The uniquecounter package

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

2016/05/16 v1.3

Abstract

This package provides a kind of counter that provides unique number values. Several counters can be created by different names. The numeric values are not limited.

Contents

1 Documentation **2**
1.1 Example . . . 2

2 Implementation **2**
2.1 Reload check and package identification . . . 2
2.2 Catcodes . . . 4

3 Test **6**
3.1 Catcode checks for loading . . . 6
3.2 Macro tests . . . 8
3.2.1 Test with LaTeX . . . 8
3.2.2 Test with plain-TeX . . . 9

4 Installation **10**
4.1 Download . . . 10
4.2 Bundle installation . . . 11
4.3 Package installation . . . 11
4.4 Refresh file name databases . . . 11
4.5 Some details for the interested . . . 11

5 Catalogue **12**

6 History **12**
[2009/09/11 v1.0] . . . 12
[2009/12/18 v1.1] . . . 12
[2011/01/30 v1.2] . . . 12
[2016/05/16 v1.3] . . . 13

7 Index **13**

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

1 Documentation

`\UniqueCounterNew {⟨name⟩}`

Macro `\UniqueCounterNew` creates a new unique counter ⟨*name*⟩. An error is thrown, if the counter already exists.

`\UniqueCounterCall {⟨name⟩} {⟨code⟩}`

Macro `\UniqueCounterCall` calls the given ⟨*code*⟩ with a new value of counter ⟨*name*⟩ as argument.

`\UniqueCounterIncrement {⟨name⟩}`

Macro `\UniqueCounterIncrement` generates a new value for the counter ⟨*name*⟩ by incrementing by one (globally).

`\UniqueCounterGet {⟨name⟩}`

Expandable macro `\UniqueCounterGet` returns the current value of counter ⟨*name*⟩

1.1 Example

```
⟨*example⟩
\documentclass{minimal}
\usepackage{uniquecounter}
\UniqueCounterNew{anchor}
\makeatletter
\newcommand*{\DefNewAnchorName}[2]{%
  % #1 is unique counter value
  % #2 is name of anchor
  \@namedef{anchor@#2}{a#1}%
}
\newcommand*{\NewAnchorName}[1]{%
  \UniqueCounterCall{anchor}\DefNewAnchorName{#1}%
}
\newcommand*{\PrintAnchorName}[1]{%
  \@nameuse{anchor@#1}%
}
\begin{document}
  \NewAnchorName{Top}%
  \NewAnchorName{Left}%
  \noindent
  Top: \PrintAnchorName{Top}\\%
  Left: \PrintAnchorName{Left}%
\end{document}
⟨/example⟩
```

2 Implementation

```
⟨*package⟩
```

2.1 Reload check and package identification

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

```
\begingroup\catcode61\catcode48\catcode32=10\relax%
  \catcode13=5 % ^^M
  \endlinechar=13 %
```

```
\catcode35=6 % #
\catcode39=12 % '
\catcode44=12 % ,
\catcode45=12 % -
\catcode46=12 % .
\catcode58=12 % :
\catcode64=11 % @
\catcode123=1 % {
\catcode125=2 % }
\expandafter\let\expandafter\x\csname ver@uniquecounter.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{uniquecounter}{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@uniquecounter.sty\endcsname
\ProvidesPackage{uniquecounter}%
  [2016/05/16 v1.3 Provide unlimited unique counter (HO)]%
```

2.2 Catcodes

```
\begingroup\catcode61\catcode48\catcode32=10\relax%
  \catcode13=5 % ^^M
  \endlinechar=13 %
  \catcode123=1 % {
  \catcode125=2 % }
  \catcode64=11 % @
  \def\x{\endgroup
    \expandafter\edef\csname uqc@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\uqc@AtEnd{%
    \uqc@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{43}{12}% +
\TMP@EnsureCode{46}{12}% .
\TMP@EnsureCode{47}{12}% /
\TMP@EnsureCode{91}{12}% [
\TMP@EnsureCode{93}{12}% ]
\TMP@EnsureCode{96}{12}% `
\edef\uqc@AtEnd{\uqc@AtEnd\noexpand\endinput}
```

```
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname RequirePackage\endcsname\relax
  \def\TMP@RequirePackage#1[#2]{%
    \begingroup\expandafter\expandafter\expandafter\endgroup
    \expandafter\ifx\csname ver@#1.sty\endcsname\relax
      \input #1.sty\relax
    \fi
  }%
  \TMP@RequirePackage{bigintcalc}[2007/11/11]%
  \TMP@RequirePackage{infwarerr}[2007/09/09]%
\else
  \RequirePackage{bigintcalc}[2007/11/11]%
  \RequirePackage{infwarerr}[2007/09/09]%
\fi
```

