

The infwarerr package

Heiko Oberdiek
<oberdiek@uni-freiburg.de>

2007/06/14 v1.1

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	4
3 Test	7
4 Installation	10
4.1 Download	10
4.2 Bundle installation	11
4.3 Package installation	11
4.4 Refresh file name databases	11
4.5 Some details for the interested	11
5 History	12
[2007/04/26 v1.0]	12
[2007/06/14 v1.1]	12
6 Index	12

1 Documentation

Package infwarerr defines the following set of commands, given as pseudo regular expression:

```
\@(Package|Class)(Info|InfoNoLine|Warning|WarningNoLine|Error)
```

The first argument is the package or class in question. The second argument contains the message text. `\MessageBreak` is supported. In case of error messages, the third argument contains the help text. Supported are L^AT_EX's message texts `\@ehc`:

```
Try typing <return> to proceed
If that doesn't work, type X <return> to quit.
```

and `\@ehd`:

You're in trouble here. Try typing <return> to proceed
 If that doesn't work, type X <return> to quit.

The package is intended for macro writers that want to write code that works for both L^AT_EX and plain-T_EX. If L^AT_EX is detected, then the macros of this package will only be wrappers for L^AT_EX's macros. Otherwise L^AT_EX's macros will be mimicked for use with plain-T_EX.

2 Implementation

1 (*package)

2.1 Reload check and package identification

Reload check, especially if the package is not used with L^AT_EX.

```

2 \begingroup
3 \catcode44 12 % ,
4 \catcode45 12 % -
5 \catcode46 12 % .
6 \catcode58 12 % :
7 \catcode64 11 % @
8 \expandafter\let\expandafter\x\csname ver@infwarerr.sty\endcsname
9 \ifcase 0%
10 \ifx\x\relax % plain
11 \else
12 \ifx\x\empty % LaTeX
13 \else
14 1%
15 \fi
16 \fi
17 \else
18 \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 [2007/06/14 v1.1 Providing info/warning/message (H0)]

```

2.2 Catcodes

```

55 \expandafter\edef\csname InfWarErr@End\endcsname{%
56   \catcode39 \the\catcode39\relax % '
57   \catcode40 \the\catcode40\relax % (
58   \catcode41 \the\catcode41\relax % )
59   \catcode44 \the\catcode44\relax % ,
60   \catcode45 \the\catcode45\relax % -
61   \catcode46 \the\catcode46\relax % .
62   \catcode58 \the\catcode58\relax % :
63   \catcode60 \the\catcode60\relax % <
64   \catcode61 \the\catcode61\relax % =
65   \catcode62 \the\catcode62\relax % >
66   \catcode64 \the\catcode64\relax % @
67   \catcode94 \the\catcode94\relax % ^
68   \noexpand\endinput
69 }
70 \catcode39 12 % '
71 \catcode40 12 % (
72 \catcode41 12 % )
73 \catcode44 12 % ,
74 \catcode45 12 % -
75 \catcode46 12 % .
76 \catcode58 12 % :
77 \catcode60 12 % <
78 \catcode61 12 % =
79 \catcode62 12 % >
80 \catcode64 11 % @
81 \catcode94 7 % ^

```

2.3 L^AT_EX detection

```

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

```


2.4 Version for L^AT_EX

```

\@PackageInfo
103 \let\@PackageInfo\PackageInfo

\@PackageInfoNoLine
104 \def\@PackageInfoNoLine#1#2{%
105     \PackageInfo{#1}{#2\@gobble}%
106 }%

\@ClassInfo
107 \let\@ClassInfo\ClassInfo

\@ClassInfoNoLine
108 \def\@ClassInfoNoLine#1#2{%
109     \ClassInfo{#1}{#2\@gobble}%
110 }%

\@PackageWarning
111 \let\@PackageWarning\PackageWarning

\@PackageWarningNoLine
112 \let\@PackageWarningNoLine\PackageWarningNoLine

\@ClassWarning
113 \let\@ClassWarning\ClassWarning

\@ClassWarningNoLine
114 \let\@ClassWarningNoLine\ClassWarningNoLine

\@PackageError
115 \let\@PackageError\PackageError

\@ClassError
116 \let\@ClassError\ClassError

117 \csname fi\endcsname
118 \InfWarErr@End
119 \fi

