The *iflang* package

Heiko Oberdiek*

2018/01/21 v1.7

Abstract

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

Contents

1 Documentation 2

2 Implementation 3

  2.1 Reload check and package identification 3
  2.2 Tools 5

    2.2.1 Provide some basic macros of L\TeX{} 5
    2.2.2 Expandable existence check for macros 5
    2.2.3 Macros for messages 5
    2.2.4 Support for etex.src 6

  2.3 \IfLanguagePatterns 7
  2.4 \IfLanguageName 7
  2.5 Check plausibility of \languagename 8

3 Installation 8

  3.1 Download 8
  3.2 Bundle installation 9
  3.3 Package installation 9
  3.4 Refresh file name databases 9
  3.5 Some details for the interested 9

4 Acknowledgement 10

5 History 10

  [2007/04/10 v1.0] 10
  [2007/04/11 v1.1] 10
  [2007/04/12 v1.2] 10
  [2007/04/26 v1.3] 10
  [2007/11/11 v1.5] 10
  [2016/05/16 v1.6] 10
  [2018/01/21 v1.7] 11

6 Index 11

*Please report any issues at https://github.com/ho-tex/oberdiek/issues
1 Documentation

Package \texttt{babel} defines \texttt{iflanguage}. 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{iflanguage} fails.

However, package \texttt{babel} and some other packages such as \texttt{german} or \texttt{ngerman} store the language name in the macro \texttt{language} 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{language}. 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{language} the \texttt{(else)} part is executed.

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

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

Format based on \texttt{babel} (\texttt{language.dat}):
If package \texttt{babel} is not used (or not yet loaded), then \texttt{babel}'s \texttt{hyphen.cfg} has set \texttt{language} 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{language} is basically garbage. Package \texttt{iflang} warns if \texttt{language} and \texttt{language} do not fit. This can be fixed by loading package \texttt{babel} previously.

Format based on \texttt{e-TpX}'s \texttt{etex.src} (\texttt{language.def}):
Unhappily it does not support \texttt{language}. Thus this package hooks into \texttt{uselanguage} to get \texttt{language} 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{language} is set to this language. Otherwise a \texttt{language} 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{et@xpatterns} (defined in \texttt{etex.src}) to find out the naming convention in use.
2 Implementation

2.1 Reload check and package identification

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

\begin{verbatim}
\catcode=10 % \relax
\catcode48=10 %
\catcode13=5 % ^^M
\catcode32=10 %
\catcode35=6 % #
\catcode39=12 % '
\catcode40=12 % ( 
\catcode41=12 % )
\catcode44=12 % ,
\catcode45=12 % -
\catcode46=12 % .
\catcode47=12 % /
\catcode48=10 %
\catcode123=1 % {
\catcode125=2 % }
\expandafter\let\expandafter\x\csname ver@iflang.sty\endcsname
\ifx\x\relax % plain-Tex, first loading
  \else
    \def\empty{}%
    \ifx\x\empty % LaTeX, first loading, % variable is initialized, but \ProvidesPackage not yet seen
      \else
        \expandafter\ifx\csname PackageInfo\endcsname\relax
          \x{iflang}{The package is already loaded}%
        \else
          \PackageInfo{#1}{#2, stopped}%
        \fi
      \fi
    \fi
  \else
    \expandafter\let\expandafter\x\csname ver@iflang.sty\endcsname
    \ifx\x\relax % plain-Tex, first loading
      \else
        \def\empty{}%
        \ifx\x\empty % LaTeX, first loading, % variable is initialized, but \ProvidesPackage not yet seen
          \else
            \expandafter\ifx\csname PackageInfo\endcsname\relax
              \x{iflang}{The package is already loaded}%
            \else
              \PackageInfo{#1}{#2, stopped}%
            \fi
          \fi
        \fi
    \fi
  \fi
\end{verbatim}

Package identification:

\begin{verbatim}
\catcode=10 % \relax
\catcode48=10 %
\catcode13=5 % ^^M
\catcode32=10 %
\catcode35=6 % #
\catcode39=12 % '
\catcode40=12 % ( 
\catcode41=12 % )
\catcode44=12 % ,
\catcode45=12 % -
\catcode46=12 % .
\catcode47=12 % /
\catcode48=10 %
\catcode123=1 % {
\catcode125=2 % }
\expandafter\let\expandafter\x\csname ver@iflang.sty\endcsname
\ifx\x\relax % plain-Tex, first loading
  \else
    \def\empty{}%
    \ifx\x\empty % LaTeX, first loading, % variable is initialized, but \ProvidesPackage not yet seen
      \else
        \expandafter\ifx\csname PackageInfo\endcsname\relax
          \x{iflang}{The package is already loaded}%
        \else
          \PackageInfo{#1}{#2, stopped}%
        \fi
      \fi
    \fi
  \else
    \expandafter\let\expandafter\x\csname ver@iflang.sty\endcsname
    \ifx\x\relax % plain-Tex, first loading
      \else
        \def\empty{}%
        \ifx\x\empty % LaTeX, first loading, % variable is initialized, but \ProvidesPackage not yet seen
          \else
            \expandafter\ifx\csname PackageInfo\endcsname\relax
              \x{iflang}{The package is already loaded}%
            \else
              \PackageInfo{#1}{#2, stopped}%
            \fi
          \fi
        \fi
    \fi
  \fi
\end{verbatim}
2.2 Tools