\uqc@IncNum

```
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname numexpr\endcsname\relax
  \def\uqc@IncNum#1{%
    \begingroup
      \count@=\csname uqc@cnt@#1\endcsname\relax
      \advance\count@\@ne
      \expandafter\xdef\csname uqc@cnt@#1\endcsname{%
        \number\count@
      }%
      \ifnum\count@=2147483647 %
        \global\expandafter\let\csname uqc@inc@#1\endcsname
        \uqc@IncBig
      \fi
    \endgroup
  }%
\else
  \def\uqc@IncNum#1{%
    \expandafter\xdef\csname uqc@cnt@#1\endcsname{%
      \number\numexpr\csname uqc@cnt@#1\endcsname+1%
    }%
    \ifnum\csname uqc@cnt@#1\endcsname=2147483647 %
      \global\expandafter\let\csname uqc@inc@#1\endcsname
      \uqc@IncBig
    \fi
  }%
\fi
```

\uqc@IncBig

```
\def\uqc@IncBig#1{%
  \expandafter\xdef\csname uqc@cnt@#1\endcsname{%
    \expandafter\expandafter\expandafter
    \BigIntCalcInc\csname uqc@cnt@#1\endcsname!%
  }%
}
```

\uqc@Def

```
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname newcommand\endcsname\relax
  \def\uqc@Def#1{\def#1##1}%
\else
  \def\uqc@Def#1{\newcommand*{#1}[1]}%
\fi
```

\UniqueCounterNew

```
\uqc@Def\UniqueCounterNew{%
  \expandafter\ifx\csname uqc@cnt@#1\endcsname\relax
    \expandafter\xdef\csname uqc@cnt@#1\endcsname{0}%
    \global\expandafter\let\csname uqc@inc@#1\endcsname\uqc@IncNum
    \@PackageInfo{uniquecounter}{New unique counter `#1'}%
  \else
    \@PackageError{uniquecounter}{Unique counter `#1' is already defined}\@ehc
  \fi
}
```

\UniqueCounterIncrement

```
\uqc@Def\UniqueCounterIncrement{%
  \expandafter\ifx\csname uqc@cnt@#1\endcsname\relax
    \@PackageError{uniquecounter}{Unique counter `#1' is undefined}\@ehc
  \else
    \csname uqc@inc@#1\endcsname{#1}%
  \fi
}
```

\UniqueCounterGet

```
\uqc@Def\UniqueCounterGet{%
  \csname uqc@cnt@#1\endcsname
}
```

\UniqueCounterCall

```
\uqc@Def\UniqueCounterCall{%
  \expandafter\ifx\csname uqc@cnt@#1\endcsname\relax
    \@PackageError{uniquecounter}{Unique counter `#1' is undefined}\@ehc
    \expandafter\uqc@Call\expandafter0%
  \else
    \UniqueCounterIncrement{#1}%
    \expandafter\expandafter\expandafter\uqc@Call
    \expandafter\expandafter\expandafter{%
      \csname uqc@cnt@#1\expandafter\endcsname\expandafter
    }%
  \fi
}
```

\uqc@Call

```
\long\def\uqc@Call#1#2{#2{#1}}%
\uqc@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 uniquecounter.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

3.2.1 Test with LaTeX

```
⟨*test2⟩
\NeedsTeXFormat{LaTeX2e}
\nofiles
\documentclass{minimal}
\usepackage{uniquecounter}[2016/05/16]
\usepackage{qstest}
\IncludeTests{*}
\LogTests{log}{*}{*}

