The \texttt{iflang} package

Heiko Oberdiek
\texttt{<heiko.oberdiek at googlemail.com>}

2007/11/11 v1.5

Abstract
This package provides expandible checks for the current language based on macro \texttt{\languagename} or hyphenation patterns.

Contents

1 Documentation 2

2 Implementation 2

2.1 Reload check and package identification 3

2.2 Tools 4

2.2.1 Provide some basic macros of \texttt{\LaTeX} 4

2.2.2 Expandible existence check for macros 5

2.2.3 Macros for messages 5

2.2.4 Support for \texttt{etex.src} 5

2.3 \texttt{\IfLanguagePatterns} 6

2.4 \texttt{\IfLanguageName} 6

2.5 Check plausibility of \texttt{\languagename} 8

3 Test 8

3.1 Catcode checks for loading 8

3.2 Test with \texttt{\LaTeX} 10

3.3 Test with plain \texttt{\TeX} and \texttt{\varepsilon-\TeX} 11

3.4 Test with plain \texttt{\TeX} and without \texttt{\varepsilon-\TeX/pdf\TeX} 11

4 Installation 13

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

5 Catalogue 14

6 Acknowledgement 15

7 History 15

\[2007/04/10\ v1.0\] 15

\[2007/04/11\ v1.1\] 15

\[2007/04/12\ v1.2\] 15

\[2007/04/26\ v1.3\] 15

\[2007/09/09\ v1.4\] 15

\[2007/11/11\ v1.5\] 15

8 Index 15
1 Documentation

Package \texttt{babel} defines \texttt{\iflanguagename}. As first argument it takes a language name and executes the second or third argument depending on the current language. This language test is based on hyphenation patterns. However, it is possible that different languages or dialects share the same patterns. In such cases \texttt{\iflanguagename} fails.

However, package \texttt{babel} and some other packages such as \texttt{german} or \texttt{ngerman} store the language name in the macro \texttt{\languagename} if \texttt{\selectlanguage} is called.

\begin{verbatim}
\IfLanguageName {⟨lang⟩} {⟨then⟩} {⟨else⟩}
\end{verbatim}

Makro \texttt{\IfLanguageName} compares language \texttt{⟨lang⟩} with the current setting of macro \texttt{\languagename}. If both contains the same name then the \texttt{(then)} part is called, otherwise the \texttt{(else)} part.

The macro is expandable. Thus it can be safely used inside \texttt{\edef} or \texttt{\csname}. If case of errors like an undefined \texttt{\languagename} the \texttt{(else)} part is executed.

Note: Macro \texttt{\IfLanguageName} relies on the fact, that \texttt{\languagename} is set correctly:

Package \texttt{babel}:
Full support of \texttt{\languagename} in its language switching commands.

Format based on \texttt{babel (language.dat)}:
If package \texttt{babel} is not used (or not yet loaded), then \texttt{babel}'s \texttt{hyphen.cfg} has set \texttt{\languagename} to the last language in \texttt{language.dat}, but \texttt{\language} (current patterns) is zero and points to the first language. Thus the value of \texttt{\languagename} is basically garbage. Package \texttt{iflang} warns if \texttt{\languagename} and \texttt{\language} do not fit. This can be fixed by loading package \texttt{babel} previously.

Format based on \texttt{-\TeX{}'s etex.src (language.def)}:
Unhappily it does not support \texttt{\languagename}. Thus this package hooks into \texttt{\uselanguage} to get \texttt{\languagename} defined and updated there. At package loading time the changed \texttt{\uselanguage} has not been called yet. Thus package \texttt{iflang} tries \texttt{USenglish}. This is the definite default language of \texttt{etex.src}. If the current patterns suit this default language, an undefined \texttt{\languagename} is set to this language. Otherwise a \texttt{\languagename} remains undefined and a warning is given.

\begin{verbatim}
\IfLanguagePatterns {⟨lang⟩} {⟨then⟩} {⟨else⟩}
\end{verbatim}

This macro behaves similar to \texttt{\IfLanguageName}. But the language test is based on the current pattern in force \texttt{⟨language⟩}. Also this macro is expandable, in case of errors the \texttt{(else)} part is called.

