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

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

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 LaTeX and plain-TeX. If LaTeX is detected, then the macros of this package will only be wrappers for LaTeX's macros. Otherwise LaTeX's macros will be mimicked for use with plain-TeX.

2 Implementation

1 ⟨∗package⟩

2.1 Reload check and package identification

Reload check, especially if the package is not used with LaTeX.

```
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    \catcode35 6 % #
19    \catcode123 1 % {
20    \catcode125 2 % }
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    \endgroup
30    \expandafter\endinput
31  \fi
32 \endgroup
```

Package identification:

```
33 \begingroup
34   \catcode35 6 % #
35   \catcode40 12 % (
36   \catcode41 12 % )
37   \catcode44 12 % ,
38   \catcode45 12 % -
39   \catcode46 12 % .
40   \catcode47 12 % /
41   \catcode58 12 % :
42   \catcode64 11 % @
```

```
  \catcode123 1 % {
  \catcode125 2 % }
  \expandafter\ifx\csname ProvidesPackage\endcsname\relax
    \def\x#1#2#3[#4]{\endgroup
      \immediate\write-1{Package: #3 #4}%
      \xdef#1{#4}%
    }%
  \else
    \def\x#1#2[#3]{\endgroup
      #2[{#3}]%
      \ifx#1\relax
        \xdef#1{#3}%
      \fi
    }%
  \fi
\expandafter\x\csname ver@infwarerr.sty\endcsname
\ProvidesPackage{infwarerr}%
  [2007/09/09 v1.2 Providing info/warning/message (HO)]
```

2.2 Catcodes

```
\begingroup
  \catcode123 1 % {
  \catcode125 2 % }
  \def\x{\endgroup
    \expandafter\edef\csname InfWarErr@AtEnd\endcsname{%
      \catcode35 \the\catcode35\relax
      \catcode64 \the\catcode64\relax
      \catcode123 \the\catcode123\relax
      \catcode125 \the\catcode125\relax
    }%
  }%
\x
\catcode35 6 % #
\catcode64 11 % @
\catcode123 1 % {
\catcode125 2 % }
\def\TMP@EnsureCode#1#2{%
  \edef\InfWarErr@AtEnd{%
    \InfWarErr@AtEnd
    \catcode#1 \the\catcode#1\relax
  }%
  \catcode#1 #2\relax
}
\TMP@EnsureCode{10}{12}% ^^J
\TMP@EnsureCode{39}{12}% '
\TMP@EnsureCode{40}{12}% (
\TMP@EnsureCode{41}{12}% )
\TMP@EnsureCode{44}{12}% ,
\TMP@EnsureCode{45}{12}% -
\TMP@EnsureCode{46}{12}% .
\TMP@EnsureCode{58}{12}% :
\TMP@EnsureCode{60}{12}% <
\TMP@EnsureCode{61}{12}% =
\TMP@EnsureCode{62}{12}% >
\TMP@EnsureCode{94}{7}% ^
```

2.3 LaTeX detection

```
\begingroup
  \let\x=Y%
  \def\check#1{%
    \@check{Generic}{#1}%
```

```
    \@check{Package}{#1}%
    \@check{Class}{#1}%
  }%
  \def\@check#1#2{%
    \expandafter\ifx\csname #1#2\endcsname\relax
      \let\x=N%
    \fi
  }%
  \check{Info}%
  \check{Warning}%
  \check{Error}%
  \@check{Package}{WarningNoLine}%
  \@check{Class}{WarningNoLine}%
  \@check{}{@ehc}%
  \@check{}{@ehd}%
\expandafter\endgroup
\ifx\x Y%
```

2.4 Version for LaTeX

\@PackageInfo

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

\@PackageInfoNoLine

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

\@ClassInfo

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

\@ClassInfoNoLine

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

\@PackageWarning

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

\@PackageWarningNoLine

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

\@ClassWarning

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

\@ClassWarningNoLine

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

\@PackageError

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

\@ClassError

```
  \let\@ClassError\ClassError

  \InfWarErr@AtEnd
  \expandafter\endinput
\fi
```

