

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 \TeX .

Contents

1	Documentation	1
2	Implementation	2
2.1	Reload check and package identification	2
2.2	Catcodes	3
2.3	\LaTeX detection	3
2.4	Version for \LaTeX	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
	[2010/04/08 v1.3]	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 \LaTeX '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 [2010/04/08 v1.3 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 T_EX

`\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 \escapechar=92 %
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\space}%
```

```

182 \def\MessageBreak{^^J#1}%
183 \newlinechar=10 %
184 \escapechar=92 %
185 \immediate\write\InfWarErr@unused{^^J#2\on@line.^^J}%
186 \endgroup
187 }

```

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

```

188 \InfWarErr@protected\def\InfWarErr@GenericError#1#2#3#4{%
189 \begingroup
190 \immediate\write\InfWarErr@unused{}%
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 \newlinechar=10 %
198 \escapechar=92 %
199 \edef\InfWarErr@help{\noexpand\errhelp{#4}}%
200 \InfWarErr@help
201 \def\MessageBreak{^^J#1}%
202 \errmessage{%
203 #2.^^J^^J%
204 #3^^J%
205 Type \space H <return> \space for immediate help.^^J ...^^J%
206 }%
207 \endgroup
208 }

```

`\@PackageInfo`

```

209 \def\@PackageInfo#1#2{%
210 \InfWarErr@GenericInfo{%
211 (#1) \@spaces\@spaces\@spaces
212 }{%
213 Package #1 Info: #2%
214 }%
215 }

```

`\@PackageInfoNoLine`

```

216 \def\@PackageInfoNoLine#1#2{%
217 \@PackageInfo{#1}{#2@gobble}%
218 }

```

`\@ClassInfo`

```

219 \def\@ClassInfo#1#2{%
220 \InfWarErr@GenericInfo{%
221 (#1) \space\space\@spaces\@spaces
222 }{%
223 Class #1 Info: #2%
224 }%
225 }

```

`\@ClassInfoNoLine`

```

226 \def\@ClassInfoNoLine#1#2{%
227 \@ClassInfo{#1}{#2@gobble}%
228 }

```

`\@PackageWarning`

```

229 \def\@PackageWarning#1#2{%

```

```

230 \InfWarErr@GenericWarning{%
231   (#1)\@spaces\@spaces\@spaces\@spaces
232 }{%
233   Package #1 Warning: #2%
234 }%
235 }

```

\@PackageWarningNoLine

```

236 \def\@PackageWarningNoLine#1#2{%
237   \@PackageWarning{#1}{#2@gobble}%
238 }

```

\@ClassWarning

```

239 \def\@ClassWarning#1#2{%
240   \InfWarErr@GenericWarning{%
241     (#1) \space\@spaces\@spaces\@spaces
242   }{%
243     Class #1 Warning: #2%
244   }%
245 }

```

\@ClassWarningNoLine

```

246 \def\@ClassWarningNoLine#1#2{%
247   \@ClassWarning{#1}{#2@gobble}%
248 }

```

\@PackageError

```

249 \def\@PackageError#1#2#3{%
250   \InfWarErr@GenericError{%
251     (#1)\@spaces\@spaces\@spaces\@spaces
252   }{%
253     Package #1 Error: #2%
254   }{%
255     See the #1 package documentation for explanation.%
256   }{#3}%
257 }

```

\@ClassError

```

258 \def\@ClassError#1#2#3{%
259   \InfWarErr@GenericError{%
260     (#1) \space\@spaces\@spaces\@spaces
261   }{%
262     Class #1 Error: #2%
263   }{%
264     See the #1 class documentation for explanation.%
265   }{#3}%
266 }

```

```

267 \InfWarErr@AtEnd
268 </package>

```

3 Test

3.1 Catcode checks for loading

```

269 (*test1)
270 \catcode'\{=1 %
271 \catcode'\}=2 %
272 \catcode'\#=6 %
273 \catcode'\@=11 %

```

```

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

```



```

336 \catcode'\]=12 %
337 \catcode'\%=14 %
338 \catcode'\ =10 %
339 \catcode13=5 %
340 \LoadCommand
341 \RestoreCatcodes
342 }
343 \Test
344 \csname @@end\endcsname
345 \end
346 </test1>

```

3.2 Macro tests