The following naming convention for the pattern are supported:

\texttt{babel/language.dat} : \texttt{\l@⟨language⟩}
\texttt{etex/src/language.def} : \texttt{\lang@⟨language⟩}

Package \texttt{iflang} looks for \texttt{\uselanguage} (defined in \texttt{etex.src}) to find out the naming convention in use.

2 Implementation

1 (*package)
2.1 Reload check and package identification

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

\begin{verbatim}
\begingroup\catcode61\catcode48\catcode32=10\relax%
\catcode13=5 \^\M
\endlinechar=13 %
\catcode35=6 % #
\catcode39=12 % '
\catcode44=12 % ,
\catcode45=12 % -
\catcode46=12 % .
\catcode8=12 % :
\catcode6=11 % @
\catcode123=1 % {
\catcode125=2 % }
\expandafter\ifx\expandafter\csname ver@iflang.sty\endcsname
\ifx\relax % plain-TeX, first loading
\else
\expandafter\def\expandafter\empty\expandafter\relax
\ifx\empty % LaTeX, first loading,
% variable is initialized, but \ProvidesPackage not yet seen
\else
\expandafter\ifx\csname PackageInfo\endcsname\relax
\def\x#1#2#3[#4]{\endgroup
\immediate\write-1{Package #1 Info: #2.}\
}\else
\def\x#1#2#3[#4]{\PackageInfo{#1}{#2, stopped}}\
\fi
\x{iflang}{The package is already loaded}
\aftergroup\endinput
\fi
\fi
\endgroup%
\end{verbatim}

Package identification:

\begin{verbatim}
\begingroup\catcode61\catcode48\catcode32=10\relax%
\catcode13=5 \^\M
\endlinechar=13 %
\catcode35=6 % #
\catcode39=12 % '
\catcode44=12 % ,
\catcode45=12 % -
\catcode46=12 % .
\catcode47=12 % /
\catcode48=12 % :
\catcode6=11 % @
\catcode123=1 % {
\catcode125=2 % }
\expandafter\ifx\expandafter\csname ProvidesPackage\endcsname\relax
\def\x#1#2#3[#4]{\endgroup
\immediate\write-1{Package: #3 #4}}\
\else
\def\x#1#2#3[#4]{\endgroup
#2[#3]}\
\ifx#1\@undefined
\xdef#1{#3}\
\fi
\fi
\endgroup%
\end{verbatim}

3
\@firstoftwo
\expandafter\ifx\csname @firstoftwo\endcsname\relax
\long\def\@firstoftwo#1#2{#1}\fi
\@secondoftwo

\@firstoftwo
\expandafter\if\csname @firstoftwo\endcsname@firstoftwo\endcsname\relax
\long\def\@firstoftwo#1\@secondoftwo{#1}\fi
\@secondoftwo

\@firstoftwo
\expandafter\if\csname @firstoftwo\endcsname\@firstoftwo\endcsname\relax
\long\def\@firstoftwo#1\@secondoftwo{#1}\fi
\@secondoftwo

2.2 Tools

2.2.1 Provide some basic macros of \LaTeX
2.2.2 Expandible existence check for macros

\IfLang@IfDefined

2.2.3 Macros for messages

2.2.4 Support for \texttt{etex.src}

\IfLang@prefix
The first `\uselanguage` that is executed as last line in `language.def` cannot be patched this way. However, `language.def` is very strict. It forces the first added and used language to be `USenglish`. Thus, if `\languagename` is not defined, we can quite safely assume `USenglish`. As additional safety precaution the actual used patterns are checked.

\begin{verbatim}
\IfLang@prefix{lang}@\%
\let\IfLang@OrgUseLanguage\uselanguage
\def\uselanguage#1{%
  \edef\languagename{#1}%
  \IfLang@OrgUseLanguage{#1}%
}
\end{verbatim}

