedBabel

```
\rc@RobustDefOne{}*\IfRefUndefinedBabel{%
  \begingroup
    \csname safe@actives@true\endcsname
  \expandafter\expandafter\expandafter\endgroup
  \expandafter\ifx\csname r@#1\endcsname\relax
    \expandafter\ltx@firstoftwo
  \else
    \expandafter\ltx@secondoftwo
  \fi
}
```

```
\rc@AtEnd%
⟨/package⟩
```

3 Test

3.1 Catcode checks for loading

```
⟨*test1⟩
\catcode`\{=1 %
\catcode`\}=2 %
\catcode`\#=6 %
\catcode`\@=11 %
\expandafter\ifx\csname count@\endcsname\relax
  \countdef\count@=255 %
\fi
\expandafter\ifx\csname @gobble\endcsname\relax
  \long\def\@gobble#1{}%
\fi
\expandafter\ifx\csname @firstofone\endcsname\relax
  \long\def\@firstofone#1{#1}%
\fi
\expandafter\ifx\csname loop\endcsname\relax
  \expandafter\@firstofone
\else
  \expandafter\@gobble
\fi
{%
  \def\loop#1\repeat{%
    \def\body{#1}%
    \iterate
  }%
  \def\iterate{%
    \body
      \let\next\iterate
    \else
      \let\next\relax
    \fi
    \next
  }%
  \let\repeat=\fi
}%
\def\RestoreCatcodes{}
\count@=0 %
\loop
  \edef\RestoreCatcodes{%
    \RestoreCatcodes
    \catcode\the\count@=\the\catcode\count@\relax
  }%
\ifnum\count@<255 %
  \advance\count@ 1 %
\repeat

\def\RangeCatcodeInvalid#1#2{%
  \count@=#1\relax
  \loop
    \catcode\count@=15 %
  \ifnum\count@<#2\relax
    \advance\count@ 1 %
  \repeat
}
\def\RangeCatcodeCheck#1#2#3{%
```

```
  \count@=#1\relax
  \loop
    \ifnum#3=\catcode\count@
    \else
      \errmessage{%
        Character \the\count@\space
        with wrong catcode \the\catcode\count@\space
        instead of \number#3%
      }%
    \fi
  \ifnum\count@<#2\relax
    \advance\count@ 1 %
  \repeat
}
\def\space{ }
\expandafter\ifx\csname LoadCommand\endcsname\relax
  \def\LoadCommand{\input refcount.sty\relax}%
\fi
\def\Test{%
  \RangeCatcodeInvalid{0}{47}%
  \RangeCatcodeInvalid{58}{64}%
  \RangeCatcodeInvalid{91}{96}%
  \RangeCatcodeInvalid{123}{255}%
  \catcode`\@=12 %
  \catcode`\\=0 %
  \catcode`\%=14 %
  \LoadCommand
  \RangeCatcodeCheck{0}{36}{15}%
  \RangeCatcodeCheck{37}{37}{14}%
  \RangeCatcodeCheck{38}{47}{15}%
  \RangeCatcodeCheck{48}{57}{12}%
  \RangeCatcodeCheck{58}{63}{15}%
  \RangeCatcodeCheck{64}{64}{12}%
  \RangeCatcodeCheck{65}{90}{11}%
  \RangeCatcodeCheck{91}{91}{15}%
  \RangeCatcodeCheck{92}{92}{0}%
  \RangeCatcodeCheck{93}{96}{15}%
  \RangeCatcodeCheck{97}{122}{11}%
  \RangeCatcodeCheck{123}{255}{15}%
  \RestoreCatcodes
}
\Test
\csname @@end\endcsname
\end
```

