

# The infwarerr package

Heiko Oberdiek  
<heiko.oberdiek at gmail.com>

2010/04/08 v1.3

## Abstract

This package provides a complete set of macros for informations, warnings and error messages with support for plain  $\text{\TeX}$ .

## Contents

<b>1</b>	<b>Documentation</b>	<b>1</b>
<b>2</b>	<b>Implementation</b>	<b>2</b>
2.1	Reload check and package identification . . . . .	2
2.2	Catcodes . . . . .	3
2.3	$\text{\LaTeX}$ detection . . . . .	4
2.4	Version for $\text{\LaTeX}$ . . . . .	4
2.5	Version for plain $\text{\TeX}$ . . . . .	5
<b>3</b>	<b>Test</b>	<b>8</b>
3.1	Catcode checks for loading . . . . .	8
3.2	Macro tests . . . . .	9
<b>4</b>	<b>Installation</b>	<b>13</b>
4.1	Download . . . . .	13
4.2	Bundle installation . . . . .	13
4.3	Package installation . . . . .	13
4.4	Refresh file name databases . . . . .	13
4.5	Some details for the interested . . . . .	14
<b>5</b>	<b>Catalogue</b>	<b>14</b>
<b>6</b>	<b>History</b>	<b>15</b>
	[2007/04/26 v1.0] . . . . .	15
	[2007/06/14 v1.1] . . . . .	15
	[2007/09/09 v1.2] . . . . .	15
	[2010/04/08 v1.3] . . . . .	15
<b>7</b>	<b>Index</b>	<b>15</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\catcode61\catcode48\catcode32=10\relax%
3 \catcode13=5 % ^~M
4 \endlinechar=13 %
5 \catcode35=6 % #
6 \catcode39=12 % '
7 \catcode44=12 % ,
8 \catcode45=12 % -
9 \catcode46=12 % .
10 \catcode58=12 % :
11 \catcode64=11 % @
12 \catcode123=1 % {
13 \catcode125=2 % }
14 \expandafter\let\expandafter\x\csname ver@infwarerr.sty\endcsname
15 \ifx\x\relax % plain-TeX, first loading
16 \else
17 \def\empty{}%
18 \ifx\x\empty % LaTeX, first loading,
19 % variable is initialized, but \ProvidesPackage not yet seen
20 \else
21 \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 \aftergroup\endinput
30 \fi
31 \fi
32 \endgroup%
```

Package identification:

```
33 \begingroup\catcode61\catcode48\catcode32=10\relax%
34 \catcode13=5 % ^~M
35 \endlinechar=13 %
36 \catcode35=6 % #
37 \catcode39=12 % '

```

```

38 \catcode40=12 % (
39 \catcode41=12 % )
40 \catcode44=12 % ,
41 \catcode45=12 % -
42 \catcode46=12 % .
43 \catcode47=12 % /
44 \catcode58=12 % :
45 \catcode64=11 % @
46 \catcode91=12 % [
47 \catcode93=12 % ]
48 \catcode123=1 % {
49 \catcode125=2 % }
50 \expandafter\ifx\csname ProvidesPackage\endcsname\relax
51   \def\x#1#2#3[#4]{\endgroup
52     \immediate\write-1{Package: #3 #4}%
53     \xdef#1{#4}%
54   }%
55 \else
56   \def\x#1#2[#3]{\endgroup
57     #2[#{#3}]%
58     \ifx#1@undefined
59       \xdef#1{#3}%
60     \fi
61     \ifx#1\relax
62       \xdef#1{#3}%
63     \fi
64   }%
65 \fi
66 \expandafter\x\csname ver@infwarerr.sty\endcsname
67 \ProvidesPackage{infwarerr}%
68 [2010/04/08 v1.3 Providing info/warning/error messages (H0)]%

```

## 2.2 Catcodes

```

69 \begingroup\catcode61\catcode48\catcode32=10\relax%
70 \catcode13=5 % ^~M
71 \endlinechar=13 %
72 \catcode123=1 % {
73 \catcode125=2 % }
74 \catcode64=11 % @
75 \def\x{\endgroup
76   \expandafter\edef\csname InfWarErr@AtEnd\endcsname{%
77     \endlinechar=\the\endlinechar\relax
78     \catcode13=\the\catcode13\relax
79     \catcode32=\the\catcode32\relax
80     \catcode35=\the\catcode35\relax
81     \catcode61=\the\catcode61\relax
82     \catcode64=\the\catcode64\relax
83     \catcode123=\the\catcode123\relax
84     \catcode125=\the\catcode125\relax
85   }%
86 }%
87 \x\catcode61\catcode48\catcode32=10\relax%
88 \catcode13=5 % ^~M
89 \endlinechar=13 %
90 \catcode35=6 % #
91 \catcode64=11 % @
92 \catcode123=1 % {
93 \catcode125=2 % }
94 \def\TMP@EnsureCode#1#2{%
95   \edef\InfWarErr@AtEnd{%
96     \InfWarErr@AtEnd

```

