

# 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

<b>1 Documentation</b>	<b>1</b>
<b>2 Implementation</b>	<b>2</b>
2.1 Reload check and package identification . . . . .	2
2.2 Catcodes . . . . .	3
2.3 L <sup>A</sup> T <sub>E</sub> X detection . . . . .	3
2.4 Version for L <sup>A</sup> T <sub>E</sub> X . . . . .	4
2.5 Version for plain-TeX . . . . .	4
<b>3 Test</b>	<b>7</b>
3.1 Catcode checks for loading . . . . .	7
3.2 Macro tests . . . . .	8
<b>4 Installation</b>	<b>11</b>
4.1 Download . . . . .	11
4.2 Bundle installation . . . . .	11
4.3 Package installation . . . . .	12
4.4 Refresh file name databases . . . . .	12
4.5 Some details for the interested . . . . .	12
<b>5 History</b>	<b>13</b>
[2007/04/26 v1.0] . . . . .	13
[2007/06/14 v1.1] . . . . .	13
[2007/09/09 v1.2] . . . . .	13
<b>6 Index</b>	<b>13</b>

## 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<sup>A</sup>T<sub>E</sub>X'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<sup>A</sup>T<sub>E</sub>X and plain-T<sub>E</sub>X. If L<sup>A</sup>T<sub>E</sub>X is detected, then the macros of this package will only be wrappers for L<sup>A</sup>T<sub>E</sub>X's macros. Otherwise L<sup>A</sup>T<sub>E</sub>X's macros will be mimicked for use with plain-T<sub>E</sub>X.

## 2 Implementation

1 <package>

### 2.1 Reload check and package identification

Reload check, especially if the package is not used with L<sup>A</sup>T<sub>E</sub>X.

```

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 \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 \endgroup
27 \expandafter\endinput
28 \fi
29 \endgroup

```

Package identification:

```

30 \begingroup
31 \catcode40 12 % (
32 \catcode41 12 % )
33 \catcode44 12 % ,
34 \catcode45 12 % -
35 \catcode46 12 % .
36 \catcode47 12 % /
37 \catcode58 12 % :
38 \catcode64 11 % @
39 \expandafter\ifx\csname ProvidesPackage\endcsname\relax
40 \def\x#1#2#3[#4]{\endgroup
41 \immediate\write-1{Package: #3 #4}%
42 \xdef#1{#4}%

```

```

43   }%
44   \else
45     \def\x#1#2[#3]{\endgroup
46       #2[#3]}%
47     \ifx#1\relax
48       \xdef#1[#3]%
49     \fi
50   }%
51   \fi
52   \expandafter\x\csname ver@infwarerr.sty\endcsname
53   \ProvidesPackage{infwarerr}%
54   [2007/09/09 v1.2 Providing info/warning/message (HO)]

```

## 2.2 Catcodes

```

55   \expandafter\edef\csname InfWarErr@AtEnd\endcsname{%
56     \catcode64 \the\catcode64\relax
57   }
58   \catcode64 11 % @
59   \def\TMP@EnsureCode#1#2{%
60     \edef\InfWarErr@AtEnd{%
61       \InfWarErr@AtEnd
62       \catcode#1 \the\catcode#1\relax
63     }%
64     \catcode#1 #2\relax
65   }
66   \TMP@EnsureCode{10}{12}% ^^J
67   \TMP@EnsureCode{39}{12}% '
68   \TMP@EnsureCode{40}{12}% (
69   \TMP@EnsureCode{41}{12}% )
70   \TMP@EnsureCode{44}{12}% ,
71   \TMP@EnsureCode{45}{12}% -
72   \TMP@EnsureCode{46}{12}% .
73   \TMP@EnsureCode{58}{12}% :
74   \TMP@EnsureCode{60}{12}% <
75   \TMP@EnsureCode{61}{12}% =
76   \TMP@EnsureCode{62}{12}% >
77   \TMP@EnsureCode{94}{7}% ^