The first `\uselanguage` that is executed as last line in `language.def` cannot be patched this way. However, `language.def` is very strict. It forces the first added and used language to be `USenglish`. Thus, if `\languagename` is not defined, we can quite safely assume `USenglish`. As additional safety precaution the actual used patterns are checked.

\begin{verbatim}
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname languagename\endcsname\relax
  \begingroup\expandafter\expandafter\expandafter\endgroup
  \expandafter\ifx\csname lang@USenglish\endcsname\relax
    \@PackageWarningNoLine{iflang}{% \string\lang@USenglish\space is missing%
    \}%
  \else
    \@PackageWarningNoLine{iflang}{% \string\languagename\space is not set,\MessageBreak current language is unknown%
    \}%
  \fi
\fi
\fi
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname languagename\endcsname\relax
  \@PackageInfoNoLine{iflang}{% \string\languagename\space is not set%
\fi
\fi
\end{verbatim}

\section{\IfLanguagePatterns}

\begin{verbatim}
\def\IfLanguagePatterns#1{%
  \ifnum\IfLang@IfDefined{\IfLang@prefix#1}{% \IfLang@prefix#1=0 \}
  \else
    \@firstoftwo
  \fi
  \else
    \@secondoftwo
  \fi
\end{verbatim}

\section{\IfLanguageName}

\begin{verbatim}
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname pdf@strcmp\endcsname\relax
  \expandafter\@firstoftwo
\else
  \expandafter\@secondoftwo
\fi
\end{verbatim}
We do not have \pdfstrcmp (and \pdfstrcmp). Thus we must define our own expandable string comparison. The following implementation is based on a \TeX pearl from David Kastrup, presented at the conference Bacho\TeX 2005: http://www-stary.gust.org.pl/pearls/2005/david-kastrup/bachotex2005-david-kastrup-pearl1.pdf

The original code allows macros inside the second string. Because also \languagename might consists of further macros, we need a variant that allows macros in the first string, too.
\begin{verbatim}
\def\IfLang@StrNil{\relax}
\def\IfLang@StrEqual#1{\number\IfLang@StrEqualStart{}{}#1\IfLang@StrNil}
\def\IfLang@StrEqualStart#1#2#3{\ifx#3\IfLang@StrNil\IfLang@StrEqualStop\fi\ifcat\noexpand#3\relax\IfLang@StrExpand{#1}{#2}{#3}\fi\IfLang@StrEqualStart{\if#3#1}{#2\fi}}
\def\IfLang@StrEqualStop{\fi#1\IfLang@StrEqualStart#2#3#4{}}
\def\IfLang@StrExpand#1#2#3\fi\IfLang@StrEqualStart#4#5{\fi\IfLang@StrEqualStart#1#2#3\fi\IfLang@StrEqualStart#4#5{}\fi\IfLang@StrEqualStart#1#2#3}\fi
\def\IfLang@StrExpand#1#2#3\IfLang@StrNil{\expandafter\IfLang@@StrExpand#3\IfLang@StrNil{#1}{#2}}
\def\IfLang@@StrExpand#1#2#3\IfLang@StrNil{\expandafter\IfLang@@@StrExpand#3\IfLang@StrNil{#1}{#2}}
\def\IfLang@@@StrExpand#1\IfLang@StrNil#2#3{\IfLang@StrEqualStart{#2}{#3}#1\IfLang@StrNil}
\end{verbatim}

\IfLanguageName
\begin{verbatim}
\def\IfLanguageName#1{\ifnum\IfLang@IfDefined{languagename}{\if\expandafter\IfLang@StrEqual\expandafter#1\languagename\else1\fi}\else\expandafter\@secondoftwo\fi}\expandafter\@firstoftwo}
\end{verbatim}

\IfLanguageName
\begin{verbatim}
\def\IfLanguageName#1{\ifnum\IfLang@IfDefined{languagename}{\pdfstrcmp#1\languagename}{1}=0 \expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi}\expandafter\@firstoftwo}
\end{verbatim}
2.5 Check plausibility of \texttt{languagename}

\begin{itemize}
\item \texttt{\expandafter\@firstoftwo}
\item \texttt{\else}
\item \texttt{\expandafter\@secondoftwo}
\item \texttt{\fi}
\end{itemize}