```

97     \catcode#1=\the\catcode#1\relax
98   }%
99   \catcode#1=#2\relax
100 }
101 \TMP@EnsureCode{10}{12}% ^^J
102 \TMP@EnsureCode{39}{12}% '
103 \TMP@EnsureCode{40}{12}% (
104 \TMP@EnsureCode{41}{12}% )
105 \TMP@EnsureCode{44}{12}% ,
106 \TMP@EnsureCode{45}{12}% -
107 \TMP@EnsureCode{46}{12}% .
108 \TMP@EnsureCode{58}{12}% :
109 \TMP@EnsureCode{60}{12}% <
110 \TMP@EnsureCode{62}{12}% >
111 \TMP@EnsureCode{94}{7}% ^
112 \edef\InfWarErr@AtEnd{\InfWarErr@AtEnd\noexpand\endinput}

```

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

```

113 \begingroup
114   \let\x=Y%
115   \def\check#1{%
116     \@check{Generic}{#1}%
117     \@check{Package}{#1}%
118     \@check{Class}{#1}%
119   }%
120   \def\@check#1#2{%
121     \expandafter\ifx\csname #1#2\endcsname\relax
122       \let\x=N%
123     \fi
124   }%
125   \check{Info}%
126   \check{Warning}%
127   \check{Error}%
128   \@check{Package}{WarningNoLine}%
129   \@check{Class}{WarningNoLine}%
130   \@check{}{@ehc}%
131   \@check{}{@ehd}%
132 \expandafter\endgroup
133 \ifx\x Y%

```

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

```

\@PackageInfo
134   \let\@PackageInfo\PackageInfo

\@PackageInfoNoLine
135   \def\@PackageInfoNoLine#1#2{%
136     \PackageInfo{#1}{#2@gobble}%
137   }%

\@ClassInfo
138   \let\@ClassInfo\ClassInfo

\@ClassInfoNoLine
139   \def\@ClassInfoNoLine#1#2{%
140     \ClassInfo{#1}{#2@gobble}%
141   }%

\@PackageWarning
142   \let\@PackageWarning\PackageWarning

\@PackageWarningNoLine
143   \let\@PackageWarningNoLine\PackageWarningNoLine

```

```

\@ClassWarning
144 \let\@ClassWarning\ClassWarning

\@ClassWarningNoLine
145 \let\@ClassWarningNoLine\ClassWarningNoLine

\@PackageError
146 \let\@PackageError\PackageError

\@ClassError
147 \let\@ClassError\ClassError

148 \expandafter\InfWarErr@AtEnd
149 \fi%

```

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

```

\InfWarErr@online
150 \begingroup\expandafter\endgroup
151 \ifcase\expandafter\ifx\csname inputlineno\endcsname\relax
152     0 %
153     \else
154         \ifnum\inputlineno<0 %
155             0 %
156         \else
157             1 %
158         \fi
159     \fi
160 \def\InfWarErr@online{}%
161 \else
162 \def\InfWarErr@online{ on input line \the\inputlineno}%
163 \fi

\InfWarErr@protected
164 \begingroup\expandafter\expandafter\expandafter\endgroup
165 \expandafter\ifx\csname protected\endcsname\relax
166 \def\InfWarErr@protected{}%
167 \else
168 \let\InfWarErr@protected\protected
169 \fi

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

\InfWarErr@ehc
171 \def\InfWarErr@ehc{%
172 Try typing \space <return> \space to proceed.\MessageBreak
173 If that doesn't work, type \space X <return> \space to quit.%
174 }

\InfWarErr@ehd
175 \def\InfWarErr@ehd{%
176 You're in trouble here. \space\@ehc
177 }

\InfWarErr@GenericInfo
178 \InfWarErr@protected\def\InfWarErr@GenericInfo#1#2{%
179 \begingroup
180 \let\online\InfWarErr@online
181 \long\def\@gobble##1{%
182 \def\space{ }%

```

```

183 \def\@spaces{\space\space\space\space}%
184 \def\MessageBreak{^^J#1}%
185 \newlinechar=10 %
186 \escapechar=92 %
187 \immediate\write-1{#2\on@line.}%
188 \endgroup
189 }

```

\InfWarErr@GenericWarning

