1 Introduction

This is the documented code for the tracklang package. See tracklang-manual.pdf for the main user guide.
### 2 Main Package Code

### 3 \LaTeX{} Code (tracklang.sty)

To ensure maximum portability this file only uses \LaTeX{} kernel commands, rather than using more convenient commands provided by packages such as etoolbox.

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{tracklang}[2022/12/13 v1.6.1 (NLCT) Track Languages]

\@tracklang@declareoption \Set up package options.
\providecommand*{\@tracklang@declareoption}[1]{%
  \DeclareOption{#1}{\TrackPredefinedDialect{#1}}%
}

Load generic code:
\input{tracklang}

There are no other options as this package will typically be loaded using \RequirePackage by a package. Explicitly adding an option at that point might create a package option clash. The declared package options are all the possible language names that might be passed as a document class option. (Also, adding any non-language options here will interfere with \@tracklang@declaredoptions.)

\let\@tracklang@declaredoptions\@declaredoptions
\ProcessOptions

Unset \@tracklang@declareoption:
\let\@tracklang@declareoption\@gobble

In the event that the language hasn’t been supplied through the package options (or through the class options, which the package options should process provided the document class has used the standard option declarations) we need to check if any of the known language packages have been loaded. This is a bit risky as it relies on the packages not changing their internal language macros. It would be easier if all the language packages could provide a reliable user interface to determine which languages (and variants) have been loaded.

\@tracklang@prelangpkgcheck@hook

In the event that any of the language packages choose to load this package and use \texttt{tracklang.tex}'s commands to identify the tracked languages and regions, then they can first define \texttt{@tracklang@prelangpkgcheck@hook} to do \texttt{endinput} which will skip the checks.

\csname @tracklang@prelangpkgcheck@hook\endcsname

\ifx\@tracklang@languages\@empty

First try \texttt{babel}. If \texttt{babel} has been loaded, the languages are stored in \texttt{\bbl@loaded}, so check if this command has been defined, and if it has add those languages.
If `translator` has been loaded, the languages are stored in `\trans@languages`.

\@ifundefined{trans@languages}
{%
Has `ngerman` been loaded?
\@ifpackageloaded{ngerman}
{%
\@tracklang@add@ngerman
}%
%
Has `german` been loaded?
\@ifpackageloaded{german}
{%
\@tracklang@add@german
}%
%
Has `polyglossia` been loaded?
\@ifpackageloaded{polyglossia}
{%
polyglossia now provides `\xpg@bcp@loaded`, which contains a comma-separated list of language tags, so use that if it exists.
\@ifundefined{xpg@bcp@loaded}
{%
Older versions: `polyglossia` sets `⟨lang⟩@loaded` for each loaded language, so check this for all known languages. I don’t know how to consistently check for variants. (Conditionals such as `\if@british@locale` are set immediately with `\setotherlanguage` but are deferred to the start of the document with `\setmainlanguage`, which is too late for `tracklang`.) Script names seem to be stored in `\xpg:scripttag⟨language⟩` but again this doesn’t seem to be set for the main language until the start of the document. New versions of `polyglossia` store the list of loaded languages in `\xpg@loaded`, so check if this is defined.
\@ifundefined{xpg@loaded}
{%
\xpg@loaded isn’t defined, so iterate over known options and check if the language has been loaded.
\PackageInfo{tracklang}{polyglossia loaded but \string\xpg@loaded\space not defined. Will attempt to track known languages.}%
\@for\this@language:=\@tracklang@declaredoptions\do{%
\@ifundefined{\this@language @loaded}
{%
}%
\@nameuse{@tracklang@add@\this@language}
}%
}%}
These are language tags, which is more convenient.

\expandafter\TrackLanguageTag{\this@language}

The \xpg@bcp@loaded list doesn’t seem to include the main language until the start of the document environment but \mainbabelname is defined after \setmainlanguage. I’m not sure if \mainbabelname was provided in the same version of polyglossia as \xpg@bcp@loaded or a different one, so have a separate conditional.

\@ifundefined{mainbabelname}{}%
{%
\let\@org@tracklang@addlatex\@tracklang@addlatex
\def\@tracklang@addlatex{%
\PackageWarning{tracklang}{Can’t detect polyglossia main language. It may not have been set yet}%
}%
\@ifundefined{@tracklang@add\mainbabelname}{}%
{%
\PackageWarning{tracklang}{Adding unknown polyglossia babel name ‘\mainbabelname’}%
\expandafter\AddTrackedLanguage{\mainbabelname}%
}%
{\@nameuse{@tracklang@add\mainbabelname}%n%
\let\@tracklang@addlatex\@org@tracklang@addlatex
}%
}%
%
None of the known packages have been loaded, so do nothing in case another package wants to load this one before setting up the language options. However,
if at this point \texttt{babel} has been loaded, then it’s an older version that hasn’t defined \texttt{\bbl@loaded} so check for this.

\begin{verbatim}
\ifpackageloaded{babel}
  \PackageInfo{tracklang}{babel loaded but \string\bbl@loaded\space not defined. Will attempt to track known languages.}%
  \foreach this@language in \tracklang@declaredoptions do{%
    \ifdefundefined{captions\this@language}\else{\nameuse{\tracklang@add\this@language}}\fi
  }%
\end{verbatim}

Add from \texttt{translator}. If \texttt{translator} has been loaded, the language names are stored in \texttt{\trans@languages} but these are labels used by \texttt{translator}, so some mapping is required.

\begin{verbatim}
\let\tracklang@add@Acadian\tracklang@add@acadian
\let\tracklang@add@French\tracklang@add@french
\let\tracklang@add@Afrikaans\tracklang@add@afrikaans
\let\tracklang@add@Dutch\tracklang@add@dutch
\let\tracklang@add@AmericanEnglish\tracklang@add@american
\let\tracklang@add@Austrian\tracklang@add@austrian
\namedef{\tracklang@add@Austrian1997}{\tracklang@add@naustrian
\let\tracklang@add@Bahasa\tracklang@add@bahasa
\let\tracklang@add@Basque\tracklang@add@basque
\let\tracklang@add@Brazilian\tracklang@add@brazil
\let\tracklang@add@Portuguese\tracklang@add@portuguese
\let\tracklang@add@Breton\tracklang@add@breton
\let\tracklang@add@BritishEnglish\tracklang@add@british
\let\tracklang@add@Bulgarian\tracklang@add@bulgarian
\let\tracklang@add@Canadian\tracklang@add@canadian
\let\tracklang@add@Canadien\tracklang@add@canadien
\let\tracklang@add@Catalan\tracklang@add@catalan
\let\tracklang@add@Croatian\tracklang@add@croatian
\let\tracklang@add@Czech\tracklang@add@czech
\let\tracklang@add@Danish\tracklang@add@danish
\let\tracklang@add@Dutch\tracklang@add@dutch
\let\tracklang@add@English\tracklang@add@english
\let\tracklang@add@Esperanto\tracklang@add@esperanto
\let\tracklang@add@Estonian\tracklang@add@estonian
\let\tracklang@add@Finnish\tracklang@add@finnish
\let\tracklang@add@French\tracklang@add@french
\let\tracklang@add@Galician\tracklang@add@galician
\end{verbatim}

5
Now iterate through the declared languages:

```
\texttt{% \@for\this@language:=\trans@languages\do{%}
  \@ifundefined{\tracklang@add\this@language}{}{
    \nameuse{\tracklang@add\this@language}}%
}%
```

Add from babel:

```
\texttt{% \@for\this@language:=\bbl@loaded\do{%}
  \@ifundefined{\tracklang@add\this@language}{%}
    \PackageWarning{tracklang}{{Adding unknown babel language \'\this@language\'}}%
    \AddTrackedLanguage{\this@language}%
  }{
    \nameuse{\tracklang@add\this@language}}%
}%
```

If \texttt{babel} has been loaded with \texttt{serbian}, then the script needs to be set to \texttt{Latn}.

\texttt{6}
4 Generic Code (tracklang.tex)

Does the category code of @ need changing?

\@tracklang@restore@at
\ifnum\catcode'@=11\relax
  \def\@tracklang@restore@at{}%
\else
  \expandafter\edef\csname @tracklang@restore@at\endcsname{\
    \noexpand\catcode'\noexpand@=\number\catcode'@\relax
  }%
  \catcode'@=11\relax
\fi

First check if this file has already been loaded:
\ifx\@tracklang@languages\undefined
\else
  \@tracklang@restore@at
\expandafter\endinput
\fi

Version info.
\expandafter\def\csname ver\@tracklang.tex\endcsname{\
  2022/12/13 v1.6.1 (NLCT) Track Languages Generic Code}

Define a long command for determining the existence of a control sequence by
its name. (\relax is considered undefined.)

