

The rerunfilecheck package

Heiko Oberdiek
<oberdiek@uni-freiburg.de>

2009/12/10 v1.0

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	5
2.3	Standard ..aux files	5
2.4	Rerun check	7
3	Test	9
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
6	Index	12

1 Documentation

L^AT_EX informs the user, when to run L^AT_EX 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 L^AT_EX, 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: L^AT_EX'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`.

`\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}[2009/12/10 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

5 (*package)

Reload check, especially if the package is not used with L^AT_EX.

```

6 \begingroup
7   \catcode44 12 % ,
8   \catcode45 12 % -
9   \catcode46 12 % .
10  \catcode58 12 % :
11  \catcode64 11 % @
12  \catcode123 1 % {
13  \catcode125 2 % }
14  \expandafter\let\expandafter\x\csname ver@rerunfilecheck.sty\endcsname
15  \ifx\x\relax % plain-TeX, first loading
16  \else
17    \def\empty{}%
18    \ifx\x\empty % LaTeX, first loading,
19      % variable is initialized, but \ProvidesPackage not yet seen
20    \else
21      \catcode35 6 % #
22      \expandafter\ifx\csname PackageInfo\endcsname\relax
23        \def\x#1#2{%
24          \immediate\write-1{Package #1 Info: #2.}%
25        }%
26      \else
27        \def\x#1#2{\PackageInfo{#1}{#2, stopped}}%
28      \fi
29      \x{rerunfilecheck}{The package is already loaded}%
30      \aftergroup\endinput
31    \fi
32  \fi
33 \endgroup

```

Package identification:

```

34 \begingroup
35   \catcode35 6 % #
36   \catcode40 12 % (
37   \catcode41 12 % )
38   \catcode44 12 % ,
39   \catcode45 12 % -
40   \catcode46 12 % .
41   \catcode47 12 % /
42   \catcode58 12 % :
43   \catcode64 11 % @
44   \catcode91 12 % [
45   \catcode93 12 % ]
46   \catcode123 1 % {
47   \catcode125 2 % }
48   \expandafter\ifx\csname ProvidesPackage\endcsname\relax
49     \def\x#1#2#3[#4]{\endgroup
50       \immediate\write-1{Package: #3 #4}%
51       \xdef#1{#4}%
52     }%
53   \else
54     \def\x#1#2[#3]{\endgroup
55       #2[{#3}]%
56       \ifx#1\@undefined

```

```

57      \xdef#1{#3}%
58      \fi
59      \ifx#1\relax
60      \xdef#1{#3}%
61      \fi
62  }%
63  \fi
64  \expandafter\x\csname ver@rerunfilecheck.sty\endcsname
65  \ProvidesPackage{rerunfilecheck}%
66  [2009/12/10 v1.0 Checksum based rerun checks on auxiliary files (HO)]
67  \begingroup
68    \catcode123 1 % {
69    \catcode125 2 % }
70    \def\x{\endgroup
71      \expandafter\edef\csname ReFiCh@AtEnd\endcsname{%
72        \catcode35 \the\catcode35\relax
73        \catcode64 \the\catcode64\relax
74        \catcode123 \the\catcode123\relax
75        \catcode125 \the\catcode125\relax
76      }%
77    }%
78    \x
79    \catcode35 6 % #
80    \catcode64 11 % @
81    \catcode123 1 % {
82    \catcode125 2 % }
83    \def\TMP@EnsureCode#1#2{%
84      \edef\ReFiCh@AtEnd{%
85        \ReFiCh@AtEnd
86        \catcode#1 \the\catcode#1\relax
87      }%
88      \catcode#1 #2\relax
89    }
90    \TMP@EnsureCode{39}{12}% '
91    \TMP@EnsureCode{40}{12}% (
92    \TMP@EnsureCode{41}{12}% )
93    \TMP@EnsureCode{42}{12}% *
94    \TMP@EnsureCode{44}{12}% ,
95    \TMP@EnsureCode{46}{12}% .
96    \TMP@EnsureCode{47}{12}% /
97    \TMP@EnsureCode{58}{12}% :
98    \TMP@EnsureCode{59}{12}% ;
99    \TMP@EnsureCode{60}{12}% <
100    \TMP@EnsureCode{61}{12}% =
101    \TMP@EnsureCode{62}{12}% >
102    \TMP@EnsureCode{91}{12}% [
103    \TMP@EnsureCode{93}{12}% ]
104    \TMP@EnsureCode{96}{12}% '
105    \g@addto@macro\ReFiCh@AtEnd{\endinput}

