The rerunfilecheck package

Heiko Oberdiek
<heiko.oberdiek at googlemail.com>

2011/04/15 v1.7

Abstract

The package provides additional rerun warnings if some auxiliary files have changed. It is based on MD5 checksum, provided by pdfTeX.

Contents

1 Documentation **1**
1.1 Options . . . 2
1.2 Interface for class/package authors . . . 2

2 Implementation **3**
2.1 Options . . . 4
2.2 Check for checksum feature . . . 4
2.3 Standard .aux files . . . 4
2.4 Rerun check . . . 7

3 Test **8**
3.1 Catcode checks for loading . . . 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 History **12**
[2009/12/10 v1.0] . . . 12
[2009/12/12 v1.1] . . . 12
[2009/12/18 v1.2] . . . 12
[2010/01/25 v1.3] . . . 12
[2010/02/22 v1.4] . . . 12
[2010/03/15 v1.5] . . . 12
[2010/03/16 v1.6] . . . 12
[2011/04/15 v1.7] . . . 12

6 Index **13**

1 Documentation

LaTeX informs the user, when to run LaTeX again, if the references have changed. It has the old references from the first reading of the .aux files already in memory, thus it compares them with the new version of the .aux file at the end of the document. However this rerun warnings are not given for the table of contents

and other data stored in the `.aux` files or other auxiliary files. Usually many of these data as the table of contents is not keep in memory. If someone wants to detect changes, he has either to keep the data in memory. This does not scale well with huge documents. Or he copies the file before they are changed. Slow I/O operations cost time.

Since version 1.30.0 pdfTeX provides `\pdfmdfivesum` and `\pdffilesize`. These features are also available in LuaTeX, provided by package pdftexcmds. Thus this package rerunfilecheck uses these features to detect file changes. This saves the packages from keeping the whole files in memory or in file copies. The drawback are different files with the same size and the same MD5 checksum (seldom, hopefully).

1.1 Options

All options are key value options of boolean type. No option or `true` turns an option on, `false` disables an option.

mainaux: Check the main `.aux` file.

partaux: Check the `.aux` files from `\include` files.

starttoc: Add the rerun checks in `\@starttoc` that is called by `\tableofcontents`, `\listoffigures`, ...

index, glossary: LaTeX's original `\makeindex` and `\makeglossary` are redefined to add the rerun checks. The options do not have an effect, if `\makeindex`/`\makeglossary` are already called or if a package or class had redefined or will redefine them.

aux: This option turns all previous options on or off. "aux" means auxiliary file.

The default for the options is `false`, because some internals must be redefined to insert the rerun checks. The options can be set in `\usepackage` or the configuration file `rerunfilecheck.cfg`. Global options are ignored (since 1.4).

`\RerunFileCheckSetup` {⟨*key value list*⟩}

Options can also be set using `\RerunFileCheckSetup`. Currently all options are disabled after the package is loaded. Thus `\RerunFileCheckSetup` makes sense in the configuration file only.

Example for the configuration file:

```
1 ⟨*cfg⟩
2 \ProvidesFile{rerunfilecheck.cfg}[2011/04/15 Default configuration]%
3 \RerunFileCheckSetup{aux}
4 ⟨/cfg⟩
```

1.2 Interface for class/package authors

`\RerunFileCheck` {⟨*file*⟩} {⟨*file closing action*⟩} {⟨*rerun warning*⟩}

If you want to add a rerun check, call `\RerunFileCheck` right before an output file is opened for writing. The macro first remembers the current checksum of ⟨*file*⟩. The file is checked again right before the end of the job. Macro `\AtVeryEndDocument` of package atveryend is used to place the check after the main aux file is closed in `\end{document}`. Before reading the file again, it must be closed. Provide the code for closing in argument ⟨*file closing action*⟩. Do not forget `\immediate` before `\openout`. Otherwise the closing action would be delayed to the next shipout that never happens (the last page is already shipped out). If the file has changed, `\RerunFileCheck` informs the user with a warning that the file has changed and says the magic word "Rerun". If the last argument

