

The infwarerr package

Heiko Oberdiek
<heiko.oberdiek at gmail.com>

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

Package identification:

```
30 \begingroup
31 \catcode35 6 % #
32 \catcode40 12 % (
33 \catcode41 12 % )
34 \catcode44 12 % ,
35 \catcode45 12 % -
36 \catcode46 12 % .
37 \catcode47 12 % /
38 \catcode58 12 % :
39 \catcode64 11 % @
40 \catcode91 12 % [
41 \catcode93 12 % ]
42 \catcode123 1 % {
```

```

43 \catcode125 2 % }
44 \expandafter\ifx\csname ProvidesPackage\endcsname\relax
45 \def\x#1#2#3[#4]{\endgroup
46 \immediate\write-1{Package: #3 #4}%
47 \xdef#1{#4}%
48 }%
49 \else
50 \def\x#1#2[#3]{\endgroup
51 #2[#{#3}]%
52 \ifx#1@undefined
53 \xdef#1{#3}%
54 \fi
55 \ifx#1\relax
56 \xdef#1{#3}%
57 \fi
58 }%
59 \fi
60 \expandafter\x\csname ver@infwarerr.sty\endcsname
61 \ProvidesPackage{infwarerr}%
62 [2007/09/09 v1.2 Providing info/warning/message (H0)]

```

2.2 Catcodes

```

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

```

2.3 L^AT_EX detection

```

98 \begingroup
99 \let\x=y%

```

```

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

```

2.4 Version for L^AT_EX

\@PackageInfo

```
119 \let\@PackageInfo\PackageInfo
```

\@PackageInfoNoLine

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

\@ClassInfo

```
123 \let\@ClassInfo\ClassInfo
```

\@ClassInfoNoLine

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

\@PackageWarning

```
127 \let\@PackageWarning\PackageWarning
```

\@PackageWarningNoLine

```
128 \let\@PackageWarningNoLine\PackageWarningNoLine
```

\@ClassWarning

```
129 \let\@ClassWarning\ClassWarning
```

\@ClassWarningNoLine

```
130 \let\@ClassWarningNoLine\ClassWarningNoLine
```

\@PackageError

```
131 \let\@PackageError\PackageError
```

\@ClassError

```
132 \let\@ClassError\ClassError
```

```
133 \InfWarErr@AtEnd
```

```
134 \expandafter\endinput
```

```
135 \fi
```

2.5 Version for plain-TeX

`\InfWarErr@online`

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

`\InfWarErr@protected`

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

`\InfWarErr@unused`

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

`\InfWarErr@ehc`

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

`\InfWarErr@ehd`

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

`\InfWarErr@GenericInfo`

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

`\InfWarErr@GenericWarning`

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

```

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

```

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

```

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

```

`\@PackageInfo`

```

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

```

`\@PackageInfoNoLine`

```

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

```

`\@ClassInfo`

```

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

```

`\@ClassInfoNoLine`

```

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

```

`\@PackageWarning`

```

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

```

```

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

```

`\@PackageWarningNoLine`

```

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

```

`\@ClassWarning`

```

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

```

`\@ClassWarningNoLine`

```

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

```

`\@PackageError`

```

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

```

`\@ClassError`

```

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

```

```

264 \InfWarErr@AtEnd
265 \</package>

```

3 Test

3.1 Catcode checks for loading

```

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

```

```

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

```



```

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

```

3.2 Macro tests

```

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

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

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