\@tracklang@ifundef
\long\def\@tracklang@ifundef#1#2#3{%
  \ifcsname#1\endcsname
  \expandafter\ifx\csname #1\endcsname\relax
    #2%
  \else
    #3%
  \fi
\else
  \expandafter\ifx\csname #1\endcsname\relax
    #2%
  \else

(The Cyrillic script is provided with serbianc.)
\ifx\captionsserbian\undefined
\else
  \SetTrackedDialectScript{serbian}{Latn}%
\fi

End of check for language packages

\fi
\fi
\}

\ifcsname is an \TeX primitive. Need to check if it's defined.
\ifx\ifcsname\undefined
Not using \TeX.
\long\def\@tracklang@ifundef#1#2#3{%
  \expandafter\ifx\csname #1\endcsname\relax
    #2%
  \else
    #3%
  \fi
}
\fi
Can't have an else part here as \TeX won't recognise \ifcsname and we'll have
an unmatched end brace.
\fi

The shell escape stuff needs the Plain \TeX version of \input. This is
\@@input if we're using \LaTeX.
\@tracklang@input
\ifx\@@input\undefined
  \let\@tracklang@input\input
\else
  \let\@tracklang@input\@@input
\fi

Provide some commands in case the \LaTeX kernel hasn't been loaded.
\@tracklang@nnil
\ifx\@nnil\undefined
  \def\@tracklang@nnil{@nil}
\else
  \let\@tracklang@nnil\@nnil
\fi
\@tracklang@for
\ifx\@for\undefined
  \long\def\@tracklang@for#1:=#2\do#3{%
    \expandafter\def\expandafter\@fortmp\expandafter{#2}%
    \ifx\@fortmp\empty
      \else
      \expandafter
      \@tracklang@forloop #2,\@nnil,\@nnil\@nil \@@ #1{#3}%
    \fi
  }
\long\def\@tracklang@forloop#1,#2,\@nnil\@nil\@nil\@nil #1{% #3}%
\fi
8
\@tracklang@ifcseqcsname\{⟨\text{cs}⟩\}{⟨\text{csname}⟩}{⟨\text{true}⟩}{⟨\text{false}⟩}

Compares two commands, the first is reference as a control sequence the second is referenced as a control sequence name. Condition is false if the second isn’t defined (no check for the existence of the first).
\fi
} else
\def\@tracklang@pkgwarn#1\#2{%
  \ifTrackLangShowWarnings
    \PackageWarning{\#1}{\#2}%
  \fi
}\fi

\@tracklang@warn
\def\@tracklang@warn#1{\@tracklang@pkgwarn{tracklang}{#1}}%

\ifTrackLangShowInfo Allow user to switch information messages on or off.
  \newif\ifTrackLangShowInfo
  \TrackLangShowInfotrue
\@tracklang@info
  \ifx\PackageInfo\undefined
    \def\@tracklang@info#1{%
      \ifTrackLangShowInfo
        \newlinechar='\^^J
        \def\MessageBreak{\^^J}
        \message{\^^Jtracklang Info: #1 on line \the\inputlineno.\^^J}%
      \fi
    }%
  \else
    \def\@tracklang@info#1{%
      \ifTrackLangShowInfo
        \PackageInfo{tracklang}{#1}%
      \fi
    }%
  \fi
}\fi

\ifTrackLangShowVerbose Allow user to switch verbose (debugging) information messages on or off. Note
  that verbose information will be off regardless of this switch if the info switch
  is off.
  \newif\ifTrackLangShowVerbose
  \TrackLangShowVerbosefalse
\@tracklang@verbose
  \def\@tracklang@verbose#1{%
    \ifTrackLangShowVerbose
      \@tracklang@info{#1}%
    \fi
  }%
\@tracklang@ifFileExists
\if\IfFileExists\undefined
\long\def\@tracklang@ifFileExists#1#2#3{%
\openin0=#1 %
\ifeof0\relax
\def\@tracklang@tmp{#3}%
\else
\closein0\relax
\edef@filef@und{#1 }%
\def\@tracklang@tmp{#2}%
\fi
\@tracklang@tmp
}
\else
\let\@tracklang@ifFileExists\IfFileExists
\fi
\TrackLangEncodingName
\TrackLangEncodingName
Provide a default encoding name to avoid repeated tests to determine whether
or not \texttt{\inputencodingname} has been defined.
\if\inputencodingname\undefined
\def\TrackLangEncodingName{utf8}
\else
\def\TrackLangEncodingName{\inputencodingname}
\fi

Provide a way to query the environment variables \texttt{LC\_ALL} or \texttt{LANG} to deter-
mine the region and language. The result is stored in \texttt{\TrackLangEnv} if it can
be obtained. If it can’t be obtained, \texttt{\TrackLangEnv} is set to empty. Also de-
dine \texttt{\TrackLangQueryOtherEnv\{\langle name\rangle\}} to query \texttt{LC\_ALL}, \langle name\rangle, \texttt{LANG}. For
example
\%
\TrackLangQueryOtherEnv\{LC\_\_MONETARY\}
%

Note that there’s not much that can be done from within \TeX for the C or
POSIX locale or a locale starting with a slash, so provide a check for them.

\@tracklang@checklocale
\def\@tracklang@checklocale{%
\if\TrackLangEnv\empty
\else
\if\TrackLangEnv\@tracklang@locale@posix
\def\TrackLangEnv{}%
\else
\if\TrackLangEnv\@tracklang@locale@c
\def\TrackLangEnv{}%
\else
\expandafter\@@tracklang@checklocale
\fi
\fi
\fi

\expandafter\@\@tracklang@checklocale

12
Check for leading slash.
\def\@@tracklang@checklocale#1#2\relax{\ifx#1/\relax\def\TrackLangEnv{}\fi\fi}

\@tracklang@locale@posix \def\@tracklang@locale@posix{POSIX}
\@tracklang@locale@c \def\@tracklang@locale@c{C}
\ifx\directlua\undefined
We can’t use Lua, so we’ll have to use the shell escape if it’s enabled. First
determine if the shell escape is available.
\@tracklang@tryshellescape No shell escape.
\def\@tracklang@tryshellescape#1{%\def\TrackLangEnv{}%\ifx\pdfshellescape\undefined%\ifx\shellescape\undefined%Can’t determine if the shell escape has been enabled.
\def\@tracklang@tryshellescape#1{%\ifx\pdfshellescape\undefined%\ifx\shellescape\undefined%
\def\TrackLangQueryOtherEnv##1{\% 
@tracklang@warn{\string\TrackLangQueryOtherEnv{##1}\space non-operational as can’t determine if the shell escape has been enabled. (Consider using eTeX or pdfTeX.}\% 
\def\TrackLangEnv{}\% }
}\% 

\ifx\shellescape\relax 
\else 
\ifnum\shellescape=0\relax 
\else 
\def@tracklang@tryshellescape#1{#1}\% 
\fi 
\fi 
\fi 
\else 
\pdfshellescape is defined. Check no one’s been messing around with it and set it to \relax. (Default no-op already set.) 
\ifx\pdfshellescape\relax 
\pdfshellescape has been set to \relax. Is it possible that \shellescape is available?
\ifx\shellescape\undefined 
\else 
\ifx\shellescape\relax 
\else 
\shellescape is available.
\ifnum\shellescape=0\relax 
\else 
\def@tracklang@tryshellescape#1{#1}\% 
\fi 
\fi 
\fi 
\else 
\ifnum\pdfshellescape=0\relax 
\else 
\def@tracklang@tryshellescape#1{#1}\% 
\fi 
\fi 
\fi 
Try the shell escape: 
@tracklang@tryshellescape {\%
Not sure if a path is likely to occur with kpsewhich but check for it just in case.

Try texosquery if available.

Try texosquery if available.

Try texosquery if available.
Try texosquery if available.