2.2.1 Provide some basic macros of \LaTeX

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

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

2.2.2 Expandible existence check for macros

\IfLang@ifdefined
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname ifcsname\endcsname\relax
\expandafter\@firstoftwo
\else
\expandafter\@secondoftwo
\fi
\def\IfLang@ifdefined#1{\expandafter\ifx\csname#1\endcsname\relax
\expandafter\@secondoftwo
\else
\expandafter\@firstoftwo
\fi
\ifnum\ifcsname#1\endcsname
1\%
\else
0\%
\fi
\expandafter\@firstoftwo
\else
\expandafter\@secondoftwo
\fi
\fi
\}
\}
\def\IfLang@ifdefined#1{\ifnum\ifcsname#1\endcsname
1\%
\else
0\%
\fi
\expandafter\@firstoftwo
\else
\expandafter\@secondoftwo
\fi
\}

2.2.3 Macros for messages

\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname RequirePackage\endcsname\relax
\input infwarnerr.sty\relax
\input pdftexcmds.sty\relax
\endinput
2.2.4 Support for \texttt{etex.src}

\IfLang@prefix

\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname et@xpatterns\endcsname\relax
\@PackageInfoNoLine{iflang}{%
Naming convention for patterns: babel%
}\def\IfLang@prefix{l@}%
\else
\@PackageInfoNoLine{iflang}{%
Naming convention for patterns: etex.src%
}\def\IfLang@prefix{lang@}%
\let\IfLang@OrgUseLanguage\uselanguage
\def\uselanguage#1{%
\edef\languagename{#1}%
\IfLang@OrgUseLanguage{#1}%
}\fi
\fi
\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
\ifnum\lang@USenglish=\language
\def\languagename{USenglish}%
\else
\@PackageWarningNoLine{iflang}{%
\string\languagename\space is not set,\MessageBreak
current language is unknown%
}\fi
\fi
\fi
\@PackageInfoNoLine{iflang}{%
\string\languagename\space is not set%
}\fi
\fi
\fi
\fi
\@PackageInfoNoLine{iflang}{%
\string\languagename\space is not set%
}\fi
\fi
\section*{\texttt{IfLanguagePatterns}}

\begin{verbatim}
def \IfLanguagePatterns#1{% 
  \ifnum\IfLang@ifDefined{\IfLang@prefix#1}\% 
  \ifnum\csname\IfLang@prefix#1\endcsname=\language 
  0\% 
  \else 
  1\% 
  \fi 
  \expandafter\@firstoftwo 
}\else 
\expandafter\@secondoftwo 
\fi 
}
\end{verbatim}

\section*{\texttt{IfLanguageName}}

\begin{verbatim}
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname pdf@strcmp\endcsname\relax
\expandafter\@firstoftwo
\else
\expandafter\@secondoftwo
\fi
{ We do not have \texttt{pdf@strcmp} (and \texttt{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: \url{http://www.gust.org.pl/projects/pearls/2005p/david-kastrup/bachotex2005-david-kastrup-pearl1.pdf}.

The original code allows macros inside the second string. Because also \texttt{language} 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 
}\end{verbatim}
2.5 Check plausibility of \languagename

\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname languagename\endcsname\relax
\else
\IfLanguagePatterns{\languagename}{}{%
\PackageWarningNoLine{iflang}{Mismatch between \string\language\space
\MessageBreak
\string{patterns} and setting of \string\languagename}
}\}%
\fi
\IfLang@AtEnd%
⟨/package⟩

3 Installation

3.1 Download

Package. This package is available on CTAN¹:

¹CTAN:pkg/iflang
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.

TDS refers to the standard “A Directory Structure for \TeX Files” (CTAN:pkg/tds). Directories with \texttt{texmf} in their name are usually organized this way.

