

The infwarerr package

Heiko Oberdiek
<oberdiek@uni-freiburg.de>

2007/09/09 v1.2

Abstract

This package provides a complete set of macros for informations, warnings and error messages with support for plain-TeX.

Contents

1 Documentation	1
2 Implementation	2
2.1 Reload check and package identification	2
2.2 Catcodes	3
2.3 L ^A T _E X detection	3
2.4 Version for L ^A T _E X	4
2.5 Version for plain-TeX	5
3 Test	7
3.1 Catcode checks for loading	7
3.2 Macro tests	9
4 Installation	12
4.1 Download	12
4.2 Bundle installation	12
4.3 Package installation	13
4.4 Refresh file name databases	13
4.5 Some details for the interested	13
5 History	14
[2007/04/26 v1.0]	14
[2007/06/14 v1.1]	14
[2007/09/09 v1.2]	14
6 Index	14

1 Documentation

Package infwarerr defines the following set of commands, given as pseudo regular expression:

`\@(Package|Class)(Info|InfoNoLine|Warning|WarningNoLine|Error)`

The first argument is the package or class in question. The second argument contains the message text. `\MessageBreak` is supported. In case of error messages, the third argument contains the help text. Supported are L^AT_EX's message texts `\@ehc`:

Try typing <return> to proceed
 If that doesn't work, type X <return> to quit.

and \@ehd:

You're in trouble here. Try typing <return> to proceed
 If that doesn't work, type X <return> to quit.

The package is intended for macro writers that want to write code that works for both L^AT_EX and plain-T_EX. If L^AT_EX is detected, then the macros of this package will only be wrappers for L^AT_EX's macros. Otherwise L^AT_EX's macros will be mimicked for use with plain-T_EX.

2 Implementation

1 (*package)

2.1 Reload check and package identification

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

```

2 \begingroup
3 \catcode44 12 % ,
4 \catcode45 12 % -
5 \catcode46 12 % .
6 \catcode58 12 % :
7 \catcode64 11 % @
8 \expandafter\let\expandafter\x\csname ver@infwarerr.sty\endcsname
9 \ifcase 0%
10 \ifx\x\relax % plain
11 \else
12 \ifx\x\empty % LaTeX
13 \else
14 1%
15 \fi
16 \fi
17 \else
18 \catcode35 6 % #
19 \catcode123 1 % {
20 \catcode125 2 % }
21 \expandafter\ifx\csname PackageInfo\endcsname\relax
22 \def\x#1#2{%
23 \immediate\write-1{Package #1 Info: #2.}%
24 }%
25 \else
26 \def\x#1#2{\PackageInfo{#1}{#2, stopped}}%
27 \fi
28 \x{infwarerr}{The package is already loaded}%
29 \endgroup
30 \expandafter\endinput
31 \fi
32 \endgroup

```

Package identification:

```

33 \begingroup
34 \catcode35 6 % #
35 \catcode40 12 % (
36 \catcode41 12 % )
37 \catcode44 12 % ,
38 \catcode45 12 % -
39 \catcode46 12 % .
40 \catcode47 12 % /
41 \catcode58 12 % :
42 \catcode64 11 % @

```

```

43 \catcode123 1 % {
44 \catcode125 2 % }
45 \expandafter\ifx\csname ProvidesPackage\endcsname\relax
46   \def\x#1#2#3[#4]{\endgroup
47     \immediate\write-1{Package: #3 #4}%
48     \xdef#1{#4}%
49   }%
50 \else
51   \def\x#1#2[#3]{\endgroup
52     #2[{#3}]%
53     \ifx#1\relax
54       \xdef#1{#3}%
55     \fi
56   }%
57 \fi
58 \expandafter\x\csname ver@infwarerr.sty\endcsname
59 \ProvidesPackage{infwarerr}%
60 [2007/09/09 v1.2 Providing info/warning/message (H0)]