\newcommand*{\CheckValue}[2]{%
  \Expect*{#2}*{\UniqueCounterGet{#1}}%
}
\newcommand*{\CheckSpace}[1]{%
  \sbox0{#1}%
  \Expect{0.0pt}*{\the\wd0}%
}

\begin{qstest}{creation}{creation}
  \CheckSpace{%
    \UniqueCounterNew{test}%
  }%
  \CheckValue{test}{0}%
\end{qstest}

\begin{qstest}{increment}{increment}
  \CheckSpace{%
    \UniqueCounterIncrement{test}%
  }%
  \CheckValue{test}{1}%
  \makeatletter
  \def\uqc@cnt@test{2147483645}%
  \CheckValue{test}{2147483645}%
  \CheckSpace{%
    \UniqueCounterIncrement{test}%
  }%
  \CheckValue{test}{2147483646}%
  \CheckSpace{%
    \UniqueCounterIncrement{test}%
  }%
  \Expect{true}*{\ifx\uqc@inc\uqc@NumInc true\else false\fi}%
  \CheckValue{test}{2147483647}%
  \CheckSpace{%
    \UniqueCounterIncrement{test}%
  }%
  \CheckValue{test}{2147483648}%
  \CheckSpace{%
    \UniqueCounterIncrement{test}%
  }%
  \CheckValue{test}{2147483649}%
\end{qstest}

\begin{qstest}{call}{call}
  \def\CheckCall#1#2{%
```

```
    \Expect{#1}{#2}%
  }%
  \CheckSpace{%
    \UniqueCounterNew{foo}%
  }%
  \CheckValue{foo}{0}%
  \def\Check#1{%
    \CheckSpace{%
      \UniqueCounterCall{foo}{\CheckCall}{#1}%
    }%
    \CheckValue{foo}{#1}%
  }%
  \Check{1}%
  \Check{2}%
  \Check{3}%
  \Check{4}%
  \Check{5}%
  \Check{6}%
  \Check{7}%
  \Check{8}%
  \Check{9}%
  \Check{10}%
  \Check{11}%
  \Check{12}%
\end{qstest}

\csname @@end\endcsname
⟨/test2⟩
```

3.2.2 Test with plain-TEX

```
⟨*test3⟩
\input uniquecounter.sty\relax
\catcode`\@=11 %
\def\CheckValue#1#2{%
  \begingroup
    \edef\A{#2}%
    \edef\B{\UniqueCounterGet{#1}}%
    \ifx\A\B
    \else
      \@PackageError{TEST}{Failed: \A\space<> \B}\@ehc
    \fi
  \endgroup
}
\def\CheckSpace#1{%
  \setbox0=\hbox{#1}%
  \ifdim\wd0=\z@
  \else
    \@PackageError{TEST}{Failed: 0.0pt <> \the\wd0}\@ehc
  \fi
}

\begingroup
  \CheckSpace{%
    \UniqueCounterNew{test}%
  }%
  \CheckValue{test}{0}%
\endgroup

\begingroup
  \CheckSpace{%
    \UniqueCounterIncrement{test}%
  }%
  \CheckValue{test}{1}%
```

```
  \def\uqc@cnt@test{2147483645}%
  \CheckValue{test}{2147483645}%
  \CheckSpace{%
    \UniqueCounterIncrement{test}%
  }%
  \CheckValue{test}{2147483646}%
  \CheckSpace{%
    \UniqueCounterIncrement{test}%
  }%
  \ifx\uqc@inc\uqc@NumInc
  \else
    \@PackageError{TEST}{Failed: wrong inc function}\@ehc
  \fi
  \CheckValue{test}{2147483647}%
  \CheckSpace{%
    \UniqueCounterIncrement{test}%
  }%
  \CheckValue{test}{2147483648}%
  \CheckSpace{%
    \UniqueCounterIncrement{test}%
  }%
  \CheckValue{test}{2147483649}%
\endgroup
\begingroup
  \def\CheckCall#1#2{%
    \begingroup
      \def\A{#1}%
      \def\B{#2}%
      \ifx\A\B
      \else
        \@PackageError{TEST}{Failed: \A\space <> \B}\@ehc
      \fi
    \endgroup
  }%
  \CheckSpace{%
    \UniqueCounterNew{foo}%
  }%
  \CheckValue{foo}{0}%
  \CheckSpace{%
    \UniqueCounterCall{foo}{\CheckCall}{1}%
  }%
  \CheckSpace{%
    \UniqueCounterCall{foo}{\CheckCall}{2}%
  }%
  \CheckValue{foo}{2}%
\endgroup
\csname @@end\endcsname\end
⟨/test3⟩
```

4 Installation

4.1 Download

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

CTAN:macros/latex/contrib/oberdiek/uniquecounter.dtx The source file.

CTAN:macros/latex/contrib/oberdiek/uniquecounter.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.

[1] http://ctan.org/pkg/uniquecounter

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 uniquecounter.dtx
```