```

347 (*test2)
348 \let\PackageInfo\relax
349 \input infwarerr.sty\relax
350 \let\OrgWrite\write
351 \chardef\WriteUnused=16 %
352 \def\msg#\{ \immediate\OrgWrite\WriteUnused}
353 \msg{File: infwarerr-test2.tex 2010/04/08 v1.3 Test file for plain-TeX}
354 \catcode64=11 %
355 </test2>

356 (*test3)
357 \NeedsTeXFormat{LaTeX2e}
358 \ProvidesFile{infwarerr-test3.tex}[2010/04/08 v1.3 Test file for LaTeX]
359 \RequirePackage{infwarerr}[2010/04/08]
360 \makeatletter
361 \let\OrgWrite\write
362 \chardef\WriteUnused=\@unused
363 \def\msg#\{ \immediate\OrgWrite\WriteUnused}
364 </test3>

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

```

```

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

```

```

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

```

```

519 \test3) \@onelevel@sanitize\ErrorMessage
520   \ifx\x\ErrorMessage
521   \else
522     \msg{ }%
523     \msg{[\ErrorMessage]}%
524     \msg{<>}%
525     \msg{[\x]}%
526     \TestFailed#1{errmessage text}%
527   \fi
528   \edef\x{\the\ErrHelpToks}%
529   \edef\y{#3}%
530   \ifx\x\y
531   \else
532     \msg{ }%
533     \msg{[\the\ErrHelpToks]}%
534     \msg{<>}%
535     \msg{[\y]}%
536     \TestFailed#1{errhelp text}%
537   \fi
538 \endgroup
539 }
540 \TestInfo
541 \TestWarning
542 \TestError
543 </test2 | test3>
544 (test2)\end
545 (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[ftp://ftp.ctan.org/tex-archive/](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.

[2010/04/08 v1.3]

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

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

Symbols	
<code>\#</code>	272, 334
<code>\%</code>	337
<code>\@</code>	273, 330
<code>\@end</code>	545
<code>\@ClassError</code>	132, 258, 480, 481
<code>\@ClassInfo</code> ...	123, 219, 227, 424, 425
<code>\@ClassInfoNoLine</code> .	124, 226, 429, 430
<code>\@ClassWarning</code> <u>129</u> , <u>239</u> , 247, 450, 451	
<code>\@ClassWarningNoLine</code> <u>130</u> , <u>246</u> , 457, 458	
<code>\@PackageError</code>	131,
<u>249</u> , 469, 470, 491, 492, 502, 503	
<code>\@PackageInfo</code> .	119, 209, 217, 414, 415
<code>\@PackageInfoNoLine</code> <u>120</u> , <u>216</u> , 419, 420	
<code>\@PackageWarning</code> <u>127</u> , <u>229</u> , 237, 436, 437	
<code>\@PackageWarningNoLine</code>	<u>128</u> , <u>236</u> , 443, 444
<code>\@TestError</code> ...	470, 481, 492, 503, 514
<code>\@check</code>	101,
102, 103, 105, 113, 114, 115, 116	
<code>\@ehc</code>	162, 193, 491
<code>\@ehd</code>	194, 502
<code>\@firstofone</code>	281, 284
<code>\@gobble</code>	121, 125, 167,
179, 217, 227, 237, 247, 278, 286	
<code>\@onelevel@sanitize</code>	518, 519
<code>\@spaces</code>	169, 181,
192, 211, 221, 231, 241, 251, 260	
<code>\@undefined</code>	52
<code>\@unused</code>	362
<code>\@write</code>	375, 377
<code>\[</code>	335
<code>\]</code>	331
<code>\{</code>	270, 332
<code>\}</code>	271, 333
<code>\]</code>	336
<code>_</code>	338
A	
<code>\advance</code>	311, 319
<code>\aftergroup</code>	26
B	
<code>\body</code>	290, 294
C	
<code>\catcode</code>	3, 4, 5, 6, 7,
8, 9, 17, 31, 32, 33, 34, 35, 36,	
37, 38, 39, 40, 41, 42, 43, 64,	
65, 68, 69, 70, 71, 75, 76, 77,	
78, 82, 84, 270, 271, 272, 273,	
308, 317, 330, 331, 332, 333,	
334, 335, 336, 337, 338, 339, 354	
<code>\chardef</code>	156, 351, 362
<code>\check</code>	100, 110, 111, 112
<code>\ClassError</code>	132
<code>\ClassInfo</code>	123, 125
<code>\ClassWarning</code>	129
<code>\ClassWarningNoLine</code>	130
<code>\clsname</code> ..	367, 424, 429, 450, 457, 480
<code>\count@</code>	275, 304,
308, 310, 311, 315, 317, 318, 319	
<code>\countdef</code>	275
<code>\csname</code> 10, 18, 44, 60, 67, 106, 137,	
151, 274, 277, 280, 283, 322, 344	
E	
<code>\empty</code>	13, 14
<code>\end</code>	345, 544
<code>\endcsname</code> 10, 18, 44, 60, 67, 106, 137,	
151, 274, 277, 280, 283, 322, 344	
<code>\endinput</code>	26, 134