```

## 2.3 L<sup>A</sup>T<sub>E</sub>X detection

```

78   \begingroup
79   \let\x=Y%
80   \def\check#1{%
81     \@check{Generic}{#1}%
82     \@check{Package}{#1}%
83     \@check{Class}{#1}%
84   }%
85   \def\@check#1#2{%
86     \expandafter\ifx\csname #1#2\endcsname\relax
87       \let\x=N%
88     \fi
89   }%
90   \check{Info}%
91   \check{Warning}%
92   \check{Error}%
93   \@check{Package}{WarningNoLine}%
94   \@check{Class}{WarningNoLine}%
95   \@check{}{@ehc}%
96   \@check{}{@ehd}%
97   \expandafter\endgroup
98   \ifx\x Y%

```

## 2.4 Version for L<sup>A</sup>T<sub>E</sub>X

```
\@PackageInfo
99 \let\@PackageInfo\PackageInfo

\@PackageInfoNoLine
100 \def\@PackageInfoNoLine#1#2{%
101   \PackageInfo{#1}{#2\@gobble}%
102 }%

\@ClassInfo
103 \let\@ClassInfo\ClassInfo

\@ClassInfoNoLine
104 \def\@ClassInfoNoLine#1#2{%
105   \ClassInfo{#1}{#2\@gobble}%
106 }%

\@PackageWarning
107 \let\@PackageWarning\PackageWarning

\@PackageWarningNoLine
108 \let\@PackageWarningNoLine\PackageWarningNoLine

\@ClassWarning
109 \let\@ClassWarning\ClassWarning

\@ClassWarningNoLine
110 \let\@ClassWarningNoLine\ClassWarningNoLine

\@PackageError
111 \let\@PackageError\PackageError

\@ClassError
112 \let\@ClassError\ClassError

113 \InfWarErr@AtEnd
114 \expandafter\endinput
115 \fi
```

## 2.5 Version for plain-T<sub>E</sub>X

```
\InfWarErr@online
116 \begingroup\expandafter\endgroup
117 \ifcase\expandafter\ifx\csname inputlineno\endcsname\relax
118   0 %
119   \else
120     \ifnum\inputlineno<0 %
121       0 %
122     \else
123       1 %
124     \fi
125   \fi
126 \def\InfWarErr@online{}%
127 \else
128 \def\InfWarErr@online{ on input line \the\inputlineno}%
129 \fi
```

```

\InfWarErr@protected
130 \begingroup\expandafter\expandafter\expandafter\endgroup
131 \expandafter\ifx\csname protected\endcsname\relax
132   \def\InfWarErr@protected{}%
133 \else
134   \let\InfWarErr@protected\protected
135 \fi

\InfWarErr@unused
136 \chardef\InfWarErr@unused=16 %

\InfWarErr@ehc
137 \def\InfWarErr@ehc{%
138   Try typing \space <return> \space to proceed.\MessageBreak
139   If that doesn't work, type \space X <return> \space to quit.%
140 }

\InfWarErr@ehd
141 \def\InfWarErr@ehd{%
142   You're in trouble here. \space\@ehc
143 }

\InfWarErr@GenericInfo
144 \InfWarErr@protected\def\InfWarErr@GenericInfo#1#2{%
145   \begingroup
146     \let\on@line\InfWarErr@online
147     \long\def\@gobble##1{%
148       \def\space{ }%
149       \def\@spaces{\space\space\space\space}%
150       \def\MessageBreak{^^J#1}%
151       \newlinechar=10 %
152       \immediate\write-1{#2\on@line.}%
153     \endgroup
154 }

\InfWarErr@GenericWarning
155 \InfWarErr@protected\def\InfWarErr@GenericWarning#1#2{%
156   \begingroup
157     \let\on@line\InfWarErr@online
158     \long\def\@gobble##1{%
159       \def\space{ }%
160       \def\@spaces{\space\space\space\space}%
161       \def\MessageBreak{^^J#1}%
162       \newlinechar=10 %
163       \immediate\write\InfWarErr@unused{^^J#2\on@line.^^J}%
164     \endgroup
165 }

\InfWarErr@GenericError Old TeX versions below 3.141 where ^^J is not working as newline in error message
and help texts are not supported.
166 \InfWarErr@protected\def\InfWarErr@GenericError#1#2#3#4{%
167   \begingroup
168     \immediate\write\InfWarErr@unused{}%
169     \newlinechar=10 %
170     \def\space{ }%
171     \def\@spaces{\space\space\space\space}%
172     \let\@ehc\InfWarErr@ehc
173     \let\@ehd\InfWarErr@ehd
174     \let\on@line\InfWarErr@online
175     \def\MessageBreak{^^J}%
176     \edef\InfWarErr@help{\noexpand\errhelp{#4}}%