2.5 Version for plain-TEX

\InfWarErr@online

```
\begingroup\expandafter\endgroup
\ifcase\expandafter\ifx\csname inputlineno\endcsname\relax
         0 %
       \else
         \ifnum\inputlineno<0 %
           0 %
         \else
           1 %
         \fi
       \fi
  \def\InfWarErr@online{}%
\else
  \def\InfWarErr@online{ on input line \the\inputlineno}%
\fi
```

\InfWarErr@protected

```
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname protected\endcsname\relax
  \def\InfWarErr@protected{}%
\else
  \let\InfWarErr@protected\protected
\fi
```

\InfWarErr@unused

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

\InfWarErr@ehc

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

\InfWarErr@ehd

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

\InfWarErr@GenericInfo

```
\InfWarErr@protected\def\InfWarErr@GenericInfo#1#2{%
  \begingroup
    \let\on@line\InfWarErr@online
    \long\def\@gobble##1{}%
    \def\space{ }%
    \def\@spaces{\space\space\space\space}%
    \def\MessageBreak{^^J#1}%
    \newlinechar=10 %
    \immediate\write-1{#2\on@line.}%
  \endgroup
}
```

\InfWarErr@GenericWarning

```
\InfWarErr@protected\def\InfWarErr@GenericWarning#1#2{%
  \begingroup
    \let\on@line\InfWarErr@online
    \long\def\@gobble##1{}%
    \def\space{ }%
    \def\@spaces{\space\space\space\space}%
    \def\MessageBreak{^^J#1}%
```

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

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

```
\InfWarErr@protected\def\InfWarErr@GenericError#1#2#3#4{%
  \begingroup
    \immediate\write\InfWarErr@unused{}%
    \newlinechar=10 %
    \def\space{ }%
    \def\@spaces{\space\space\space\space}%
    \let\@ehc\InfWarErr@ehc
    \let\@ehd\InfWarErr@ehd
    \let\on@line\InfWarErr@online
    \def\MessageBreak{^^J}%
    \edef\InfWarErr@help{\noexpand\errhelp{#4}}%
    \InfWarErr@help
    \def\MessageBreak{^^J#1}%
    \errmessage{%
      #2.^^J^^J%
      #3^^J%
      Type \space H <return> \space for immediate help.^^J ...^^J%
    }%
  \endgroup
}
```

`\@PackageInfo`

```
\def\@PackageInfo#1#2{%
  \InfWarErr@GenericInfo{%
    (#1) \@spaces\@spaces\@spaces
  }{%
    Package #1 Info: #2%
  }%
}
```

`\@PackageInfoNoLine`

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

`\@ClassInfo`

```
\def\@ClassInfo#1#2{%
  \InfWarErr@GenericInfo{%
    (#1) \space\space\@spaces\@spaces
  }{%
    Class #1 Info: #2%
  }%
}
```

`\@ClassInfoNoLine`

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

`\@PackageWarning`

```
\def\@PackageWarning#1#2{%
  \InfWarErr@GenericWarning{%
    (#1)\@spaces\@spaces\@spaces\@spaces
  }{%
```

```
    Package #1 Warning: #2%
  }%
}
```

\@PackageWarningNoLine

```
\def\@PackageWarningNoLine#1#2{%
  \@PackageWarning{#1}{#2\@gobble}%
}
```

\@ClassWarning

```
\def\@ClassWarning#1#2{%
  \InfWarErr@GenericWarning{%
    (#1) \space\@spaces\@spaces\@spaces
  }{%
    Class #1 Warning: #2%
  }%
}
```

\@ClassWarningNoLine

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

\@PackageError

```
\def\@PackageError#1#2#3{%
  \InfWarErr@GenericError{%
    (#1)\@spaces\@spaces\@spaces\@spaces
  }{%
    Package #1 Error: #2%
  }{%
    See the #1 package documentation for explanation.%
  }{#3}%
}
```