⟨*rerun warning*⟩ is not empty, then the rerun sentence is replaced by it. Usually the phrase "to get something right" is added. As example the relevant part of the redefined `\makeindex` is shown, see package code:

```
\newwrite\@indexfile
\RerunFileCheck{\jobname.idx}{%
  \immediate\closeout\@indexfile
}{%
  Rerun LaTeX/makeindex to get index right%
}%
\immediate\openout\@indexfile=\jobname.idx %
```

2 Implementation

```
⟨*package⟩
\begingroup\catcode61\catcode48\catcode32=10\relax%
  \catcode13=5 % ^^M
  \endlinechar=13 %
  \catcode123=1 % {
  \catcode125=2 % }
  \catcode64=11 % @
  \def\x{\endgroup
    \expandafter\edef\csname ReFiCh@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\ReFiCh@AtEnd{%
    \ReFiCh@AtEnd
    \catcode#1=\the\catcode#1\relax
  }%
  \catcode#1=#2\relax
}
\TMP@EnsureCode{39}{12}% '
\TMP@EnsureCode{40}{12}% (
\TMP@EnsureCode{41}{12}% )
\TMP@EnsureCode{42}{12}% *
\TMP@EnsureCode{44}{12}% ,
\TMP@EnsureCode{46}{12}% .
\TMP@EnsureCode{47}{12}% /
\TMP@EnsureCode{58}{12}% :
\TMP@EnsureCode{59}{12}% ;
\TMP@EnsureCode{60}{12}% <
\TMP@EnsureCode{62}{12}% >
\TMP@EnsureCode{91}{12}% [
\TMP@EnsureCode{93}{12}% ]
\TMP@EnsureCode{96}{12}% `
\edef\ReFiCh@AtEnd{\ReFiCh@AtEnd\noexpand\endinput}
```

Package identification.

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{rerunfilecheck}%
  [2011/04/15 v1.7 Rerun checks for auxiliary files (HO)]
```

2.1 Options

```
\RequirePackage{kvoptions}[2010/02/22]
\SetupKeyvalOptions{%
  family=rerunfilecheck,%
  prefix=ReFiCh@%
}
```

\RerunFileCheckSetup

```
\newcommand*{\RerunFileCheckSetup}{%
  \setkeys{rerunfilecheck}%
}

\DeclareBoolOption{mainaux}
\DeclareBoolOption{partaux}
\DeclareBoolOption{starttoc}
\DeclareBoolOption{index}
\DeclareBoolOption{glossary}
\define@key{rerunfilecheck}{aux}[true]{%
  \RerunFileCheckSetup{%
    mainaux={#1},%
    partaux={#1},%
    starttoc={#1},%
    index={#1},%
    glossary={#1}%
  }%
}

\InputIfFileExists{rerunfilecheck.cfg}{}{}
\ProcessLocalKeyvalOptions*
```

\ReFiCh@DisableOption

```
\def\ReFiCh@DisableOption{%
  \DisableKeyvalOption[%
    action=warning,%
    package=rerunfilecheck%
  ]{rerunfilecheck}%
}
```

2.2 Check for checksum feature

```
\RequirePackage{infwarerr}[2007/09/09]
\RequirePackage{pdftexcmds}[2009/04/10]

\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname pdf@filemdfivesum\endcsname\relax
  \@PackageInfoNoLine{rerunfilecheck}{%
    Feature \string\pdfmdfivesum\space is not available\MessageBreak
    (e.g. pdfTeX or LuaTeX with package 'pdftexcmds').\MessageBreak
    Therefore file contents cannot be checked efficiently\MessageBreak
    and the loading of the package is aborted%
  }%
  \newcommand*{\RerunFileCheck}[3]{}%
  \renewcommand*{\RerunFileCheckSetup}[1]{}%
  \expandafter\ReFiCh@AtEnd
\fi%
```

2.3 Standard .aux files

