The \texttt{refcount} package

Heiko Oberdiek∗

\texttt{<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. \texttt{ref} or \texttt{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 \texttt{hyperref}, \texttt{nameref}, \texttt{titleref}, and \texttt{babel} are supported.

Contents

1 Usage \hspace{1cm} 2
\hspace{0.5cm} 1.1 Setting counters \hspace{1cm} 2
\hspace{0.5cm} 1.2 Expandable commands \hspace{1cm} 2
\hspace{0.5cm} 1.3 Undefined references \hspace{1cm} 3
\hspace{0.5cm} \hspace{0.5cm} 1.3.1 Check for undefined references \hspace{1cm} 3
\hspace{0.5cm} 1.4 Notes \hspace{1cm} 3

2 Implementation \hspace{1cm} 3
\hspace{0.5cm} 2.1 Loading packages \hspace{1cm} 5
\hspace{0.5cm} 2.2 Defining commands \hspace{1cm} 5
\hspace{0.5cm} 2.3 \texttt{setrefcountdefault} \hspace{1cm} 7
\hspace{0.5cm} 2.4 \texttt{refused} \hspace{1cm} 7
\hspace{0.5cm} 2.5 Setting counters by reference data \hspace{1cm} 7
\hspace{0.5cm} \hspace{0.5cm} 2.5.1 Generic setting \hspace{1cm} 7
\hspace{0.5cm} \hspace{0.5cm} 2.5.2 User commands \hspace{1cm} 8
\hspace{0.5cm} 2.6 Extracting references \hspace{1cm} 8
\hspace{0.5cm} 2.7 Macros for checking undefined references \hspace{1cm} 10

3 Test \hspace{1cm} 11
\hspace{0.5cm} 3.1 Catcode checks for loading \hspace{1cm} 11
\hspace{0.5cm} 3.2 Macro tests \hspace{1cm} 12
\hspace{0.5cm} 3.3 Test with package \texttt{titleref} \hspace{1cm} 15

4 Installation \hspace{1cm} 17
\hspace{0.5cm} 4.1 Download \hspace{1cm} 17
\hspace{0.5cm} 4.2 Bundle installation \hspace{1cm} 17
\hspace{0.5cm} 4.3 Package installation \hspace{1cm} 17
\hspace{0.5cm} 4.4 Refresh file name databases \hspace{1cm} 18
\hspace{0.5cm} 4.5 Some details for the interested \hspace{1cm} 18

5 Catalogue \hspace{1cm} 18

∗Please report any issues at https://github.com/ho-tex/oberdiek/issues
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:

<table>
<thead>
<tr>
<th>Key</th>
<th>Description</th>
<th>Package</th>
</tr>
</thead>
<tbody>
<tr>
<td>⟨empty⟩</td>
<td>same as <code>\ref</code></td>
<td>\LaTeX</td>
</tr>
<tr>
<td>page</td>
<td>same as <code>\pageref</code></td>
<td>\LaTeX</td>
</tr>
<tr>
<td>title</td>
<td>section and caption titles</td>
<td>titleref</td>
</tr>
<tr>
<td>name</td>
<td>section and caption titles</td>
<td>nameref</td>
</tr>
<tr>
<td>anchor</td>
<td>anchor name</td>
<td>hyperref</td>
</tr>
<tr>
<td>url</td>
<td>url/file</td>
<td>hyperref/xr</td>
</tr>
</tbody>
</table>
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:

```latex
\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

```latex
\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.

```latex
\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