```

2.2 Catcodes

```

61 \begingroup
62 \catcode123 1 % {
63 \catcode125 2 % }
64 \def\x{\endgroup
65   \expandafter\edef\csname InfWarErr@AtEnd\endcsname{%
66     \catcode35 \the\catcode35\relax
67     \catcode64 \the\catcode64\relax
68     \catcode123 \the\catcode123\relax
69     \catcode125 \the\catcode125\relax
70   }%
71 }%
72 \x
73 \catcode35 6 % #
74 \catcode64 11 % @
75 \catcode123 1 % {
76 \catcode125 2 % }
77 \def\TMP@EnsureCode#1#2{%
78   \edef\InfWarErr@AtEnd{%
79     \InfWarErr@AtEnd
80     \catcode#1 \the\catcode#1\relax
81   }%
82   \catcode#1 #2\relax
83 }
84 \TMP@EnsureCode{10}{12}% ^^J
85 \TMP@EnsureCode{39}{12}% '
86 \TMP@EnsureCode{40}{12}% (
87 \TMP@EnsureCode{41}{12}% )
88 \TMP@EnsureCode{44}{12}% ,
89 \TMP@EnsureCode{45}{12}% -
90 \TMP@EnsureCode{46}{12}% .
91 \TMP@EnsureCode{58}{12}% :
92 \TMP@EnsureCode{60}{12}% <
93 \TMP@EnsureCode{61}{12}% =
94 \TMP@EnsureCode{62}{12}% >
95 \TMP@EnsureCode{94}{7}% ^

```

2.3 L^AT_EX detection

```

96 \begingroup
97 \let\x=Y%
98 \def\check#1{%
99   \@check{Generic}{#1}%

```

```

100     \@check{Package}{#1}%
101     \@check{Class}{#1}%
102 }%
103 \def\@check#1#2{%
104     \expandafter\ifx\csname #1#2\endcsname\relax
105         \let\x=N%
106         \fi
107 }%
108 \@check{Info}%
109 \@check{Warning}%
110 \@check{Error}%
111 \@check{Package}{WarningNoLine}%
112 \@check{Class}{WarningNoLine}%
113 \@check{}{@ehc}%
114 \@check{}{@ehd}%
115 \expandafter\endgroup
116 \ifx\x Y%

```

2.4 Version for L^AT_EX

\@PackageInfo

```
117 \let\@PackageInfo\PackageInfo
```

\@PackageInfoNoLine

```

118 \def\@PackageInfoNoLine#1#2{%
119     \PackageInfo{#1}{#2\@gobble}%
120 }%

```

\@ClassInfo

```
121 \let\@ClassInfo\ClassInfo
```

\@ClassInfoNoLine

```

122 \def\@ClassInfoNoLine#1#2{%
123     \ClassInfo{#1}{#2\@gobble}%
124 }%

```

\@PackageWarning

```
125 \let\@PackageWarning\PackageWarning
```

\@PackageWarningNoLine

```
126 \let\@PackageWarningNoLine\PackageWarningNoLine
```

\@ClassWarning

```
127 \let\@ClassWarning\ClassWarning
```

\@ClassWarningNoLine

```
128 \let\@ClassWarningNoLine\ClassWarningNoLine
```

\@PackageError

```
129 \let\@PackageError\PackageError
```

\@ClassError

```
130 \let\@ClassError\ClassError
```

```
131 \InfWarErr@AtEnd
```

```
132 \expandafter\endinput
```

```
133 \fi
```

2.5 Version for plain-TeX

`\InfWarErr@online`

```
134 \begingroup\expandafter\endgroup
135 \ifcase\expandafter\ifx\csname inputlineno\endcsname\relax
136     0 %
137     \else
138     \ifnum\inputlineno<0 %
139         0 %
140     \else
141         1 %
142     \fi
143     \fi
144 \def\InfWarErr@online{}%
145 \else
146 \def\InfWarErr@online{ on input line \the\inputlineno}%
147 \fi
```