```

```

177 \InfWarErr@help
178 \def\MessageBreak{^^J#1}%
179 \errmessage{%
180   #2.^^J^^J%
181   #3^^J%
182   Type \space H <return> \space for immediate help.^^J ...^^J%
183 }%
184 \endgroup
185 }

```

\@PackageInfo

```

186 \def\@PackageInfo#1#2{%
187   \InfWarErr@GenericInfo{%
188     (#1) \@spaces\@spaces\@spaces
189   }{%
190     Package #1 Info: #2%
191   }%
192 }

```

\@PackageInfoNoLine

```

193 \def\@PackageInfoNoLine#1#2{%
194   \@PackageInfo{#1}{#2\@gobble}%
195 }

```

\@ClassInfo

```

196 \def\@ClassInfo#1#2{%
197   \InfWarErr@GenericInfo{%
198     (#1) \space\space\@spaces\@spaces
199   }{%
200     Class #1 Info: #2%
201   }%
202 }

```

\@ClassInfoNoLine

```

203 \def\@ClassInfoNoLine#1#2{%
204   \@ClassInfo{#1}{#2\@gobble}%
205 }

```

\@PackageWarning

```

206 \def\@PackageWarning#1#2{%
207   \InfWarErr@GenericWarning{%
208     (#1)\@spaces\@spaces\@spaces\@spaces
209   }{%
210     Package #1 Warning: #2%
211   }%
212 }

```

\@PackageWarningNoLine

```

213 \def\@PackageWarningNoLine#1#2{%
214   \@PackageWarning{#1}{#2\@gobble}%
215 }

```

\@ClassWarning

```

216 \def\@ClassWarning#1#2{%
217   \InfWarErr@GenericWarning{%
218     (#1) \space\@spaces\@spaces\@spaces
219   }{%
220     Class #1 Warning: #2%
221   }%
222 }

```

\@ClassWarningNoLine

```
223 \def\@ClassWarningNoLine#1#2{%
224   \@ClassWarning{#1}{#2@gobble}%
225 }
```

\@PackageError

```
226 \def\@PackageError#1#2#3{%
227   \InfWarErr@GenericError{%
228     (#1)\@spaces\@spaces\@spaces\@spaces
229   }{%
230     Package #1 Error: #2%
231   }{%
232     See the #1 package documentation for explanation.%
233   }{#3}%
234 }
```

\@ClassError

```
235 \def\@ClassError#1#2#3{%
236   \InfWarErr@GenericError{%
237     (#1) \space\@spaces\@spaces\@spaces
238   }{%
239     Class #1 Error: #2%
240   }{%
241     See the #1 class documentation for explanation.%
242   }{#3}%
243 }

244 \InfWarErr@AtEnd
245 \</package>
```

## 3 Test

### 3.1 Catcode checks for loading

```
246 \<test1>

247 \catcode'\@=11 %
248 \def\RestoreCatcodes{}
249 \count@=0 %
250 \loop
251   \edef\RestoreCatcodes{%
252     \RestoreCatcodes
253     \catcode\the\count@=\the\catcode\count@\relax
254   }%
255 \ifnum\count@<255 %
256   \advance\count@\@ne
257 \repeat
258
259 \def\RangeCatcodeInvalid#1#2{%
260   \count@=#1\relax
261   \loop
262     \catcode\count@=15 %
263     \ifnum\count@<#2\relax
264       \advance\count@\@ne
265     \repeat
266 }
267 \def\Test{%
268   \RangeCatcodeInvalid{0}{47}%
269   \RangeCatcodeInvalid{58}{64}%
270   \RangeCatcodeInvalid{91}{96}%
271   \RangeCatcodeInvalid{123}{255}%
272   \catcode'\@=12 %
```

```

273 \catcode'\=0 %
274 \catcode'\{=1 %
275 \catcode'\}=2 %
276 \catcode'\#=6 %
277 \catcode'\[=12 %
278 \catcode'\]=12 %
279 \catcode'\%=14 %
280 \catcode'\ =10 %
281 \catcode13=5 %
282 \input infwarerr.sty\relax
283 \RestoreCatcodes
284 }
285 \Test
286 \csname @@end\endcsname
287 \end
288 </test1>