\ifx\TeXOSQueryLocale\undefined
\@tracklang@warn{Locale environment variables unavailable (tried \texttt{LC\string\_ALL}, #1 and \texttt{LANG})}%
\else
\@tracklang@info{Using texosquery to find locale}%
\TeXOSQueryLocale\TrackLangEnv
\ifx\TrackLangEnv\empty
\@tracklang@warn{Locale can’t be found (tried querying \texttt{LC\string\_ALL}, #1 and \texttt{LANG} variables and tried using texosquery)}%
\fi
\fi
\fi
\fi
\fi
\else
\directlua
\TrackLangQueryEnv
\ifx\TrackLangEnv\empty
Try texosquery if available.
\fi
\fi
\fi
\fi
\else
\directlua
\TrackLangQueryEnv
\edef\TrackLangEnv{\directlua{
l = os.getenv("LC\_ALL")
if l == nil or l == "" or l == "C" or l == "POSIX" or string.find(l, "/") then
  l = os.getenv("LANG")
  if l == nil or l == "" or l == "C" or l == "POSIX" or string.find(l, "/") then
    l=os.setlocale(nil)
    if l == nil or l == "C" or l == "POSIX" or string.find(l, "/") then
      l = ""
    end
  end
end
tex.print(l)}}%
\ifx\TrackLangEnv\empty
Try texosquery if available.
\fi
\fi\TeXOSQueryLocale\undefined
\@tracklang@warn{Locale can’t be found through Lua (tried querying \texttt{LC\string\_ALL} and \texttt{LANG} variables and
os.setlocale(nil))%%
\else
\TeXOSQueryLocale\TrackLangEnv
\ifx\TrackLangEnv\empty
 \@tracklang@warn{Locale can’t be found through Lua
 (tried querying LC\string_ALL and LANG variables and
 os.setlocale(nil) and tried using texosquery)}%%
 \fi
 \fi
 \fi
\fi
\fi
\TrackLangQueryOtherEnv
\def\TrackLangQueryOtherEnv#1{%
 \edef\TrackLangEnv{\directlua{
 l = os.getenv("LC_ALL")
 if l == nil or l == "" or l == "C" or l == "POSIX"
 or string.find(l, "/") then
 l = os.getenv("#1")
 if l == nil or l == "" or l == "C" or l == "POSIX"
 or string.find(l, "/") then
 l = os.getenv("LANG")
 if l == nil or l == "" or l == "C" or l == "POSIX"
 or string.find(l, "/") then
 l= os.setlocale(nil)
 if l == nil or l == "C" or l == "POSIX"
 or string.find(l, "/") then
 l = ""
 end
 end
 end
 tex.print(l)}%}
 \ifx\TrackLangEnv\empty
 Try texosquery if available.
 \elseif\TeXOSQueryLocale\undefined
 \@tracklang@warn{Locale can’t be found through Lua
 (tried querying LC\string_ALL, #1 and LANG variables and
 os.setlocale(nil))}%%
 \else
 \TeXOSQueryLocale\TrackLangEnv
 \ifx\TrackLangEnv\empty
 \@tracklang@warn{Locale can’t be found through Lua
 (tried querying LC\string_ALL, #1 and LANG variables and
 os.setlocale(nil) and tried using texosquery)}%%
 \fi
 \fi
 \fi
}
Allowed formats for the localisation environment variables are

\[\langle iso-lang\rangle[.\langle iso-territory\rangle][.\langle encoding\rangle][@\langle modifier\rangle]\]

(where the square brackets above indicate an optional component not that there are literal square brackets.) This is a bit fiddly, so it needs to be broken up into manageable chunks.

\TrackLangParseFromEnv Parse \TrackLangEnv, if it has been set, and set \TrackLangEnvLang, \TrackLangEnvTerritory and \TrackLangEnvCodeSet. If the information is unavailable, the relevant commands will be set to empty. Use \TrackLangFromEnv to query, parse and set.

\def\TrackLangParseFromEnv{%
  \ifx\TrackLangEnv\undefined
    \@tracklang@warn{\string\TrackLangParseFromEnv\space non-operational as \string\TrackLangEnv\space hasn't been defined}%
    \def\TrackLangEnvLang{}
    \def\TrackLangEnvTerritory{}
    \def\TrackLangEnvCodeSet{}
    \def\TrackLangEnvModifier{}
  \else
    \ifx\TrackLangEnv\empty
      \@tracklang@warn{\string\TrackLangParseFromEnv\space non-operational as \string\TrackLangEnv\space is empty}%
      \def\TrackLangEnvLang{}
      \def\TrackLangEnvTerritory{}
      \def\TrackLangEnvCodeSet{}
      \def\TrackLangEnvModifier{}
    \else
      \@tracklang@parse@locale{\TrackLangEnv}%
      \let\TrackLangEnvLang@TrackLangEnvLang
      \let\TrackLangEnvTerritory@TrackLangEnvTerritory
      \let\TrackLangEnvCodeSet@TrackLangEnvCodeSet
      \let\TrackLangEnvModifier@TrackLangEnvModifier
    \fi
  \fi
}\@tracklang@parse@locale Parse localisation format.

\def\@tracklang@parse@locale#1{%
  Initialise.
  \def\@TrackLangEnvLang{}
  \def\@TrackLangEnvSubLang{}
  \def\@TrackLangEnvFirstSubLang{}
  \def\@TrackLangEnvTerritory{}
  \def\@TrackLangEnvCodeSet{}
  \def\@TrackLangEnvModifier{}
}
Just in case argument is empty or \relax.
\expandafter\ifx\expandafter\relax#1\relax
\else
Parse codeset and modifier first.
\expandafter\@tracklang@parseenv
#1..\relax\@tracklang@end@parseenv\@tracklang@result
\fi
\fi
}
\@tracklang@split@underscoreorhyp
Split oneither an underscore or a hyphen and store the result in \@tracklang@split@pre
and \@tracklang@split@post
\def\@tracklang@split@underscoreorhyp#1{%
First try to split on an underscore.
\@tracklang@split@underscore{#1}%
If the post part was empty, try to split on hyphen instead.
\ifx\@tracklang@split@post\empty
\@tracklang@split@hyphen{#1}%
If the post part was empty, maybe the underscore has had its category code
changed to 12.
\ifx\@tracklang@split@post\empty
\@tracklang@split@otherunderscore{#1}%
\fi
\fi
\}
\@tracklang@split@underscore
Split on an underscore and store the results in \@tracklang@split@pre
and \@tracklang@split@post. First make sure that the underscore has its normal
subscript category code.
{
\catcode`\_8\relax
\gdef\@tracklang@split@underscore#1{%
\@@tracklang@split@underscore#1__\relax\@tracklang@end@split@underscore
19
\@tracklang@split@underscore As above but where underscore has catcode 12.
\catcode'\_12\relax
\gdef\@tracklang@split@otherunderscore{\@tracklang@split@otherunderscore@remainder\relax\@tracklang@end@split@otherunderscore}
\gdef\@tracklang@split@otherunderscore@remainder\relax{\def\@tracklang@split@post{#1}}
\gdef\@tracklang@split@otherunderscore{\@tracklang@split@otherunderscore@remainder#1__\relax}
\gdef\@tracklang@split@otherunderscore@remainder#1__\relax{\def\@tracklang@split@post{#1}}

\@tracklang@split@hyphen Split on a hyphen and store the results in \@tracklang@split@pre and \@tracklang@split@post
\catcode'\-12\relax
\gdef\@tracklang@split@hyphen\relax{\@tracklang@split@hyphen@remainder\relax\@tracklang@end@split@hyphen}
\gdef\@tracklang@split@hyphen@remainder\relax{\def\@tracklang@split@post{#1}}
\gdef\@tracklang@split@hyphen@remainder#1--\relax{\def\@tracklang@split@post{#1}}
\@tracklang@parseenv Parse for the codeset. The first argument will be the language and (optionally) the territory. So the final argument is the control sequence to use to store the first argument, which can then be split.

\gdef\@tracklang@parseenv#1.#2.#3\@tracklang@end@parseenv#4{%
  \def\@TrackLangEnvCodeSet{#2}%
  \def#4{#1}%
  \ifx\@TrackLangEnvCodeSet\empty
    \tracklangparsemod#4%
  \else
    \tracklangparsemod\@TrackLangEnvCodeSet
  \fi
}

\tracklangparsemod Extract the modifier from the code set. The @ is rather awkward as we need to change its category code as it’s likely to be set to 12 within \TrackLangEnv. So change the category code of @ to 12, but this means we can’t use it in the command name, so although these are private internal commands they don’t look like internal commands.)

\{\catcode\@=12\relax
  \gdef\tracklangparsemod#1{
    \expandafter\tracklangparseenvatmod#1@@\relax\tracklangendparseenvatmod
    \let#1\tracklangtmp
  }%
  \gdef\tracklangparseenvatmod#1@#2@#3\tracklangendparseenvatmod{%
    \def\tracklangtmp{#1}%
  \csname @TrackLangEnvModifier\endcsname{#2}%
  \csname @tracklang@sanitize\expandafter\endcsname\csname @TrackLangEnvModifier\endcsname
}

\@tracklang@languages Provide a list to keep track of all the languages.

\def\@tracklang@languages{}

\@tracklang@dialects Provide a list to keep track of all the dialects. Here the “dialect” isn’t necessarily an actual dialect but may be a root language or a synonym. It will usually correspond to the language name as specified by the user in the package option.

