

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\@undefined
54 \xdef#1{#3}%
55 \fi
56 \ifx#1\relax
57 \xdef#1{#3}%
58 \fi
59 }%
60 \fi
61 \expandafter\x\csname ver@infwarerr.sty\endcsname
62 \ProvidesPackage{infwarerr}%
63 [2007/09/09 v1.2 Providing info/warning/message (H0)]

```

2.2 Catcodes

```

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

```

2.3 L^AT_EX detection

```

99 \begingroup

```

```

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

```

2.4 Version for L^AT_EX

\@PackageInfo

```
120 \let\@PackageInfo\PackageInfo
```

\@PackageInfoNoLine

```

121 \def\@PackageInfoNoLine#1#2{%
122   \PackageInfo{#1}{#2@gobble}%
123 }%

```

\@ClassInfo

```
124 \let\@ClassInfo\ClassInfo
```

\@ClassInfoNoLine

```

125 \def\@ClassInfoNoLine#1#2{%
126   \ClassInfo{#1}{#2@gobble}%
127 }%

```

\@PackageWarning

```
128 \let\@PackageWarning\PackageWarning
```

\@PackageWarningNoLine

```
129 \let\@PackageWarningNoLine\PackageWarningNoLine
```

\@ClassWarning

```
130 \let\@ClassWarning\ClassWarning
```

\@ClassWarningNoLine

```
131 \let\@ClassWarningNoLine\ClassWarningNoLine
```

\@PackageError

```
132 \let\@PackageError\PackageError
```

\@ClassError

```
133 \let\@ClassError\ClassError
```

```
134 \InfWarErr@AtEnd
```

```
135 \expandafter\endinput
```

```
136 \fi
```

2.5 Version for plain-TeX

`\InfWarErr@online`

```

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

```

`\InfWarErr@protected`

```

151 \begingroup\expandafter\expandafter\expandafter\endgroup
152 \expandafter\ifx\csname protected\endcsname\relax
153 \def\InfWarErr@protected{}%
154 \else
155 \let\InfWarErr@protected\protected
156 \fi

```

`\InfWarErr@unused`

```

157 \chardef\InfWarErr@unused=16 %

```

`\InfWarErr@ehc`

```

158 \def\InfWarErr@ehc{%
159 Try typing \space <return> \space to proceed.\MessageBreak
160 If that doesn't work, type \space X <return> \space to quit.%
161 }

```

`\InfWarErr@ehd`

```

162 \def\InfWarErr@ehd{%
163 You're in trouble here. \space\@ehc
164 }

```

`\InfWarErr@GenericInfo`

```

165 \InfWarErr@protected\def\InfWarErr@GenericInfo#1#2{%
166 \begingroup
167 \let\on@line\InfWarErr@online
168 \long\def\@gobble##1{}%
169 \def\space{ }%
170 \def\@spaces{\space\space\space}%
171 \def\MessageBreak{^^J#1}%
172 \newlinechar=10 %
173 \immediate\write-1{#2\on@line.}%
174 \endgroup
175 }

```

`\InfWarErr@GenericWarning`

```

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

```

```

183     \newlinechar=10 %
184     \immediate\write\InfWarErr@unused{^^J#2\on@line.^^J}%
185 \endgroup
186 }

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

187 \InfWarErr@protected\def\InfWarErr@GenericError#1#2#3#4{%
188   \begingroup
189     \immediate\write\InfWarErr@unused{}%
190     \newlinechar=10 %
191     \def\space{ }%
192     \def\@spaces{\space\space\space\space}%
193     \let\@ehc\InfWarErr@ehc
194     \let\@ehd\InfWarErr@ehd
195     \let\on@line\InfWarErr@online
196     \def\MessageBreak{^^J}%
197     \edef\InfWarErr@help{\noexpand\errhelp{#4}}%
198     \InfWarErr@help
199     \def\MessageBreak{^^J#1}%
200     \errmessage{%
201       #2.^^J^^J%
202       #3^^J%
203       Type \space H <return> \space for immediate help.^^J ...^^J%
204     }%
205   \endgroup
206 }

\@PackageInfo

207 \def\@PackageInfo#1#2{%
208   \InfWarErr@GenericInfo{%
209     (#1) \@spaces\@spaces\@spaces
210   }{%
211     Package #1 Info: #2%
212   }%
213 }

\@PackageInfoNoLine

214 \def\@PackageInfoNoLine#1#2{%
215   \@PackageInfo{#1}{#2@gobble}%
216 }

\@ClassInfo

217 \def\@ClassInfo#1#2{%
218   \InfWarErr@GenericInfo{%
219     (#1) \space\space\@spaces\@spaces
220   }{%
221     Class #1 Info: #2%
222   }%
223 }