```

190 \InfWarErr@protected\def\InfWarErr@GenericWarning#1#2{%
191 \begingroup
192 \let\on@line\InfWarErr@online
193 \long\def\@gobble##1{%
194 \def\space{ }%
195 \def\@spaces{\space\space\space\space}%
196 \def\MessageBreak{^^J#1}%
197 \newlinechar=10 %
198 \escapechar=92 %
199 \immediate\write\InfWarErr@unused{^^J#2\on@line.^^J}%
200 \endgroup
201 }

```

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

```

202 \InfWarErr@protected\def\InfWarErr@GenericError#1#2#3#4{%
203 \begingroup
204 \immediate\write\InfWarErr@unused{%
205 \def\space{ }%
206 \def\@spaces{\space\space\space\space}%
207 \let\@ehc\InfWarErr@ehc
208 \let\@ehd\InfWarErr@ehd
209 \let\on@line\InfWarErr@online
210 \def\MessageBreak{^^J}%
211 \newlinechar=10 %
212 \escapechar=92 %
213 \edef\InfWarErr@help{\noexpand\errhelp{#4}}%
214 \InfWarErr@help
215 \def\MessageBreak{^^J#1}%
216 \errmessage{%
217 #2.^^J^^J%
218 #3^^J%
219 Type \space H <return> \space for immediate help.^^J ...^^J%
220 }%
221 \endgroup
222 }

```

\@PackageInfo

```

223 \def\@PackageInfo#1#2{%
224 \InfWarErr@GenericInfo{%
225 (#1) \@spaces\@spaces\@spaces
226 }{%
227 Package #1 Info: #2%
228 }%
229 }

```

\@PackageInfoNoLine

```

230 \def\@PackageInfoNoLine#1#2{%
231 \@PackageInfo{#1}{#2\@gobble}%
232 }

```

\@ClassInfo

```

233 \def\@ClassInfo#1#2{%

```

```

234 \InfWarErr@GenericInfo{%
235     (#1) \space\space\@spaces\@spaces
236 }{%
237     Class #1 Info: #2%
238 }%
239 }

\@ClassInfoNoLine
240 \def\@ClassInfoNoLine#1#2{%
241     \@ClassInfo{#1}{#2\@gobble}%
242 }

\@PackageWarning
243 \def\@PackageWarning#1#2{%
244     \InfWarErr@GenericWarning{%
245         (#1)\@spaces\@spaces\@spaces\@spaces
246     }{%
247         Package #1 Warning: #2%
248     }%
249 }

\@PackageWarningNoLine
250 \def\@PackageWarningNoLine#1#2{%
251     \@PackageWarning{#1}{#2\@gobble}%
252 }

\@ClassWarning
253 \def\@ClassWarning#1#2{%
254     \InfWarErr@GenericWarning{%
255         (#1) \space\@spaces\@spaces\@spaces
256     }{%
257         Class #1 Warning: #2%
258     }%
259 }

\@ClassWarningNoLine
260 \def\@ClassWarningNoLine#1#2{%
261     \@ClassWarning{#1}{#2\@gobble}%
262 }

\@PackageError
263 \def\@PackageError#1#2#3{%
264     \InfWarErr@GenericError{%
265         (#1)\@spaces\@spaces\@spaces\@spaces
266     }{%
267         Package #1 Error: #2%
268     }{%
269         See the #1 package documentation for explanation.%
270     }{#3}%
271 }

\@ClassError
272 \def\@ClassError#1#2#3{%
273     \InfWarErr@GenericError{%
274         (#1) \space\@spaces\@spaces\@spaces
275     }{%
276         Class #1 Error: #2%
277     }{%
278         See the #1 class documentation for explanation.%
279     }{#3}%
280 }

281 \InfWarErr@AtEnd%
282 \</package>

```

## 3 Test

### 3.1 Catcode checks for loading

```
283 {*test1}
284 \catcode`\{=1 %
285 \catcode`\}=2 %
286 \catcode`\#=6 %
287 \catcode`\@=11 %
288 \expandafter\ifx\csname count@\endcsname\relax
289 \countdef\count@=255 %
290 \fi
291 \expandafter\ifx\csname @gobble\endcsname\relax
292 \long\def\@gobble#1{%
293 \fi
294 \expandafter\ifx\csname @firstofone\endcsname\relax
295 \long\def\@firstofone#1{#1}%
296 \fi
297 \expandafter\ifx\csname loop\endcsname\relax
298 \expandafter\@firstofone
299 \else
300 \expandafter\@gobble
301 \fi
302 {%
303 \def\loop#1\repeat{%
304 \def\body{#1}%
305 \iterate
306 }%
307 \def\iterate{%
308 \body
309 \let\next\iterate
310 \else
311 \let\next\relax
312 \fi
313 \next
314 }%
315 \let\repeat=\fi
316 }%
317 \def\RestoreCatcodes{}
318 \count@=0 %
319 \loop
320 \edef\RestoreCatcodes{%
321 \RestoreCatcodes
322 \catcode\the\count@=\the\catcode\count@\relax
323 }%
324 \ifnum\count@<255 %
325 \advance\count@ 1 %
326 \repeat
327
328 \def\RangeCatcodeInvalid#1#2{%
329 \count@=#1\relax
330 \loop
331 \catcode\count@=15 %
332 \ifnum\count@<#2\relax
333 \advance\count@ 1 %
334 \repeat
335 }
336 \def\RangeCatcodeCheck#1#2#3{%
337 \count@=#1\relax
338 \loop
339 \ifnum#3=\catcode\count@
340 \else
```