`\InfWarErr@protected`

```
148 \begingroup\expandafter\expandafter\expandafter\endgroup
149 \expandafter\ifx\csname protected\endcsname\relax
150 \def\InfWarErr@protected{}%
151 \else
152 \let\InfWarErr@protected\protected
153 \fi
```

`\InfWarErr@unused`

```
154 \chardef\InfWarErr@unused=16 %
```

`\InfWarErr@ehc`

```
155 \def\InfWarErr@ehc{%
156   Try typing \space <return> \space to proceed.\MessageBreak
157   If that doesn't work, type \space X <return> \space to quit.%
158 }
```

`\InfWarErr@ehd`

```
159 \def\InfWarErr@ehd{%
160   You're in trouble here. \space\@ehc
161 }
```

`\InfWarErr@GenericInfo`

```
162 \InfWarErr@protected\def\InfWarErr@GenericInfo#1#2{%
163   \begingroup
164     \let\on@line\InfWarErr@online
165     \long\def\@gobble##1{%
166       \def\space{ }%
167       \def\@spaces{\space\space\space\space}%
168       \def\MessageBreak{^^J#1}%
169       \newlinechar=10 %
170       \immediate\write-1{#2\on@line.}%
171     \endgroup
172 }
```

`\InfWarErr@GenericWarning`

```
173 \InfWarErr@protected\def\InfWarErr@GenericWarning#1#2{%
174   \begingroup
175     \let\on@line\InfWarErr@online
176     \long\def\@gobble##1{%
177       \def\space{ }%
178       \def\@spaces{\space\space\space\space}%
179       \def\MessageBreak{^^J#1}%
180     \endgroup
181 }
```

```

180     \newlinechar=10 %
181     \immediate\write\InfWarErr@unused{^^J#2\on@line.^^J}%
182 \endgroup
183 }

\InfWarErr@GenericError Old TeX versions below 3.141 where ^^J is not working as newline in error message
and help texts are not supported.

184 \InfWarErr@protected\def\InfWarErr@GenericError#1#2#3#4{%
185   \begingroup
186     \immediate\write\InfWarErr@unused{}%
187     \newlinechar=10 %
188     \def\space{ }%
189     \def\@spaces{\space\space\space\space}%
190     \let\@ehc\InfWarErr@ehc
191     \let\@ehd\InfWarErr@ehd
192     \let\on@line\InfWarErr@online
193     \def\MessageBreak{^^J}%
194     \edef\InfWarErr@help{\noexpand\errhelp{#4}}%
195     \InfWarErr@help
196     \def\MessageBreak{^^J#1}%
197     \errmessage{%
198       #2.^^J^^J%
199       #3^^J%
200       Type \space H <return> \space for immediate help.^^J ...^^J%
201     }%
202   \endgroup
203 }

\@PackageInfo

204 \def\@PackageInfo#1#2{%
205   \InfWarErr@GenericInfo{%
206     (#1) \@spaces\@spaces\@spaces
207   }{%
208     Package #1 Info: #2%
209   }%
210 }

\@PackageInfoNoLine

211 \def\@PackageInfoNoLine#1#2{%
212   \@PackageInfo{#1}{#2@gobble}%
213 }

\@ClassInfo

214 \def\@ClassInfo#1#2{%
215   \InfWarErr@GenericInfo{%
216     (#1) \space\space\@spaces\@spaces
217   }{%
218     Class #1 Info: #2%
219   }%
220 }

\@ClassInfoNoLine

221 \def\@ClassInfoNoLine#1#2{%
222   \@ClassInfo{#1}{#2@gobble}%
223 }

\@PackageWarning

224 \def\@PackageWarning#1#2{%
225   \InfWarErr@GenericWarning{%
226     (#1)\@spaces\@spaces\@spaces\@spaces
227   }{%