```
\ifReFiCh@partaux
```

```
  \let\ReFiCh@org@include\@include
  \def\@include#1 {%
    \if@filesw
      \RerunFileCheck{#1.aux}{}{}%
    \fi
    \ReFiCh@org@include{#1} %
  }%
\fi
\ifReFiCh@mainaux
  \AtBeginDocument{%
    \ReFiCh@mainauxfalse
  }%
  \ifReFiCh@mainaux
    \AtEndOfPackage{%
      \RerunFileCheck{\jobname.aux}{}{}%
    }%
  \else
    \if@filesw
      \@PackageWarningNoLine{rerunfilecheck}{%
        Main aux file check is disabled,\MessageBreak
        because the file is already opened.\MessageBreak
        Load the package before \string\begin{document}%
      }%
    \fi
  \fi
\fi
\ifReFiCh@starttoc
  \let\ReFiCh@org@starttoc\@starttoc
  \def\@starttoc#1{%
    \if@filesw
      \RerunFileCheck{\jobname.#1}{%
        \@ifundefined{tf@#1}{%
        }{%
          \immediate\closeout\csname tf@#1\endcsname
        }%
      }{}%
    \fi
    \ReFiCh@org@starttoc{#1}%
  }%
\fi
\ifReFiCh@index
  \ifx\makeindex\@empty
    \@PackageWarningNoLine{rerunfilecheck}{%
      Option 'index' ignored,\MessageBreak
      because \string\makeindex\space has already been called%
    }%
  \else
    \def\ReFiCh@temp{%
      \newwrite\@indexfile
      \immediate\openout\@indexfile=\jobname.idx %
      \def\index{%
        \@bsphack
        \begingroup
        \@sanitize
        \@wrindex
      }%
      \typeout{Writing index file \jobname.idx}%
      \let\makeindex\@empty
    }%
    \ifx\ReFiCh@temp\makeindex
      \def\makeindex{%
        \newwrite\@indexfile
```

```
        \RerunFileCheck{\jobname.idx}{%
          \immediate\closeout\@indexfile
        }{%
          Rerun LaTeX/makeindex to get index right%
        }%
        \immediate\openout\@indexfile=\jobname.idx %
        \def\index{%
          \@bsphack
          \begingroup
          \@sanitize
          \@wrindex
        }%
        \typeout{Writing index file \jobname.idx}%
        \let\makeindex\@empty
      }%
    \else
      \@PackageInfoNoLine{rerunfilecheck}{%
        Option `index': unsupported version of \string\makeindex
      }%
    \fi
  \fi
\fi
\ifReFiCh@glossary
  \ifx\makeglossary\@empty
    \@PackageWarningNoLine{rerunfilecheck}{%
      Option `glossary' ignored,\MessageBreak
      because \string\makeglossary\space has already been called%
    }%
  \else
    \def\ReFiCh@temp{%
      \newwrite\@glossaryfile
      \immediate\openout\@glossaryfile=\jobname.glo %
      \def\glossary{%
        \@bsphack
        \begingroup
        \@sanitize
        \@wrglossary
      }%
      \typeout{Writing glossary file \jobname.glo }%
      \let\makeglossary\@empty
    }%
    \ifx\ReFiCh@temp\makeglossary
      \def\ReFiCh@temp{%
        \newwrite\@glossaryfile
        \RerunFileCheck{\jobname.glo}{%
          \immediate\closeout\@glossaryfile
        }{%
          Rerun LaTeX/makeindex to get glossary right%
        }%
        \immediate\openout\@glossaryfile=\jobname.glo %
        \def\glossary{%
          \@bsphack
          \begingroup
          \@sanitize
          \@wrglossary
        }%
        \typeout{Writing glossary file \jobname.glo}%
        \let\makeglossary\@empty
      }%
    \else
      \@PackageInfoNoLine{rerunfilecheck}{%
        Option `glossary': unsupported version of \string\makeglossary
```