```

341 \errmessage{%
342     Character \the\count@\space
343     with wrong catcode \the\catcode\count@\space
344     instead of \number#3%
345 }%
346 \fi
347 \ifnum\count@<#2\relax
348 \advance\count@ 1 %
349 \repeat
350 }
351 \def\space{ }
352 \expandafter\ifx\csname LoadCommand\endcsname\relax
353 \def\LoadCommand{\input infwarerr.sty\relax}%
354 \fi
355 \def\Test{%
356 \RangeCatcodeInvalid{0}{47}%
357 \RangeCatcodeInvalid{58}{64}%
358 \RangeCatcodeInvalid{91}{96}%
359 \RangeCatcodeInvalid{123}{255}%
360 \catcode`\@=12 %
361 \catcode`\=0 %
362 \catcode`\%=14 %
363 \LoadCommand
364 \RangeCatcodeCheck{0}{36}{15}%
365 \RangeCatcodeCheck{37}{37}{14}%
366 \RangeCatcodeCheck{38}{47}{15}%
367 \RangeCatcodeCheck{48}{57}{12}%
368 \RangeCatcodeCheck{58}{63}{15}%
369 \RangeCatcodeCheck{64}{64}{12}%
370 \RangeCatcodeCheck{65}{90}{11}%
371 \RangeCatcodeCheck{91}{91}{15}%
372 \RangeCatcodeCheck{92}{92}{0}%
373 \RangeCatcodeCheck{93}{96}{15}%
374 \RangeCatcodeCheck{97}{122}{11}%
375 \RangeCatcodeCheck{123}{255}{15}%
376 \RestoreCatcodes
377 }
378 \Test
379 \csname @@end\endcsname
380 \end
381 </test1>

```

## 3.2 Macro tests

```

382 <*test2>
383 \let\PackageInfo\relax
384 \input infwarerr.sty\relax
385 \let\OrgWrite\write
386 \chardef\WriteUnused=16 %
387 \def\msg#{\immediate\OrgWrite\WriteUnused}
388 \msg{File: infwarerr-test2.tex 2010/04/08 v1.3 Test file for plain-TeX}
389 \catcode64=11 %
390 </test2>

391 <*test3>
392 \NeedsTeXFormat{LaTeX2e}
393 \ProvidesFile{infwarerr-test3.tex}[2010/04/08 v1.3 Test file for LaTeX]
394 \RequirePackage{infwarerr}[2010/04/08]
395 \makeatletter
396 \let\OrgWrite\write
397 \chardef\WriteUnused=\@unused
398 \def\msg#{\immediate\OrgWrite\WriteUnused}
399 </test3>

```

```

400 <*test2j test3>
401 \def\pkgname{PACKAGE}
402 \def\clsname{CLASS}
403 \def\msgtext{MESSAGE\MessageBreak SECOND LINE}
404 \def\hlptext{HELP\MessageBreak SECOND LINE}
405
406 \let\OrgWrite\write
407 \let\OrgErrMsg\errmessage
408
409 \def\write#1{%
410   \@write{#1}%
411 }
412 \def\@write#1#2{%
413   \OrgWrite#1{#2}%
414   \xdef\WriteNum{\number#1}%
415   \xdef\WriteText{#2}%
416 }
417 \def\TestFailed#1#2{%
418   \OrgErrMsg{Test (\string#1) failed: #2}%
419 }
420 \def\TestWrite#1#2#3{%
421   \ifnum\WriteNum=#2\relax
422   \else
423     \TestFailed#1{write stream number: \WriteNum <> #2}%
424   \fi
425   \begingroup
426     \let~\space
427     \edef\TestString{#3}%
428     \ifx\WriteText\TestString
429     \else
430       \msg{}%
431       \msg{[\WriteText]}%
432       \msg{<>}%
433       \msg{[#3]}% hash-ok
434       \TestFailed#1{write text}%
435     \fi
436   \endgroup
437 }
438 \newtoks\ErrHelpToks
439 \def\errmessage#1{%
440   \global\ErrHelpToks=\expandafter{\the\errhelp}%
441   \msg{***** error message *****}%
442   \msg{! #1}%
443   \msg{***** help message *****}%
444   \msg{\the\errhelp}%
445   \msg{*****}%
446   \xdef\ErrorMessage{! #1}%
447 }
448 \def\TestInfo{%
449   \@PackageInfo\pkgname\msgtext
450   \TestWrite\@PackageInfo{-1}{%
451     Package PACKAGE Info: MESSAGE^^J%
452     (PACKAGE)~~~~~SECOND LINE on input line \the\inputlineno.%
453   }%
454   \@PackageInfoNoLine\pkgname\msgtext
455   \TestWrite\@PackageInfoNoLine{-1}{%
456     Package PACKAGE Info: MESSAGE^^J%
457     (PACKAGE)~~~~~SECOND LINE.%
458   }%
459   \@ClassInfo\clsname\msgtext
460   \TestWrite\@ClassInfo{-1}{%
461     Class CLASS Info: MESSAGE^^J%

```