3 Test

3.1 Catcode checks for loading

\begin{itemize}
\item \texttt{\catcode`\{=1}
\item \texttt{\catcode`\}=2}
\item \texttt{\catcode`\#=6}
\item \texttt{\catcode`\@=11}
\item \texttt{\countdef\count@=255}
\item \texttt{\long\def\@gobble#1{}}
\item \texttt{\long\def\@firstofone#1{#1}}
\item \texttt{\expandafter\@firstofone}
\item \texttt{\else}
\item \texttt{\let\next\relax}
\end{itemize}
\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
}\def\RangeCatcodeCheck#1#2#3{%
  \count@=#1 \relax
  \loop
    \ifnum#3=\catcode\count@
      \else
        \errmessage{%
          Character \the\count@ \space
          with wrong \ catcode \the\catcode\count@ \space
        instead of \ number #3%}
      \fi
    \ifnum\count@<#2 \relax
      \advance\count@ 1 \%
    \repeat
}\def\space{ }
\expandafter\ifx\csname LoadCommand\endcsname\relax
  \def\LoadCommand{\input iflang.sty \relax}%
\fi
\def\Test{%
  \RangeCatcodeInvalid{0}{47} \%
  \RangeCatcodeInvalid{58}{64} \%
  \RangeCatcodeInvalid{91}{96} \%
  \RangeCatcodeInvalid{123}{255} \%
  \catcode`@=12 \%
  \catcode`\=0 \%
  \catcode`\%=14 \%
  \LoadCommand
  \RangeCatcodeCheck{0}{36}{15} \%
  \RangeCatcodeCheck{37}{37}{14} \%
  \RangeCatcodeCheck{38}{47}{15} \%
  \RangeCatcodeCheck{48}{57}{12} \%
  \RangeCatcodeCheck{58}{63}{15} \%
  \RangeCatcodeCheck{64}{64}{12} \%
  \RangeCatcodeCheck{65}{90}{11} \%
  \RangeCatcodeCheck{91}{91}{15} \%
  \RangeCatcodeCheck{92}{92}{0} \%
  \RangeCatcodeCheck{93}{96}{15} \%
  \RangeCatcodeCheck{97}{122}{11} \%
  \RangeCatcodeCheck{123}{255}{15} \%
  \RestoreCatcodes
}\Test
3.2 Test with \LaTeX

\begin{document}
\begin{qstest}{IfLanguagePatterns}{language, pattern}
\def\test#1#2{\
  \Expect*{\IfLanguagePatterns{#1}{true}{false}}{#2}\
}\test{ngerman}{true}\test{naustrian}{true}\test{english}{false}\test{foobar}{false}
\end{qstest}

\begin{qstest}{IfLanguageName}{language, name}
\def\test#1#2{\
  \Expect*{\IfLanguageName{#1}{true}{false}}{#2}\
}\test{ngerman}{true}\test{naustrian}{false}\selectlanguage{naustrian}\test{ngerman}{false}\test{naustrian}{true}\test{foobar}{false}\edef\languagename{\string naustrian}\test{naustrian}{true}\test{ngerman}{false}\test{naustrian}{true}\test{naustrian}{false}\test{naustrian}{true}\test{naustrian}{false}\test{\languagename}{true}\test{\languagename}{false}\test{\languagename}{true}\test{\languagename}{false}\test{\languagename}{true}\test{\languagename}{false}\test{\languagename}{true}\test{\languagename}{false}\test{\languagename}{true}\test{\languagename}{false}\test{\languagename}{true}\test{\languagename}{false}\test{\languagename}{true}\test{\languagename}{false}\test{\languagename}{true}\test{\languagename}{false}\test{\languagename}{true}\test{\languagename}{false}\test{\languagename}{true}\test{\languagename}{false}\test{\languagename}{true}\test{\languagename}{false}\test{\languagename}{true}\test{\languagename}{false}\test{\languagename}{true}\test{\languagename}{false}
\end{qstest}
\end{document}
3.3 Test with plain \TeX and -\TeX