```

```

228     Package #1 Warning: #2%
229 }%
230 }

\@PackageWarningNoLine

231 \def\@PackageWarningNoLine#1#2{%
232   \@PackageWarning{#1}{#2\@gobble}%
233 }

\@ClassWarning

234 \def\@ClassWarning#1#2{%
235   \InfWarErr@GenericWarning{%
236     (#1) \space\@spaces\@spaces\@spaces
237   }{%
238     Class #1 Warning: #2%
239   }%
240 }

\@ClassWarningNoLine

241 \def\@ClassWarningNoLine#1#2{%
242   \@ClassWarning{#1}{#2\@gobble}%
243 }

\@PackageError

244 \def\@PackageError#1#2#3{%
245   \InfWarErr@GenericError{%
246     (#1)\@spaces\@spaces\@spaces\@spaces
247   }{%
248     Package #1 Error: #2%
249   }{%
250     See the #1 package documentation for explanation.%
251   }{#3}%
252 }

\@ClassError

253 \def\@ClassError#1#2#3{%
254   \InfWarErr@GenericError{%
255     (#1) \space\@spaces\@spaces\@spaces
256   }{%
257     Class #1 Error: #2%
258   }{%
259     See the #1 class documentation for explanation.%
260   }{#3}%
261 }

262 \InfWarErr@AtEnd
263 </package>

```

3 Test

3.1 Catcode checks for loading

```

264 (*test1)
265 \catcode'\{=1 %
266 \catcode'\}=2 %
267 \catcode'\#=6 %
268 \catcode'\@=11 %
269 \expandafter\ifx\csname count@\endcsname\relax
270   \countdef\count@=255 %
271 \fi

```

```

272 \expandafter\ifx\csname @gobble\endcsname\relax
273   \long\def\@gobble#1{}%
274 \fi
275 \expandafter\ifx\csname @firstofone\endcsname\relax
276   \long\def\@firstofone#1{#1}%
277 \fi
278 \expandafter\ifx\csname loop\endcsname\relax
279   \expandafter\@firstofone
280 \else
281   \expandafter\@gobble
282 \fi
283 {%
284   \def\loop#1\repeat{%
285     \def\body{#1}%
286     \iterate
287   }%
288   \def\iterate{%
289     \body
290     \let\next\iterate
291   \else
292     \let\next\relax
293   \fi
294   \next
295   }%
296   \let\repeat=\fi
297 }%
298 \def\RestoreCatcodes{}
299 \count@=0 %
300 \loop
301   \edef\RestoreCatcodes{%
302     \RestoreCatcodes
303     \catcode\the\count@=\the\catcode\count@\relax
304   }%
305   \ifnum\count@<255 %
306     \advance\count@ 1 %
307 \repeat
308
309 \def\RangeCatcodeInvalid#1#2{%
310   \count@=#1\relax
311   \loop
312     \catcode\count@=15 %
313     \ifnum\count@<#2\relax
314       \advance\count@ 1 %
315     \repeat
316 }
317 \expandafter\ifx\csname LoadCommand\endcsname\relax
318   \def\LoadCommand{\input infwarerr.sty\relax}%
319 \fi
320 \def\Test{%
321   \RangeCatcodeInvalid{0}{47}%
322   \RangeCatcodeInvalid{58}{64}%
323   \RangeCatcodeInvalid{91}{96}%
324   \RangeCatcodeInvalid{123}{255}%
325   \catcode'\@=12 %
326   \catcode'\=0 %
327   \catcode'\{=1 %
328   \catcode'\}=2 %
329   \catcode'\#=6 %
330   \catcode'\[=12 %
331   \catcode'\]=12 %
332   \catcode'\%=14 %
333   \catcode'\ =10 %

```

```

334 \catcode13=5 %
335 \LoadCommand
336 \RestoreCatcodes
337 }
338 \Test
339 \csname @@end\endcsname
340 \end
341 \test1\