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

```
uniquecounter.sty             → tex/generic/oberdiek/uniquecounter.sty
uniquecounter.pdf             → doc/latex/oberdiek/uniquecounter.pdf
uniquecounter-example.tex     → doc/latex/oberdiek/uniquecounter-example.tex
test/uniquecounter-test1.tex → doc/latex/oberdiek/test/uniquecounter-test1.tex
test/uniquecounter-test2.tex → doc/latex/oberdiek/test/uniquecounter-test2.tex
test/uniquecounter-test3.tex → doc/latex/oberdiek/test/uniquecounter-test3.tex
uniquecounter.dtx             → source/latex/oberdiek/uniquecounter.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{uniquecounter.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 uniquecounter.dtx
makeindex -s gind.ist uniquecounter.idx
pdflatex uniquecounter.dtx
makeindex -s gind.ist uniquecounter.idx
pdflatex uniquecounter.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 uniquecounter.xml.

```
⟨*catalogue⟩
<?xml version='1.0' encoding='us-ascii'?>
<!DOCTYPE entry SYSTEM 'catalogue.dtd'>
<entry datestamp='$Date$' modifier='$Author$' id='uniquecounter'>
  <name>uniquecounter</name>
  <caption>Provides unlimited unique counter.</caption>
  <authorref id='auth:oberdiek'/>
  <copyright owner='Heiko Oberdiek' year='2009,2011'/>
  <license type='lppl1.3'/>
  <version number='1.3'/>
  <description>
    This package provides a kind of counter that provides unique
    number values. Several counters can be created with different names.
    The numeric values are not limited.
    <p/>
    The package is part of the <xref refid='oberdiek'>oberdiek</xref>
    bundle.
  </description>
  <documentation details='Package documentation'
      href='ctan:/macros/latex/contrib/oberdiek/uniquecounter.pdf'/>
  <ctan file='true' path='/macros/latex/contrib/oberdiek/uniquecounter.dtx'/>
  <miktex location='oberdiek'/>
  <texlive location='oberdiek'/>
  <install path='/macros/latex/contrib/oberdiek/oberdiek.tds.zip'/>
</entry>
⟨/catalogue⟩
```

6 History

[2009/09/11 v1.0]

- First public version.

[2009/12/18 v1.1]

- Bug fix in \UniqueCounterCall for values > 9 (bug report of Lev Bishop).

[2011/01/30 v1.2]

- Already loaded package files are not input in plain TeX.

[2016/05/16 v1.3]

- 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

\# 224
\% 300
\@ 225, 298, 403
\@PackageError 193, 198, 208, 410, 418, 445, 464
\@PackageInfo 191
\@ehc . 193, 198, 208, 410, 418, 445, 464
\@firstofone 233, 236
\@gobble 230, 238
\@namedef 9
\@nameuse 15
\@ne 154
\@undefined 82
\\ 21, 299
\{ 222
\} 223

A

\A 406, 408, 410, 460, 462, 464
\advance 154, 263, 271, 286
\aftergroup 53

B

\B 407, 408, 410, 461, 462, 464
\begin 17, 337, 344, 371
\BigIntCalcInc 178
\body 242, 246

C