```

2.5 Version for plain-T_EX

```

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

```


\InfWarErr@protected

```
134 \begingroup\expandafter\expandafter\expandafter\endgroup
135 \expandafter\ifx\csname protected\endcsname\relax
136   \def\InfWarErr@protected{}%
137 \else
138   \let\InfWarErr@protected\protected
139 \fi
```

\InfWarErr@unused

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

\InfWarErr@ehc

```
141 \def\InfWarErr@ehc{%
142   Try typing \space <return> \space to proceed.\MessageBreak
143   If that doesn't work, type \space X <return> \space to quit.%
144 }
```

\InfWarErr@ehd

```
145 \def\InfWarErr@ehd{%
146   You're in trouble here. \space\@ehc
147 }
```

\InfWarErr@GenericInfo

```
148 \InfWarErr@protected\def\InfWarErr@GenericInfo#1#2{%
149   \begingroup
150     \let\on@line\InfWarErr@online
151     \long\def\@gobble##1{%
152       \def\space{ }%
153       \def\@spaces{\space\space\space\space}%
154       \def\MessageBreak{^^J#1}%
155       \newlinechar=10 %
156       \immediate\write-1{#2\on@line.}%
157     \endgroup
158 }
```

\InfWarErr@GenericWarning

```
159 \InfWarErr@protected\def\InfWarErr@GenericWarning#1#2{%
160   \begingroup
161     \let\on@line\InfWarErr@online
162     \long\def\@gobble##1{%
163       \def\space{ }%
164       \def\@spaces{\space\space\space\space}%
165       \def\MessageBreak{^^J#1}%
166       \newlinechar=10 %
167       \immediate\write\InfWarErr@unused{^^J#2\on@line.^^J}%
168     \endgroup
169 }
```

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

```
170 \InfWarErr@protected\def\InfWarErr@GenericError#1#2#3#4{%
171   \begingroup
172     \immediate\write\InfWarErr@unused{}%
173     \newlinechar=10 %
174     \def\space{ }%
175     \def\@spaces{\space\space\space\space}%
176     \let\@ehc\InfWarErr@ehc
177     \let\@ehd\InfWarErr@ehd
178     \let\on@line\InfWarErr@online
179     \def\MessageBreak{^^J}%
180     \edef\InfWarErr@help{\noexpand\errhelp{#4}}%
```



```

181 \InfWarErr@help
182 \def\MessageBreak{^^J#1}%
183 \errmessage{%
184   #2.^^J^^J%
185   #3^^J%
186   Type \space H <return> \space for immediate help.^^J ...^^J%
187 }%
188 \endgroup
189 }

```

\@PackageInfo

```

190 \def\@PackageInfo#1#2{%
191   \InfWarErr@GenericInfo{%
192     (#1) \@spaces\@spaces\@spaces
193   }{%
194     Package #1 Info: #2%
195   }%
196 }

```

\@PackageInfoNoLine

```

197 \def\@PackageInfoNoLine#1#2{%
198   \@PackageInfo{#1}{#2\@gobble}%
199 }

```

\@ClassInfo

```

200 \def\@ClassInfo#1#2{%
201   \InfWarErr@GenericInfo{%
202     (#1) \space\space\@spaces\@spaces
203   }{%
204     Class #1 Info: #2%
205   }%
206 }

```

\@ClassInfoNoLine

```

207 \def\@ClassInfoNoLine#1#2{%
208   \@ClassInfo{#1}{#2\@gobble}%
209 }

```

\@PackageWarning

```

210 \def\@PackageWarning#1#2{%
211   \InfWarErr@GenericWarning{%
212     (#1)\@spaces\@spaces\@spaces\@spaces
213   }{%
214     Package #1 Warning: #2%
215   }%
216 }