⟨/test1⟩

3.2 Macro tests

```
⟨*test2⟩
\errorcontextlines=10000 %
\showboxbreadth=10000 %
\showboxdepth=10000 %
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname RequirePackage\endcsname\relax
  \input refcount.sty\relax
\else
  \RequirePackage{refcount}[2016/05/16]%
\fi
\catcode`\@=11 %
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname @onelevel@sanitize\endcsname\relax
  \begingroup\expandafter\expandafter\expandafter\endgroup
  \expandafter\ifx\csname detokenize\endcsname\relax
```

```
    \def\strip@prefix#1->{}%
    \def\@onelevel@sanitize#1{%
      \edef#1{%
        \expandafter\strip@prefix\meaning#1%
      }%
    }%
  \else
    \def\@onelevel@sanitize#1{%
      \edef#1{%
        \detokenize\expandafter{#1}%
      }%
    }%
  \fi
\fi
\def\msg#{\immediate\write16}
\def\empty{}
\def\space{ }
\def\r@foo{{\empty 1}{\empty 2}}
\long\def\test#1#2{%
  \begingroup
    \setbox0=\hbox{%
      \def\TestTask{#1}%
      \@onelevel@sanitize\TestTask
      \msg{* \TestTask}%
      \expandafter\expandafter\expandafter\def
      \expandafter\expandafter\expandafter\TestResult
      \expandafter\expandafter\expandafter{%
        #1%
      }%
      \def\TestExpected{#2}%
      \ifx\TestResult\TestExpected
        \msg{ \space ok.}%
      \else
        \@onelevel@sanitize\TestResult
        \@onelevel@sanitize\TestExpected
        \msg{ \space Result: \space\space[\TestResult]}%
        \msg{ \space Expected: [\TestExpected]}%
        \errmessage{Test failed!}%
      \fi
    }%
    \ifdim\wd0=0pt %
    \else
      \showbox0 %
    \fi
  \endgroup
}
\test{\getrefnumber{foo}}{\empty 1}
\test{\getpagerefnumber{foo}}{\empty 2}
\test{\getrefbykeydefault{foo}{}{\empty default}}{\empty 1}
\test{\getrefbykeydefault{foo}{page}{\empty default}}{\empty 2}
\test{\getrefbykeydefault{foo}{name}{\empty default}}{\empty default}
\test{\getrefbykeydefault{foo}{anchor}{\empty default}}{\empty default}
\test{\getrefbykeydefault{foo}{url}{\empty default}}{\empty default}
\test{\getrefbykeydefault{foo}{title}{\empty default}}{\empty default}
\msg{}
\def\r@foo{{}{}{}{}{}{}{}{}{}{}}
\def\Test#1#2\\{%
  \test{#1{foo}#2}{}%
}
\def\TestGroup{%
  \Test\getrefnumber\\%
  \Test\getpagerefnumber\\%
```

```
  \Test\getrefbykeydefault{}{}\\%
  \Test\getrefbykeydefault{page}{}\\%
  \Test\getrefbykeydefault{anchor}{}\\%
  \Test\getrefbykeydefault{name}{}\\%
  \Test\getrefbykeydefault{url}{}\\%
}
\TestGroup
\Test\getrefbykeydefault{title}{}\\%
\msg{}
\def\r@foo{\par\par\par\par\par\par\par\par}
\long\def\Test#1#2\\{%
  \test{#1{foo}#2}{\par}%
}
\TestGroup
\test{\getrefbykeydefault{title}{}{}}{}
\msg{}
\def\r@foo{{ }{ }{ }{ }{ }}
\def\Test#1#2\\{%
  \test{#1{foo}#2}{ }%
}
\TestGroup
\msg{}
\long\def\TestDefault#1{%
  \begingroup
    \setrefcountdefault{#1}%
    \test{\getrefnumber{foo}}{#1}%
    \test{\getpagerefnumber{foo}}{#1}%
  \endgroup
}
\def\TestDefaultX{%
  \TestDefault{}%
  \TestDefault{\par}%
  \TestDefault{ }%
  \TestDefault{\space}%
}
\let\r@foo\@undefined
\TestDefaultX
\let\r@foo\relax
\TestDefaultX
\def\r@foo{}
\TestDefaultX
\msg{}
\long\def\Test#1#2#3#4{%
  \begingroup
    \def\TestTask{#1}%
    \@onelevel@sanitize\TestTask
    \msg{* [\TestTask]}%
    \edef\TestResultA{\IfRefUndefinedExpandable{#1}{#2}{#3}}%
    \IfRefUndefinedBabel{#1}{%
      \def\TestResultB{#2}%
    }{%
      \def\TestResultB{#3}%
    }%
    \def\TestExpected{#4}%
    \ifx\TestResultA\TestExpected
      \msg{ \space ok.}%
    \else
      \begingroup
        \@onelevel@sanitize\TestResultA
        \@onelevel@sanitize\TestExpected
        \msg{ \space Result: \space\space[\TestResultA]}%
        \msg{ \space Expected: [\TestExpected]}%
```

```
      \errmessage{Test failed!}%
    \endgroup
  \fi
  \ifx\TestResultB\TestExpected
    \msg{ \space ok.}%
  \else
    \begingroup
      \@onelevel@sanitize\TestResultB
      \@onelevel@sanitize\TestExpected
      \msg{ \space Result: \space\space[\TestResultB]}%
      \msg{ \space Expected: [\TestExpected]}%
      \errmessage{Test failed!}%
    \endgroup
  \fi
 \endgroup
}
\begingroup
 \def\r@foo{{}{}}%
 \let\r@bar\@undefined
 \let\r@xyz\relax
 \Test{foo}{true}{false}{false}%
 \Test{bar}{true}{false}{true}%
 \Test{xyz}{true}{false}{true}%
\endgroup
\csname @@end\endcsname\end
⟨/test2⟩
```