\def\@tracklang@dialects{}
\@tracklang@ifinlist{(item)}{(list)}{(true part)}{(false part)}

Checks if \texttt{(item)} is in \texttt{(list)}. (Performs a one-level expansion on \texttt{(list)} but no expansion on \texttt{(item)}.)

\def\@tracklang@ifinlist#1#2#3#4{%
\def\@tracklang@doifinlist##1,#1,##2\end@tracklang@doifinlist{%
\def\@before{##1}%
\def\@after{##2}%
\expandafter\@tracklang@doifinlist\expandafter,#2,#1,\@nil\end@tracklang@doifinlist
\ifx\@after\@tracklang@nnil
not found
#4%
\else
found
#3%
\fi%
}

\@tracklang@add{(item)}{(list cs)}

Adds an item to the list given by \texttt{(list cs)}. Does nothing if \texttt{(item)} is empty or is already in the list. The \texttt{(item)} is fully expanded before being added.

\def\@tracklang@add#1#2{%
First find out if the item is empty.
\edef\@tracklang@element{#1}%
\if\@tracklang@element\empty
Item is empty, so do nothing.
\else
\expandafter\@tracklang@ifinlist\expandafter{\@tracklang@element}{#2}%
\fi
\if\@tracklang@element\empty
Already in list, so do nothing.
\elm%
\else
Not in list, so add.
\if\empty#2\relax
\let#2\@tracklang@element
\else
\edef#2{#2,\@tracklang@element}%
\fi
\fi
\AddTrackedDialect{(dialect name)}{(language name)}

Add a dialect. (v1.3 switched from unexpanded to expanded def. All labels should be expandable.)
\def\AddTrackedDialect#1#2{
  \@tracklang@add{#1}{\@tracklang@dialects}\
  \@tracklang@add{#2}{\@tracklang@languages}\
  \@tracklang@enamedef{\@tracklang@fromdialect@#1}{#2}\
  \@tracklang@ifundef{\@tracklang@todialect@#2}{}{\
    \def\@tracklang@lang{#1}\
    \expandafter\@tracklang@add\expandafter\@tracklang@lang\cdot\csname @tracklang@todialect@#2\endcsname\
  }\
}

Provide a convenient way of referencing the last dialect to be tracked.
\edef\TrackLangLastTrackedDialect{#1}\

\AddTrackedLanguage{(language name)}

Add a dialect.
\def\AddTrackedLanguage#1{
  \AddTrackedDialect{#1}{#1}\
}

4.2 Known Languages
\@tracklang@known@langs
List of known (root) languages (that may or may not be tracked).
\def\@tracklang@known@langs{}

\TrackLangNewLanguage{(language name)}{(639-1 code)}{(639-2 (T))}{(639-2 (B))}{(639-3)}{(3166-1)}{(default script)}

Identifies a new language that may be tracked. The code arguments may be empty if not available. (v1.3 switched from unexpanded to expanded def. All labels should be expandable.) Most root languages don’t have an associated country code as they’re spoken in multiple regions. The (default script) is
the default script identified with the ISO 15924 alpha script code. To reduce overheads, only define 639-3 if there’s no 639-1 or 639-2 code.

```
def\TrackLangNewLanguage#1#2#3#4#5#6#7{% 
@tracklang@add(#1){\@tracklang@known@langs}% 
edef\@tracklang@tmp(#2)% 
@if\@tracklang@tmp\empty
else
 \@tracklang@enamedef{@tracklang@knownisolang@#2}{#1}%;
 \@tracklang@enamedef{@tracklang@knowniso@639@1@#1}{#2}%;
 \@tracklang@enamedef{@tracklang@knowniso@639@2@#1}{#3}%;
 \fi
edef\@tracklang@tmp(#4)% 
@if\@tracklang@tmp\empty
else
 \@tracklang@enamedef{@tracklang@knownisolang@#4}{#1}%;
 \@tracklang@enamedef{@tracklang@knowniso@639@2B@#1}{#4}%;
 \fi
edef\@tracklang@tmp(#5)% 
@if\@tracklang@tmp\empty
else
 \@tracklang@enamedef{@tracklang@knownisolang@#5}{#1}%;
 \@tracklang@enamedef{@tracklang@knowniso@639@3@#1}{#5}%;
 \fi
edef\@tracklang@tmp(#6)% 
@if\@tracklang@tmp\empty
else
 \@tracklang@enamedef{@tracklang@knowniso@03160@#1}{#6}%;
 \fi
edef\@tracklang@tmp(#7)% 
@if\@tracklang@tmp\empty
else
 \@tracklang@enamedef{@tracklang@knowniso@script@0@#1}{#7}%;
 \fi
}
```

```
\TrackLangIfKnownLang{(language)}{(true)}{(false)}
```

Tests if ⟨language⟩ is known (but not necessarily tracked).

```
def\TrackLangIfKnownLang#1#2#3 {% 
\expandafter@tracklang@ifinlist\expandafter{#1}\@tracklang@known@langs}% 
{#2}% 
{#3}%
}
```
\TrackLangIfKnownIsoTwoLetterLang{(language)}{(true)}{(false)}

Checks if the given language has an ISO 639-1 language code (but is not necessarily tracked).
\begin{verbatim}
def\TrackLangIfKnownIsoTwoLetterLang#1#2#3{\
  @tracklang@ifundef{@tracklang@knowniso063901@#1}#3#2
}
\end{verbatim}

\TrackLangGetKnownIsoTwoLetterLang{(language)}

Gets the ISO 639-1 language code for the given language.
\begin{verbatim}
def\TrackLangGetKnownIsoTwoLetterLang#1{\
  @tracklang@nameuse{@tracklang@knowniso063901@#1}
}
\end{verbatim}

\TrackLangIfKnownIsoThreeLetterLang{(language)}{(true)}{(false)}

Checks if the given language has an ISO 639-2 language code (but is not necessarily tracked).
\begin{verbatim}
def\TrackLangIfKnownIsoThreeLetterLang#1#2#3{\
  @tracklang@ifundef{@tracklang@knowniso063902@#1}#3#2
}
\end{verbatim}

\TrackLangGetKnownIsoThreeLetterLang{(language)}

Gets the ISO 639-2 language code.
\begin{verbatim}
def\TrackLangGetKnownIsoThreeLetterLang#1{\
  @tracklang@nameuse{@tracklang@knowniso063902@#1}
}
\end{verbatim}

\TrackLangIfKnownIsoThreeLetterLangB{(language)}{(true)}{(false)}

Checks if the given language has an ISO 639-2 (B) language code (but is not necessarily tracked).
\def\TrackLangIfKnownIsoThreeLetterLangB#1#2#3{% 
  @tracklang@ifundef{@tracklang@knowniso063902B#1}% 
  {#3}% 
  {#2}% 
}\TrackLangGetKnownIsoThreeLetterLangB{⟨language⟩}

Gets the ISO 639-2 (B) language code.
\def\TrackLangGetKnownIsoThreeLetterLangB#1{% 
  @tracklang@nameuse{@tracklang@knowniso063902B#1}% 
}\TrackLangGetKnownLangFromIso{⟨ISO code⟩}{⟨true⟩}{⟨false⟩}

Checks if the given ISO language code (639-1 or 639-2 or 639-3) is recognised (but not necessarily tracked).
\def\TrackLangGetKnownLangFromIso#1#2#3{% 
  @tracklang@nameuse{@tracklang@knownisolang#1}% 
}\TrackLangGetKnownLangFromIso{⟨ISO code⟩}

Gets the root language label from the given ISO code (639-1 or 639-2).
\def\TrackLangGetKnownLangFromIso#1{% 
  @tracklang@nameuse{@tracklang@knownisolang#1}% 
}\TrackLangIfHasKnownCountry{⟨language⟩}{⟨true⟩}{⟨false⟩}

Checks if the given language has an ISO 3166-1 country code (but is not necessarily tracked).
\def\TrackLangIfHasKnownCountry#1#2#3{% 
  @tracklang@ifundef{@tracklang@knowniso031660#1}% 
  {#3}% 
  {#2}% 
}\TrackLangGetKnownCountry{⟨language⟩}
Fetches the ISO 3166-1 country code for the given language.
```latex
\def\TrackLangGetKnownCountry#1{\%\@tracklang@nameuse{@tracklang@knowniso@3166@#1}%%%%}
```

\TrackLangGetDefaultScript

Gets the default script for the given root language label.
```latex
\def\TrackLangGetDefaultScript#1{\%\@tracklang@nameuse{@tracklang@knowniso@script@#1}%%%%}
```