```

462 (CLASS)~~~~~SECOND LINE on input line \the\inputlineno.%
463 }%
464 \@ClassInfoNoLine\clsname\msgtext
465 \TestWrite\@ClassInfoNoLine{-1}{%
466 Class CLASS Info: MESSAGE^^J%
467 (CLASS)~~~~~SECOND LINE.%
468 }%
469 }
470 \def\TestWarning{%
471 \@PackageWarning\pkgname\msgtext
472 \TestWrite\@PackageWarning\WriteUnused{%
473 ^^J%
474 Package PACKAGE Warning: MESSAGE^^J%
475 (PACKAGE)~~~~~SECOND LINE on input line \the\inputlineno.%
476 ^^J%
477 }%
478 \@PackageWarningNoLine\pkgname\msgtext
479 \TestWrite\@PackageWarningNoLine\WriteUnused{%
480 ^^J%
481 Package PACKAGE Warning: MESSAGE^^J%
482 (PACKAGE)~~~~~SECOND LINE.%
483 ^^J%
484 }%
485 \@ClassWarning\clsname\msgtext
486 \TestWrite\@ClassWarning\WriteUnused{%
487 ^^J%
488 Class CLASS Warning: MESSAGE^^J%
489 (CLASS)~~~~~SECOND LINE on input line \the\inputlineno.%
490 ^^J%
491 }%
492 \@ClassWarningNoLine\clsname\msgtext
493 \TestWrite\@ClassWarningNoLine\WriteUnused{%
494 ^^J%
495 Class CLASS Warning: MESSAGE^^J%
496 (CLASS)~~~~~SECOND LINE.%
497 ^^J%
498 }%
499 }
500 \def\TestError{%
501 \test3 \let\write\OrgWrite
502 \test3 \def\plaindots{}
503 \test2 \def\plaindots{.^^J ...^^J}
504 \@PackageError\pkgname\msgtext\hlptext
505 \@TestError\@PackageError{%
506 ! Package PACKAGE Error: MESSAGE^^J%
507 (PACKAGE)~~~~~SECOND LINE.^^J%
508 ^^J%
509 See the PACKAGE package documentation for explanation.^^J%
510 Type~H <return>~~for immediate help\plaindots
511 }{%
512 HELP^^J%
513 SECOND LINE%
514 }%
515 \@ClassError\clsname\msgtext\hlptext
516 \@TestError\@ClassError{%
517 ! Class CLASS Error: MESSAGE^^J%
518 (CLASS)~~~~~SECOND LINE.^^J%
519 ^^J%
520 See the CLASS class documentation for explanation.^^J%
521 Type~H <return>~~for immediate help\plaindots
522 }{%
523 HELP^^J%

```