```

3.2 Macro tests

```

342 (*test2\
343 \let\PackageInfo\relax
344 \input infwarerr.sty\relax
345 \let\OrgWrite\write
346 \chardef\WriteUnused=16 %
347 \def\msg#{\immediate\OrgWrite\WriteUnused}
348 \msg{File: infwarerr-test2.tex 2007/09/09 v1.2 Test file for plain-Tex}
349 \catcode64=11 %
350 \test2\

351 (*test3\
352 \NeedsTeXFormat{LaTeX2e}
353 \ProvidesFile{infwarerr-test3.tex}[2007/09/09 v1.2 Test file for LaTeX]
354 \RequirePackage{infwarerr}[2007/09/09]
355 \makeatletter
356 \let\OrgWrite\write
357 \chardef\WriteUnused=\@unused
358 \def\msg#{\immediate\OrgWrite\WriteUnused}
359 \test3\

360 (*test2 | test3\
361 \def\pkgname{PACKAGE}
362 \def\clsname{CLASS}
363 \def\msgtext{MESSAGE\MessageBreak SECOND LINE}
364 \def\hlptext{HELP\MessageBreak SECOND LINE}
365
366 \let\OrgWrite\write
367 \let\OrgErrorMessage\errmessage
368
369 \def\write#1#{%
370   \@write{#1}%
371 }
372 \def\@write#1#2{%
373   \OrgWrite#1{#2}%
374   \xdef\WriteNum{\number#1}%
375   \xdef\WriteText{#2}%
376 }
377 \def\TestFailed#1#2{%
378   \OrgErrorMessage{Test (\string#1) failed: #2}%
379 }
380 \def\TestWrite#1#2#3{%
381   \ifnum\WriteNum=#2\relax
382   \else
383     \TestFailed#1{write stream number: \WriteNum <> #2}%
384   \fi
385   \begingroup
386     \let~\space
387     \edef\TestString{#3}%
388     \ifx\WriteText\TestString
389     \else
390       \msg{}%
391       \msg{[\WriteText]}%
392       \msg{<>}%

```

```

393     \msg{[#3]}% hash-ok
394     \TestFailed#1{write text}%
395     \fi
396 \endgroup
397 }
398 \newtoks\ErrHelpToks
399 \def\errmessage#1{%
400   \global\ErrHelpToks=\expandafter{\the\errhelp}%
401   \msg{***** error message *****}%
402   \msg{! #1}%
403   \msg{***** help message *****}%
404   \msg{\the\errhelp}%
405   \msg{*****}%
406   \xdef\ErrorMessage{! #1}%
407 }
408 \def\TestInfo{%
409   \@PackageInfo\pkgname\msgtext
410   \TestWrite\@PackageInfo{-1}{%
411     Package PACKAGE Info: MESSAGE^^J%
412     (PACKAGE)~~~~~SECOND LINE on input line \the\inputlineno.%
413   }%
414   \@PackageInfoNoLine\pkgname\msgtext
415   \TestWrite\@PackageInfoNoLine{-1}{%
416     Package PACKAGE Info: MESSAGE^^J%
417     (PACKAGE)~~~~~SECOND LINE.%
418   }%
419   \@ClassInfo\clsname\msgtext
420   \TestWrite\@ClassInfo{-1}{%
421     Class CLASS Info: MESSAGE^^J%
422     (CLASS)~~~~~SECOND LINE on input line \the\inputlineno.%
423   }%
424   \@ClassInfoNoLine\clsname\msgtext
425   \TestWrite\@ClassInfoNoLine{-1}{%
426     Class CLASS Info: MESSAGE^^J%
427     (CLASS)~~~~~SECOND LINE.%
428   }%
429 }
430 \def\TestWarning{%
431   \@PackageWarning\pkgname\msgtext
432   \TestWrite\@PackageWarning\WriteUnused{%
433     ^^J%
434     Package PACKAGE Warning: MESSAGE^^J%
435     (PACKAGE)~~~~~SECOND LINE on input line \the\inputlineno.%
436     ^^J%
437   }%
438   \@PackageWarningNoLine\pkgname\msgtext
439   \TestWrite\@PackageWarningNoLine\WriteUnused{%
440     ^^J%
441     Package PACKAGE Warning: MESSAGE^^J%
442     (PACKAGE)~~~~~SECOND LINE.%
443     ^^J%
444   }%
445   \@ClassWarning\clsname\msgtext
446   \TestWrite\@ClassWarning\WriteUnused{%
447     ^^J%
448     Class CLASS Warning: MESSAGE^^J%
449     (CLASS)~~~~~SECOND LINE on input line \the\inputlineno.%
450     ^^J%
451   }%
452   \@ClassWarningNoLine\clsname\msgtext
453   \TestWrite\@ClassWarningNoLine\WriteUnused{%
454     ^^J%