\TrackLangIfHasDefaultScript

If there’s a default script for \texttt{⟨language⟩}, do \texttt{⟨true⟩} otherwise do \texttt{⟨false⟩}.
```latex
\def\TrackLangIfHasDefaultScript#1#2#3{\%\@tracklang@ifundef{@tracklang@knowniso@script@#1}{#3}{#2}%%%%}
```

4.3 Mappings

\AddTrackedIsoLanguage

 Adds a mapping between the given ISO code and language name. There may be multiple mappings from an ISO code to a language name, but only one mapping from a language name to an ISO code. (v1.3 switched from unexpanded to expanded def. All labels should be expandable.)
```latex
\def\AddTrackedIsoLanguage#1#2#3{\%\@tracklang@enamedef{@tracklang@#1@isofromlang@#3}{#2}%%%%
\@tracklang@ifundef{@tracklang@#1@isotolang@#2}{}{\@tracklang@enamedef{@tracklang@#1@isotolang@#2}{#3}%%%%
{\%\def\@tracklang@lang{#3}%%%%
\expandafter\@tracklang@add\expandafter\@tracklang@lang\csname @tracklang@#1@isotolang@#2\endcsname
}}}%%%%}
```

\TrackedLanguageFromIsoCode

 Fetches the language label (or labels) associated with the given code.
```latex
\def\TrackedLanguageFromIsoCode#1#2{\%\@tracklang@nameuse{@tracklang@knowniso@1@isotolang@#2}%%%%}
```
\TrackedIsoCodeFromLanguage{(code type)}{(language)}
Fetches the code associated with the given language or dialect.
\def\TrackedIsoCodeFromLanguage#1#2{% 
  \@tracklang@nameuse{@tracklang@#1@isofromlang@#2}\
}%

\TrackedLanguageFromDialect{(dialect)}
Fetches the language name from the given dialect.
\def\TrackedLanguageFromDialect#1{% 
  \@tracklang@nameuse{@tracklang@fromdialect@#1}\
}%

\TrackedDialectsFromLanguage{(root language label)}
Fetches the tracked dialects whose language is given by ⟨root language label⟩.
\def\TrackedDialectsFromLanguage#1{% 
  \@tracklang@nameuse{@tracklang@todialect@#1}\
}%

\TwoLetterIsoCountryCode
\def\TwoLetterIsoCountryCode{3166-1}

\TwoLetterIsoLanguageCode
\def\TwoLetterIsoLanguageCode{639-1}

\ThreeLetterIsoLanguageCode
\def\ThreeLetterIsoLanguageCode{639-2}

\ThreeLetterExtIsoLanguageCode
\def\ThreeLetterExtIsoLanguageCode{639-3}

\SetTrackedDialectModifier{(dialect)}{(value)}
Set the modifier for ⟨dialect⟩. (For example, old or new.) Arguments are expanded.
\def\SetTrackedDialectModifier#1#2{% 
  \@tracklang@enamedef{@tracklang@modifier@#1}{#2}\
}%
\GetTrackedDialectModifier{(dialect)}

Get the modifier for \textit{dialect}.

\def\GetTrackedDialectModifier#1{%
  \@tracklang@nameuse{\@tracklang@modifier@#1}%
}\IfHasTrackedDialectModifier{(dialect)}{(true)}{(false)}

If there’s a modifier for \textit{dialect}, do \textit{true} otherwise do \textit{false}.

\def\IfHasTrackedDialectModifier#1#2#3{%
  \@tracklang@ifundef{\@tracklang@modifier@#1}{#3}{#2}%
}\SetTrackedDialectScript{(dialect)}{(value)}

Set the script for \textit{dialect}. (For example, Latn or Cyril.) Arguments are expanded.

\def\SetTrackedDialectScript#1#2{%
  \@tracklang@enamedef{\@tracklang@script@#1}{#2}%
}\GetTrackedDialectScript{(dialect)}

Get the script for \textit{dialect}.

\def\GetTrackedDialectScript#1{%
  \@tracklang@nameuse{\@tracklang@script@#1}%
}\IfHasTrackedDialectScript{(dialect)}{(true)}{(false)}

If there’s a script for \textit{dialect}, do \textit{true} otherwise do \textit{false}.

\def\IfHasTrackedDialectScript#1#2#3{%
  \@tracklang@ifundef{\@tracklang@script@#1}{#3}{#2}%
}\IfTrackedDialectIsScriptCs{(dialect)}{(cs)}{(true)}{(false)}

\IfTrackedDialectIsScriptCs
If the given tracked dialect has an associated script and that script code matches the replacement text for the control sequence \( cs \) then do \( \text{true} \) otherwise to \( \text{false} \). If the tracked dialect doesn’t have an associated script then the default script for the root language is tested. The use of a control sequence allows \( \text{ifx} \) for the test, which means that this command can expand. The supplementary package \texttt{tracklang-script} provides control sequences for known ISO 15924 codes.

\begin{verbatim}
def\IfTrackedDialectIsScriptCs#1#2#3#4{%
  \IfHasTrackedDialectScript{#1}%
  {%\%
    \expandafter\ifx\expandafter#2\csname @tracklang@script@#1\endcsname #3%
    \else #4%
    \fi
  %}
  {%\%
    \TrackLangIfHasDefaultScript{\TrackedLanguageFromDialect{#1}}%\%
    {%\%
      \expandafter\ifx\expandafter
      #2\csname @tracklang@knowniso@script@\TrackedLanguageFromDialect{#1}\endcsname #3%
      \else #4%
      \fi
    %}
    {#4}%
  %}
}
\end{verbatim}

\SetTrackedDialectVariant{\langle dialect \rangle}{\langle value \rangle}

Set the modifier for \langle dialect \rangle. (For example, old or new.) Arguments are expanded.

\begin{verbatim}
def\SetTrackedDialectVariant#1#2{%
  \@tracklang@enamedef{@tracklang@variant@#1}{#2}%
}
\end{verbatim}

\GetTrackedDialectVariant{\langle dialect \rangle}

Get the modifier for \langle dialect \rangle.

\begin{verbatim}
def\GetTrackedDialectVariant#1{%
  \@tracklang@nameuse{@tracklang@variant@#1}%
}
\end{verbatim}
\IfHasTrackedDialectVariant{\texttt{dialect}}{(true)}{(false)}

If there's a modifier for \texttt{dialect}, do \texttt{true} otherwise do \texttt{false}.
\texttt{\def\IfHasTrackedDialectVariant#1#2#3{%}
  \@tracklang@ifundef{@tracklang@variant@#1}{#3}{#2}%
}\}

\SetTrackedDialectSubLang{\texttt{dialect}}{(\texttt{value})}

Set the sublang for \texttt{dialect}. Arguments are expanded.
\texttt{\def\SetTrackedDialectSubLang#1#2{%}
  \@tracklang@enamedef{@tracklang@sublang@#1}{#2}%
}\}

\GetTrackedDialectSubLang{\texttt{dialect}}

Get the sublang for \texttt{dialect}.
\texttt{\def\GetTrackedDialectSubLang#1{%}
  \@tracklang@nameuse{@tracklang@sublang@#1}%
}\}

\IfHasTrackedDialectSubLang{\texttt{dialect}}{(true)}{(false)}

If there's a sublang for \texttt{dialect}, do \texttt{true} otherwise do \texttt{false}.
\texttt{\def\IfHasTrackedDialectSubLang#1#2#3{%}
  \@tracklang@ifundef{@tracklang@sublang@#1}{#3}{#2}%
}\}

\SetTrackedDialectAdditional{\texttt{dialect}}{(\texttt{value})}

Set the extra for \texttt{dialect}. Arguments are expanded.
\texttt{\def\SetTrackedDialectAdditional#1#2{%}
  \@tracklang@enamedef{@tracklang@extra@#1}{#2}%
}\}

\GetTrackedDialectAdditional{\texttt{dialect}}

Get the extra for \texttt{dialect}.
\texttt{\def\GetTrackedDialectAdditional#1{%}
  \@tracklang@nameuse{@tracklang@extra@#1}%
}\}
\IfHasTrackedDialectAdditional\{\langle\textit{dialect}\rangle\}\{\langle\textit{true}\rangle\}\{\langle\textit{false}\rangle\}

If there's extra info for \langle\textit{dialect}\rangle, do \langle\textit{true}\rangle otherwise do \langle\textit{false}\rangle.

\def\IfHasTrackedDialectAdditional#1#2#3{%
  \@tracklang@ifundef{@tracklang@extra@#1}{#3}{#2}%
}\GetTrackedLanguageTag\{\langle\textit{dialect}\rangle\}

Get the language tag for \langle\textit{dialect}\rangle.