```

2.1 Options

```

106 \RequirePackage{kvoptions}[2009/07/17]
107 \SetupKeyvalOptions{%
108   family=rerunfilecheck,%
109   prefix=ReFiCh%
110 }

```

\RerunFileCheckSetup

```

111 \newcommand*\RerunFileCheckSetup{%
112   \setkeys{rerunfilecheck}%
113 }
114 \DeclareBoolOption{mainaux}

```

```

115 \DeclareBoolOption{partaux}
116 \DeclareBoolOption{starttoc}
117 \DeclareBoolOption{index}
118 \DeclareBoolOption{glossary}
119 \define@key{rerunfilecheck}{aux}[true]{%
120   \RerunFileCheckSetup{%
121     mainaux={#1},%
122     partaux={#1},%
123     starttoc={#1},%
124     index={#1},%
125     glossary={#1}%
126   }%
127 }

128 \InputIfFileExists{rerunfilecheck.cfg}{-}{-}
129 \ProcessKeyvalOptions*

```

\ReFiCh@DisableOption

```

130 \def\ReFiCh@DisableOption{%
131   \DisableKeyvalOption[%
132     action=warning,%
133     package=rerunfilecheck%
134   ]{rerunfilecheck}%
135 }

```

2.2 Check for checksum feature

```

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

138 \begingroup\expandafter\expandafter\expandafter\endgroup
139 \expandafter\ifx\csname pdf@filemdfivesum\endcsname\relax
140   \@PackageInfoNoLine{rerunfilecheck}{%
141     Feature \string\pdfmdfivesum\space is not available\MessageBreak
142     (e.g. pdfTeX or LuaTeX with package ‘pdftexcmds’).\MessageBreak
143     Therefore file contents cannot be checked efficiently\MessageBreak
144     and the loading of the package is aborted%
145   }%
146   \newcommand*{\RerunFileCheck}[3]{}%
147   \renewcommand*{\RerunFileCheckSetup}[1]{}%
148   \expandafter\ReFiCh@AtEnd
149 \fi

```

2.3 Standard .aux files

```

150 \ifReFiCh@partaux
151   \let\ReFiCh@org@include\@include
152   \def\@include#1 {%
153     \if@filesw
154       \RerunFileCheck{#1.aux}{-}{-}%
155     \fi
156     \ReFiCh@org@include{#1} %
157   }%
158 \fi

159 \ifReFiCh@mainaux
160   \AtBeginDocument{%
161     \ReFiCh@mainauxfalse
162   }%
163   \ifReFiCh@mainaux
164     \AtEndOfPackage{%
165       \RerunFileCheck{\jobname.aux}{-}{-}%
166     }%
167   \else
168     \if@filesw

```

```

169     \@PackageWarningNoLine{rerunfilecheck}{%
170         Main aux file check is disabled,\MessageBreak
171         because the file is already opened.\MessageBreak
172         Load the package before \string\begin{document}%
173     }%
174 \fi
175 \fi
176 \fi
177 \ifReFiCh@starttoc
178     \let\ReFiCh@org@starttoc\@starttoc
179     \def\@starttoc#1{%
180         \if@filesw
181             \RerunFileCheck{\jobname.#1}{%
182                 \@ifundefined{tf@#1}{%
183                     }{%
184                         \immediate\closeout\csname tf@#1\endcsname
185                     }%
186                 }{}%
187             \fi
188             \ReFiCh@org@starttoc{#1}%
189         }%
190 \fi
191 \ifReFiCh@index
192     \ifx\makeindex\@empty
193         \@PackageWarningNoLine{rerunfilecheck}{%
194             Option 'index' ignored,\MessageBreak
195             because \string\makeindex\space has already been called%
196         }%
197     \else
198         \def\ReFiCh@temp{%
199             \newwrite\@indexfile
200             \immediate\openout\@indexfile=\jobname.idx %
201             \def\index{%
202                 \@bsphack
203                 \begingroup
204                 \@sanitize
205                 \@wrindex
206             }%
207             \typeout{Writing index file \jobname.idx}%
208             \let\makeindex\@empty
209         }%
210         \ifx\ReFiCh@temp\makeindex
211             \def\makeindex{%
212                 \newwrite\@indexfile
213                 \RerunFileCheck{\jobname.idx}{%
214                     \immediate\closeout\@indexfile
215                 }{%
216                     Rerun LaTeX/makeindex to get index right%
217                 }%
218                 \immediate\openout\@indexfile=\jobname.idx %
219                 \def\index{%
220                     \@bsphack
221                     \begingroup
222                     \@sanitize
223                     \@wrindex
224                 }%
225                 \typeout{Writing index file \jobname.idx}%
226                 \let\makeindex\@empty
227             }%
228         \else
229             \@PackageInfoNoLine{rerunfilecheck}{%
230                 Option 'index': unsupported version of \string\makeindex