\catcode 26, 27, 29, 30, 31, 32, 33, 34, 35, 36, 37, 57, 58, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 93, 94, 96, 97, 98, 102, 103, 104, 105, 106, 107, 108, 111, 112, 114, 115, 116, 117, 121, 123, 222, 223, 224, 225, 260, 269, 277, 281, 298, 299, 300, 403
\Check 379, 385, 386, 387, 388, 389, 390, 391, 392, 393, 394, 395, 396
\CheckCall 372, 381, 458, 473, 476
\CheckSpace 332, 338, 345, 352, 356, 361, 365, 375, 380, 414, 423, 430, 436, 440, 448, 452, 468, 472, 475
\CheckValue 329, 341, 348, 351, 355, 360, 364, 368, 378, 383, 404, 426, 433, 435, 439, 447, 451, 455, 471, 478
\count@ 153, 154, 156, 158, 227, 256, 260, 262, 263, 267, 269, 270, 271, 275, 277, 280, 281, 285, 286
\countdef 227
\csname 38, 45, 74, 90, 100, 136, 139, 150, 153, 155, 159, 166, 167, 169, 170, 176, 178, 182, 188, 189, 190, 197, 200, 204, 207, 214, 226, 229, 232, 235, 290, 317, 399, 480

D

\DefNewAnchorName 6, 12
\documentclass 2, 323

E

\empty 41, 42
\end 23, 318, 342, 369, 397, 480
\endcsname 38, 45, 74, 90, 100, 136, 139, 150, 153, 155, 159, 166, 167, 169, 170, 176, 178, 182, 188, 189, 190, 197, 200, 204, 207, 214, 226, 229, 232, 235, 290, 317, 399, 480
\endinput 53, 134
\endlinechar 28, 59, 95, 101, 113
\errmessage 279
\Expect 330, 334, 359, 373

H

\hbox 415

I

\ifdim 416
\ifnum 158, 169, 262, 270, 277, 285
\ifx 39, 42, 45, 74, 82, 85, 136, 139, 150, 182, 188, 197, 207, 226, 229, 232, 235, 290, 359, 408, 443, 462
\immediate 47, 76
\IncludeTests 326
\input 140, 291, 402
\iterate 243, 245, 247

L

\LoadCommand 291, 301
\LogTests 327
\loop 241, 257, 268, 276

M

\makeatletter 5, 349

N

\NeedsTeXFormat 321
\NewAnchorName 11, 18, 19

\newcommand . . 6, 11, 14, 185, 329, 332
\next 247, 249, 251
\nofiles 322
\noindent 20
\number 156, 167, 282
\numexpr 167

P

\PackageInfo 50
\PrintAnchorName 14, 21, 22
\ProvidesPackage 43, 91

R

\RangeCatcodeCheck 274, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313
\RangeCatcodeInvalid 266, 294, 295, 296, 297
\repeat 241, 253, 264, 272, 287
\RequirePackage 146, 147
\RestoreCatcodes . . 255, 258, 259, 314

S

\sbox 333
\setbox 415
\space 280, 281, 289, 410, 464

T

\Test 293, 316
\the 101, 102, 103, 104, 105, 106, 107, 108, 121, 260, 280, 281, 334, 418
\TMP@EnsureCode . . 118, 125, 126, 127, 128, 129, 130, 131, 132, 133
\TMP@RequirePackage . . 137, 143, 144

U

\UniqueCounterCall 2, 12, 206, 381, 473, 476
\UniqueCounterGet . . . 2, 203, 330, 407
\UniqueCounterIncrement 2, 196, 211, 346, 353, 357, 362, 366, 431, 437, 441, 449, 453
\UniqueCounterNew 2, 4, 187, 339, 376, 424, 469
\uqc@AtEnd 119, 120, 134, 219
\uqc@Call 209, 212, 218
\uqc@cnt@test 350, 434
\uqc@Def 181, 187, 196, 203, 206
\uqc@inc 359, 443
\uqc@IncBig 160, 171, 175
\uqc@IncNum 149, 190
\uqc@NumInc 359, 443
\usepackage 3, 324, 325

W

\wd 334, 416, 418
\write 47, 76

X

\x 38, 39, 42, 46, 50, 52, 75, 80, 90, 99, 111

Z

\z@ 416