\def\GetTrackedLanguageTag#1{%
  \IfTrackedLanguageHasIsoCode{639-1}{\TrackedLanguageFromDialect{#1}}%  
  {\TrackedIsoCodeFromLanguage{639-1}{\TrackedLanguageFromDialect{#1}}}%  
  {\%  
    \IfTrackedLanguageHasIsoCode{639-2}{\TrackedLanguageFromDialect{#1}}%  
    {\TrackedIsoCodeFromLanguage{639-2}{\TrackedLanguageFromDialect{#1}}}%  
    {\%  
      \IfTrackedLanguageHasIsoCode{639-3}{\TrackedLanguageFromDialect{#1}}%  
      {\TrackedIsoCodeFromLanguage{639-3}{\TrackedLanguageFromDialect{#1}}}%  
      {und}% undefined  
    }%  
  }%  
  \@tracklang@ifundef{@tracklang@sublang@#1}{}%  
  {-%\csname @tracklang@sublang@#1\endcsname}  
  \@tracklang@ifundef{@tracklang@script@#1}{}%  
  {-%\csname @tracklang@script@#1\endcsname}  
  \IfTrackedLanguageHasIsoCode{3166-1}{#1}%  
  {-%\TrackedIsoCodeFromLanguage{3166-1}{#1}}%  
  {}  
  \@tracklang@ifundef{@tracklang@variant@#1}{}%  
  {-%\csname @tracklang@variant@#1\endcsname}%  
  \@tracklang@ifundef{@tracklang@extra@#1}{}%  
  {-%\csname @tracklang@extra@#1\endcsname}%
\}

\setCurrentTrackedDialect\{\langle\textit{dialect}\rangle\}

Provided for use by language hooks to establish the current tracked dialect. This command doesn't change \texttt{\textbackslash language\name} or hyphenation patterns etc. It just provides convenient commands that can be accessed. The argument may
be a \texttt{tracklang} dialect label or the language hook label from which a \texttt{tracklang} dialect label can be obtained or the root language label.

\begin{verbatim}
def\SetCurrentTrackedDialect\#1\
    \edef\CurrentTrackedDialect{\#1}\
    \@tracklang@verbose{Setting current tracked dialect ‘\CurrentTrackedDialect’}\
    \IfTrackedDialect{\CurrentTrackedDialect}{}{\
        \@tracklang@verbose{Not a recognised tracklang label}\
    }
\end{verbatim}

Has a mapping from this dialect to a \texttt{tracklang} dialect been supplied?

\begin{verbatim}
    \IfHookHasMappingFromTrackedDialect{\CurrentTrackedDialect}{}{\
        \IfTrackedDialect{\CurrentTrackedDialect}{}{\
            \@tracklang@verbose{Not a recognised tracklang label}\
        }\
    }
\end{verbatim}

Has the root language name been supplied?

\begin{verbatim}
    \IfTrackedLanguage{\#1}{\
        \@tracklang@verbose{Supplied label is a root language. Finding last tracked dialect for language ‘\#1’}\
        \edef\@tracklang@dialects{\TrackedDialectsFromLanguage{\#1}}\
        \@tracklang@for\@tracklang@dialect:=\@tracklang@dialects\do{\
            \let\CurrentTrackedDialect\@tracklang@dialect\
        }\
        \@tracklang@verbose{No tracklang dialect found for label ‘\#1’}\
    }\
\end{verbatim}

Has the root language name been supplied?

\begin{verbatim}
    \IfTrackedLanguage{\#1}{\
        \@tracklang@verbose{Supplied label is a root language. Finding last tracked dialect for language ‘\#1’}\
        \edef\@tracklang@dialects{\TrackedDialectsFromLanguage{\#1}}\
        \@tracklang@for\@tracklang@dialect:=\@tracklang@dialects\do{\
            \let\CurrentTrackedDialect\@tracklang@dialect\
        }\
        \@tracklang@verbose{No tracklang dialect found for label ‘\#1’}\
    }\
\end{verbatim}

Has the root language name been supplied?

\begin{verbatim}
    \IfTrackedLanguage{\#1}{\
        \@tracklang@verbose{Supplied label is a root language. Finding last tracked dialect for language ‘\#1’}\
        \edef\@tracklang@dialects{\TrackedDialectsFromLanguage{\#1}}\
        \@tracklang@for\@tracklang@dialect:=\@tracklang@dialects\do{\
            \let\CurrentTrackedDialect\@tracklang@dialect\
        }\
        \@tracklang@verbose{No mapping found}\
    }
\end{verbatim}

Has the root language name been supplied?

\begin{verbatim}
    \IfTrackedLanguage{\#1}{\
        \@tracklang@verbose{Supplied label is a root language. Finding last tracked dialect for language ‘\#1’}\
        \edef\@tracklang@dialects{\TrackedDialectsFromLanguage{\#1}}\
        \@tracklang@for\@tracklang@dialect:=\@tracklang@dialects\do{\
            \let\CurrentTrackedDialect\@tracklang@dialect\
        }\
        \@tracklang@verbose{No tracklang dialect found for label ‘\#1’}\
    }\
\end{verbatim}

Has the root language name been supplied?

\begin{verbatim}
    \IfTrackedLanguage{\#1}{\
        \@tracklang@verbose{Supplied label is a root language. Finding last tracked dialect for language ‘\#1’}\
        \edef\@tracklang@dialects{\TrackedDialectsFromLanguage{\#1}}\
        \@tracklang@for\@tracklang@dialect:=\@tracklang@dialects\do{\
            \let\CurrentTrackedDialect\@tracklang@dialect\
        }\
        \@tracklang@verbose{No mapping found}\
    }
\end{verbatim}
Get the default script if not set.
\IfHasTrackedDialectScript{\CurrentTrackedDialect}{%
  \edef\CurrentTrackedDialectScript{\GetTrackedDialectScript{\CurrentTrackedDialect}}%
}{%
  \edef\CurrentTrackedDialectScript{\TrackLangGetDefaultScript{\CurrentTrackedLanguage}}%
}%
\edef\CurrentTrackedDialectSubLang{\GetTrackedDialectSubLang{\CurrentTrackedDialect}}%
\edef\CurrentTrackedDialectAdditional{\GetTrackedDialectAdditional{\CurrentTrackedDialect}}%
\edef\CurrentTrackedLanguageTag{\GetTrackedLanguageTag{\CurrentTrackedDialect}}%

Region code.
\IfTrackedLanguageHasIsoCode{3166-1}{\CurrentTrackedDialect}{%
  \edef\CurrentTrackedRegion{\TrackedIsoCodeFromLanguage{3166-1}{\CurrentTrackedDialect}}%
}{%}
\edef\CurrentTrackedRegion{}%

Language code.
\IfTrackedLanguageHasIsoCode{639-1}{\CurrentTrackedLanguage}{%
  \edef\CurrentTrackedIsoCode{\TrackedIsoCodeFromLanguage{639-1}{\CurrentTrackedLanguage}}%
}{%}
\IfTrackedLanguageHasIsoCode{639-2}{\CurrentTrackedLanguage}{%
  \edef\CurrentTrackedIsoCode{}}%
\AddTrackedLanguageIsoCodes{\{language\}}

Adds the 639-1, 639-2 and 639-3 ISO codes for the given language, which must have previously been declared using \TrackLangNewLanguage.
As above but adds the 3166-1 country code if provided. Most root languages don’t have an associated country code as they’re spoken in multiple regions. Some of those that do have an associated region code may also be spoken as a minority language elsewhere, so this is separate from the previous command. If a regionless setting is required, use \TrackLocale instead of \TrackPredefinedDialect.