```

```

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

```

```

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

```

```

519     \msg{}%
520     \msg{[\ErrMessage]}%
521     \msg{<>}%
522     \msg{[\x]}%
523     \TestFailed#1{errmessage text}%
524     \fi
525     \edef\x{\the\ErrHelpToks}%
526     \edef\y{#3}%
527     \ifx\x\y
528     \else
529     \msg{}%
530     \msg{[\the\ErrHelpToks]}%
531     \msg{<>}%
532     \msg{[\y]}%
533     \TestFailed#1{errhelp text}%
534     \fi
535 \endgroup
536 }
537 \TestInfo
538 \TestWarning
539 \TestError
540 </test2 | test3>
541 <test2>\end
542 <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 \TeX Files” ([CTAN:tds/tds.pdf](#)). Directories with `texmf` in their name are usually organized this way.

4.2 Bundle installation

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

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

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

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

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 , mi \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> 269, 331	<code>\advance</code> 308, 316
<code>\%</code> 334	<code>\aftergroup</code> 26
<code>\@</code> 270, 327	B
<code>\@@end</code> 542	<code>\body</code> 287, 291
<code>\@ClassError</code> <u>132</u> , <u>255</u> , 477, 478	C
<code>\@ClassInfo</code> ... <u>123</u> , <u>216</u> , <u>224</u> , 421, 422	<code>\catcode</code> 3, 4, 5, 6, 7,
<code>\@ClassInfoNoLine</code> . <u>124</u> , <u>223</u> , 426, 427	8, 9, 17, 31, 32, 33, 34, 35, 36,
<code>\@ClassWarning</code> <u>129</u> , <u>236</u> , <u>244</u> , 447, 448	37, 38, 39, 40, 41, 42, 43, 64,
<code>\@ClassWarningNoLine</code> <u>130</u> , <u>243</u> , 454, 455	65, 68, 69, 70, 71, 75, 76, 77,
<code>\@PackageError</code> <u>131</u> ,	78, 82, 84, 267, 268, 269, 270,
<u>246</u> , 466, 467, 488, 489, 499, 500	305, 314, 327, 328, 329, 330,
<code>\@PackageInfo</code> . <u>119</u> , <u>206</u> , 214, 411, 412	331, 332, 333, 334, 335, 336, 351
<code>\@PackageInfoNoLine</code> <u>120</u> , <u>213</u> , 416, 417	<code>\chardef</code> 156, 348, 359
<code>\@PackageWarning</code> <u>127</u> , <u>226</u> , 234, 433, 434	<code>\check</code> 100, 110, 111, 112
<code>\@PackageWarningNoLine</code> <u>128</u> , <u>233</u> , 440, 441	<code>\ClassError</code> 132
<code>\@TestError</code> ... 467, 478, 489, 500, 511	<code>\ClassInfo</code> 123, 125
<code>\@check</code> 101,	<code>\ClassWarning</code> 129
102, 103, 105, 113, 114, 115, 116	<code>\ClassWarningNoLine</code> 130
<code>\@ehc</code> 162, 192, 488	<code>\clsname</code> .. 364, 421, 426, 447, 454, 477
<code>\@ehd</code> 193, 499	<code>\count@</code> 272, 301,
<code>\@firstofone</code> 278, 281	305, 307, 308, 312, 314, 315, 316
<code>\@gobble</code> <u>121</u> , <u>125</u> , 167,	<code>\countdef</code> 272
<u>178</u> , 214, 224, 234, 244, 275, 283	<code>\csname</code> 10, 18, 44, 60, 67, 106, 137,
<code>\@onelevel@sanitize</code> 515, 516	151, 271, 274, 277, 280, 319, 341
<code>\@spaces</code> 169, 180,	E
<u>191</u> , 208, 218, 228, 238, 248, 257	<code>\empty</code> 13, 14
<code>\@undefined</code> 52	<code>\end</code> 342, 541
<code>\@unused</code> 359	<code>\endcsname</code> 10, 18, 44, 60, 67, 106, 137,
<code>\@write</code> 372, 374	151, 271, 274, 277, 280, 319, 341
<code>\[</code> 332	<code>\endinput</code> 26, 134
<code>\]</code> 328	<code>\errhelp</code> 196, 402, 406
<code>\{</code> 267, 329	<code>\ErrHelpToks</code> 400, 402, 525, 530
<code>\}</code> 268, 330	<code>\ErrorMessage</code> 408, 516, 517, 520
<code>\]</code> 333	<code>\errmessage</code> 199, 369, 401
<code>_</code> 335	H
	<code>\hlptext</code> 366, 466, 477