\begin{verbatim}
(*test4*)
%% Format `etex' based on `language.def'
\input iflang.sty
\catcode64=12
\def\TestGeneric#1#2#3{%
\begingroup
  \edef\x{#1{#2}{true}{false}}%
  \edef\y{#3}%
  \ifx\x\y\else
    \errmessage{Failed test: \string#1{#2} <> #3}%
  \fi
\endgroup}
\def\TestPatterns{\TestGeneric\IfLanguagePatterns}
\def\TestName{\TestGeneric\IfLanguageName}
\TestPatterns{USenglish}{true}
\TestPatterns{ngerman}{false}
\TestName{USenglish}{true}
\TestName{ngerman}{false}
\uselanguage{ngerman}
\TestPatterns{USenglish}{false}
\TestPatterns{ngerman}{true}
\TestName{USenglish}{false}
\TestName{ngerman}{true}
\csname @@end\endcsname
\end
\end{verbatim}

3.4 Test with plain \TeX and without -\TeX/pdf\TeX
\let\ifcsname\UNDEFINED\let\pdfstrcmp\UNDEFINED
\input iflang.sty
\catcode64=11
\def\TestDefined#1{\IfLang@IfDefined{foobar}{}{}
\ifx\foobar#1\else\errmessage{Failed test: \string\foobar <> \string#1}\fi}
\let\foobar\relax\TestDefined\relax\let\foobar\UNDEFINED\TestDefined\relax
\def\strip@prefix#1>{}
\def\@onelevel@sanitize#1{\edef#1{\expandafter\strip@prefix\meaning#1}}
\def\TestCompare#1#2#3{\begingroup\edef\x{\if\IfLang@StrEqual{#1}{#2}true\elsefalse\fi}
\def\expect{#3}\ifx\x\expect\else\def\a{#1}\@onelevel@sanitize\a\def\b{#2}\@onelevel@sanitize\b\errmessage{Failed test: \a='\b' <> \expect}\fi\endgroup}
\def\TestCompare\junk\junk{true}\TestCompare{}{}{true}\TestCompare{a}{b}{false}\TestCompare{aa}{bb}{false}\def\a{ax}\def\b{bx}\def\c{\a\b}\def\d{\c\b}\def\exch#1#2#2#1\def\gobble#1{}
\TestCompare\gobble a\{}{true}\TestCompare\{}{true}\TestCompare\{}{true}\def\exch xyb{ayxb}\TestCompare\c\c{true}\TestCompare\d\c\b{true}\TestCompare\c\{}{true}\TestCompare\{}\c\b{true}\TestCompare\gobble\a\{}{true}\TestCompare\{}\gobble a\{}{true}\TestCompare{axexch\ xyz\ b}{true}\TestCompare\c\c\{}{true}\TestCompare\{}\c\b\{}{true}\csname @@end\endcsname\end
4 Installation

4.1 Download

Package. This package is available on CTAN:


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) 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 iflang.dtx
```

TDS. Now the different files must be moved into the different directories in your installation TDS tree (also known as texmf):

```
iflang.sty -> tex/generic/oberdiek/iflang.sty
test/iflang-test1.tex -> doc/latex/oberdiek/test/iflang-test1.tex
test/iflang-test2.tex -> doc/latex/oberdiek/test/iflang-test2.tex
test/iflang-test3.tex -> doc/latex/oberdiek/test/iflang-test3.tex
test/iflang-test4.tex -> doc/latex/oberdiek/test/iflang-test4.tex
iflang.dtx -> source/latex/oberdiek/iflang.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.