3.2 Bundle installation

Unpacking. Unpack the \texttt{oberdiek.tds.zip} in the TDS tree (also known as \texttt{texmf} tree) of your choice. Example (linux):

```sh
unzip oberdiek.tds.zip -d ~/texmf
```

3.3 Package installation

Unpacking. The \texttt{.dtx} file is a self-extracting \texttt{docstrip} archive. The files are extracted by running the \texttt{.dtx} through plain \TeX:

```sh
tex iflang.dtx
```

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

- \texttt{iflang.sty} → \texttt{tex/generic/oberdiek/iflang.sty}
- \texttt{iflang.pdf} → \texttt{doc/latex/oberdiek/iflang.pdf}
- \texttt{iflang.dtx} → \texttt{source/latex/oberdiek/iflang.dtx}

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

3.4 Refresh file name databases

If your \TeX distribution (\TeX Live, MiK\TeX, ...) relies on file name databases, you must refresh these. For example, \TeX Live users run \texttt{texhash} or \texttt{mktexlsr}.

3.5 Some details for the interested

Unpacking with \LaTeX. The \texttt{.dtx} chooses its action depending on the format:

- \texttt{plain \TeX}: Run \texttt{docstrip} and extract the files.
- \texttt{\LaTeX}: Generate the documentation.

If you insist on using \LaTeX for \texttt{docstrip} (really, \texttt{docstrip} does not need \LaTeX), then inform the autodetect routine about your intention:

```sh
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 \texttt{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:

\begin{verbatim}
pdflatex iflang.dtx
makeindex -s gind.ist iflang.idx
pdflatex iflang.dtx
makeindex -s gind.ist iflang.idx
pdflatex iflang.dtx
\end{verbatim}

4 Acknowledgement

I wish to thank:

Markus Kohm Useful hints for version 1.2.

5 History

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

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

[2007/04/12 v1.2]
• Initialization of \texttt{\textbackslash languagename} in case of \texttt{etex.src}.
• Some sanity tests added.
• Documentation improved.

[2007/04/26 v1.3]
• Use of package \texttt{infwarerr}.

[2007/09/09 v1.4]
• Bug fix: \texttt{\textbackslash IfLang@StrEqual} → \texttt{\textbackslash IfLangStrEqual} (Gabriele Balducci).
• Catcode section rewritten.

[2007/11/11 v1.5]
• Use of package \texttt{pdftexcmds} for \LaTeX{} support.

[2016/05/16 v1.6]
• Documentation updates.
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
\@PackageInfoNoLine ... 159, 164, 194
\@PackageWarningNoLine 177, 184, 274
\@firstoftwo ................................ 111,
119, 128, 143, 206, 213, 254, 264
\@secondoftwo ......................... 114,
121, 126, 145, 208, 215, 256, 266
\@undefined ....................... 58
\aftergroup ................................ 29
\catcode 2, 3, 5, 6, 7, 8, 9, 10, 11, 12,
13, 33, 34, 36, 37, 38, 39, 40, 41,
42, 43, 44, 45, 46, 47, 48, 49, 69,
70, 72, 73, 74, 78, 79, 80, 81, 82,
83, 84, 87, 88, 90, 91, 92, 93, 97, 99
\csname .................................. 14, 21, 50, 66,
76, 111, 114, 118, 125, 134, 150,
158, 174, 176, 193, 200, 212, 271
\empty .................................... 17, 18
\endcsname 14, 21, 50, 66, 76, 111,
114, 118, 125, 133, 134, 150,
158, 174, 176, 193, 200, 212, 271
\endinput ................................ 29, 110
\endlinechar .......... 4, 35, 71, 77, 89
\if .................................. 220, 247
\ifcat .................................. 226
\ifcsname ................................ 133
\IfLang@StrExpand .... 240, 242
\IfLang@StrExpand ..... 237, 239
\IfLang@AtEnd .......... 95, 96, 110, 281
\IfLang@IfDefined .... 117, 199, 246, 261
\IfLang@OrgUseLanguage ...... 168, 171
\IfLang@prefix .......... 157, 190, 200
\IfLang@StrEqual ........ 219, 247
\IfLang@StrEqualStart .... 220, 222, 229, 231, 235, 243
\IfLang@StrEqualStop .... 224, 231
\IfLang@StrExpand .......... 227, 235
\PackageInfo ................. 26
\pdf@strcmp .................. 262
\ProvidesPackage ........ 19, 67
\RequirePackage .......... 154, 155
\space .......................... 178, 185, 195, 275
\the .................. 77, 78, 79, 80, 81, 82, 83, 84, 97
\Tmp@EnsureCode ........... 94, 101, 102,
\uselanguage ............ 168, 169
\write .......................... 23, 52
\x 14, 15, 18, 22, 26, 28, 51, 56, 66, 75, 87