3.3 Test with package **titleref**

```
⟨*test3⟩
\NeedsTeXFormat{LaTeX2e}
\documentclass{article}
\usepackage{refcount}[2016/05/16]
⟨test4⟩\usepackage{nameref}
\usepackage{titleref}
\begin{document}
\section{Hello World}
\label{sec:hello}
\section{\hbox{xy}}
\label{sec:foo}
%
\makeatletter
\@ifundefined{r@sec:hello}{%
 \typeout{==> Compile twice!}%
}{%
 \def\test#1#2{%
  \begingroup
   \def\TestTask{#1}%
   \@onelevel@sanitize\TestTask
   \typeout{* \TestTask}%
   \expandafter\expandafter\expandafter\def
   \expandafter\expandafter\expandafter\TestResult
   \expandafter\expandafter\expandafter{%
    #1%
   }%
   \def\TestExpected{#2}%
   \ifx\TestResult\TestExpected
    \typeout{ \space ok.}%
   \else
    \@onelevel@sanitize\TestResult
    \@onelevel@sanitize\TestExpected
    \typeout{ \space Result: \space\space[\TestResult]}%
    \typeout{ \space Expected: [\TestExpected]}%
```

```
      \errmessage{Test failed!}%
    \fi
  \endgroup
 }%
 \test{\getrefbykeydefault{sec:hello}{title}{}}{Hello World}%
 \test{\getrefbykeydefault{sec:foo}{title}{}}{\hbox{xy}}%
 \begingroup
  \def\hbox#1{[#1]}% hash-ok
  \test{\getrefbykeydefault{sec:foo}{title}{}}{\hbox{xy}}%
 \endgroup
}
\makeatother
\end{document}
⟨/test3⟩
⟨*test5⟩
\NeedsTeXFormat{LaTeX2e}
\documentclass{book}
\usepackage{refcount}[2016/05/16]
\usepackage{zref-runs}
\newcounter{test}
\begin{document}
\ifnum\zruns>1 %
 \makeatletter
 \def\Test#1#2#3{%
  \begingroup
   \setcounter{test}{10}%
   \sbox0{%
    #1{test}{#2}%
    \ifnum#3=\value{test}%
    \else
     \PackageError{test}{\string#1{#2} <> #3 (\the\value{test})}%
    \fi
   }%
   \ifdim\wd0=0pt %
   \else
    \PackageError{test}{Non-empty box}\@ehc
   \fi
  \endgroup
 }%
 \makeatother
 \Test\setcounterpageref{ch:two}{1}%
 \Test\setcounterpageref{ch:three}{3}%
 \Test\setcounterpageref{ch:four}{5}%
 \Test\setcounterpageref{ch:five}{7}%
 \Test\setcounterpageref{ch:six}{9}%
 \Test\setcounterpageref{ch:seven}{13}%
 \Test\addtocounterpageref{ch:two}{11}%
 \Test\addtocounterpageref{ch:three}{13}%
 \Test\addtocounterpageref{ch:four}{15}%
 \Test\addtocounterpageref{ch:five}{17}%
 \Test\addtocounterpageref{ch:six}{19}%
 \Test\addtocounterpageref{ch:seven}{23}%
 \Test\setcounterref{ch:two}{1}%
 \Test\setcounterref{ch:three}{2}%
 \Test\setcounterref{ch:four}{11}%
 \Test\addtocounterref{ch:two}{11}%
 \Test\addtocounterref{ch:three}{12}%
 \Test\addtocounterref{ch:four}{21}%
\fi
\frontmatter
\chapter{Chapter one}\label{ch:one}
\cleardoublepage
```

```
\mainmatter
\chapter{Chapter two}\label{ch:two}
\cleardoublepage
\chapter{Chapter three}\label{ch:three}
\cleardoublepage
\setcounter{chapter}{10}
\chapter{Chapter four}\label{ch:four}
\cleardoublepage
\appendix
\chapter{Chapter five}\label{ch:five}
\cleardoublepage
\chapter{Chapter six}\label{ch:six}
\cleardoublepage
\null
\cleardoublepage
\chapter{Chapter seven}\label{ch:seven}
\end{document}
⟨/test5⟩
```

4 Installation

4.1 Download

Package. This package is available on CTAN[1]:

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 TEX 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 TEX:

```
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 mktexlsr.

4.5 Some details for the interested

Unpacking with LATEX. The .dtx chooses its action depending on the format:

plain TEX: Run docstrip and extract the files.

LATEX: Generate the documentation.

If you insist on using LATEX for docstrip (really, docstrip does not need LATEX), 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 TEX 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>
```

```
<caption>Counter operations with label references.</caption>
<authorref id='auth:oberdiek'/>
<copyright owner='Heiko Oberdiek' year='1998,2000,2006,2008,2010,2011'/>
<license type='lppl1.3'/>
<version number='3.5'/>
<description>
  Provides commands <tt>\setcounterref</tt> and
  <tt>\addtocounterref</tt> which use the section (or whatever)
  number from the reference as the value to put into the counter, as
  in:

  <pre>
  ...\label{sec:foo}
  ...
  \setcounterref{foonum}{sec:foo}
  </pre>
  Commands <tt>\setcounterpageref</tt> and
  <tt>\addtocounterpageref</tt> do the corresponding thing with the
  page reference of the label.
  <p/>
  No <tt>.ins</tt> file is distributed; process the
  <tt>.dtx</tt> with plain TeX to create one.
  <p/>
  The package is part of the <xref refid='oberdiek'>oberdiek</xref>
  bundle.
</description>
<documentation details='Package documentation'
    href='ctan:/macros/latex/contrib/oberdiek/refcount.pdf'/>
<ctan file='true' path='/macros/latex/contrib/oberdiek/refcount.dtx'/>
<miktex location='oberdiek'/>
<texlive location='oberdiek'/>
<install path='/macros/latex/contrib/oberdiek/oberdiek.tds.zip'/>
</entry>
⟨/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

[2]Url: 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

\# 365
\% 441
\@ 366, 439, 471
\@PackageError 127
\@PackageWarning 222
\@ehc 130, 695
\@empty 318
\@firstofone 374, 377
\@gobble 371, 379
\@ifdefinable 136
\@ifundefined 639
\@latex@warning 228
\@nil . . 245, 270, 284, 311, 317, 318, 320, 324, 328, 332, 336, 340, 343
\@onelevel@sanitize 477, 483, 498, 509, 510, 583, 596, 597, 607, 608, 645, 656, 657
\@undefined 58, 573, 618
\\ 440, 532, 536, 537, 538, 539, 540, 541, 542, 545, 548, 555
\{ 363
\} 364

A

\addtocounter 253, 259
\addtocounterpageref 258, 706, 707, 708, 709, 710, 711, 762
\addtocounterref 252, 715, 716, 717, 752
\advance 404, 412, 427
\aftergroup 29
\appendix 730

B

\begin 632, 680
\body 383, 387

C

\catcode . . 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, 69, 70, 72, 73, 74, 78, 79, 80, 81, 82, 83, 84, 87, 88, 90, 91, 92, 93, 97, 99, 363, 364, 365, 366, 401, 410, 418, 422, 439, 440, 441, 471

\chapter 720, 723, 725, 728, 731, 733, 737
\cleardoublepage 721, 724, 726, 729, 732, 734, 736
\count 220
\count@ 368, 397, 401, 403, 404, 408, 410, 411, 412, 416, 418, 421, 422, 426, 427
\countdef 368
\csname 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

\DeclareRobustCommand 153, 158
\detokenize 485
\documentclass 628, 676

E

\empty 17, 18, 491, 493, 522, 523, 524, 525, 526, 527, 528, 529
\end 459, 624, 672, 738
\endcsname 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
\endinput 29, 110
\endlinechar 4, 35, 71, 77, 89
\errmessage 420, 513, 600, 611, 660
\errorcontextlines 462

F

\frontmatter 719

G

\G@refundefinedtrue 200
\getpagerefnumber . 273, 523, 537, 564
\getrefbykeydefault 287, 524, 525, 526, 527, 528, 529, 538, 539, 540, 541, 542, 545, 552, 664, 665, 668
\getrefnumber 261, 522, 536, 563

