The \texttt{iflang} package

Heiko Oberdiek\textsuperscript{*}

2018/01/21 v1.7

Abstract

This package provides expandible checks for the current language based on macro \texttt{\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 \texttt{\LaTeX} 5

2.2.2 Expandable existence check for macros 5

2.2.3 Macros for messages 5

2.2.4 Support for \texttt{etex.src} 6

2.3 \texttt{\IfLanguagePatterns} 7

2.4 \texttt{\IfLanguageName} 7

2.5 Check plausibility of \texttt{\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

\textsuperscript{*}Please report any issues at \url{https://github.com/ho-tex/oberdiek/issues}
1 Documentation

Package `babel` defines \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 \iflanguage fails.

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

\IfLanguageName{⟨lang⟩}{⟨then⟩}{⟨else⟩}

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

The macro is expandable. Thus it can be safely used inside \edef or \csname. If case of errors like an undefined \language the ⟨else⟩ part is executed.

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

Package `babel`:

Full support of \language in its language switching commands.

Format based on `babel` (language.dat):

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

Format based on e-\TeX's `etex.src` (language.def):

Unhappily it does not support \language. Thus this package hooks into `\uselanguage` to get \language defined and updated there. At package loading time the changed `\uselanguage` has not been called yet. Thus package `iflang` tries `USenglish`. This is the definite default language of `etex.src`. If the current patterns suit this default language, an undefined \language is set to this language. Otherwise a \language remains undefined and a warning is given.

\IfLanguagePatterns{⟨lang⟩}{⟨then⟩}{⟨else⟩}

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

The following naming convention for the pattern are supported:

`babel/language.dat` : \l@⟨language⟩

`etex.src/language.def` : \lang@⟨language⟩

Package `iflang` looks for \et@xpatterns (defined in `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}
\begin{group}
\catcode13=5 \relax\% \\
\endlinechar=13 \%
\catcode35=6 \% # \\
\catcode39=12 \% \\
\catcode40=12 \% ( \\
\catcode41=12 \% ) \\
\catcode44=12 \% , \\
\catcode45=12 \% - \\
\catcode46=12 \% . \\
\catcode58=12 \% : \\
\catcode64=11 \% @ \\
\catcode123=1 \% { \\
\catcode125=2 \% }
\expandafter\ifx\csname ver@iflang.sty\endcsname\relax
\else
\def\empty{}\%
\ifx\empty \LaTeX, first loading,
\% variable is initialized, but \ProvidesPackage not yet seen
\else
\expandafter\ifx\csname PackageInfo\endcsname\relax
\def\x#1#2{\immediate\write-1{Package #1 Info: #2.}}\%
\else
\def\x#1#2{\PackageInfo{#1}{#2, stopped}}\%
\fi
\x{iflang}{The package is already loaded}\%
\aftergroup\endinput
\fi
\endgroup
\end{verbatim}

Package identification:

\begin{verbatim}
\begin{group}
\catcode13=5 \relax\% \\
\endlinechar=13 \%
\catcode35=6 \% # \\
\catcode39=12 \% \\
\catcode40=12 \% ( \\
\catcode41=12 \% ) \\
\catcode44=12 \% , \\
\catcode45=12 \% - \\
\catcode46=12 \% . \\
\catcode58=12 \% : \\
\catcode64=11 \% @ \\
\catcode91=12 \% [ \\
\catcode93=12 \% ]
\expandafter\ifx\csname ProvidesPackage\endcsname\relax
\else
\def\x#1#2#3[#4]{\endgroup
\def\x#1#2#3[#4]{\endgroup
\end{verbatim}

3
\immediate\write-1{Package: #3 #4}\%
\xdef#1{#4}\%
\else
\def\x#1#2[#3]{\endgroup #2[#3]\%
\ifx#1\undefined
\xdef#1{#3}\%
\fi
\ifx#1\relax
\xdef#1{#3}\%
\fi
}\fi
\expandafter\x\csname ver@iflang.sty\endcsname
\ProvidesPackage{iflang}[
2018/01/21 v1.7 Checks for the current language (HO)]%
\begingroup\catcode61\catcode48\catcode32=10\relax\
\catcode13=5 % ^^M
\endlinechar=13 %
\catcode35=6 % #
\catcode64=11 % @
\catcode123=1 % {
\catcode125=2 % }
\def\x{\endgroup
\expandafter\edef\csname IfLang@AtEnd\endcsname{\
\endlinechar=\the\endlinechar\relax
\catcode13=\the\catcode13\relax
\catcode32=\the\catcode32\relax
\catcode35=\the\catcode35\relax
\catcode61=\the\catcode61\relax
\catcode64=\the\catcode64\relax
\catcode123=\the\catcode123\relax
\catcode125=\the\catcode125\relax
}
}\%\%
\x\catcode61\catcode48\catcode32=10\relax%
\catcode13=5 % ^^M
\endlinechar=13 %
\catcode35=6 % #
\catcode64=11 % @
\catcode123=1 % {
\catcode125=2 % }
\def\TMP@EnsureCode#1#2{\%
\edef\IfLang@AtEnd{\IfLang@AtEnd
\catcode#1=\the\catcode#1\relax
\catcode#1=#2\relax
}
}\%
\TMP@EnsureCode{39}{12}\
\TMP@EnsureCode{40}{12}\
\TMP@EnsureCode{41}{12}\
\TMP@EnsureCode{44}{12}\
\TMP@EnsureCode{46}{12}\
\TMP@EnsureCode{47}{12}\
\TMP@EnsureCode{58}{12}\
\TMP@EnsureCode{91}{12}\
\TMP@EnsureCode{93}{12}]
}
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
}{\%
\def\IfLang@ifDefined#1{\%}
\}

2.2.3 Macros for messages

\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname RequirePackage\endcsname\relax
\input infwarerr.sty\relax
\input pdftexcmds.sty\relax
\
2.2.4 Support for 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
\fi
\fi
\fi
\fi
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname languagename\endcsname\relax
\@PackageInfoNoLine{iflang}{% \string\languagename\space is not set%
\fi
\fi
2.3 \IfLanguagePatterns

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

2.4 \IfLanguageName

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

The orignal code allows macros inside the second string. Because also \language might consists of further macros, we need a variant that allows macros in the first string, too.