\@ClassError

```
\def\@ClassError#1#2#3{%
  \InfWarErr@GenericError{%
    (#1) \space\@spaces\@spaces\@spaces
  }{%
    Class #1 Error: #2%
  }{%
    See the #1 class documentation for explanation.%
  }{#3}%
}

\InfWarErr@AtEnd
```

⟨/package⟩

3 Test

3.1 Catcode checks for loading

⟨*test1⟩

```
\catcode`\{=1 %
\catcode`\}=2 %
\catcode`\#=6 %
\catcode`\@=11 %
\expandafter\ifx\csname count@\endcsname\relax
  \countdef\count@=255 %
\fi
```

```
\expandafter\ifx\csname @gobble\endcsname\relax
  \long\def\@gobble#1{}%
\fi
\expandafter\ifx\csname @firstofone\endcsname\relax
  \long\def\@firstofone#1{#1}%
\fi
\expandafter\ifx\csname loop\endcsname\relax
  \expandafter\@firstofone
\else
  \expandafter\@gobble
\fi
{%
  \def\loop#1\repeat{%
    \def\body{#1}%
    \iterate
  }%
  \def\iterate{%
    \body
      \let\next\iterate
    \else
      \let\next\relax
    \fi
    \next
  }%
  \let\repeat=\fi
}%
\def\RestoreCatcodes{}
\count@=0 %
\loop
  \edef\RestoreCatcodes{%
    \RestoreCatcodes
    \catcode\the\count@=\the\catcode\count@\relax
  }%
\ifnum\count@<255 %
  \advance\count@ 1 %
\repeat

\def\RangeCatcodeInvalid#1#2{%
  \count@=#1\relax
  \loop
    \catcode\count@=15 %
  \ifnum\count@<#2\relax
    \advance\count@ 1 %
  \repeat
}
\expandafter\ifx\csname LoadCommand\endcsname\relax
  \def\LoadCommand{\input infwarerr.sty\relax}%
\fi
\def\Test{%
  \RangeCatcodeInvalid{0}{47}%
  \RangeCatcodeInvalid{58}{64}%
  \RangeCatcodeInvalid{91}{96}%
  \RangeCatcodeInvalid{123}{255}%
  \catcode`\@=12 %
  \catcode`\\=0 %
  \catcode`\{=1 %
  \catcode`\}=2 %
  \catcode`\#=6 %
  \catcode`\[=12 %
  \catcode`\]=12 %
  \catcode`\%=14 %
  \catcode`\ =10 %
```

```
  \catcode13=5 %
  \LoadCommand
  \RestoreCatcodes
}
\Test
\csname @@end\endcsname
\end
```

⟨/test1⟩

3.2 Macro tests

⟨*test2⟩

```
\let\PackageInfo\relax
\input infwarerr.sty\relax
\let\OrgWrite\write
\chardef\WriteUnused=16 %
\def\msg#{\immediate\OrgWrite\WriteUnused}
\msg{File: infwarerr-test2.tex 2007/09/09 v1.2 Test file for plain-TeX}
\catcode64=11 %
```

⟨/test2⟩

⟨*test3⟩

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesFile{infwarerr-test3.tex}[2007/09/09 v1.2 Test file for LaTeX]
\RequirePackage{infwarerr}[2007/09/09]
\makeatletter
\let\OrgWrite\write
\chardef\WriteUnused=\@unused
\def\msg#{\immediate\OrgWrite\WriteUnused}
```

⟨/test3⟩

⟨*test2 | test3⟩

```
\def\pkgname{PACKAGE}
\def\clsname{CLASS}
\def\msgtext{MESSAGE\MessageBreak SECOND LINE}
\def\hlptext{HELP\MessageBreak SECOND LINE}

\let\OrgWrite\write
\let\OrgErrMessage\errmessage

\def\write#1#{%
  \@write{#1}%
}
\def\@write#1#2{%
  \OrgWrite#1{#2}%
  \xdef\WriteNum{\number#1}%
  \xdef\WriteText{#2}%
}
\def\TestFailed#1#2{%
  \OrgErrMessage{Test (\string#1) failed: #2}%
}
\def\TestWrite#1#2#3{%
  \ifnum\WriteNum=#2\relax
  \else
    \TestFailed#1{write stream number: \WriteNum <> #2}%
  \fi
  \begingroup
    \let~\space
    \edef\TestString{#3}%
    \ifx\WriteText\TestString
    \else
      \msg{}%
      \msg{[\WriteText]}%
      \msg{<>}%
```