```
\begin{verbatim}
1 \langle*package⟩

Reload check, especially if the package is not used with \LaTeX{}.
2 \begingroup\catcode61=10\relax\% 3 \catcode13=5 \% "M
4 \catcode13\endlinechar=10 \%
5 \catcode35=6 \%
6 \catcode35\catcode32=10 \%relax%
7 \catcode39=12 \%
8 \catcode44=12 \%
9 \catcode46=12 \%
\end{verbatim}
```
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

\ifx@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}{%
\else
\fi
\endinput}
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{%
\protected\def#1{#2}%
}%
%}
def\rc@RobustDefOne#1#2#3#4{%
\rc@IfDefinable#3{%
\protected\def#1##1{#4}%
}%
%}
def\rc@RobustDefZero#1#2{%
\rc@IfDefinable#1{%
\protected\def#1{#2}%
}%
%}
def\rc@RobustDefOne#1#2#3#4{%
\rc@IfDefinable#3{%
\protected\def#1##1{#4}%
}%
%}
def\rc@RobustDefZero#1#2{%
\rc@IfDefinable#1{%
\protected\def#1{#2}%
}%
%}
def\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
%}
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@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%
}%
}
\endgroup
\rc@WarningUndefined

2.5 Setting counters by reference data

2.5.1 Generic setting

\rc@set  Generic command for \{set,addto\}counter{page, }ref:
2.5.2 User commands

\setcounter
\addtocounter
\setcounterpageref
\addtocounterpageref

2.6 Extracting references

\getrefnumber
2.7 Macros for checking undefined references

\IfRefUndefinedExpandable
\newcommand*{\IfRefUndefinedExpandable}[1]{\@ifundefined{r@#1}\@firstoftwo\@secondoftwo}

\IfRefUndefinedBabel
\RobustDefOne{}\IfRefUndefinedBabel{\begingroup\csname safe@actives@true\endcsname\expandafter\expandafter\expandafter\endgroup\expandafter\ifx\csname r@#1\endcsname\relax\@firstoftwo\else\@secondoftwo\fi}\else\expandafter\expandafter\expandafter\endgroup\expandafter\ifx\csname r@#1\endcsname\relax\@firstoftwo\else\@secondoftwo\fi\fi
3 Test

3.1 Catcode checks for loading

\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
\else
\expandafter\@gobble
\fi
\def\loop#1\repeat{%
\def\body{#1}%
\iterate
}
\def\iterate{%
\body
\let
ext\iterate
\else
\let
ext\relax
\fi
\next
}
\let
repeat=\fi
\def\RangeCatcodeInvalid#1#2{%
\count@=#1\relax
\loop
\catcode\count@=15 %
\ifnum\count@<#2\relax
\advance\count@ 1 %
\repeat
}
\def\RangeCatcodeCheck#1#2#3{%
\def\RangeCatcodeInvalid#1#2{%
\count@=#1\relax
\loop
\catcode\count@=1 %
\ifnum\count@<255 %
\advance\count@ 1 %
\repeat
}
\def\RangeCatcodeCheck#1#2#3{%
3.2 Macro tests
\def\strip@prefix#1->{% 
\def\@onelevel@sanitize#1{% 
  \edef#1{\expandafter\strip@prefix\meaning#1}% 
}\fi 
\else 
  \def\@onelevel@sanitize#1{% 
    \edef#1{\detokenize\expandafter{#1}}% 
  }% 
}\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: \space\space\TestExpected} \errmessage{Test failed!} \fi \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}{empty default}}{\empty 2}\test{\getrefbykeydefault{foo}{page}{empty default}}{\empty 1}\test{\getrefbykeydefault{foo}{name}{empty default}}{\empty 2}\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}
\texttt{\_Test\getrefbykeydefault{\{\}}{\}}
\%
\texttt{\Test\getrefbykeydefault{\{page\}}{\}}
\%
\texttt{\Test\getrefbykeydefault{\{anchor\}}{\}}
\%
\texttt{\Test\getrefbykeydefault{\{name\}}{\}}
\%
\texttt{\Test\getrefbykeydefault{\{url\}}{\}}
\%
\}
\texttt{\TestGroup}
\texttt{\Test\getrefbykeydefault{\{title\}}{\}}
\%
\texttt{\msg{}}
\texttt{\def\r@foo{\par\par\par\par\par\par\par\par}}
\%
\texttt{\long\def\Test#1#2{}}
\%
\texttt{\test{#1{foo}#2}{\par}}
\%
\texttt{\TestGroup}
\texttt{\test{\getrefbykeydefault{\{title\}}{\}}{\}}
\%
\texttt{\msg{}}
\texttt{\long\def\Test#1#2{}}
\%
\texttt{\test{#1{foo}#2}{}}
\%
\texttt{\TestGroup}
\texttt{\msg{}}
\texttt{\def\r@foo{{ }{ }{ }{ }{ }}}
\%
\texttt{\long\def\TestDefault#1{}}
\%
\begin{group}
\texttt{\setrefcountdefault{#1}}
\%
\texttt{\test{\getrefnumber{foo}}{#1}}
\%
\texttt{\test{\getpagerefnumber{foo}}{#1}}
\%
\end{group}
\%
\texttt{\TestDefaultX{}}
\%
\texttt{\let\r@foo@undefined}
\%
\texttt{\TestDefaultX{}}
\%
\texttt{\let\r@foo@relax}
\%
\texttt{\TestDefaultX{}}
\%
\texttt{\def\r@foo{}}
\%
\texttt{\TestDefaultX{}}