```

```

455     Class CLASS Warning: MESSAGE^^J%
456     (CLASS)~~~~~SECOND LINE.%
457     ^^J%
458 }%
459 }
460 \def\TestError{%
461 (test3) \let\write\OrgWrite
462 (test3) \def\plainedots{}
463 (test2) \def\plainedots{.^^J ...^^J}
464 \@PackageError\pkgname\msgtext\hlptext
465 \@TestError\@PackageError{%
466     ! Package PACKAGE Error: MESSAGE^^J%
467     (PACKAGE)~~~~~SECOND LINE.^^J%
468     ^^J%
469     See the PACKAGE package documentation for explanation.^^J%
470     Type~~H <return>~~for immediate help\plainedots
471 }{%
472     HELP^^J%
473     SECOND LINE%
474 }%
475 \@ClassError\clsname\msgtext\hlptext
476 \@TestError\@ClassError{%
477     ! Class CLASS Error: MESSAGE^^J%
478     (CLASS)~~~~~SECOND LINE.^^J%
479     ^^J%
480     See the CLASS class documentation for explanation.^^J%
481     Type~~H <return>~~for immediate help\plainedots
482 }{%
483     HELP^^J%
484     SECOND LINE%
485 }%
486 \@PackageError\pkgname\msgtext\@ehc
487 \@TestError\@PackageError{%
488     ! Package PACKAGE Error: MESSAGE^^J%
489     (PACKAGE)~~~~~SECOND LINE.^^J%
490     ^^J%
491     See the PACKAGE package documentation for explanation.^^J%
492     Type~~H <return>~~for immediate help\plainedots
493 }{%
494     Try typing~~<return>~~to proceed.^^J%
495     If that doesn't work, type~~X <return>~~to quit.%
496 }%
497 \@PackageError\pkgname\msgtext\@ehd
498 \@TestError\@PackageError{%
499     ! Package PACKAGE Error: MESSAGE^^J%
500     (PACKAGE)~~~~~SECOND LINE.^^J%
501     ^^J%
502     See the PACKAGE package documentation for explanation.^^J%
503     Type~~H <return>~~for immediate help\plainedots
504 }{%
505     You're in trouble here.~~Try typing~~<return>~~to proceed.^^J%
506     If that doesn't work, type~~X <return>~~to quit.%
507 }%
508 }
509 \def\@TestError#1#2#3{%
510     \begingroup
511     \let~\space
512     \edef\x{#2}%
513 (test3) \@onelevel@sanitize\x
514 (test3) \@onelevel@sanitize\ErrMsgage
515     \ifx\x\ErrMsgage
516     \else

```

```

517     \msg{}%
518     \msg{[\ErrMessage]}%
519     \msg{<>}%
520     \msg{[\x]}%
521     \TestFailed#1{errmessage text}%
522 \fi
523 \edef\x{\the\ErrHelpToks}%
524 \edef\y{#3}%
525 \ifx\x\y
526 \else
527     \msg{}%
528     \msg{[\the\ErrHelpToks]}%
529     \msg{<>}%
530     \msg{[\y]}%
531     \TestFailed#1{errhelp text}%
532 \fi
533 \endgroup
534 }
535 \TestInfo
536 \TestWarning
537 \TestError
538 </test2 | test3>
539 <test2>\end
540 <test3>\@@end

```

4 Installation

4.1 Download

Package. This package is available on CTAN¹:

[CTAN:macros/latex/contrib/oberdiek/infwarerr.dtx](#) The source file.