```
      \msg{[#3]}% hash-ok
      \TestFailed#1{write text}%
    \fi
  \endgroup
}
\newtoks\ErrHelpToks
\def\errmessage#1{%
  \global\ErrHelpToks=\expandafter{\the\errhelp}%
  \msg{**************** error message ****************}%
  \msg{! #1}%
  \msg{**************** help message *****************}%
  \msg{\the\errhelp}%
  \msg{***********************************************}%
  \xdef\ErrMessage{! #1}%
}
\def\TestInfo{%
  \@PackageInfo\pkgname\msgtext
  \TestWrite\@PackageInfo{-1}{%
    Package PACKAGE Info: MESSAGE^^J%
    (PACKAGE)~~~~~~~~~~~~~SECOND LINE on input line \the\inputlineno.%
  }%
  \@PackageInfoNoLine\pkgname\msgtext
  \TestWrite\@PackageInfoNoLine{-1}{%
    Package PACKAGE Info: MESSAGE^^J%
    (PACKAGE)~~~~~~~~~~~~~SECOND LINE.%
  }%
  \@ClassInfo\clsname\msgtext
  \TestWrite\@ClassInfo{-1}{%
    Class CLASS Info: MESSAGE^^J%
    (CLASS)~~~~~~~~~~~SECOND LINE on input line \the\inputlineno.%
  }%
  \@ClassInfoNoLine\clsname\msgtext
  \TestWrite\@ClassInfoNoLine{-1}{%
    Class CLASS Info: MESSAGE^^J%
    (CLASS)~~~~~~~~~~~SECOND LINE.%
  }%
}
\def\TestWarning{%
  \@PackageWarning\pkgname\msgtext
  \TestWrite\@PackageWarning\WriteUnused{%
    ^^J%
    Package PACKAGE Warning: MESSAGE^^J%
    (PACKAGE)~~~~~~~~~~~~~~~~SECOND LINE on input line \the\inputlineno.%
    ^^J%
  }%
  \@PackageWarningNoLine\pkgname\msgtext
  \TestWrite\@PackageWarningNoLine\WriteUnused{%
    ^^J%
    Package PACKAGE Warning: MESSAGE^^J%
    (PACKAGE)~~~~~~~~~~~~~~~~SECOND LINE.%
    ^^J%
  }%
  \@ClassWarning\clsname\msgtext
  \TestWrite\@ClassWarning\WriteUnused{%
    ^^J%
    Class CLASS Warning: MESSAGE^^J%
    (CLASS)~~~~~~~~~~~~~~SECOND LINE on input line \the\inputlineno.%
    ^^J%
  }%
  \@ClassWarningNoLine\clsname\msgtext
  \TestWrite\@ClassWarningNoLine\WriteUnused{%
    ^^J%
```