\%
\texttt{\msg{}}
\texttt{\long\def\Test#1#2#3#4{}}
\%
\begin{group}
\texttt{\def\TestTask{\#1}}
\%
\texttt{\@onelevel@sanitize\TestTask}
\%
\texttt{\msg{\star{\TestTask}}}
\%
\texttt{\edef\TestResultA{\IfRefUndefinedExpandable{\#1}{\#2}{\#3}{}}}\%
\texttt{\IfRefUndefinedBabel{\#1}{}}\%
\texttt{\def\TestResultB{\#2}{}}\%
\texttt{\def\TestExpected{\#3}}\%
\texttt{\ifx\TestResultA\TestExpected}
\%
\texttt{\msg{\space ok.}}\%
\texttt{\end{group}
\texttt{\elsee}
\begin{group}
\texttt{\@onelevel@sanitize\TestResultA}
\%
\texttt{\@onelevel@sanitize\TestExpected}
\%
\texttt{\msg{\space Result: \TestResultA}}\%
\texttt{\msg{\space Expected: \TestExpected}}\%
\end{group}
\texttt{\input{test2}}

\section{Hello World}
\label{sec:hello}

\section{\hbox{xy}}
\label{sec:foo}

\begin{verbatim}
\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!}%
\endgroup
\fi
\endgroup
}
\end{verbatim}

3.3 Test with package titleref

\NeedsTeXFormat{LaTeX2e}
\documentclass{article}
\usepackage{refcount}[2016/05/16]
\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]}%
\endgroup
\fi
\endgroup
}
\end{verbatim}

\end{document}
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{#1{test}{#2}%
687 \ifnum#3=\value{test}%
688 \else
689 \PackageError{test}{\string#1{#2} <> #3 (\the\value{test})}%
690 \fi
691 %
692 \ifdim\wd0=0pt %
693 \else
694 \PackageError{test}{Non-empty box}@ehc
695 \fi
696 \endgroup
697 }%
698 \makeatother
699 \Test\setcounterpageref{ch:two}{1}%
700 \Test\setcounterpageref{ch:three}{3}%
701 \Test\setcounterpageref{ch:four}{5}%
702 \Test\setcounterpageref{ch:five}{7}%
703 \Test\setcounterpageref{ch:six}{9}%
704 \Test\setcounterpageref{ch:seven}{13}%
705 \Test\addtocounterpageref{ch:two}{1}%
706 \Test\addtocounterpageref{ch:three}{1}%
707 \Test\addtocounterpageref{ch:four}{11}%
708 \Test\addtocounterpageref{ch:five}{17}%
709 \Test\addtocounterpageref{ch:six}{19}%
710 \Test\addtocounterpageref{ch:seven}{23}%
711 \Test\setcounterref{ch:two}{1}%
712 \Test\setcounterref{ch:three}{2}%
713 \Test\setcounterref{ch:four}{11}%
714 \Test\addtocounterref{ch:two}{11}%
715 \Test\addtocounterref{ch:three}{12}%
716 \Test\addtocounterref{ch:four}{21}%
717 \fi
718 \frontmatter
719 \chapter{Chapter one}\label{ch:one}
720 \cleardoublepage
4 Installation

4.1 Download

Package. This package is available on CTAN:\(^1\):


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 *L*AT*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 *L*AT*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 (\TeX\, \LaTeX\, \ldots) relies on file name databases, you must refresh these. For example, \TeX\ 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`.

```
<?xml version='1.0' encoding='us-ascii'?>
<!DOCTYPE entry SYSTEM 'catalogue.dtd'>
<entry datestamp='$Date$' modifier='$Author$' id='refcount'>
  <name>refcount</name>
</entry>
```

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>
```
Provides commands \texttt{\setcounterref} and \texttt{\addtocounterref} which use the section (or whatever) number from the reference as the value to put into the counter, as in:

\begin{verbatim}
...\label{sec:foo}
...
\setcounterref{foonum}{sec:foo}
\end{verbatim}

Commands \texttt{\setcounterpageref} and \texttt{\addtocounterpageref} do the corresponding thing with the page reference of the label.

No \texttt{.ins} file is distributed; process the \texttt{.dtx} with plain \TeX{} to create one.

The package is part of the \texttt{xref refid='oberdiek'}oberdiek\texttt{xref} bundle.

6 History

[1998/04/08 v1.0]

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

[2000/09/07 v2.0]

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

[2006/02/20 v3.0]

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

\textsuperscript{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, \getpageregnumber, \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 \setaddtocounter{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
\@sbc 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, 344
\@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