```

```

231     }%
232     \fi
233 \fi
234 \fi
235 \ifReFiCh@glossary
236   \ifx\makeglossary\@empty
237     \@PackageWarningNoLine{rerunfilecheck}{%
238       Option 'glossary' ignored,\MessageBreak
239       because \string\makeglossary\space has already been called%
240     }%
241   \else
242     \def\ReFiCh@temp{%
243       \newwrite\@glossaryfile
244       \immediate\openout\@glossaryfile=\jobname.glo %
245       \def\glossary{%
246         \@bsphack
247         \begingroup
248         \@sanitize
249         \@wrglossary
250       }%
251       \typeout{Writing glossary file \jobname.glo }%
252       \let\makeglossary\@empty
253     }%
254     \ifx\ReFiCh@temp\makeglossary
255       \def\ReFiCh@temp{%
256         \newwrite\@glossaryfile
257         \RerunFileCheck{\jobname.glo}{%
258           \immediate\closeout\@glossaryfile
259         }{%
260           Rerun LaTeX/makeindex to get glossary right%
261         }%
262         \immediate\openout\@glossaryfile=\jobname.glo %
263         \def\glossary{%
264           \@bsphack
265           \begingroup
266           \@sanitize
267           \@wrglossary
268         }%
269         \typeout{Writing glossary file \jobname.glo}%
270         \let\makeglossary\@empty
271       }%
272     \else
273       \@PackageInfoNoLine{rerunfilecheck}{%
274         Option 'glossary': unsupported version of \string\makeglossary
275       }%
276     \fi
277 \fi
278 \fi
279 \ReFiCh@DisableOption{mainaux}
280 \ReFiCh@DisableOption{partaux}
281 \ReFiCh@DisableOption{starttoc}
282 \ReFiCh@DisableOption{index}
283 \ReFiCh@DisableOption{glossary}
284 \ReFiCh@DisableOption{aux}

```

2.4 Rerun check

```

285 \RequirePackage{atveryend}[2009/12/07]
286 \RequirePackage{uniquecounter}[2009/09/11]

```

\ReFiCh@Checksum

```

287 \begingroup\expandafter\expandafter\expandafter\endgroup
288 \expandafter\ifx\csname pdf@filesize\endcsname\relax
289   \def\ReFiCh@Checksum{%

```

```

290     \pdf@filemdfivesum
291 }%
292 \else
293   \def\ReFiCh@Checksum#1{%
294     \pdf@filemdfivesum{#1}%
295     \ReFiCh@Separator
296     \pdf@filesize{#1}%
297 }%
298 \fi

\ReFiCh@NoFile

299 \def\ReFiCh@Separator{;}

\ReFiCh@NoFile

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

301 \UniqueCounterNew{rerunfilecheck}

\RerunFileCheck

302 \newcommand*{\RerunFileCheck}{%
303   \UniqueCounterCall{rerunfilecheck}\ReFiCh@RerunFileCheck
304 }

\ReFiCh@RerunFileCheck

305 \def\ReFiCh@RerunFileCheck#1{%
306   \expandafter\ReFiCh@@RerunFileCheck\csname ReFiCh@#1\endcsname
307 }

\ReFiCh@@RerunFileCheck

308 \def\ReFiCh@@RerunFileCheck#1#2#3#4{%
309   \xdef#1{\ReFiCh@Checksum{#2}}%
310   \ifx#1\ReFiCh@Separator
311     \global\let#1\ReFiCh@NoFile
312   \fi
313   \AtVeryEndDocument{%
314     \begingroup
315     #3%
316     \edef\x{\ReFiCh@Checksum{#2}}%
317     \ifx\x\ReFiCh@Separator
318       \let\x\ReFiCh@NoFile
319     \fi
320     \ifx#1\x
321       \@PackageInfoNoLine{rerunfilecheck}{%
322         File ‘#2’ has not changed.\MessageBreak
323         Checksum: \x
324       }%
325     \else
326       \ifnum
327         \ReFiCh@IsAux#2\relax.aux\relax\@nil
328         \ifx#1\ReFiCh@NoFile 1\else 0\fi
329         \ifx\x\ReFiCh@AuxEmptyUnix 1%
330         \else
331           \ifx\x\ReFiCh@AuxEmptyDos 1\fi
332         \fi
333         =111 %
334         \@PackageInfoNoLine{rerunfilecheck}{%
335           File ‘#2’ is empty .aux file.\MessageBreak
336           Before: #1\MessageBreak
337           After: \space\x
338         }%
339       \else
340         \@PackageWarningNoLine{rerunfilecheck}{%