```

\@PackageWarningNoLine

```

217 \def\@PackageWarningNoLine#1#2{%
218   \@PackageWarning{#1}{#2\@gobble}%
219 }

```

\@ClassWarning

```

220 \def\@ClassWarning#1#2{%
221   \InfWarErr@GenericWarning{%
222     (#1) \space\@spaces\@spaces\@spaces
223   }{%
224     Class #1 Warning: #2%
225   }%
226 }

```


\@ClassWarningNoLine

```
227 \def\@ClassWarningNoLine#1#2{%
228   \@ClassWarning{#1}{#2@gobble}%
229 }
```

\@PackageError

```
230 \def\@PackageError#1#2#3{%
231   \InfWarErr@GenericError{%
232     (#1)\@spaces\@spaces\@spaces\@spaces
233   }{%
234     Package #1 Error: #2%
235   }{%
236     See the #1 package documentation for explanation.%
237   }{#3}%
238 }
```

\@ClassError

```
239 \def\@ClassError#1#2#3{%
240   \InfWarErr@GenericError{%
241     (#1) \space\@spaces\@spaces\@spaces
242   }{%
243     Class #1 Error: #2%
244   }{%
245     See the #1 class documentation for explanation.%
246   }{#3}%
247 }
```

```
248 \InfWarErr@End
249 \endpackage
```

3 Test

```
250 <*test1>
251 \let\PackageInfo\relax
252 \input infwarerr.sty\relax
253 \let\OrgWrite\write
254 \chardef\WriteUnused=16 %
255 \def\msg#\{\immediate\OrgWrite\WriteUnused}
256 \msg{File: infwarerr-test1.tex 2007/06/14 v1.1 Test file for plain-TeX}
257 \catcode64=11 %
258 </test1>

259 <*test2>
260 \NeedsTeXFormat{LaTeX2e}
261 \ProvidesFile{infwarerr-test2.tex}[2007/06/14 v1.1 Test file for LaTeX]
262 \RequirePackage{infwarerr}
263 \makeatletter
264 \let\OrgWrite\write
265 \chardef\WriteUnused=\@unused
266 \def\msg#\{\immediate\OrgWrite\WriteUnused}
267 </test2>

268 <*test1 | test2>
269 \def\pkgname{PACKAGE}
270 \def\clsname{CLASS}
271 \def\msgtext{MESSAGE\MessageBreak SECOND LINE}
272 \def\hlptext{HELP\MessageBreak SECOND LINE}
273
274 \let\OrgWrite\write
275 \let\OrgErrorMessage\errmessage
276
277 \def\write#1#{%
```



```

278 \@write{#1}%
279 }
280 \def\@write#1#2{%
281   \OrgWrite#1{#2}%
282   \xdef\WriteNum{\number#1}%
283   \xdef\WriteText{#2}%
284 }
285 \def\TestFailed#1#2{%
286   \OrgErrMessage{Test (\string#1) failed: #2}%
287 }
288 \def\TestWrite#1#2#3{%
289   \ifnum\WriteNum=#2\relax
290   \else
291     \TestFailed#1{write stream number: \WriteNum <> #2}%
292   \fi
293   \begingroup
294     \let~\space
295     \edef\TestString{#3}%
296     \ifx\WriteText\TestString
297     \else
298       \msg{ }%
299       \msg{[\WriteText]}%
300       \msg{<>}%
301       \msg{[#3]}% hash-ok
302       \TestFailed#1{write text}%
303     \fi
304   \endgroup
305 }
306 \newtoks\ErrHelpToks
307 \def\errmessage#1{%
308   \global\ErrHelpToks=\expandafter{\the\errhelp}%
309   \msg{***** error message *****}%
310   \msg{! #1}%
311   \msg{***** help message *****}%
312   \msg{\the\errhelp}%
313   \msg{*****}%
314   \xdef\ErrorMessage{! #1}%
315 }
316 \def\TestInfo{%
317   \@PackageInfo\pkgname\msgtext
318   \TestWrite\@PackageInfo{-1}{%
319     Package PACKAGE Info: MESSAGE^^J%
320     (PACKAGE)~~~~~SECOND LINE on input line \the\inputlineno.%
321   }%
322   \@PackageInfoNoLine\pkgname\msgtext
323   \TestWrite\@PackageInfoNoLine{-1}{%
324     Package PACKAGE Info: MESSAGE^^J%
325     (PACKAGE)~~~~~SECOND LINE.%
326   }%
327   \@ClassInfo\clsname\msgtext
328   \TestWrite\@ClassInfo{-1}{%
329     Class CLASS Info: MESSAGE^^J%
330     (CLASS)~~~~~SECOND LINE on input line \the\inputlineno.%
331   }%
332   \@ClassInfoNoLine\clsname\msgtext
333   \TestWrite\@ClassInfoNoLine{-1}{%
334     Class CLASS Info: MESSAGE^^J%
335     (CLASS)~~~~~SECOND LINE.%
336   }%
337 }
338 \def\TestWarning{%
339   \@PackageWarning\pkgname\msgtext