---

1ftp://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:

```bash
pdftk iflang.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
latex \let\install=y\input{iflang.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:

```latex
\PassOptionsToClass{a4paper}{article}
```

An example follows how to generate the documentation with `pdf\LaTeX`:

```bash
pdflatex iflang.dtd
makeindex -s gind.ist iflang.idx
pdflatex iflang.dtd
makeindex -s gind.ist iflang.idx
pdflatex iflang.dtd
```

5 Catalogue

The following XML file can be used as source for the \TeX 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 `iflang.xml`.

```xml
<?xml version='1.0' encoding='us-ascii'?>
<!DOCTYPE entry SYSTEM 'catalogue.dtd'>
<entry datestamp='$Date$' modifier='$Author$' id='iflang'>
  <name>iflang</name>
  <caption>Expandable checks for the current language.</caption>
  <authorref id='auth:oberdiek'/>
  <copyright owner='Heiko Oberdiek' year='2007'/>
  <license type='lppl1.3'/>
  <version number='1.5'/>
  <description>
    This package provides expandable checks for the current language
    based on macro `<tt>\languagename</tt>` or hyphenation patterns.
  </description>
  <documentation details='Package documentation' href='ctan:/macros/latex/contrib/oberdiek/iflang.pdf'/>
  <ctan file='true' path='/macros/latex/contrib/oberdiek/iflang.dtd'/>
  <miktex location='oberdiek'/>
  <texlive location='oberdiek'/>
  <install path='/macros/latex/contrib/oberdiek/oberdiek.tds.zip'/>
</entry>
```

14
6 Acknowledgement

I wish to thank:

Markus Kohm Useful hints for version 1.2.

7 History

[2007/04/10 v1.0]
- First public version.

[2007/04/11 v1.1]
- Line ends sanitized.

[2007/04/12 v1.2]
- Initialization of \languagename in case of etex.src.
- Some sanity tests added.
- Documentation improved.

[2007/04/26 v1.3]
- Use of package infwarerr.

[2007/09/09 v1.4]
- Bug fix: \IfLang@StrEqual \rightarrow \IfLangStrEqual (Gabriele Balducci).
- Catcode section rewritten.

[2007/11/11 v1.5]
- Use of package pdftexcmds for LuaTEX support.

8 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.

<table>
<thead>
<tr>
<th>Symbols</th>
<th># ........................................... 286 } ....................................... 255</th>
</tr>
</thead>
<tbody>
<tr>
<td>%</td>
<td>........................................................................ 362 A</td>
</tr>
<tr>
<td>@</td>
<td>........................................... 287, 360 \a ................................... 529, 530, 533, 541, 543</td>
</tr>
<tr>
<td>\PackageInfoNoLine</td>
<td>159, 164, 194 \advance ......................... 325, 333, 348</td>
</tr>
<tr>
<td>\PackageWarningNoLine</td>
<td>177, 184, 274 \aftergroup ........................ 29</td>
</tr>
<tr>
<td>\ifstofone</td>
<td>........................................... 295, 298 \ifstoftwo .......................... 111,</td>
</tr>
<tr>
<td>\ifstofone</td>
<td>........................................... 111, 119, 128, 143, 206, 213, 254, 264 \b ........ 531, 532, 533, 542, 543, 544, 551</td>
</tr>
<tr>
<td>\gobble</td>
<td>........................................... 292, 300 \begin .............................. 392, 393, 402, 447</td>
</tr>
<tr>
<td>\onelevel@sanitize</td>
<td>423, 514, 530, 532 \body ............................ 304, 308</td>
</tr>
<tr>
<td>\seconftwo</td>
<td>........................................... 114, 121, 126, 145, 208, 215, 256, 266 \c ........ 543, 544, 550, 551</td>
</tr>
<tr>
<td>\undefined</td>
<td>........................................... 58 \catcode 2, 3, 5, 6, 7, 8, 9, 10, 11, 12,</td>
</tr>
<tr>
<td>\</td>
<td>........................................... 361  13, 33, 34, 36, 37, 38, 39, 40, 41,</td>
</tr>
<tr>
<td>{</td>
<td>........................................... 284</td>
</tr>
</tbody>
</table>

15