```



```

341         File '#2' has changed.%
342         \ifx\#4\%
343         \space Rerun%
344         \else
345         \MessageBreak
346         #4%
347         \fi
348     }%
349     \@PackageInfoNoLine{rerunfilecheck}{%
350     Checksums for '#2':\MessageBreak
351     Before: #1\MessageBreak
352     After: \space\x
353     }%
354 \fi
355 \fi
356 \endgroup
357 }%
358 }

359 \def\ReFiCh@IsAux#1.aux\relax#2\@nil{%
360 \ifx\hbox#2\hbox
361     0%
362 \else
363     1%
364 \fi
365 }

366 \def\ReFiCh@AuxEmptyUnix{A94A2480D3289E625EEA47CD1B285758;8}%
367 \@onelevel@sanitize\ReFiCh@AuxEmptyUnix

368 \def\ReFiCh@AuxEmptyDos{A62A15ECE803E2EBE94952FCC9933BC0;9}%
369 \@onelevel@sanitize\ReFiCh@AuxEmptyDos

370 \ReFiCh@AtEnd
371 \</package>

```

3 Test

3.1 Catcode checks for loading

```

372 <*test1>

373 \catcode'\{=1 %
374 \catcode'\}=2 %
375 \catcode'\#=6 %
376 \catcode'\@=11 %
377 \expandafter\ifx\csname count@\endcsname\relax
378 \countdef\count@=255 %
379 \fi
380 \expandafter\ifx\csname @gobble\endcsname\relax
381 \long\def\@gobble#1{}%
382 \fi
383 \expandafter\ifx\csname @firstofone\endcsname\relax
384 \long\def\@firstofone#1{#1}%
385 \fi
386 \expandafter\ifx\csname loop\endcsname\relax
387 \expandafter\@firstofone
388 \else
389 \expandafter\@gobble
390 \fi
391 {%
392 \def\loop#1\repeat{%
393 \def\body{#1}%
394 \iterate

```

```

395 }%
396 \def\iterate{%
397   \body
398   \let\next\iterate
399   \else
400   \let\next\relax
401   \fi
402   \next
403 }%
404 \let\repeat=\fi
405 }%
406 \def\RestoreCatcodes{}
407 \count@=0 %
408 \loop
409   \edef\RestoreCatcodes{%
410     \RestoreCatcodes
411     \catcode\the\count@=\the\catcode\count@\relax
412   }%
413   \ifnum\count@<255 %
414     \advance\count@ 1 %
415   \repeat
416
417 \def\RangeCatcodeInvalid#1#2{%
418   \count@=#1\relax
419   \loop
420     \catcode\count@=15 %
421     \ifnum\count@<#2\relax
422       \advance\count@ 1 %
423     \repeat
424 }
425 \expandafter\ifx\csname LoadCommand\endcsname\relax
426 \def\LoadCommand{\input rerunfilecheck.sty\relax}%
427 \fi
428 \def\Test{%
429   \RangeCatcodeInvalid{0}{47}%
430   \RangeCatcodeInvalid{58}{64}%
431   \RangeCatcodeInvalid{91}{96}%
432   \RangeCatcodeInvalid{123}{255}%
433   \catcode'\@=12 %
434   \catcode'\=0 %
435   \catcode'\{=1 %
436   \catcode'\}=2 %
437   \catcode'\#=6 %
438   \catcode'\[=12 %
439   \catcode'\]=12 %
440   \catcode'\%=14 %
441   \catcode'\ =10 %
442   \catcode13=5 %
443   \LoadCommand
444   \RestoreCatcodes
445 }
446 \Test
447 \csname @@end\endcsname
448 \end
449 </test1>

```

4 Installation

4.1 Download

Package. This package is available on CTAN¹:

¹<ftp://ftp.ctan.org/tex-archive/>

[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 T_EX Files” ([CTAN:tds/tds.pdf](#)). Directories with `texmf` in their name are usually organized this way.