\@ClassInfoNoLine

224 \def\@ClassInfoNoLine#1#2{%
225   \@ClassInfo{#1}{#2@gobble}%
226 }

\@PackageWarning

227 \def\@PackageWarning#1#2{%
228   \InfWarErr@GenericWarning{%
229     (#1)\@spaces\@spaces\@spaces\@spaces
230   }{%

```

```

231     Package #1 Warning: #2%
232 }%
233 }

\@PackageWarningNoLine

234 \def\@PackageWarningNoLine#1#2{%
235   \@PackageWarning{#1}{#2\@gobble}%
236 }

\@ClassWarning

237 \def\@ClassWarning#1#2{%
238   \InfWarErr@GenericWarning{%
239     (#1) \space\@spaces\@spaces\@spaces
240   }{%
241     Class #1 Warning: #2%
242   }%
243 }

\@ClassWarningNoLine

244 \def\@ClassWarningNoLine#1#2{%
245   \@ClassWarning{#1}{#2\@gobble}%
246 }

\@PackageError

247 \def\@PackageError#1#2#3{%
248   \InfWarErr@GenericError{%
249     (#1)\@spaces\@spaces\@spaces\@spaces
250   }{%
251     Package #1 Error: #2%
252   }{%
253     See the #1 package documentation for explanation.%
254   }{#3}%
255 }

\@ClassError

256 \def\@ClassError#1#2#3{%
257   \InfWarErr@GenericError{%
258     (#1) \space\@spaces\@spaces\@spaces
259   }{%
260     Class #1 Error: #2%
261   }{%
262     See the #1 class documentation for explanation.%
263   }{#3}%
264 }

265 \InfWarErr@AtEnd
266 \</package>

```

3 Test

3.1 Catcode checks for loading

```

267 \test1\
268 \catcode'\{=1 %
269 \catcode'\}=2 %
270 \catcode'\#=6 %
271 \catcode'\@=11 %
272 \expandafter\ifx\csname count@\endcsname\relax
273   \countdef\count@=255 %
274 \fi

```

```

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

```



```

337 \catcode13=5 %
338 \LoadCommand
339 \RestoreCatcodes
340 }
341 \Test
342 \csname @@end\endcsname
343 \end
344 </test1>

```

3.2 Macro tests

```

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

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

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

```

```

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

```

```

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

```

```

520     \msg{}%
521     \msg{[\ErrMessage]}%
522     \msg{<>}%
523     \msg{[\x]}%
524     \TestFailed#1{errmessage text}%
525 \fi
526 \edef\x{\the\ErrHelpToks}%
527 \edef\y{#3}%
528 \ifx\x\y
529     \else
530         \msg{}%
531         \msg{[\the\ErrHelpToks]}%
532         \msg{<>}%
533         \msg{[\y]}%
534         \TestFailed#1{errhelp text}%
535 \fi
536 \endgroup
537 }
538 \TestInfo
539 \TestWarning
540 \TestError
541 </test2 | test3>
542 <test2>\end
543 <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	A
<code>\#</code> 270, 332	<code>\advance</code> 309, 317
<code>\%</code> 335	
<code>\@</code> 271, 328	B
<code>\@@end</code> 543	<code>\body</code> 288, 292
<code>\@ClassError</code> 133, 256, 478, 479	
<code>\@ClassInfo</code> ... 124, 217, 225, 422, 423	C
<code>\@ClassInfoNoLine</code> . 125, 224, 427, 428	<code>\catcode</code> 3, 4, 5,
<code>\@ClassWarning</code> 130, 237, 245, 448, 449	6, 7, 18, 19, 20, 34, 35, 36, 37,
<code>\@ClassWarningNoLine</code> 131, 244, 455, 456	38, 39, 40, 41, 42, 43, 44, 65,
<code>\@PackageError</code> 132,	66, 69, 70, 71, 72, 76, 77, 78,
247, 467, 468, 489, 490, 500, 501	79, 83, 85, 268, 269, 270, 271,
<code>\@PackageInfo</code> . 120, 207, 215, 412, 413	306, 315, 328, 329, 330, 331,
<code>\@PackageInfoNoLine</code> 121, 214, 417, 418	332, 333, 334, 335, 336, 337, 352
<code>\@PackageWarning</code> 128, 227, 235, 434, 435	<code>\chardef</code> 157, 349, 360
<code>\@PackageWarningNoLine</code> 129, 234, 441, 442	<code>\check</code> 101, 111, 112, 113
<code>\@TestError</code> ... 468, 479, 490, 501, 512	<code>\ClassError</code> 133
<code>\@check</code> 102,	<code>\ClassInfo</code> 124, 126
103, 104, 106, 114, 115, 116, 117	<code>\ClassWarning</code> 130
<code>\@ehc</code> 163, 193, 489	<code>\ClassWarningNoLine</code> 131
<code>\@ehd</code> 194, 500	<code>\clsname</code> .. 365, 422, 427, 448, 455, 478
<code>\@firstofone</code> 279, 282	<code>\count@</code> 273, 302,
<code>\@gobble</code> 122, 126, 168,	306, 308, 309, 313, 315, 316, 317
179, 215, 225, 235, 245, 276, 284	<code>\countdef</code> 273
<code>\@onelevel@sanitize</code> 516, 517	<code>\csname</code> . 8, 21, 45, 61, 68, 107, 138,
<code>\@spaces</code> 170, 181,	152, 272, 275, 278, 281, 320, 342
192, 209, 219, 229, 239, 249, 258	
<code>\@undefined</code> 53	E
<code>\@unused</code> 360	<code>\empty</code> 12
<code>\@write</code> 373, 375	<code>\end</code> 343, 542
<code>\[</code> 333	<code>\endcsname</code> 8, 21, 45, 61, 68, 107, 138,
<code>\</code> 329	152, 272, 275, 278, 281, 320, 342
<code>\{</code> 268, 330	<code>\endinput</code> 30, 135
<code>\}</code> 269, 331	<code>\errhelp</code> 197, 403, 407
<code>\]</code> 334	<code>\ErrHelpToks</code> 401, 403, 526, 531
	<code>\ErrorMessage</code> 409, 517, 518, 521
	<code>\errmessage</code> 200, 370, 402
	H
<code>_</code> 336	<code>\hlptext</code> 367, 467, 478