```
      }%
    \fi
  \fi
\fi
\ReFiCh@DisableOption{mainaux}
\ReFiCh@DisableOption{partaux}
\ReFiCh@DisableOption{starttoc}
\ReFiCh@DisableOption{index}
\ReFiCh@DisableOption{glossary}
\ReFiCh@DisableOption{aux}
```

2.4 Rerun check

```
\RequirePackage{atveryend}[2011/04/15]
\RequirePackage{uniquecounter}[2009/12/18]
```

\ReFiCh@CheckSum

```
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname pdf@filesize\endcsname\relax
  \def\ReFiCh@CheckSum{%
    \pdf@filemdfivesum
  }%
\else
  \def\ReFiCh@CheckSum#1{%
    \pdf@filemdfivesum{#1}%
    \ReFiCh@Separator
    \pdf@filesize{#1}%
  }%
\fi
```

\ReFiCh@NoFile

```
\def\ReFiCh@Separator{;}
```

\ReFiCh@NoFile

```
\def\ReFiCh@NoFile{<no file>}
```

```
\UniqueCounterNew{rerunfilecheck}
```

\RerunFileCheck

```
\newcommand*{\RerunFileCheck}{%
  \UniqueCounterCall{rerunfilecheck}\ReFiCh@RerunFileCheck
}
```

\ReFiCh@RerunFileCheck

```
\def\ReFiCh@RerunFileCheck#1{%
  \expandafter\ReFiCh@@RerunFileCheck\csname ReFiCh@#1\endcsname
}
```

\ReFiCh@Check

```
\def\ReFiCh@Check#1#2#3{%
%  \IfFileExists{#3}{%
    #1\edef#2{\ReFiCh@CheckSum{#3}}%
    \ifx#2\ReFiCh@Separator
      #1\let#2\ReFiCh@NoFile
    \fi
%  }{%
%    #1\let#2\ReFiCh@NoFile
%  }%
}
```

\ReFiCh@@RerunFileCheck

```
\def\ReFiCh@@RerunFileCheck#1#2#3#4{%
```

```
  \ReFiCh@Check\global#1{#2}%
  \AtEndAfterFileList{%
    \begingroup
      #3%
      \ReFiCh@Check{}\x{#2}%
      \ifx#1\x
        \@PackageInfoNoLine{rerunfilecheck}{%
          File `#2' has not changed.\MessageBreak
          Checksum: \x
        }%
      \else
        \ifnum
          \ReFiCh@IsAux#2\relax.aux\relax\@nil
          \ifx#1\ReFiCh@NoFile 1\else 0\fi
          \ifx\x\ReFiCh@AuxEmptyUnix 1%
          \else
            \ifx\x\ReFiCh@AuxEmptyDos 1\fi
          \fi
        =111 %
          \@PackageInfoNoLine{rerunfilecheck}{%
            File `#2' is empty .aux file.\MessageBreak
            Before: #1\MessageBreak
            After: \space\x
          }%
        \else
          \@PackageWarningNoLine{rerunfilecheck}{%
            File `#2' has changed.%
            \ifx\\#4\\%
              \space Rerun%
            \else
              \MessageBreak
              #4%
            \fi
          }%
          \@PackageInfoNoLine{rerunfilecheck}{%
            Checksums for `#2':\MessageBreak
            Before: #1\MessageBreak
            After: \space\x
          }%
        \fi
      \fi
    \endgroup
  }%
}
\def\ReFiCh@IsAux#1.aux\relax#2\@nil{%
  \ifx\hbox#2\hbox
    0%
  \else
    1%
  \fi
}
\def\ReFiCh@AuxEmptyUnix{A94A2480D3289E625EEA47CD1B285758;8}%
\@onelevel@sanitize\ReFiCh@AuxEmptyUnix
\def\ReFiCh@AuxEmptyDos{A62A15ECE803E2EBE94952FCC9933BC0;9}%
\@onelevel@sanitize\ReFiCh@AuxEmptyDos
\ReFiCh@AtEnd%
⟨/package⟩
```

3 Test