```
    Class CLASS Warning: MESSAGE^^J%
    (CLASS)~~~~~~~~~~~~~~SECOND LINE.%
    ^^J%
  }%
}
\def\TestError{%
⟨test3⟩  \let\write\OrgWrite
⟨test3⟩  \def\plaindots{}
⟨test2⟩  \def\plaindots{.^^J ...^^J}
  \@PackageError\pkgname\msgtext\hlptext
  \@TestError\@PackageError{%
    ! Package PACKAGE Error: MESSAGE^^J%
    (PACKAGE)~~~~~~~~~~~~~~~~SECOND LINE.^^J%
    ^^J%
    See the PACKAGE package documentation for explanation.^^J%
    Type~~H <return>~~for immediate help\plaindots
  }{%
    HELP^^J%
    SECOND LINE%
  }%
  \@ClassError\clsname\msgtext\hlptext
  \@TestError\@ClassError{%
    ! Class CLASS Error: MESSAGE^^J%
    (CLASS)~~~~~~~~~~~~~~SECOND LINE.^^J%
    ^^J%
    See the CLASS class documentation for explanation.^^J%
    Type~~H <return>~~for immediate help\plaindots
  }{%
    HELP^^J%
    SECOND LINE%
  }%
  \@PackageError\pkgname\msgtext\@ehc
  \@TestError\@PackageError{%
    ! Package PACKAGE Error: MESSAGE^^J%
    (PACKAGE)~~~~~~~~~~~~~~~~SECOND LINE.^^J%
    ^^J%
    See the PACKAGE package documentation for explanation.^^J%
    Type~~H <return>~~for immediate help\plaindots
  }{%
    Try typing~~<return>~~to proceed.^^J%
    If that doesn't work, type~~X <return>~~to quit.%
  }%
  \@PackageError\pkgname\msgtext\@ehd
  \@TestError\@PackageError{%
    ! Package PACKAGE Error: MESSAGE^^J%
    (PACKAGE)~~~~~~~~~~~~~~~~SECOND LINE.^^J%
    ^^J%
    See the PACKAGE package documentation for explanation.^^J%
    Type~~H <return>~~for immediate help\plaindots
  }{%
    You're in trouble here.~~Try typing~~<return>~~to proceed.^^J%
    If that doesn't work, type~~X <return>~~to quit.%
  }%
}
\def\@TestError#1#2#3{%
  \begingroup
    \let~\space
    \edef\x{#2}%
⟨test3⟩  \@onelevel@sanitize\x
⟨test3⟩  \@onelevel@sanitize\ErrMessage
    \ifx\x\ErrMessage
    \else
```

```
      \msg{}%
      \msg{[\ErrMessage]}%
      \msg{<>}%
      \msg{[\x]}%
      \TestFailed#1{errmessage text}%
    \fi
    \edef\x{\the\ErrHelpToks}%
    \edef\y{#3}%
    \ifx\x\y
    \else
      \msg{}%
      \msg{[\the\ErrHelpToks]}%
      \msg{<>}%
      \msg{[\y]}%
      \TestFailed#1{errhelp text}%
    \fi
  \endgroup
}
\TestInfo
\TestWarning
\TestError
⟨/test2 | test3⟩
⟨test2⟩\end
⟨test3⟩\@@end
```

4 Installation

4.1 Download

Package. This package is available on CTAN[1]:

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/

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 (teTEX, mikTEX, ...) relies on file name databases, you must refresh these. For example, teTEX users run `texhash` or `mktexlsr`.

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 pdfLATEX:

```
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

\# 267, 329
\% 332
\@ 268, 325
\@@end 540
\@ClassError 130, 253, 475, 476
\@ClassInfo . . . 121, 214, 222, 419, 420
\@ClassInfoNoLine . 122, 221, 424, 425
\@ClassWarning 127, 234, 242, 445, 446
\@ClassWarningNoLine 128, 241, 452, 453
\@PackageError 129, 244, 464, 465, 486, 487, 497, 498
\@PackageInfo . 117, 204, 212, 409, 410
\@PackageInfoNoLine 118, 211, 414, 415
\@PackageWarning 125, 224, 232, 431, 432
\@PackageWarningNoLine 126, 231, 438, 439
\@TestError . . . 465, 476, 487, 498, 509
\@check 99, 100, 101, 103, 111, 112, 113, 114
\@ehc 160, 190, 486
\@ehd 191, 497
\@firstofone 276, 279
\@gobble 119, 123, 165, 176, 212, 222, 232, 242, 273, 281
\@onelevel@sanitize 513, 514
\@spaces 167, 178, 189, 206, 216, 226, 236, 246, 255
\@unused 357
\@write 370, 372
\[. 330
\\ 326
\{ 265, 327
\} 266, 328
\] 331
\␣ 333