\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  
  \else  
    \ifcat\noexpand#3\relax
    \IfLang@StrExpand{#1}{#2}%3  
    \fi  
  \fi  
  \IfLang@StrEqualStart{\if#3#1}{#2\fi}%  
}%  
\def\IfLang@StrEqualStop\fi\IfLang@StrEqualStart#2#3#4{  
  \fi  
  \#2\#4\relax\#313  
}  
\def\IfLang@StrExpand#1#2#3{\IfLang@StrEqualStart#4#5{  
  \fi  
  \IfLang@StrExpand{#1}{#2}\#3%  
}  
\def\IfLang@StrExpand#1#2#3\IfLang@StrNil{%  
  \expandafter\IfLang@StrExpand#3\IfLang@StrNil#1}{#2}%
2.5 Check plausibility of \languagename

\begin{verbatim}
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname languagename\endcsname\relax
\else
\IfLanguagePatterns{\languagename}{}{\@PackageWarningNoLine{iflang}{Mismatch between \string\language space and setting of \string\languagename}}\fi
\endgroup
\end{verbatim}

3 Installation

3.1 Download

Package. This package is available on CTAN\textsuperscript{1}:  
\footnotesize\textsuperscript{1}CTAN: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 T\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):

\begin{verbatim}
unzip oberdiek.tds.zip -d ~/texmf
\end{verbatim}

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:\

\begin{verbatim}
tex iflang.dtx
\end{verbatim}

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

\begin{verbatim}
iflang.sty → tex/generic/oberdiek/iflang.sty
iflang.pdf → doc/latex/oberdiek/iflang.pdf
iflang.dtx → source/latex/oberdiek/iflang.dtx
\end{verbatim}

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:

plain \TeX: Run \texttt{docstrip} and extract the files.

\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:

\begin{verbatim}
latex \let\install=y\input{iflang.dtx}
\end{verbatim}

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 pdfLa\TeX{}:

\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\textbackslash StrEqual} → \texttt{\textbackslash IfLangStrEqual} (Gabriele Balducci).
  
• Catcode section rewritten.

[2007/11/11 v1.5]

• Use of package \texttt{pdftexcmds} for Lua\TeX{} 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 \ldots 159, 164, 194
\@PackageWarningNoLine 177, 184, 274
\@firstoftwo \ldots 111, 119, 128, 143, 206, 213, 254, 264
\@secondoftwo \ldots 114, 121, 126, 145, 208, 215, 256, 266
\@undefined \ldots 58
\aftergroup \ldots 29
\catcode \ldots 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 \ldots 14, 21, 50, 66, 76, 111, 114, 118, 125, 134, 150, 158, 174, 176, 193, 200, 212, 271
\empty \ldots 17, 18
\endcsname \ldots 14, 21, 50, 66, 76, 111, 114, 118, 125, 134, 150, 158, 174, 176, 193, 200, 212, 271
\endinput \ldots 29, 110
\endlinechar \ldots 4, 35, 71, 77, 89
\if \ldots 220, 247
\ifcat \ldots 226
\ifcsname \ldots 133
\IfLang@@StrExpand \ldots 240, 242
\IfLang@@StrExpand \ldots 237, 239
\IfLang@AtEnd \ldots 95, 96, 110, 281
\IfLang@IfDefined \ldots 117, 199, 246, 261
\IfLang@OrgUseLanguage \ldots 168, 171
\IfLang@prefix \ldots 157, 190, 200
\IfLang@StrEqual \ldots 219, 247
\IfLang@StrEqualStart \ldots \ldots 220, 222, 229, 231, 235, 243
\IfLang@StrEqualStop \ldots 224, 231
\IfLang@StrExpand \ldots 227, 235
\IfLanguageName \ldots 2, 245, 260
\IfLanguagePatterns \ldots 2, 198, 273
\Ifnum \ldots 133, 181, 199, 200, 246, 261
\Ifx \ldots 15, 18, 21, 50, 58, 61, 111, 114, 118, 125, 134, 150, 158, 174, 176, 193, 212, 223, 271
\immediate \ldots 23, 52
\input \ldots 151, 152
\lang@USenglish \ldots 178, 181
\language \ldots 181, 200, 275
\languagename \ldots 170, 182, 185, 195, 248, 262, 273, 277
\MessageBreak \ldots 185, 276
\number \ldots 220
\PackageInfo \ldots 26
\pdfstrcmp \ldots 262
\ProvidesPackage \ldots 19, 67
\RequirePackage \ldots 154, 155
\space \ldots 178, 185, 195, 275
\struetrue \ldots 220
\string \ldots 132
\the \ldots 77, 78, 79, 80, 81, 82, 83, 84
\tmp@EnsureCode \ldots 94, 101, 102, 103, 104, 105, 106, 107, 108, 109
\uselanguage \ldots 168, 169
\write \ldots 23, 52
\x \ldots 14, 15, 18, 22, 26, 28, 51, 56, 66, 75, 87

[2018/01/21 v1.7]
• Fix test for etex.src.