\errhelp	199, 405, 409	\OrgWrite	350, 352, 361, 363, 371, 378, 466
\ErrHelpToks	403, 405, 528, 533		
\ErrMsgMessage	411, 519, 520, 523	P	
\errmessage	202, 372, 404	\PackageError	131
\escapechar	172, 184, 198	\PackageInfo	23, 119, 121, 348
		\PackageWarning	127
H		\PackageWarningNoLine	128
\hlptext	369, 469, 480	\pkgname	366, 414, 419, 436, 443, 469, 491, 502
		\plaindots	467, 468, 475, 486, 497, 508
I		\protected	154
\ifcase	137	\ProvidesFile	358
\ifnum	140, 310, 318, 386	\ProvidesPackage	15, 61
\ifx	11, 14, 18, 44, 52, 55, 106, 118, 137, 151, 274, 277, 280, 283, 322, 393, 520, 530	R	
\immediate	20, 46, 173, 185, 190, 352, 363	\RangeCatcodeInvalid	314, 326, 327, 328, 329
\InfWarErr@AtEnd	80, 81, 133, 267	\repeat	289, 301, 312, 320
\InfWarErr@ehc	157, 193	\RequirePackage	359
\InfWarErr@ehd	161, 194	\RestoreCatcodes	303, 306, 307, 341
\InfWarErr@GenericError	188, 250, 259	S	
\InfWarErr@GenericInfo	164, 210, 220	\space	158, 159, 162, 168, 169, 180, 181, 191, 192, 205, 221, 241, 260, 391, 516
\InfWarErr@GenericWarning	176, 230, 240	T	
\InfWarErr@help	199, 200	\Test	325, 343
\InfWarErr@online	136, 166, 178, 195	\TestError	465, 542
\InfWarErr@protected	150, 164, 176, 188	\TestFailed	382, 388, 399, 526, 536
\InfWarErr@unused	156, 185, 190	\TestInfo	413, 540
\input	323, 349	\TestString	392, 393
\inputlineno	140, 148, 417, 427, 440, 454	\TestWarning	435, 541
\iterate	291, 293, 295	\TestWrite	385, 415, 420, 425, 430, 437, 444, 451, 458
		\the	68, 69, 70, 71, 82, 148, 308, 405, 409, 417, 427, 440, 454, 528, 533
L		\TMP@EnsureCode	79, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97
\LoadCommand	323, 340	W	
\loop	289, 305, 316	\write	20, 46, 173, 185, 190, 350, 361, 371, 374, 466
		\WriteNum	379, 386, 388
M		\WriteText	380, 393, 396
\makeatletter	360	\WriteUnused	351, 352, 362, 363, 437, 444, 451, 458
\MessageBreak	158, 170, 182, 196, 201, 368, 369	X	
\msg	352, 353, 363, 395, 396, 397, 398, 406, 407, 408, 409, 410, 522, 523, 524, 525, 532, 533, 534, 535	\x	10, 11, 14, 19, 23, 25, 45, 50, 60, 66, 74, 99, 107, 118, 517, 518, 520, 525, 528, 530
\msgtext	368, 414, 419, 424, 429, 436, 443, 450, 457, 469, 480, 491, 502	Y	
		\y	529, 530, 535
N			
\NeedsTeXFormat	357		
\newlinechar	171, 183, 197		
\newtoks	403		
\next	295, 297, 299		
\number	379		
O			
\on@line	166, 173, 178, 185, 195		
\OrgErrorMessage	372, 383		