I			
<code>\ifcase</code>	137	<code>\PackageInfo</code>	
<code>\ifnum</code>	140, 307, 315, 383	<code>\PackageWarning</code>	
<code>\ifx</code>	11, 14, 18, 44, 52, 55, 106, 118, 137, 151, 271, 274, 277, 280, 319, 390, 517, 527	<code>\PackageWarningNoLine</code>	
<code>\immediate</code>	20, 46, 172, 183, 188, 349, 360	<code>\pkgname</code>	
<code>\InfWarErr@AtEnd</code>	80, 81, 133, 264	411, 416, 433, 440, 466, 488, 499	
<code>\InfWarErr@ehc</code>	157, 192	<code>\plaindots</code>	
<code>\InfWarErr@ehd</code>	161, 193	464, 465, 472, 483, 494, 505	
<code>\InfWarErr@GenericError</code>	186, 247, 256	<code>\protected</code>	
<code>\InfWarErr@GenericInfo</code>	164, 207, 217	154	
<code>\InfWarErr@GenericWarning</code>	175, 227, 237	<code>\ProvidesFile</code>	
<code>\InfWarErr@help</code>	196, 197	355	
<code>\InfWarErr@online</code>	136, 166, 177, 194	<code>\ProvidesPackage</code>	
<code>\InfWarErr@protected</code>	150, 164, 175, 186	15, 61	
<code>\InfWarErr@unused</code>	156, 183, 188	R	
<code>\input</code>	320, 346	<code>\RangeCatcodeInvalid</code>	
<code>\inputlineno</code>	140, 148, 414, 424, 437, 451	311, 323, 324, 325, 326	
<code>\iterate</code>	288, 290, 292	<code>\repeat</code>	
L		286, 298, 309, 317	
<code>\LoadCommand</code>	320, 337	<code>\RequirePackage</code>	
<code>\loop</code>	286, 302, 313	356	
M		<code>\RestoreCatcodes</code> ..	
<code>\makeatletter</code>	357	300, 303, 304, 338	
<code>\MessageBreak</code>	158, 170, 181, 195, 198, 365, 366	S	
<code>\msg</code>	349, 350, 360, 392, 393, 394, 395, 403, 404, 405, 406, 407, 519, 520, 521, 522, 529, 530, 531, 532	<code>\space</code>	
<code>\msgtext</code>	365, 411, 416, 421, 426, 433, 440, 447, 454, 466, 477, 488, 499	158, 159, 162, 168, 169, 179, 180, 190, 191, 202, 218, 238, 257, 388, 513	
N		T	
<code>\NeedsTeXFormat</code>	354	<code>\Test</code>	
<code>\newlinechar</code>	171, 182, 189	322, 340	
<code>\newtoks</code>	400	<code>\TestError</code>	
<code>\next</code>	292, 294, 296	462, 539	
<code>\number</code>	376	<code>\TestFailed</code> ...	
O		379, 385, 396, 523, 533	
<code>\on@line</code>	166, 172, 177, 183, 194	<code>\TestInfo</code>	
<code>\OrgErrorMessage</code>	369, 380	410, 537	
<code>\OrgWrite</code>	347, 349, 358, 360, 368, 375, 463	<code>\TestString</code>	
P		389, 390	
<code>\PackageError</code>	131	<code>\TestWarning</code>	
R		432, 538	
S		<code>\TestWrite</code>	
T		382, 412, 417, 422, 427, 434, 441, 448, 455	
<code>\Test</code>	322, 340	<code>\the</code>	
<code>\TestError</code>	462, 539	68, 69, 70, 71, 82, 148, 305, 402, 406, 414, 424, 437, 451, 525, 530	
<code>\TestFailed</code> ...	379, 385, 396, 523, 533	<code>\TMP@EnsureCode</code>	
<code>\TestInfo</code>	410, 537	79, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97	
<code>\TestString</code>	389, 390	W	
<code>\TestWarning</code>	432, 538	<code>\write</code>	
<code>\TestWrite</code>	382, 412, 417, 422, 427, 434, 441, 448, 455	20, 46, 172, 183, 188, 347, 358, 368, 371, 463	
<code>\the</code>	68, 69, 70, 71, 82, 148, 305, 402, 406, 414, 424, 437, 451, 525, 530	<code>\WriteNum</code>	
<code>\TMP@EnsureCode</code>	79, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97	376, 383, 385	
W		<code>\WriteText</code>	
<code>\write</code>	20, 46, 172, 183, 188, 347, 358, 368, 371, 463	377, 390, 393	
<code>\WriteNum</code>	376, 383, 385	<code>\WriteUnused</code>	
<code>\WriteText</code>	377, 390, 393	348, 349, 359, 360, 434, 441, 448, 455	
<code>\WriteUnused</code>	348, 349, 359, 360, 434, 441, 448, 455	X	
X		<code>\x</code>	
<code>\x</code>	10, 11, 14, 19, 23, 25, 45, 50, 60, 66, 74, 99, 107, 118, 514, 515, 517, 522, 525, 527	Y	
Y		<code>\y</code>	
<code>\y</code>	526, 527, 532	Z	