```

## 3.2 Macro tests

```

289 <*test2>
290 \let\PackageInfo\relax
291 \input infwarerr.sty\relax
292 \let\OrgWrite\write
293 \chardef\WriteUnused=16 %
294 \def\msg#\{\immediate\OrgWrite\WriteUnused}
295 \msg{File: infwarerr-test2.tex 2007/09/09 v1.2 Test file for plain-Tex}
296 \catcode64=11 %
297 </test2>

298 <*test3>
299 \NeedsTeXFormat{LaTeX2e}
300 \ProvidesFile{infwarerr-test3.tex}[2007/09/09 v1.2 Test file for LaTeX]
301 \RequirePackage{infwarerr}[2007/09/09]
302 \makeatletter
303 \let\OrgWrite\write
304 \chardef\WriteUnused=\@unused
305 \def\msg#\{\immediate\OrgWrite\WriteUnused}
306 </test3>

307 <*test2 | test3>
308 \def\pkgname{PACKAGE}
309 \def\clsname{CLASS}
310 \def\msgtext{MESSAGE\MessageBreak SECOND LINE}
311 \def\hlptext{HELP\MessageBreak SECOND LINE}
312
313 \let\OrgWrite\write
314 \let\OrgErrMessage\errmessage
315
316 \def\write#1#{%
317   \@write{#1}%
318 }
319 \def\@write#1#2{%
320   \OrgWrite#1{#2}%
321   \xdef\WriteNum{\number#1}%
322   \xdef\WriteText{#2}%
323 }
324 \def\TestFailed#1#2{%
325   \OrgErrMessage{Test (\string#1) failed: #2}%
326 }
327 \def\TestWrite#1#2#3{%
328   \ifnum\WriteNum=#2\relax
329   \else
330     \TestFailed#1{write stream number: \WriteNum <> #2}%
331   \fi

```



```

332 \begingroup
333   \let~\space
334   \edef\TestString{#3}%
335   \ifx\WriteText\TestString
336   \else
337     \msg{}%
338     \msg{[\WriteText]}%
339     \msg{<>}%
340     \msg{[#3]}% hash-ok
341     \TestFailed#1{write text}%
342   \fi
343 \endgroup
344 }
345 \newtoks\ErrHelpToks
346 \def\errmessage#1{%
347   \global\ErrHelpToks=\expandafter{\the\errhelp}%
348   \msg{***** error message *****}%
349   \msg{! #1}%
350   \msg{***** help message *****}%
351   \msg{\the\errhelp}%
352   \msg{*****}%
353   \xdef\ErrorMessage{! #1}%
354 }
355 \def\TestInfo{%
356   \@PackageInfo\pkgname\msgtext
357   \TestWrite\@PackageInfo{-1}{%
358     Package PACKAGE Info: MESSAGE^^J%
359     (PACKAGE)~~~~~SECOND LINE on input line \the\inputlineno.%
360   }%
361   \@PackageInfoNoLine\pkgname\msgtext
362   \TestWrite\@PackageInfoNoLine{-1}{%
363     Package PACKAGE Info: MESSAGE^^J%
364     (PACKAGE)~~~~~SECOND LINE.%
365   }%
366   \@ClassInfo\clsname\msgtext
367   \TestWrite\@ClassInfo{-1}{%
368     Class CLASS Info: MESSAGE^^J%
369     (CLASS)~~~~~SECOND LINE on input line \the\inputlineno.%
370   }%
371   \@ClassInfoNoLine\clsname\msgtext
372   \TestWrite\@ClassInfoNoLine{-1}{%
373     Class CLASS Info: MESSAGE^^J%
374     (CLASS)~~~~~SECOND LINE.%
375   }%
376 }
377 \def\TestWarning{%
378   \@PackageWarning\pkgname\msgtext
379   \TestWrite\@PackageWarning\WriteUnused{%
380     ^^J%
381     Package PACKAGE Warning: MESSAGE^^J%
382     (PACKAGE)~~~~~SECOND LINE on input line \the\inputlineno.%
383     ^^J%
384   }%
385   \@PackageWarningNoLine\pkgname\msgtext
386   \TestWrite\@PackageWarningNoLine\WriteUnused{%
387     ^^J%
388     Package PACKAGE Warning: MESSAGE^^J%
389     (PACKAGE)~~~~~SECOND LINE.%
390     ^^J%
391   }%
392   \@ClassWarning\clsname\msgtext
393   \TestWrite\@ClassWarning\WriteUnused{%