[CTAN:macros/latex/contrib/oberdiek/infwarerr.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/
```

¹<http://ftp.ctan.org/tex-archive/>

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 infwarerr.dtx
```

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

```
infwarerr.sty      → tex/generic/oberdiek/infwarerr.sty
infwarerr.pdf      → doc/latex/oberdiek/infwarerr.pdf
test/infwarerr-test1.tex → doc/latex/oberdiek/test/infwarerr-test1.tex
test/infwarerr-test2.tex → doc/latex/oberdiek/test/infwarerr-test2.tex
test/infwarerr-test3.tex → doc/latex/oberdiek/test/infwarerr-test3.tex
infwarerr.dtx      → source/latex/oberdiek/infwarerr.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 (`te \TeX` , `mik \TeX` , ...) relies on file name databases, you must refresh these. For example, `te \TeX` users run `texhash` or `mktextlsr`.

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 infwarerr.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{infwarerr.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 `pdf \LaTeX` :

```
pdflatex infwarerr.dtx
makeindex -s gind.ist infwarerr.idx
pdflatex infwarerr.dtx
makeindex -s gind.ist infwarerr.idx
pdflatex infwarerr.dtx
```

5 History

[2007/04/26 v1.0]

- First version.

[2007/06/14 v1.1]

- Small fixes.

[2007/09/09 v1.2]

- Catcode section extended.

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	B
\# 267, 329	\body 285, 289
\% 332	
\@ 268, 325	C
\@end 540	\catcode 3, 4, 5,
\@ClassError 130, 253, 475, 476	6, 7, 18, 19, 20, 34, 35, 36, 37,
\@ClassInfo ... 121, 214, 222, 419, 420	38, 39, 40, 41, 42, 43, 44, 62,
\@ClassInfoNoLine . 122, 221, 424, 425	63, 66, 67, 68, 69, 73, 74, 75,
\@ClassWarning 127, 234, 242, 445, 446	76, 80, 82, 265, 266, 267, 268,
\@ClassWarningNoLine 128, 241, 452, 453	303, 312, 325, 326, 327, 328,
\@PackageError 129,	329, 330, 331, 332, 333, 334, 349
244, 464, 465, 486, 487, 497, 498	\chardef 154, 346, 357
\@PackageInfo . 117, 204, 212, 409, 410	\check 98, 108, 109, 110
\@PackageInfoNoLine 118, 211, 414, 415	\ClassError 130
\@PackageWarning 125, 224, 232, 431, 432	\ClassInfo 121, 123
\@PackageWarningNoLine 126, 231, 438, 439	\ClassWarning 127
\@TestError ... 465, 476, 487, 498, 509	\ClassWarningNoLine 128
\@check 99,	\clsname .. 362, 419, 424, 445, 452, 475
100, 101, 103, 111, 112, 113, 114	\count@ 270, 299,
\@ehc 160, 190, 486	303, 305, 306, 310, 312, 313, 314
\@ehd 191, 497	\countdef 270
\@firstofone 276, 279	\csname . 8, 21, 45, 58, 65, 104, 135,
\@gobble 119, 123, 165,	149, 269, 272, 275, 278, 317, 339
176, 212, 222, 232, 242, 273, 281	E
\@onelevel@sanitize 513, 514	\empty 12
\@spaces 167, 178,	\end 340, 539
189, 206, 216, 226, 236, 246, 255	\endcsname 8, 21, 45, 58, 65, 104, 135,
\@unused 357	149, 269, 272, 275, 278, 317, 339
\@write 370, 372	\endinput 30, 132
\[..... 330	\errhelp 194, 400, 404
\[..... 326	\ErrHelpToks 398, 400, 523, 528
\{ 265, 327	\ErrorMessage 406, 514, 515, 518
\} 266, 328	\errmessage 197, 367, 399