```

524     SECOND LINE%
525 }%
526 \@PackageError\pkgname\msgtext\@ehc
527 \@TestError\@PackageError{%
528     ! Package PACKAGE Error: MESSAGE^^J%
529     (PACKAGE)~~~~~SECOND LINE.^^J%
530     ^^J%
531     See the PACKAGE package documentation for explanation.^^J%
532     Type~H <return>~~for immediate help\plaindots
533 }{%
534     Try typing~~<return>~~to proceed.^^J%
535     If that doesn't work, type~X <return>~~to quit.%
536 }%
537 \@PackageError\pkgname\msgtext\@ehd
538 \@TestError\@PackageError{%
539     ! Package PACKAGE Error: MESSAGE^^J%
540     (PACKAGE)~~~~~SECOND LINE.^^J%
541     ^^J%
542     See the PACKAGE package documentation for explanation.^^J%
543     Type~H <return>~~for immediate help\plaindots
544 }{%
545     You're in trouble here.~~Try typing~~<return>~~to proceed.^^J%
546     If that doesn't work, type~X <return>~~to quit.%
547 }%
548 }
549 \def\@TestError#1#2#3{%
550     \begingroup
551     \let~\space
552     \edef\x{#2}%
553     \test3 \@onelevel@sanitize\x
554     \test3 \@onelevel@sanitize\ErrMessage
555     \ifx\x\ErrMessage
556     \else
557         \msg{}%
558         \msg{[\ErrMessage]}%
559         \msg{<>}%
560         \msg{[\x]}%
561         \TestFailed#1{errmessage text}%
562     \fi
563     \edef\x{the\ErrHelpToks}%
564     \edef\y{#3}%
565     \ifx\x\y
566     \else
567         \msg{}%
568         \msg{[the\ErrHelpToks]}%
569         \msg{<>}%
570         \msg{[\y]}%
571         \TestFailed#1{errhelp text}%
572     \fi
573     \endgroup
574 }
575 \TestInfo
576 \TestWarning
577 \TestError
578 </test2j test3>
579 <test2>\end
580 <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:install/macros/latex/contrib/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
```

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

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

### 4.3 Package installation

**Unpacking.** The `.dtx` file is a self-extracting `docstrip` archive. The files are extracted by running the `.dtx` through plain T<sub>E</sub>X:

```
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):

<code>infwarerr.sty</code>	→ <code>tex/generic/oberdiek/infwarerr.sty</code>
<code>infwarerr.pdf</code>	→ <code>doc/latex/oberdiek/infwarerr.pdf</code>
<code>test/infwarerr-test1.tex</code>	→ <code>doc/latex/oberdiek/test/infwarerr-test1.tex</code>
<code>test/infwarerr-test2.tex</code>	→ <code>doc/latex/oberdiek/test/infwarerr-test2.tex</code>
<code>test/infwarerr-test3.tex</code>	→ <code>doc/latex/oberdiek/test/infwarerr-test3.tex</code>
<code>infwarerr.dtx</code>	→ <code>source/latex/oberdiek/infwarerr.dtx</code>

If you have a `docstrip.cfg` that configures and enables `docstrip`’s TDS installing feature, then some files can already be in the right place, see the documentation of `docstrip`.

### 4.4 Refresh file name databases

If your T<sub>E</sub>X distribution (teT<sub>E</sub>X, miK<sub>T</sub>E<sub>X</sub>, ...) relies on file name databases, you must refresh these. For example, teT<sub>E</sub>X users run `texhash` or `mktextlsr`.

---

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

## 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 L<sup>A</sup>T<sub>E</sub>X.** The `.dtx` chooses its action depending on the format:

**plain T<sub>E</sub>X:** Run `docstrip` and extract the files.

**L<sup>A</sup>T<sub>E</sub>X:** Generate the documentation.

If you insist on using L<sup>A</sup>T<sub>E</sub>X for `docstrip` (really, `docstrip` does not need L<sup>A</sup>T<sub>E</sub>X), 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 `pdfLATEX`:

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

## 5 Catalogue

The following XML file can be used as source for the [T<sub>E</sub>X Catalogue](#). The elements `caption` and `description` are imported from the original XML file from the Catalogue. The name of the XML file in the Catalogue is `infwarerr.xml`.

```
581 <?catalogue>
582 <?xml version='1.0' encoding='us-ascii'?>
583 <!DOCTYPE entry SYSTEM 'catalogue.dtd'>
584 <entry datestamp='$Date$' modifier='$Author$' id='infwarerr'>
585   <name>infwarerr</name>
586   <caption>Complete set of information/warning/error messages.</caption>
587   <authorref id='auth:oberdiek'>/>
588   <copyright owner='Heiko Oberdiek' year='2007,2010'>/>
589   <license type='lppl1.3'>/>
590   <version number='1.3'>/>
591   <description>
592     This package provides a complete set of macros for information,
593     warning and error messages. Under LATEX, the commands are
594     wrappers for the corresponding LATEX commands; under Plain TEX
595     they are also available as complete implementations.
596   <p/>
597   The package is part of the <xref refid='oberdiek'>oberdiek</xref>
598   bundle.
599 </description>
600 <documentation details='Package documentation'
601   href='ctan:/macros/latex/contrib/oberdiek/infwarerr.pdf'>/>
```

```

602 <ctan file='true' path='/macros/latex/contrib/oberdiek/infwarerr.dtx'>
603 <miktex location='oberdiek'>
604 <texlive location='oberdiek'>
605 <install path='/macros/latex/contrib/oberdiek/oberdiek.tds.zip'>
606 </entry>
607 </catalogue>

```

## 6 History

[2007/04/26 v1.0]

- First version.

[2007/06/14 v1.1]

- Small fixes.

[2007/09/09 v1.2]

- Catcode section extended.