```

```

394    ^^J%
395    Class CLASS Warning: MESSAGE^^J%
396    (CLASS)~~~~~SECOND LINE on input line \the\inputlineno.%
397    ^^J%
398  }%
399  \@ClassWarningNoLine\clsname\msgtext
400  \TestWrite\@ClassWarningNoLine\WriteUnused{%
401    ^^J%
402    Class CLASS Warning: MESSAGE^^J%
403    (CLASS)~~~~~SECOND LINE.%
404    ^^J%
405  }%
406 }
407 \def\TestError{%
408 (test3) \let\write\OrgWrite
409 (test3) \def\plaindots{}
410 (test2) \def\plaindots{.^^J ...^^J}
411 \@PackageError\pkgname\msgtext\hlptext
412 \@TestError\@PackageError{%
413   ! Package PACKAGE Error: MESSAGE^^J%
414   (PACKAGE)~~~~~SECOND LINE.^^J%
415   ^^J%
416   See the PACKAGE package documentation for explanation.^^J%
417   Type~~H <return>~~for immediate help\plaindots
418 }{%
419   HELP^^J%
420   SECOND LINE%
421 }%
422 \@ClassError\clsname\msgtext\hlptext
423 \@TestError\@ClassError{%
424   ! Class CLASS Error: MESSAGE^^J%
425   (CLASS)~~~~~SECOND LINE.^^J%
426   ^^J%
427   See the CLASS class documentation for explanation.^^J%
428   Type~~H <return>~~for immediate help\plaindots
429 }{%
430   HELP^^J%
431   SECOND LINE%
432 }%
433 \@PackageError\pkgname\msgtext\@ehc
434 \@TestError\@PackageError{%
435   ! Package PACKAGE Error: MESSAGE^^J%
436   (PACKAGE)~~~~~SECOND LINE.^^J%
437   ^^J%
438   See the PACKAGE package documentation for explanation.^^J%
439   Type~~H <return>~~for immediate help\plaindots
440 }{%
441   Try typing~~<return>~~to proceed.^^J%
442   If that doesn't work, type~~X <return>~~to quit.%
443 }%
444 \@PackageError\pkgname\msgtext\@ehd
445 \@TestError\@PackageError{%
446   ! Package PACKAGE Error: MESSAGE^^J%
447   (PACKAGE)~~~~~SECOND LINE.^^J%
448   ^^J%
449   See the PACKAGE package documentation for explanation.^^J%
450   Type~~H <return>~~for immediate help\plaindots
451 }{%
452   You're in trouble here.~~Try typing~~<return>~~to proceed.^^J%
453   If that doesn't work, type~~X <return>~~to quit.%
454 }%
455 }

```

```

456 \def\@TestError#1#2#3{%
457   \begingroup
458     \let~\space
459     \edef\x{#2}%
460     \test3) \@onelevel@sanitize\x
461     \test3) \@onelevel@sanitize\ErrorMessage
462     \ifx\x\ErrorMessage
463     \else
464       \msg{}%
465       \msg{[\ErrorMessage]}%
466       \msg{<>}%
467       \msg{[\x]}%
468       \TestFailed#1{errmessage text}%
469     \fi
470     \edef\x{\the\ErrHelpToks}%
471     \edef\y{#3}%
472     \ifx\x\y
473     \else
474       \msg{}%
475       \msg{[\the\ErrHelpToks]}%
476       \msg{<>}%
477       \msg{[\y]}%
478       \TestFailed#1{errhelp text}%
479     \fi
480   \endgroup
481 }
482 \TestInfo
483 \TestWarning
484 \TestError
485 </test2 | test3>
486 <test2>\end
487 <test3>\@@end

```

## 4 Installation

### 4.1 Download

**Package.** This package is available on CTAN<sup>1</sup>:

[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:macros/latex/contrib/oberdiek/oberdiek-tds.zip](#)