A

\advance 306, 314

B

\body 285, 289

C

\catcode 3, 4, 5, 6, 7, 18, 19, 20, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 62, 63, 66, 67, 68, 69, 73, 74, 75, 76, 80, 82, 265, 266, 267, 268, 303, 312, 325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 349
\chardef 154, 346, 357
\check 98, 108, 109, 110
\ClassError 130
\ClassInfo 121, 123
\ClassWarning 127
\ClassWarningNoLine 128
\clsname . . 362, 419, 424, 445, 452, 475
\count@ 270, 299, 303, 305, 306, 310, 312, 313, 314
\countdef 270
\csname . 8, 21, 45, 58, 65, 104, 135, 149, 269, 272, 275, 278, 317, 339

E

\empty 12
\end 340, 539
\endcsname 8, 21, 45, 58, 65, 104, 135, 149, 269, 272, 275, 278, 317, 339
\endinput 30, 132
\errhelp 194, 400, 404
\ErrHelpToks 398, 400, 523, 528
\ErrMessage 406, 514, 515, 518
\errmessage 197, 367, 399

H

\hlptext 364, 464, 475

I

\ifcase 9, 135
\ifnum 138, 305, 313, 381

\ifx 10, 12, 21, 45, 53, 104, 116, 135, 149, 269, 272, 275, 278, 317, 388, 515, 525
\immediate 23, 47, 170, 181, 186, 347, 358
\InfWarErr@AtEnd 78, 79, 131, 262
\InfWarErr@ehc 155, 190
\InfWarErr@ehd 159, 191
\InfWarErr@GenericError 184, 245, 254
\InfWarErr@GenericInfo 162, 205, 215
\InfWarErr@GenericWarning 173, 225, 235
\InfWarErr@help 194, 195
\InfWarErr@online . 134, 164, 175, 192
\InfWarErr@protected 148, 162, 173, 184
\InfWarErr@unused 154, 181, 186
\input 318, 344
\inputlineno 138, 146, 412, 422, 435, 449
\iterate 286, 288, 290

L

\LoadCommand 318, 335
\loop 284, 300, 311

M

\makeatletter 355
\MessageBreak 156, 168, 179, 193, 196, 363, 364
\msg 347, 348, 358, 390, 391, 392, 393, 401, 402, 403, 404, 405, 517, 518, 519, 520, 527, 528, 529, 530
\msgtext 363, 409, 414, 419, 424, 431, 438, 445, 452, 464, 475, 486, 497

N

\NeedsTeXFormat 352
\newlinechar 169, 180, 187
\newtoks 398
\next 290, 292, 294
\number 374

O

\on@line 164, 170, 175, 181, 192
\OrgErrMessage 367, 378
\OrgWrite 345, 347, 356, 358, 366, 373, 461

P

\PackageError 129
\PackageInfo 26, 117, 119, 343
\PackageWarning 125
\PackageWarningNoLine 126
\pkgname 361, 409, 414, 431, 438, 464, 486, 497
\plaindots 462, 463, 470, 481, 492, 503
\protected 152
\ProvidesFile 353
\ProvidesPackage 59

R

\RangeCatcodeInvalid 309, 321, 322, 323, 324
\repeat 284, 296, 307, 315
\RequirePackage 354
\RestoreCatcodes . . 298, 301, 302, 336

S

\space 156, 157, 160, 166, 167, 177, 178, 188, 189, 200, 216, 236, 255, 386, 511

T

\Test 320, 338
\TestError 460, 537
\TestFailed . . . 377, 383, 394, 521, 531
\TestInfo 408, 535
\TestString 387, 388
\TestWarning 430, 536
\TestWrite 380, 410, 415, 420, 425, 432, 439, 446, 453
\the 66, 67, 68, 69, 80, 146, 303, 400, 404, 412, 422, 435, 449, 523, 528
\TMP@EnsureCode 77, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95

W

\write 23, 47, 170, 181, 186, 345, 356, 366, 369, 461
\WriteNum 374, 381, 383
\WriteText 375, 388, 391
\WriteUnused 346, 347, 357, 358, 432, 439, 446, 453

X

\x 8, 10, 12, 22, 26, 28, 46, 51, 58, 64, 72, 97, 105, 116, 512, 513, 515, 520, 523, 525

Y

\y 524, 525, 530