4.4 Tracking Languages and Dialects

The commands here are provided to indicate that a language or dialect is active (tracked) in the document.
\TrackPredefinedDialect{\langle dialect label \rangle}

Track a predefined language or dialect.
\def\TrackPredefinedDialect#1{\%
  \@tracklang@ifundef{@tracklang@add@#1}\%
  {\@tracklang@err{Dialect or language ‘#1’ is not predefined}\%
    {\@tracklang@nameuse{@tracklang@add@#1}}\%
  }
}

\@tracklang@hassecondchar Check if second argument is present (non-empty and not \relax).
\def\@tracklang@hassecondchar#1#2\@end@tracklang@hassecondchar#3#4{\%
  \ifx\relax#2\relax
    #4\%
  \else
    #3\%
  \fi
}

\@tracklang@hasthirdchar Check if third argument is present (non-empty and not \relax).
\def\@tracklang@hasthirdchar#1#2#3\@end@tracklang@hasthirdchar#4#5{\%
  \ifx\relax#3\relax
    #5\%
  \else
    #4\%
  \fi
}

\@tracklang@hasfourthchar Check if fourth argument is present (non-empty and not \relax).
\def\@tracklang@hasfourthchar#1#2#3#4\@end@tracklang@hasfourthchar#5#6{\%
  \ifx\relax#4\relax
    #6\%
  \else
    #5\%
  \fi
}

\@tracklang@hasfifthchar Check if fifth argument is present (non-empty and not \relax).
\def\@tracklang@hasfifthchar#1#2#3#4#5\@end@tracklang@hasfifthchar#6#7{\%
  \ifx\relax#5\relax
    #7\%
  \else
    #6\%
  \fi
}

\@tracklang@hasninthchar Check if ninth argument is present (non-empty and not \relax).
\@tracklang@ifalldigits Check if the argument only consists of digits (no sign).
\def\@tracklang@ifalldigits#1{\%
  \expandafter\ifx\relax#1\relax
  \expandafter\@tracklang@secondoftwo
  \else
  \expandafter\@@tracklang@ifalldigits#1\@tracklang@nnil
  \fi
}\ @@tracklang@ifalldigits
\def\@@tracklang@ifalldigits#1{\%
  \ifx#1\@tracklang@nnil
    \def\@tracklang@next{\expandafter\@tracklang@firstoftwo}%
  \else
    \@tracklang@ifdigit{#1}{%\%
      \let\@tracklang@next{\@tracklang@hassecondchar}\%
      \def\@tracklang@next\@tracklang@ifalldigits\%
      \expandafter\@tracklang@secondoftwo\%
    }%\%
  \fi
  \@tracklang@next
}\ @@tracklang@ifalldigits
\@tracklang@ifalphanumeric Check if argument is an alphanumeric (0,...,9) or (a,...,z) or (A,...,Z).
\def\@tracklang@ifalphanumeric#1#2#3{\%
  \@tracklang@ifalpha{#1}{#2}{#3}
\} \@tracklang@ifalphanumeric
\TrackLangIfAlphaNumericChar\TrackLangIfAlphaNumericChar{⟨tag⟩}{⟨true⟩}{⟨false⟩}
\TrackLangIfAlphaNumericChar Check if the argument is a single alphanumeric character.
\def\TrackLangIfAlphaNumericChar#1#2#3{\%
  \expandafter\ifx\expandafter\relax#1\relax
    Tag empty or \relax.
    #3\%
  \else
    \expandafter\@tracklang@hassecondchar#1\relax\relax
    \@end@tracklang@hassecondchar
    #3\%
  \end\expandafter\@tracklang@hassecondchar\%
  }\%
}
Check if the argument is a language tag (two or three letter lower case).
\def\TrackLangIfLanguageTag#1#2#3{\%\expandafter\@tracklang@hasthirdchar#1\relax\relax\relax\@end@tracklang@hasthirdchar\{\%
Has 3 or more characters.
\expandafter\@tracklang@hasfourthchar#1\relax\@end@tracklang@hasfourthchar{#3}\%
Has 3 characters. Are they all lower case?
\expandafter\@tracklang@iflanguage@iii@tag#1{#2}{#3}\%
\}
Has less than 3 characters.
\expandafter\@tracklang@hassecondchar#1\relax\relax\@end@tracklang@hassecondchar{\%
Has two characters. Are they both lower case?
\expandafter\@tracklang@iflanguage@ii@tag#1{#2}{#3}\%
\}
\@tracklang@iflanguage@ii@tag
\def\@tracklang@iflanguage@ii@tag#1#2#3#4{\%
\ifnum\lccode'#1='#1\relax\ifnum\lccode'#2='#2\relax
#3\%
\else
#4\%
\fi
\else
#4\%
\fi
\}
\@tracklang@iflanguage@iii@tag
\def\@tracklang@iflanguage@iii@tag#1#2#3#4#5{\%
\ifnum\lccode'#1='#1\relax\ifnum\lccode'#2='#2\relax
#3\%
\else
#4\%
\fi
\else
#4\%
\fi
\}
Check if the argument is a region tag (two letter upper case or three digit numeric).
\def\TrackLangIfRegionTag#1#2#3{%\expandafter\@tracklang@hasthirdchar#1\relax\relax\relax\@end@tracklang@hasthirdchar{%Has 3 or more characters. Is it a three digit numeric code?
\expandafter\@tracklang@hasfourthchar#1\relax\@end@tracklang@hasfourthchar{%Has 4 or more characters.
 #3%
 }%{Has 3 characters. Are they all digits?
 \@tracklang@ifalldigits#1\{#2\}#3%}
}{Has less than 3 characters.
\expandafter\@tracklang@hassecondchar#1\relax\relax\@end@tracklang@hassecondchar{%Has two characters. Are they both upper case?
\expandafter\@tracklang@ifregion@ii@tag#1\{#2\}#3%
}{#3}%}
}
Check if the argument is a script tag (four letter title case).

\def\TrackLangIfScriptTag#1#2#3{%#3}
{\%#3}
{\%#3}
Has less than 5 characters.
\expandafter\@tracklang@hasfifthchar\relax\relax\relax\relax\relax
\@end\tracklang@hasfifthchar
{\%#3}
{\%
Has four characters. Are they title case? (First letter upper case, others lower case.)
\expandafter\@tracklang@ifscripttag\relax\relax\relax\relax\relax
\@end\tracklang@ifscripttag
{\%#3}
{\%
\TrackLangIfVariantTag\{tag\}\{{true}\}\{{false}\}

Check if the argument is a variant tag.
\def\TrackLangIfVariantTag#1#2#3{%
\expandafter\@tracklang@hasfifthchar#1\relax\relax\relax\relax\relax
\@end@tracklang@hasfifthchar
}%

Has at least 5 characters. Does it have a maximum of 8?
\expandafter\@tracklang@hasninthchar#1\relax\relax\relax\relax\relax\relax
\@end@tracklang@hasninthchar

%(#3)%
%(#2)%
%
%

Less than 5 characters.
\expandafter\@tracklang@hasfourthchar#1\relax\relax\relax\relax\relax\relax\relax
\@end@tracklang@hasfourthchar
%

Has 4 characters.
\expandafter\@tracklang@ifvariant@iv@tag#1{#2}{#3}%
four character variant starting with a digit.

\def\@tracklang@ifvariant@iv@tag#1#2#3#4#5#6{%\@tracklang@ifdigit{#1}% {#5} {#6}%}

\@tracklang@parse@extlang \@TrackLangEnvSubLang, \@tracklang@split@pre and \@tracklang@split@post should be initialised before use. This assumes the tag is well formed.
\def\@tracklang@parse@extlang{%\TrackLangIfLanguageTag{\@tracklang@split@pre} {\ifx\@TrackLangEnvSubLang\empty \let\@TrackLangEnvSubLang\@tracklang@split@pre \let\@TrackLangEnvFirstSubLang\@TrackLangEnvSubLang \else \edef\@TrackLangEnvSubLang{\@TrackLangEnvSubLang-\@tracklang@split@pre}\fi} \ifx\@tracklang@split@post\empty \else \expandafter\@tracklang@split@underscoreorhyp\expandafter{\@tracklang@split@post}\fi \fi
\@tracklang@parse@variant \@TrackLangEnvVariant, \@tracklang@split@pre and \@tracklang@split@post should be initialised before use.
\def\@tracklang@parse@variant{%\TrackLangIfVariantTag{\@tracklang@split@pre} {\ifx\@TrackLangEnvVariant\empty \let\@TrackLangEnvVariant\@tracklang@split@pre \else \edef\@TrackLangEnvVariant{\@TrackLangEnvVariant-\@tracklang@split@pre}\fi}
Split again if there's more.

\ifx\@tracklang@split@post\empty
  \else
    \expandafter\@tracklang@split@underscoreorhyp\expandafter
    {\@tracklang@split@post}\
  \fi
\fi
\%
{%}
}

\TrackLanguageTag{⟨tag⟩}

Parse RFC 5646 language tag (assumes regular and well-formed). See also https://tools.ietf.org/html/rfc5646. Ensure ⟨tag⟩ is fully-expanded. Warn if argument is empty.

\def\TrackLanguageTag#1{\def\@tracklang@tag{#1}
  \ifx\@tracklang@tag\empty
    \@tracklang@warn{Empty tag in \string\TrackLanguageTag}\
  \else
    \expandafter\@TrackLanguageTag\expandafter{\@tracklang@tag}\
  \fi
}\%

\@TrackLanguageTag Argument must be expanded.
\def\@TrackLanguageTag#1{\def\@tracklang@tag{#1}
First check if it’s predefined.
  \@tracklang@ifundef{@tracklang@add@#1}{
    \% Parse language tag.
    \@tracklang@parselangtag{#1}\
    \% Track this information.
    \@tracklang@track@locale\
  }{\% Predefined tag.
    \@tracklang@nameuse{@tracklang@add@#1}\
  }\%
}
Like \TrackLanguageTag but does (fail code) if the tag doesn’t contain a valid language code. If successful, does (success code) after tracking the language.