4.2 Bundle installation

Unpacking. Unpack the `oberdiek.tds.zip` in the TDS tree (also known as `texmf` tree) of your choice. Example (linux):

```
unzip oberdiek.tds.zip -d ~/texmf
```

Script installation. Check the directory `TDS:scripts/oberdiek/` for scripts that need further installation steps. Package `attachfile2` comes with the Perl script `pdfatfi.pl` that should be installed in such a way that it can be called as `pdfatfi`. Example (linux):

```
chmod +x scripts/oberdiek/pdfatfi.pl
cp scripts/oberdiek/pdfatfi.pl /usr/local/bin/
```

4.3 Package installation

Unpacking. The `.dtx` file is a self-extracting `docstrip` archive. The files are extracted by running the `.dtx` through plain-T_EX:

```
tex rerunfilecheck.dtx
```

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

<code>rerunfilecheck.sty</code>	→ <code>tex/generic/oberdiek/rerunfilecheck.sty</code>
<code>rerunfilecheck.pdf</code>	→ <code>doc/latex/oberdiek/rerunfilecheck.pdf</code>
<code>rerunfilecheck-example.cfg</code>	→ <code>doc/latex/oberdiek/rerunfilecheck-example.cfg</code>
<code>test/rerunfilecheck-test1.tex</code>	→ <code>doc/latex/oberdiek/test/rerunfilecheck-test1.tex</code>
<code>rerunfilecheck.dtx</code>	→ <code>source/latex/oberdiek/rerunfilecheck.dtx</code>

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 T_EX distribution (teT_EX, miK_TE_X, ...) relies on file name databases, you must refresh these. For example, teT_EX 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 L^AT_EX. The .dtx chooses its action depending on the format:

plain-T_EX: Run docstrip and extract the files.

L^AT_EX: Generate the documentation.

If you insist on using L^AT_EX for docstrip (really, docstrip does not need L^AT_EX), then inform the autodetect routine about your intention:

```
latex \let\install=y\input{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 pdfL^AT_EX:

```
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.

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; numbers in roman refer to the code lines where the entry is used.

Symbols		
\#	375, 437	\@wrglossary 249, 267
\%	440	\@wrindex 205, 223
\@	376, 433	\[..... 438
\@PackageInfoNoLine		\] 342, 434
..... 140, 229, 273, 321, 334, 349		\{ 373, 435
\@PackageWarningNoLine		\} 374, 436
..... 169, 193, 237, 340		\] 439
\@bsphack	202, 220, 246, 264	
\@empty	192, 208, 226, 236, 252, 270	_ 441
\@firstofone	384, 387	
\@glossaryfile	243, 244, 256, 258, 262	
\@gobble	381, 389	A
\@ifundefined	182	\advance 414, 422
\@include	151, 152	\aftergroup 30
\@indexfile	199, 200, 212, 214, 218	\AtBeginDocument 160
\@nil	327, 359	\AtEndOfPackage 164
\@onelevel@sanitize	367, 369	\AtVeryEndDocument 313
\@sanitize	204, 222, 248, 266	
\@starttoc	178, 179	B
\@undefined	56	\begin 172
		\body 393, 397