[2010/04/08 v1.3]

- Ensure `\escapechar=92` (backslash) for infos/warnings/errors.

## 7 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; plain numbers refer to the code lines where the entry is used.

Symbols	
<code>\#</code> .....	286
<code>\%</code> .....	362
<code>\@</code> .....	287, 360
<code>\@@end</code> .....	580
<code>\@ClassError</code> .....	<u>147</u> , <u>272</u> , 515, 516
<code>\@ClassInfo</code> ...	<u>138</u> , <u>233</u> , <u>241</u> , 459, 460
<code>\@ClassInfoNoLine</code> .	<u>139</u> , <u>240</u> , 464, 465
<code>\@ClassWarning</code> .	<u>144</u> , <u>253</u> , <u>261</u> , 485, 486
<code>\@ClassWarningNoLine</code>	<u>145</u> , <u>260</u> , 492, 493
<code>\@PackageError</code> .....	<u>146</u> , 263, 504, 505, 526, 527, 537, 538
<code>\@PackageInfo</code> ..	<u>134</u> , <u>223</u> , <u>231</u> , 449, 450
<code>\@PackageInfoNoLine</code>	<u>135</u> , <u>230</u> , 454, 455
<code>\@PackageWarning</code>	<u>142</u> , <u>243</u> , <u>251</u> , 471, 472
<code>\@PackageWarningNoLine</code> .....	<u>143</u> , <u>250</u> , 478, 479
<code>\@TestError</code> ...	505, 516, 527, 538, 549
<code>\@check</code> .....	116, 117, 118, 120, 128, 129, 130, 131
<code>\@ehc</code> .....	176, 207, 526
<code>\@ehd</code> .....	208, 537
<code>\@firstofone</code> .....	295, 298
<code>\@gobble</code> .....	136, 140, 181, 193, 231, 241, 251, 261, 292, 300
<code>\@onelevel@sanitize</code> .....	553, 554
<code>\@spaces</code> .....	183, 195, 206, 225, 235, 245, 255, 265, 274
<code>\@undefined</code> .....	58
<code>\@unused</code> .....	397
<code>\@write</code> .....	410, 412
<code>\%</code> .....	361
<code>\{</code> .....	284
<code>\}</code> .....	285
A	
<code>\advance</code> .....	325, 333, 348
<code>\aftergroup</code> .....	29
B	
<code>\body</code> .....	304, 308
C	
<code>\catcode</code> ..	2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13, 33, 34, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 69, 70, 72, 73, 74, 78, 79, 80, 81, 82, 83, 84, 87, 88, 90, 91, 92, 93, 97, 99, 284, 285, 286, 287, 322, 331, 339, 343, 360, 361, 362, 389
<code>\chardef</code> .....	170, 386, 397
<code>\check</code> .....	115, 125, 126, 127
<code>\ClassError</code> .....	147
<code>\ClassInfo</code> .....	138, 140
<code>\ClassWarning</code> .....	144
<code>\ClassWarningNoLine</code> .....	145
<code>\clsname</code> ..	402, 459, 464, 485, 492, 515