H

\hbox 496, 635, 665, 667, 668

I

\ifcase 177
\ifdim 516, 693
\ifnum 403, 411, 418, 426, 681, 688
\IfRefUndefinedBabel 350, 586
\IfRefUndefinedExpandable . 3, 347, 585
\ifx 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
\immediate 23, 52, 490
\input 113, 114, 432, 467
\iterate 384, 386, 388

L

\label 634, 636, 720, 723, 725, 728, 731, 733, 737, 757
\LoadCommand 432, 442
\loop 382, 398, 409, 417
\ltx@car 250, 253
\ltx@carsecond 256, 259
\ltx@firstofone 122, 125
\ltx@firstoftwo 348, 355
\ltx@gobble 131
\ltx@IfUndefined 119, 138, 139, 174, 195, 199, 209, 217
\ltx@ifundefined . . . 219, 263, 275, 348
\ltx@ReturnAfterElseFi 296, 304
\ltx@ReturnAfterFi 301, 309
\ltx@secondoftwo 348, 357
\ltx@undefined 121
\ltx@zero 264, 276, 297, 305, 321, 325, 329, 333, 337, 344

M

\mainmatter 722
\makeatletter 638, 682
\makeatother 671, 699
\meaning 479
\MessageBreak 128
\msg 490, 499, 507, 511, 512, 530, 546, 553, 559, 579, 584, 593, 598, 599, 604, 609, 610

N

\NeedsTeXFormat 627, 675
\newcommand 189
\newcounter 679
\next 388, 390, 392
\null 735
\number 220, 423

P

\PackageError 690, 695
\PackageInfo 26
\par 547, 549, 569
\protect 200
\protected 165, 170
\ProvidesPackage 19, 67

R

\r@bar 618
\r@foo 493, 531, 547, 554, 573, 575, 577, 617
\r@xyz 619
\RangeCatcodeCheck 415, 443, 444, 445, 446, 447, 448, 449, 450, 451, 452, 453, 454
\RangeCatcodeInvalid 407, 435, 436, 437, 438
\rc@@extract@title 341, 343
\rc@@set 237, 240
\rc@AtEnd 95, 96, 110, 360
\rc@default 192, 242, 245, 265, 270, 277, 283, 284
\rc@extract@ 267, 320
\rc@extract@anchor 332
\rc@extract@name 328
\rc@extract@page 279, 324
\rc@extract@title 340

\rc@extract@url 336
\rc@generic 311, 317
\rc@getrefbykeydefault 289, 293
\rc@IfDefinable 119, 141, 146, 152, 157, 164, 169, 176
\rc@InternalError 184
\rc@newcommand 174, 261, 273, 287, 347
\rc@RobustDefOne 138, 191, 196, 206, 350
\rc@RobustDefZero 138, 249, 252, 255, 258
\rc@set 233, 250, 253, 256, 259
\rc@WarningUndefined . . 201, 212, 217
\refused 195, 236
\repeat 382, 394, 405, 413, 428
\RequirePackage 116, 117, 469
\RestoreCatcodes . . 396, 399, 400, 455
\romannumeral 262, 274, 288

S

\sbox 686
\section 633, 635
\setbox 496
\setcounter 250, 256, 685, 727
\setcounterpageref 255, 700, 701, 702, 703, 704, 705, 761
\setcounterref 249, 712, 713, 714, 751, 759
\setrefcountdefault 191, 194, 562
\showbox 518
\showboxbreadth 463
\showboxdepth 464
\space . 223, 229, 421, 422, 430, 492, 507, 511, 512, 571, 593, 598, 599, 604, 609, 610, 654, 658, 659
\strip@prefix 476, 479

T

\Test 434, 457, 532, 536, 537, 538, 539, 540, 541, 542, 545, 548, 555, 580, 620, 621, 622, 683, 700, 701, 702, 703, 704, 705, 706, 707, 708, 709, 710, 711, 712, 713, 714, 715, 716, 717
\test . . 494, 522, 523, 524, 525, 526, 527, 528, 529, 533, 549, 552, 556, 563, 564, 642, 664, 665, 668
\TestDefault . . . 560, 568, 569, 570, 571
\TestDefaultX 567, 574, 576, 578
\TestExpected 505, 506, 510, 512, 591, 592, 597, 599, 603, 608, 610, 652, 653, 657, 659
\TestGroup 535, 544, 551, 558
\TestResult 501, 506, 509, 511, 648, 653, 656, 658
\TestResultA 585, 592, 596, 598
\TestResultB . . . 587, 589, 603, 607, 609
\TestTask 497, 498, 499, 582, 583, 584, 644, 645, 646
\the 77, 78, 79, 80, 81, 82, 83, 84, 97, 401, 421, 422, 690
\thepage 220, 223, 229
\TMP@EnsureCode . . . 94, 101, 102, 103, 104, 105, 106, 107, 108, 109
\TR@TitleReference 318, 343
\typeout 640, 646, 654, 658, 659

U

\usepackage 629, 630, 631, 677, 678

V

\value 688, 690

W

\wd 516, 693
\write 23, 52, 490

X

\x 14, 15, 18, 22, 26, 28, 51, 56, 66, 75, 87

Z

\zruns 681