```
⟨*test1⟩
\def\LoadCommand{\RequirePackage{rerunfilecheck}[2011/04/15]}
⟨/test1⟩
```

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 rerunfilecheck.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⟩

4 Installation

4.1 Download

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

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

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

[1] ftp://ftp.ctan.org/tex-archive/

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 rerunfilecheck.dtx
```

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

```
rerunfilecheck.sty            → tex/latex/oberdiek/rerunfilecheck.sty
rerunfilecheck.pdf            → doc/latex/oberdiek/rerunfilecheck.pdf
rerunfilecheck-example.cfg    → doc/latex/oberdiek/rerunfilecheck-example.cfg
test/rerunfilecheck-test1.tex → doc/latex/oberdiek/test/rerunfilecheck-test1.tex
rerunfilecheck.dtx            → source/latex/oberdiek/rerunfilecheck.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

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

```
pdftk rerunfilecheck.pdf unpack_files output .
```

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{rerunfilecheck.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 rerunfilecheck.dtx
makeindex -s gind.ist rerunfilecheck.idx
pdflatex rerunfilecheck.dtx
makeindex -s gind.ist rerunfilecheck.idx
pdflatex rerunfilecheck.dtx
```

5 History

[2009/12/10 v1.0]

- The first version.

[2009/12/12 v1.1]

- Short info shortened.

[2009/12/18 v1.2]

- Required date for package uniquecounter updated because of bug in this package.

[2010/01/25 v1.3]

- Moved from `TDS:*/generic/*` to `TDS:*/latex/*`.

[2010/02/22 v1.4]

- The options of this package are recognized only if they are package options. Global options are ignored. This avoids name clashes with class and other package options (for example, class option 'index=totoc').

[2010/03/15 v1.5]

- Call of `\pdffivesum` is wrapped in `\IfFileExists` to avoid calls of mktextex if this feature is enabled. However `\IfFileExists` has file name limitations.

[2010/03/16 v1.6]

- Reverted to version 1.4 and `\IfFileExists` wrapper of version 1.5 is removed.

[2011/04/15 v1.7]

- Using `\AtEndAfterFileList` of package atveryend 2011/04/15 v1.6 instead of `\AtVeryEndDocument`.

6 Index

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

Symbols

\# 332
\% 408
\@ 333, 406
\@PackageInfoNoLine 90, 179, 223, 275, 288, 303
\@PackageWarningNoLine 119, 143, 187, 294
\@bsphack 152, 170, 196, 214
\@empty . . . 142, 158, 176, 186, 202, 220
\@firstofone 341, 344
\@glossaryfile 193, 194, 206, 208, 212
\@gobble 338, 346
\@ifundefined 132
\@include 101, 102
\@indexfile . . . 149, 150, 162, 164, 168
\@nil 281, 313
\@onelevel@sanitize 321, 323
\@sanitize 154, 172, 198, 216
\@starttoc 128, 129
\@wrglossary 199, 217
\@wrindex 155, 173
\\ 296, 407
\{ 330
\} 331

A

\advance 371, 379, 394
\AtBeginDocument 110
\AtEndAfterFileList 270
\AtEndOfPackage 114

B

\begin 122
\body 350, 354

C

\catcode 6, 7, 9, 10, 11, 15, 16, 17, 18, 19, 20, 21, 24, 25, 27, 28, 29, 30, 34, 36, 330, 331, 332, 333, 368, 377, 385, 389, 406, 407, 408
\closeout 134, 164, 208
\count@ 335, 364, 368, 370, 371, 375, 377, 378, 379, 383, 385, 388, 389, 393, 394
\countdef 335
\csname 13, 89, 134, 238, 256, 334, 337, 340, 343, 398, 425