\] 331	H
	\hlptext 364, 464, 475
_ 333	I
A	\ifcase 9, 135
\advance 306, 314	\ifnum 138, 305, 313, 381

<code>\ifx</code>	10, 12, 21, 45, 53, 104, 116, 135, 149, 269, 272, 275, 278, 317, 388, 515, 525
<code>\immediate</code>	23, 47, 170, 181, 186, 347, 358
<code>\InfWarErr@AtEnd</code>	78, 79, 131, 262
<code>\InfWarErr@ehc</code>	155, 190
<code>\InfWarErr@ehd</code>	159, 191
<code>\InfWarErr@GenericError</code>	184, 245, 254
<code>\InfWarErr@GenericInfo</code>	162, 205, 215
<code>\InfWarErr@GenericWarning</code>	173, 225, 235
<code>\InfWarErr@help</code>	194, 195
<code>\InfWarErr@online</code> . .	134, 164, 175, 192
<code>\InfWarErr@protected</code>	148, 162, 173, 184
<code>\InfWarErr@unused</code>	154, 181, 186
<code>\input</code>	318, 344
<code>\inputlineno</code>	138, 146, 412, 422, 435, 449
<code>\iterate</code>	286, 288, 290
L	
<code>\LoadCommand</code>	318, 335
<code>\loop</code>	284, 300, 311
M	
<code>\makeatletter</code>	355
<code>\MessageBreak</code>	156, 168, 179, 193, 196, 363, 364
<code>\msg</code>	347, 348, 358, 390, 391, 392, 393, 401, 402, 403, 404, 405, 517, 518, 519, 520, 527, 528, 529, 530
<code>\msgtext</code>	363, 409, 414, 419, 424, 431, 438, 445, 452, 464, 475, 486, 497
N	
<code>\NeedsTeXFormat</code>	352
<code>\newlinechar</code>	169, 180, 187
<code>\newtoks</code>	398
<code>\next</code>	290, 292, 294
<code>\number</code>	374
O	
<code>\on@line</code>	164, 170, 175, 181, 192
<code>\OrgErrMessage</code>	367, 378
<code>\OrgWrite</code>	345, 347, 356, 358, 366, 373, 461
P	
<code>\PackageError</code>	129
<code>\PackageInfo</code>	26, 117, 119, 343
<code>\PackageWarning</code>	125
<code>\PackageWarningNoLine</code>	126
<code>\pkgname</code>	361, 409, 414, 431, 438, 464, 486, 497
<code>\plaindots</code>	462, 463, 470, 481, 492, 503
<code>\protected</code>	152
<code>\ProvidesFile</code>	353
<code>\ProvidesPackage</code>	59
R	
<code>\RangeCatcodeInvalid</code>	309, 321, 322, 323, 324
<code>\repeat</code>	284, 296, 307, 315
<code>\RequirePackage</code>	354
<code>\RestoreCatcodes</code> . .	298, 301, 302, 336
S	
<code>\space</code>	156, 157, 160, 166, 167, 177, 178, 188, 189, 200, 216, 236, 255, 386, 511
T	
<code>\Test</code>	320, 338
<code>\TestError</code>	460, 537
<code>\TestFailed</code> . . .	377, 383, 394, 521, 531
<code>\TestInfo</code>	408, 535
<code>\TestString</code>	387, 388
<code>\TestWarning</code>	430, 536
<code>\TestWrite</code>	380, 410, 415, 420, 425, 432, 439, 446, 453
<code>\the</code>	66, 67, 68, 69, 80, 146, 303, 400, 404, 412, 422, 435, 449, 523, 528
<code>\TMP@EnsureCode</code>	77, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95
W	
<code>\write</code>	23, 47, 170, 181, 186, 345, 356, 366, 369, 461
<code>\WriteNum</code>	374, 381, 383
<code>\WriteText</code>	375, 388, 391
<code>\WriteUnused</code>	346, 347, 357, 358, 432, 439, 446, 453
X	
<code>\x</code>	8, 10, 12, 22, 26, 28, 46, 51, 58, 64, 72, 97, 105, 116, 512, 513, 515, 520, 523, 525
Y	
<code>\y</code>	524, 525, 530