<code>\count@</code> . . . . .	289, 318, 322, 324, 325, 329, 331, 332, 333, 337, 339, 342, 343, 347, 348
<code>\countdef</code> . . . . .	289
<code>\csname</code> . . . . .	14, 21, 50, 66, 76, 121, 151, 165, 288, 291, 294, 297, 352, 379
<b>E</b>	
<code>\empty</code> . . . . .	17, 18
<code>\end</code> . . . . .	380, 579
<code>\endcsname</code> 14, 21, 50, 66, 76, 121, 151, 165, 288, 291, 294, 297, 352, 379	
<code>\endinput</code> . . . . .	29, 112
<code>\newlinechar</code> . . . . .	4, 35, 71, 77, 89
<code>\errhelp</code> . . . . .	213, 440, 444
<code>\ErrHelpToks</code> . . . . .	438, 440, 563, 568
<code>\ErrMsgMessage</code> . . . . .	446, 554, 555, 558
<code>\errmessage</code> . . . . .	216, 341, 407, 439
<code>\escapechar</code> . . . . .	186, 198, 212
<b>H</b>	
<code>\hlptext</code> . . . . .	404, 504, 515
<b>I</b>	
<code>\ifcase</code> . . . . .	151
<code>\ifnum</code> . . . . .	154, 324, 332, 339, 347, 421
<code>\ifx</code> . . . . .	15, 18, 21, 50, 58, 61, 121, 133, 151, 165, 288, 291, 294, 297, 352, 428, 555, 565
<code>\immediate</code> 23, 52, 187, 199, 204, 387, 398	
<code>\InfWarErr@AtEnd</code> . . . . .	95, 96, 112, 148, 281
<code>\InfWarErr@ehc</code> . . . . .	171, 207
<code>\InfWarErr@ehd</code> . . . . .	175, 208
<code>\InfWarErr@GenericError</code> 202, 264, 273	
<code>\InfWarErr@GenericInfo</code> . . . . .	178, 224, 234
<code>\InfWarErr@GenericWarning</code> . . . . .	190, 244, 254
<code>\InfWarErr@help</code> . . . . .	213, 214
<code>\InfWarErr@online</code> . . . . .	150, 180, 192, 209
<code>\InfWarErr@protected</code> 164, 178, 190, 202	
<code>\InfWarErr@unused</code> . . . . .	170, 199, 204
<code>\input</code> . . . . .	353, 384
<code>\inputlineno</code> 154, 162, 452, 462, 475, 489	
<code>\iterate</code> . . . . .	305, 307, 309
<b>L</b>	
<code>\LoadCommand</code> . . . . .	353, 363
<code>\loop</code> . . . . .	303, 319, 330, 338
<b>M</b>	
<code>\makeatletter</code> . . . . .	395
<code>\MessageBreak</code> . . . . .	172, 184, 196, 210, 215, 403, 404
<code>\msg</code> 387, 388, 398, 430, 431, 432, 433, 441, 442, 443, 444, 445, 557, 558, 559, 560, 567, 568, 569, 570	
<code>\msgtext</code> 403, 449, 454, 459, 464, 471, 478, 485, 492, 504, 515, 526, 537	
<b>N</b>	
<code>\NeedsTeXFormat</code> . . . . .	392
<code>\newlinechar</code> . . . . .	185, 197, 211
<code>\newtoks</code> . . . . .	438
<code>\next</code> . . . . .	309, 311, 313
<code>\number</code> . . . . .	344, 414
<b>O</b>	
<code>\on@line</code> . . . . .	180, 187, 192, 199, 209
<code>\OrgErrMsg</code> . . . . .	407, 418
<code>\OrgWrite</code> . . . . .	385, 387, 396, 398, 406, 413, 501
<b>P</b>	
<code>\PackageError</code> . . . . .	146
<code>\PackageInfo</code> . . . . .	26, 134, 136, 383
<code>\PackageWarning</code> . . . . .	142
<code>\PackageWarningNoLine</code> . . . . .	143
<code>\pkgname</code> . . . . .	401, 449, 454, 471, 478, 504, 526, 537
<code>\plaindots</code> . . . . .	502, 503, 510, 521, 532, 543
<code>\protected</code> . . . . .	168
<code>\ProvidesFile</code> . . . . .	393
<code>\ProvidesPackage</code> . . . . .	19, 67
<b>R</b>	
<code>\RangeCatcodeCheck</code> . . . . .	336, 364, 365, 366, 367, 368, 369, 370, 371, 372, 373, 374, 375
<code>\RangeCatcodeInvalid</code> . . . . .	328, 356, 357, 358, 359
<code>\repeat</code> . . . . .	303, 315, 326, 334, 349
<code>\RequirePackage</code> . . . . .	394
<code>\RestoreCatcodes</code> . . . . .	317, 320, 321, 376
<b>S</b>	
<code>\space</code> . . . . .	172, 173, 176, 182, 183, 194, 195, 205, 206, 219, 235, 255, 274, 342, 343, 351, 426, 551
<b>T</b>	
<code>\Test</code> . . . . .	355, 378
<code>\TestError</code> . . . . .	500, 577
<code>\TestFailed</code> . . . . .	417, 423, 434, 561, 571
<code>\TestInfo</code> . . . . .	448, 575
<code>\TestString</code> . . . . .	427, 428
<code>\TestWarning</code> . . . . .	470, 576
<code>\TestWrite</code> . . . . .	420, 450, 455, 460, 465, 472, 479, 486, 493
<code>\the</code> . . . . .	77, 78, 79, 80, 81, 82, 83, 84, 97, 162, 322, 342, 343, 440, 444, 452, 462, 475, 489, 563, 568
<code>\TMP@EnsureCode</code> . . . . .	94, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111
<b>W</b>	
<code>\write</code> . . . . .	23, 52, 187, 199, 204, 385, 396, 406, 409, 501
<code>\WriteNum</code> . . . . .	414, 421, 423
<code>\WriteText</code> . . . . .	415, 428, 431
<code>\WriteUnused</code> . . . . .	386, 387, 397, 398, 472, 479, 486, 493
<b>X</b>	
<code>\x</code> . . . . .	14, 15, 18, 22, 26, 28, 51, 56, 66, 75, 87, 114, 122, 133, 552, 553, 555, 560, 563, 565
<b>Y</b>	
<code>\y</code> . . . . .	564, 565, 570