I		\PackageInfo 26, 120, 122, 346	
\ifcase 9, 138		\PackageWarning 128	
\ifnum 141, 308, 316, 384		\PackageWarningNoLine 129	
\ifx 10, 12, 21, 45,		\pkgname 364,	
53, 56, 107, 119, 138, 152, 272,		412, 417, 434, 441, 467, 489, 500	
275, 278, 281, 320, 391, 518, 528		\plaindots 465, 466, 473, 484, 495, 506	
\immediate 23, 47, 173, 184, 189, 350, 361		\protected 155	
\InfWarErr@AtEnd 81, 82, 134, 265		\ProvidesFile 356	
\InfWarErr@ehc 158, 193		\ProvidesPackage 62	
\InfWarErr@ehd 162, 194			
\InfWarErr@GenericError 187, 248, 257		R	
\InfWarErr@GenericInfo 165, 208, 218		\RangeCatcodeInvalid	
\InfWarErr@GenericWarning 312, 324, 325, 326, 327	
. 176, 228, 238		\repeat 287, 299, 310, 318	
\InfWarErr@help 197, 198		\RequirePackage 357	
\InfWarErr@online . . 137, 167, 178, 195		\RestoreCatcodes . . 301, 304, 305, 339	
\InfWarErr@protected 151, 165, 176, 187			
\InfWarErr@unused . . . 157, 184, 189		S	
\input 321, 347		\space 159, 160,	
\inputlineno 141, 149, 415, 425, 438, 452		163, 169, 170, 180, 181, 191,	
\iterate 289, 291, 293		192, 203, 219, 239, 258, 389, 514	
L		T	
\LoadCommand 321, 338		\Test 323, 341	
\loop 287, 303, 314		\TestError 463, 540	
M		\TestFailed . . . 380, 386, 397, 524, 534	
\makeatletter 358		\TestInfo 411, 538	
\MessageBreak		\TestString 390, 391	
. . . 159, 171, 182, 196, 199, 366, 367		\TestWarning 433, 539	
\msg 350, 351, 361, 393, 394, 395, 396,		\TestWrite 383, 413,	
404, 405, 406, 407, 408, 520,		418, 423, 428, 435, 442, 449, 456	
521, 522, 523, 530, 531, 532, 533		\the 69, 70, 71, 72, 83, 149, 306, 403,	
\msgtext 366, 412, 417, 422, 427, 434,		407, 415, 425, 438, 452, 526, 531	
441, 448, 455, 467, 478, 489, 500		\TMP@EnsureCode 80, 87, 88,	
N		89, 90, 91, 92, 93, 94, 95, 96, 97, 98	
\NeedsTeXFormat 355		W	
\newlinechar 172, 183, 190		\write 23, 47, 173,	
\newtoks 401		184, 189, 348, 359, 369, 372, 464	
\next 293, 295, 297		\WriteNum 377, 384, 386	
\number 377		\WriteText 378, 391, 394	
O		\WriteUnused 349,	
\on@line 167, 173, 178, 184, 195		350, 360, 361, 435, 442, 449, 456	
\OrgErrMsg 370, 381		X	
\OrgWrite		\x 8, 10, 12, 22, 26,	
. . . 348, 350, 359, 361, 369, 376, 464		28, 46, 51, 61, 67, 75, 100, 108,	
P		119, 515, 516, 518, 523, 526, 528	
\PackageError 132		Y	
Y		\y 527, 528, 533	