C		M	
<code>\catcode</code>	7, 8, 9, 10, 11, 12, 13, 21, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 68, 69, 72, 73, 74, 75, 79, 80, 81, 82, 86, 88, 373, 374, 375, 376, 411, 420, 433, 434, 435, 436, 437, 438, 439, 440, 441, 442	<code>\makeglossary</code>	236, 239, 252, 254, 270, 274
<code>\closeout</code>	184, 214, 258	<code>\makeindex</code>	192, 195, 208, 210, 211, 226, 230
<code>\count@</code>	378, 407, 411, 413, 414, 418, 420, 421, 422	<code>\MessageBreak</code>	141, 142, 143, 170, 171, 194, 238, 322, 335, 336, 345, 350, 351
<code>\countdef</code>	378	N	
<code>\csname</code>	14, 22, 48, 64, 71, 139, 184, 288, 306, 377, 380, 383, 386, 425, 447	<code>\newcommand</code>	111, 146, 302
D		<code>\newwrite</code>	199, 212, 243, 256
<code>\DeclareBoolOption</code>	114, 115, 116, 117, 118	<code>\next</code>	398, 400, 402
<code>\define@key</code>	119	O	
<code>\DisableKeyvalOption</code>	131	<code>\openout</code>	200, 218, 244, 262
E		P	
<code>\empty</code>	17, 18	<code>\PackageInfo</code>	27
<code>\end</code>	448	<code>\pdf@filemdfivesum</code>	290, 294
<code>\endcsname</code>	14, 22, 48, 64, 71, 139, 184, 288, 306, 377, 380, 383, 386, 425, 447	<code>\pdf@filesize</code>	296
<code>\endinput</code>	30, 105	<code>\pdfmdfivesum</code>	141
G		<code>\ProcessKeyvalOptions</code>	129
<code>\g@addto@macro</code>	105	<code>\ProvidesFile</code>	2
<code>\glossary</code>	245, 263	<code>\ProvidesPackage</code>	19, 65
H		R	
<code>\hbox</code>	360	<code>\RangeCatcodeInvalid</code>	417, 429, 430, 431, 432
I		<code>\ReFiCh@@RerunFileCheck</code> ...	306, 308
<code>\if@filesw</code>	153, 168, 180	<code>\ReFiCh@AtEnd</code> ...	84, 85, 105, 148, 370
<code>\ifnum</code>	326, 413, 421	<code>\ReFiCh@AuxEmptyDos</code> ...	331, 368, 369
<code>\ifReFiCh@glossary</code>	235	<code>\ReFiCh@AuxEmptyUnix</code> ..	329, 366, 367
<code>\ifReFiCh@index</code>	191	<code>\ReFiCh@Checksum</code>	287, 309, 316
<code>\ifReFiCh@mainaux</code>	159, 163	<code>\ReFiCh@DisableOption</code>	130, 279, 280, 281, 282, 283, 284
<code>\ifReFiCh@partaux</code>	150	<code>\ReFiCh@IsAux</code>	327, 359
<code>\ifReFiCh@starttoc</code>	177	<code>\ReFiCh@mainauxfalse</code>	161
<code>\ifx</code>	15, 18, 22, 48, 56, 59, 139, 192, 210, 236, 254, 288, 310, 317, 320, 328, 329, 331, 342, 360, 377, 380, 383, 386, 425	<code>\ReFiCh@NoFile</code>	299, 300, 311, 318, 328
<code>\immediate</code>	24, 50, 184, 200, 214, 218, 244, 258, 262	<code>\ReFiCh@org@include</code>	151, 156
<code>\index</code>	201, 219	<code>\ReFiCh@org@starttoc</code>	178, 188
<code>\input</code>	426	<code>\ReFiCh@RerunFileCheck</code> ...	303, 305
<code>\InputIfFileExists</code>	128	<code>\ReFiCh@Separator</code> ..	295, 299, 310, 317
<code>\iterate</code>	394, 396, 398	<code>\ReFiCh@temp</code> ..	198, 210, 242, 254, 255
J		<code>\renewcommand</code>	147
<code>\jobname</code> ..	165, 181, 200, 207, 213, 218, 225, 244, 251, 257, 262, 269	<code>\repeat</code>	392, 404, 415, 423
L		<code>\RequirePackage</code> ..	106, 136, 137, 285, 286
<code>\LoadCommand</code>	426, 443	<code>\RerunFileCheck</code>	2, 146, 154, 165, 181, 213, 257, 302
<code>\loop</code>	392, 408, 419	<code>\RerunFileCheckSetup</code> ..	2, 3, 111, 120, 147
T		<code>\RestoreCatcodes</code> ..	406, 409, 410, 444
<code>\Test</code>	428, 446	S	
<code>\the</code>	72, 73, 74, 75, 86, 411	<code>\setkeys</code>	112
<code>\TMP@EnsureCode</code>	83, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104	<code>\SetupKeyvalOptions</code>	107
<code>\typeout</code>	207, 225, 251, 269	<code>\space</code>	141, 195, 239, 337, 343, 352

U		X	
\UniqueCounterCall 303	\x 14, 15, 18, 23, 27,
\UniqueCounterNew 301		29, 49, 54, 64, 70, 78, 316, 317,
			318, 320, 323, 329, 331, 337, 352
W			
\write 24, 50		