*TDS* refers to the standard “A Directory Structure for T<sub>E</sub>X 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
```

---

<sup>1</sup><http://ftp.ctan.org/tex-archive/>

**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- $\text{\TeX}$ :

```
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
infwarerr-test1.tex → doc/latex/oberdiek/infwarerr-test1.tex
infwarerr-test2.tex → doc/latex/oberdiek/infwarerr-test2.tex
infwarerr-test3.tex → doc/latex/oberdiek/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  $\text{\TeX}$  distribution (te $\text{\TeX}$ , mi $\text{\TeX}$ , ...) relies on file name databases, you must refresh these. For example, te $\text{\TeX}$  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  $\text{\LaTeX}$ .** The `.dtx` chooses its action depending on the format:

**plain- $\text{\TeX}$ :** Run `docstrip` and extract the files.

**$\text{\LaTeX}$ :** Generate the documentation.

If you insist on using  $\text{\LaTeX}$  for `docstrip` (really, `docstrip` does not need  $\text{\LaTeX}$ ), 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 pdfL<sup>A</sup>T<sub>E</sub>X:

```
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	
\#	276
\%	279
\@	247, 272
\@end	487
\@ClassError	112, 235, 422, 423
\@ClassInfo	103, 196, 204, 366, 367
\@ClassInfoNoLine	104, 203, 371, 372
\@ClassWarning	109, 216, 224, 392, 393
\@ClassWarningNoLine	110, 223, 399, 400
\@PackageError	111, 226, 411, 412, 433, 434, 444, 445
\@PackageInfo	99, 186, 194, 356, 357
\@PackageInfoNoLine	100, 193, 361, 362
\@PackageWarning	107, 206, 214, 378, 379
\@PackageWarningNoLine	108, 213, 385, 386
\@TestError	412, 423, 434, 445, 456
\@check	81, 82, 83, 85, 93, 94, 95, 96
\@ehc	142, 172, 433
\@ehd	173, 444
\@gobble	101, 105, 147, 158, 194, 204, 214, 224
\@ne	256, 264
\@onelevel@sanitize	460, 461
\@spaces	149, 160, 171, 188, 198, 208, 218, 228, 237
\@unused	304
\@write	317, 319
\[	277
\	273
\{	274
\}	275
\]	278
\_	280
<b>A</b>	
\advance	256, 264
<b>C</b>	
\catcode	3, 4, 5, 6, 7, 31, 32, 33, 34, 35, 36, 37, 38, 56, 58, 62, 64, 247, 253, 262, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 296
\chardef	136, 293, 304
\check	80, 90, 91, 92
\ClassError	112
\ClassInfo	103, 105
\ClassWarning	109
\ClassWarningNoLine	110
\clsname	309, 366, 371, 392, 399, 422
\count@	249, 253, 255, 256, 260, 262, 263, 264
\csname	8, 18, 39, 52, 55, 86, 117, 131, 286
<b>E</b>	
\empty	12
\end	287, 486
\endcsname	8, 18, 39, 52, 55, 86, 117, 131, 286
\endinput	27, 114
\errhelp	176, 347, 351

<code>\ErrHelpToks</code> . . . . .	345, 347, 470, 475		
<code>\ErrMsgMessage</code> . . . . .	353, 461, 462, 465		
<code>\errmessage</code> . . . . .	179, 314, 346		
<b>H</b>			
<code>\hlptext</code> . . . . .	311, 411, 422		
<b>I</b>			
<code>\ifcase</code> . . . . .	9, 117		
<code>\ifnum</code> . . . . .	120, 255, 263, 328		
<code>\ifx</code> . . . . .	10, 12, 18, 39,		
	47, 86, 98, 117, 131, 335, 462, 472		
<code>\immediate</code> . . . . .	20, 41, 152, 163, 168, 294, 305		
<code>\InfWarErr@AtEnd</code> . . . . .	60, 61, 113, 244		
<code>\InfWarErr@ehc</code> . . . . .	137, 172		
<code>\InfWarErr@ehd</code> . . . . .	141, 173		