```



```

340 \TestWrite\@PackageWarning\WriteUnused{%
341   ^^J%
342   Package PACKAGE Warning: MESSAGE^^J%
343   (PACKAGE)~~~~~SECOND LINE on input line \the\inputlineno.%
344   ^^J%
345 }%
346 \@PackageWarningNoLine\pkgname\msgtext
347 \TestWrite\@PackageWarningNoLine\WriteUnused{%
348   ^^J%
349   Package PACKAGE Warning: MESSAGE^^J%
350   (PACKAGE)~~~~~SECOND LINE.%
351   ^^J%
352 }%
353 \@ClassWarning\clsname\msgtext
354 \TestWrite\@ClassWarning\WriteUnused{%
355   ^^J%
356   Class CLASS Warning: MESSAGE^^J%
357   (CLASS)~~~~~SECOND LINE on input line \the\inputlineno.%
358   ^^J%
359 }%
360 \@ClassWarningNoLine\clsname\msgtext
361 \TestWrite\@ClassWarningNoLine\WriteUnused{%
362   ^^J%
363   Class CLASS Warning: MESSAGE^^J%
364   (CLASS)~~~~~SECOND LINE.%
365   ^^J%
366 }%
367 }
368 \def\TestError{%
369 (test2) \let\write\OrgWrite
370 (test2) \def\plaindots{}
371 (test1) \def\plaindots{.^^J ...^^J}
372 \@PackageError\pkgname\msgtext\hlptext
373 \@TestError\@PackageError{%
374   ! Package PACKAGE Error: MESSAGE^^J%
375   (PACKAGE)~~~~~SECOND LINE.^^J%
376   ^^J%
377   See the PACKAGE package documentation for explanation.^^J%
378   Type~~H <return>~~for immediate help\plaindots
379 }{%
380   HELP^^J%
381   SECOND LINE%
382 }%
383 \@ClassError\clsname\msgtext\hlptext
384 \@TestError\@ClassError{%
385   ! Class CLASS Error: MESSAGE^^J%
386   (CLASS)~~~~~SECOND LINE.^^J%
387   ^^J%
388   See the CLASS class documentation for explanation.^^J%
389   Type~~H <return>~~for immediate help\plaindots
390 }{%
391   HELP^^J%
392   SECOND LINE%
393 }%
394 \@PackageError\pkgname\msgtext\@ehc
395 \@TestError\@PackageError{%
396   ! Package PACKAGE Error: MESSAGE^^J%
397   (PACKAGE)~~~~~SECOND LINE.^^J%
398   ^^J%
399   See the PACKAGE package documentation for explanation.^^J%
400   Type~~H <return>~~for immediate help\plaindots
401 }{%