D

\DeclareBoolOption . . 64, 65, 66, 67, 68
\define@key 69
\DisableKeyvalOption 81

E

\end 426
\endcsname 13, 89, 134, 238, 256, 334, 337, 340, 343, 398, 425
\endinput 52
\endlinechar 8, 14, 26
\errmessage 387

G

\glossary 195, 213

H

\hbox 314

I

\if@filesw 103, 118, 130
\IfFileExists 259
\ifnum 280, 370, 378, 385, 393
\ifReFiCh@glossary 185
\ifReFiCh@index 141
\ifReFiCh@mainaux 109, 113
\ifReFiCh@partaux 100
\ifReFiCh@starttoc 127
\ifx 89, 142, 160, 186, 204, 238, 261, 274, 282, 283, 285, 296, 314, 334, 337, 340, 343, 398
\immediate 134, 150, 164, 168, 194, 208, 212
\index 151, 169
\input 399
\InputIfFileExists 78
\iterate 351, 353, 355

J

\jobname . . 115, 131, 150, 157, 163, 168, 175, 194, 201, 207, 212, 219

L

\LoadCommand 327, 399, 409
\loop 349, 365, 376, 384

M

\makeglossary 186, 189, 202, 204, 220, 224
\makeindex 142, 145, 158, 160, 161, 176, 180
\MessageBreak 91, 92, 93, 120, 121, 144, 188, 276, 289, 290, 299, 304, 305

N

\NeedsTeXFormat 53
\newcommand 61, 96, 252
\newwrite 149, 162, 193, 206
\next 355, 357, 359
\number 390

O

\openout 150, 168, 194, 212

P

\pdf@filemdfivesum 240, 244
\pdf@filesize 246
\pdfmdfivesum 91
\ProcessLocalKeyvalOptions 79
\ProvidesFile 2
\ProvidesPackage 54

R

\RangeCatcodeCheck 382, 410, 411, 412, 413, 414, 415, 416, 417, 418, 419, 420, 421
\RangeCatcodeInvalid 374, 402, 403, 404, 405
\ReFiCh@@RerunFileCheck . . . 256, 268
\ReFiCh@AtEnd 32, 33, 52, 98, 324
\ReFiCh@AuxEmptyDos . . . 285, 322, 323
\ReFiCh@AuxEmptyUnix . . 283, 320, 321
\ReFiCh@Check 258, 269, 273
\ReFiCh@CheckSum 237, 260
\ReFiCh@DisableOption 80, 229, 230, 231, 232, 233, 234
\ReFiCh@IsAux 281, 313
\ReFiCh@mainauxfalse 111
\ReFiCh@NoFile 249, 250, 262, 265, 282
\ReFiCh@org@include 101, 106
\ReFiCh@org@starttoc 128, 138
\ReFiCh@RerunFileCheck 253, 255
\ReFiCh@Separator 245, 249, 261
\ReFiCh@temp . . 148, 160, 192, 204, 205
\renewcommand 97
\repeat 349, 361, 372, 380, 395
\RequirePackage 56, 86, 87, 235, 236, 327
\RerunFileCheck *2*, 96, 104, 115, 131, 163, 207, 252
\RerunFileCheckSetup . *2*, 3, 61, 70, 97
\RestoreCatcodes . . 363, 366, 367, 422

S

\setkeys 62
\SetupKeyvalOptions 57
\space 91, 145, 189, 291, 297, 306, 388, 389, 397

T

\Test 401, 424
\the 14, 15, 16, 17, 18, 19, 20, 21, 34, 368, 388, 389
\TMP@EnsureCode 31, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51
\typeout 157, 175, 201, 219

U

\UniqueCounterCall 253
\UniqueCounterNew 251

X

\x 12, 24, 273, 274, 277, 283, 285, 291, 306