<code>\InfWarErr@GenericError</code> . . . . .	166, 227, 236		
<code>\InfWarErr@GenericInfo</code> . . . . .	144, 187, 197		
<code>\InfWarErr@GenericWarning</code> . . . . .	155, 207, 217		
<code>\InfWarErr@help</code> . . . . .	176, 177		
<code>\InfWarErr@online</code> . . . . .	116, 146, 157, 174		
<code>\InfWarErr@protected</code> . . . . .	130, 144, 155, 166		
<code>\InfWarErr@unused</code> . . . . .	136, 163, 168		
<code>\input</code> . . . . .	282, 291		
<code>\inputlineno</code> . . . . .	120, 128, 359, 369, 382, 396		
<b>L</b>			
<code>\loop</code> . . . . .	250, 261		
<b>M</b>			
<code>\makeatletter</code> . . . . .	302		
<code>\MessageBreak</code> . . . . .	138, 150, 161, 175, 178, 310, 311		
<code>\msg</code> . . . . .	294, 295, 305, 337, 338, 339, 340,		
	348, 349, 350, 351, 352, 464,		
	465, 466, 467, 474, 475, 476, 477		
<code>\msgtext</code> . . . . .	310, 356, 361, 366, 371, 378,		
	385, 392, 399, 411, 422, 433, 444		
<b>N</b>			
<code>\NeedsTeXFormat</code> . . . . .	299		
<code>\newlinechar</code> . . . . .	151, 162, 169		
<code>\newtoks</code> . . . . .	345		
<code>\number</code> . . . . .	321		
<b>O</b>			
<code>\on@line</code> . . . . .	146, 152, 157, 163, 174		
<code>\OrgErrMsg</code> . . . . .	314, 325		
<code>\OrgWrite</code> . . . . .	292, 294, 303, 305, 313, 320, 408		
<b>P</b>			
<code>\PackageError</code> . . . . .	111		
<code>\PackageInfo</code> . . . . .	23, 99, 101, 290		
<code>\PackageWarning</code> . . . . .	107		
<code>\PackageWarningNoLine</code> . . . . .	108		
<code>\pkgname</code> . . . . .	308,		
	356, 361, 378, 385, 411, 433, 444		
<code>\plaindots</code> . . . . .	409, 410, 417, 428, 439, 450		
<code>\protected</code> . . . . .	134		
<code>\ProvidesFile</code> . . . . .	300		
<code>\ProvidesPackage</code> . . . . .	53		
<b>R</b>			
<code>\RangeCatcodeInvalid</code> . . . . .	259, 268, 269, 270, 271		
<code>\repeat</code> . . . . .	257, 265		
<code>\RequirePackage</code> . . . . .	301		
<code>\RestoreCatcodes</code> . . . . .	248, 251, 252, 283		
<b>S</b>			
<code>\space</code> . . . . .	138, 139,		
	142, 148, 149, 159, 160, 170,		
	171, 182, 198, 218, 237, 333, 458		
<b>T</b>			
<code>\Test</code> . . . . .	267, 285		
<code>\TestError</code> . . . . .	407, 484		
<code>\TestFailed</code> . . . . .	324, 330, 341, 468, 478		
<code>\TestInfo</code> . . . . .	355, 482		
<code>\TestString</code> . . . . .	334, 335		
<code>\TestWarning</code> . . . . .	377, 483		
<code>\TestWrite</code> . . . . .	327, 357,		
	362, 367, 372, 379, 386, 393, 400		
<code>\the</code> . . . . .	56, 62, 128, 253, 347,		
	351, 359, 369, 382, 396, 470, 475		
<code>\TMP@EnsureCode</code> . . . . .	59, 66, 67,		
	68, 69, 70, 71, 72, 73, 74, 75, 76, 77		
<b>W</b>			
<code>\write</code> . . . . .	20, 41, 152,		
	163, 168, 292, 303, 313, 316, 408		
<code>\WriteNum</code> . . . . .	321, 328, 330		
<code>\WriteText</code> . . . . .	322, 335, 338		
<code>\WriteUnused</code> . . . . .	293,		
	294, 304, 305, 379, 386, 393, 400		
<b>X</b>			
<code>\x</code> . . . . .	8, 10, 12, 19, 23, 25, 40, 45, 52, 79,		
	87, 98, 459, 460, 462, 467, 470, 472		
<b>Y</b>			
<code>\y</code> . . . . .	471, 472, 477		