```



```

402   Try typing~~<return>~~to proceed.^^J%
403   If that doesn't work, type~~X <return>~~to quit.%
404 }%
405 \@PackageError\pkgname\msgtext\@ehd
406 \@TestError\@PackageError{%
407   ! Package PACKAGE Error: MESSAGE^^J%
408   (PACKAGE)~~~~~SECOND LINE.^^J%
409   ^^J%
410   See the PACKAGE package documentation for explanation.^^J%
411   Type~~H <return>~~for immediate help\plaindots
412 }{%
413   You're in trouble here.~~Try typing~~<return>~~to proceed.^^J%
414   If that doesn't work, type~~X <return>~~to quit.%
415 }%
416 }
417 \def\@TestError#1#2#3{%
418   \begingroup
419     \let~\space
420     \edef\x{#2}%
421     \test2 \@onelevel@sanitize\x
422     \test2 \@onelevel@sanitize\ErrorMessage
423     \ifx\x\ErrorMessage
424       \else
425         \msg{}%
426         \msg{[\ErrorMessage]}%
427         \msg{<>}%
428         \msg{[\x]}%
429         \TestFailed#1{errmessage text}%
430       \fi
431       \edef\x{\the\ErrHelpToks}%
432       \edef\y{#3}%
433       \ifx\x\y
434         \else
435           \msg{}%
436           \msg{[\the\ErrHelpToks]}%
437           \msg{<>}%
438           \msg{[\y]}%
439           \TestFailed#1{errhelp text}%
440         \fi
441       \endgroup
442 }
443 \TestInfo
444 \TestWarning
445 \TestError
446 </test1 | test2>
447 <test1>\end
448 <test2>\@@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.

¹<http://ftp.ctan.org/tex-archive/>

CTAN:macros/latex/contrib/oberdiek/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/
```

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
infwarerr-test1.tex → doc/latex/oberdiek/infwarerr-test1.tex
infwarerr-test2.tex → doc/latex/oberdiek/infwarerr-test2.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 pdfL^AT_EX:

```
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

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		64, 65, 66, 67, 70, 71, 72, 73,
\@@end	448	74, 75, 76, 77, 78, 79, 80, 81, 257
\@ClassError	<u>116</u> , 239, 383, 384	\chardef
\@ClassInfo ...	<u>107</u> , 200, 208, 327, 328	\check
\@ClassInfoNoLine .	<u>108</u> , 207, 332, 333	\ClassError
\@ClassWarning	<u>113</u> , 220, 228, 353, 354	\ClassInfo
\@ClassWarningNoLine	<u>114</u> , 227, 360, 361	\ClassWarning
\@PackageError	<u>115</u> ,	\ClassWarningNoLine
	230, 372, 373, 394, 395, 405, 406	\clsname ..
\@PackageInfo .	<u>103</u> , <u>190</u> , 198, 317, 318	\csname 8, 18, 39, 52, 55, 90, 117, 121, 135
\@PackageInfoNoLine	<u>104</u> , <u>197</u> , 322, 323	
\@PackageWarning	<u>111</u> , <u>210</u> , 218, 339, 340	
\@PackageWarningNoLine		
	<u>112</u> , 217, 346, 347	
\@TestError ...	373, 384, 395, 406, 417	
\@check ..	85, 86, 87, 89, 97, 98, 99, 100	
\@ehc	146, 176, 394	
\@ehd	177, 405	
\@gobble	105,	
	109, 151, 162, 198, 208, 218, 228	
\@onelevel@sanitize	421, 422	
\@spaces	153, 164,	
	175, 192, 202, 212, 222, 232, 241	
\@unused	265	
\@write	278, 280	
C		
\catcode	3, 4, 5, 6,	
	7, 31, 32, 33, 34, 35, 36, 37, 38,	
	56, 57, 58, 59, 60, 61, 62, 63,	
E		
\empty	12	
\end	447	
\endcsname	8,	
	18, 39, 52, 55, 90, 117, 121, 135	
\endinput	27, 68	
\errhelp	180, 308, 312	
\ErrHelpToks	306, 308, 431, 436	
\ErrorMessage	314, 422, 423, 426	
\errmessage	183, 275, 307	
H		
\hlptext	272, 372, 383	
I		
\ifcase	9, 121	
\ifnum	124, 289	
\ifx	10, 12, 18, 39,	
	47, 90, 102, 121, 135, 296, 423, 433	