\def\TrackIfKnownLanguage#1#2#3{\edef\@tracklang@tag{#1}\ifx\@tracklang@tag\empty #3\else \expandafter\TrackIfKnownLanguage\expandafter{\@tracklang@tag}{#2}{#3}\fi}\}

\@TrackIfKnownLanguage Argument must be expanded.
\def\@TrackIfKnownLanguage#1#2#3{\if\@tracklang@ifundef{\@tracklang@add@#1}{\%\First check if it’s predefined.\%\@tracklang@ifundef{\@tracklang@add@#1}{\%\%\Parse language tag.\%\@tracklang@parselangtag{#1}\%\Track this information if \TrackLangEnvLang is a known ISO language code.\%\TrackLangIfKnownLangFromIso{\@TrackLangEnvLang}{\@tracklang@track@locale#2}{#3}\%\%\Predefined tag.\%\@tracklang@nameuse{\@tracklang@add@#1\%#2}\%\%\}}\First split to determine language code.\%\@tracklang@split@underscoreorhyp{#1}\%\%\}}
Save the result.
\let\@TrackLangEnvLang\@tracklang@split@pre
Is there anything else?
\iffalse\@tracklang@split@post\empty
That’s it.
\else
Split again.
\expandafter\@tracklang@split@underscoreorhyp\expandafter
{\@tracklang@split@post}\
Is this an extension to the language tag?
\@tracklang@parse@extlang
Does this fit the format for a script?
\TrackLangIfScriptTag{\@tracklang@split@pre}\
\%\
Found script.
\let\@TrackLangEnvScript\@tracklang@split@pre
Split again if there’s more.
\iffalse\@tracklang@split@post\empty
\else
\expandafter\@tracklang@split@underscoreorhyp\expandafter
{\@tracklang@split@post}\
\fi
\%\
\%
Does this fit the format for a region?
\TrackLangIfRegionTag{\@tracklang@split@pre}\
\%\
Found region. Is it a 2 letter alpha or a 3 digit numeric code?
\expandafter\@tracklang@hasthirdchar\@tracklang@split@pre
\relax\relax\relax
\end@tracklang@hasthirdchar
\%
Is three digit numeric code. We need the mappings. Has \texttt{tracklang-region-codes.tex} been loaded?
\iffalse\TrackLangIfKnownNumericRegion\undefined
\@tracklang@input tracklang-region-codes.tex
\fi
\TrackLangIfKnownNumericRegion{\@tracklang@split@pre}\
\%\
\edef\@TrackLangEnvTerritory{\TrackLangNumericToAlphaIIRegion{\@tracklang@split@pre}}\
\%\
\%
Is two letter alpha code.
\let\@TrackLangEnvTerritory\@tracklang@split@pre
\@tracklang@warn{Unrecognised numeric region code
  ‘\@tracklang@split@pre’}%
%
Is two letter alpha code.
\let\@TrackLangEnvTerritory\@tracklang@split@pre
\@tracklang@warn{Unrecognised numeric region code
  ‘\@tracklang@split@pre’}%
\expandafter\@tracklang@split@underscoreorhyp\expandafter
\{\@tracklang@split@post}%
%
Parse for variant.
\@tracklang@parse@variant
Anything left can go in additional.
\let\@TrackLangEnvAdditional\@tracklang@split@post
\fi
%
\expandafter\@tracklang@split@underscoreorhyp\expandafter
\{\@tracklang@split@post}%
%
\def\@tracklang@extra@region@filelist{}%
\def\@tracklang@extra@script@filelist{}%
\@tracklang@input@extra@region@files
\@tracklang@input@extra@script@filelist

\TrackLangAddExtraRegionFile Add a file name to the list of extra region code files that should be input by
tracklang-regions-codes.tex
\def\TrackLangAddExtraRegionFile#1{%
  \ifx\TrackLangRegionMap\undefined
    \else
    \TrackLangRegionRegionCodesTexture has already been loaded so input the file now.
    \@tracklang@input #1\relax
  \fi
  \@tracklang@add{#1}{\@tracklang@extra@region@filelist}%
}%

\@tracklang@extra@region@filelist
\def\@tracklang@extra@script@filelist{}
Add the given file name to the list of files that should be automatically loaded by tracklang-scripts.tex

`\def\TrackLangAddExtraScriptFile#1{%\ifx\TrackLangScriptMap\undefined\else\@tracklang@input #1\relax\fi\@tracklang@add{#1}{\@tracklang@extra@script@filelist}\}%`
Get the root language label.
\edef\@tracklang@lang{\TrackLangGetKnownLangFromIso\@TrackLangEnvLang}\

Get the default script for this language.
\edef\@tracklang@defscript{\TrackLangGetDefaultScript\@tracklang@lang}\

Keep track of best match.
\def\@tracklang@bestmatch{0}\

Get the list of tracked dialects for this language.
\edef\@tracklang@dialects{\TrackedDialectsFromLanguage\@tracklang@lang}\

For each dialect in this list, check if it matches.
\@tracklang@for\@tracklang@dialect:=\@tracklang@dialects\do{%

Does the script match? (Initialise to no.)
\def\@tracklang@currentmatch{0}\
\edef\@tracklang@tmp{%
  \GetTrackedDialectScript{\@tracklang@dialect}%%
}\ifx\@tracklang@tmp\@TrackLangEnvScript
Script matches.
\def\@tracklang@currentmatch{1}\
\else
Script doesn’t match. If no script has been provided, does this dialect’s script match the default for this language?
\ifx\@TrackLangEnvScript\empty
  \ifx\@tracklang@tmp\@tracklang@defscript
    Default script matches.
    \def\@tracklang@currentmatch{1}\
  \fi
\fi

Does the sub-language match?
\edef\@tracklang@tmp{%
  \GetTrackedDialectSubLang{\@tracklang@dialect}%%
}\ifx\@tracklang@tmp\@TrackLangEnvSubLang
Sub-language matches.
\def\@tracklang@currentmatch{\@tracklang@currentmatch 1}\
\else
Sub-language doesn’t match.
\edef\@tracklang@currentmatch{\@tracklang@currentmatch 0}\
\fi

Does the variant match?
\edef\@tracklang@tmp{%
  \GetTrackedDialectVariant{\@tracklang@dialect}%%
}\ifx\@tracklang@tmp\@TrackLangEnvVariant

50
Variant matches.
\edef\@tracklang@currentmatch{\@tracklang@currentmatch 1}\% 
\else

Variant doesn’t match.
\edef\@tracklang@currentmatch{\@tracklang@currentmatch 0}\%
\fi

Does the region match?
\edef\@tracklang@tmp{%
  \TrackedIsoCodeFromLanguage{3166-1}{\@tracklang@dialect}%%
  \ifx\@tracklang@tmp\@TrackLangEnvTerritory

Region matches.
\edef\@tracklang@currentmatch{\@tracklang@currentmatch 1}\% 
\else

Region doesn’t match.
\edef\@tracklang@currentmatch{\@tracklang@currentmatch 0}\%
\fi

Do all four match?
\ifx\@tracklang@currentmatch\@tracklang@fullmatch

Found it.
\let#2\@tracklang@dialect
\else

Not a complete match. Is this the best match so far?
\ifnum\@tracklang@currentmatch>\@tracklang@bestmatch\relax
  \let\TrackedDialectClosestSubMatch\@tracklang@dialect
  \let\@tracklang@bestmatch\@tracklang@currentmatch
\fi
\fi
\}%
%
%
\@tracklang@fullmatch (Used to identify a full match for script, sub-language, variant and region.)
\def\@tracklang@fullmatch{1111}

\TrackLangFromEnv This command performs the following steps: query environment variable (if \TrackLangEnv not already set), parse \TrackLangEnv (if it has been set), and add the dialect (if recognised).

Note that this works slightly differently from just using \TrackLangQueryEnv followed by \TrackLangParseFromEnv and \TrackPredefinedDialect.

\def\TrackLangFromEnv{%
Initialise.
\def\TrackLangEnvLang{}
\def\TrackLangEnvTerritory{}
\def\TrackLangEnvCodeSet{}
\def\TrackLangEnvModifier{}

51
If \TrackQueryEnv is empty, assume \TrackQueryEnv has already been attempted but failed, so don’t bother retrying.

\ifx\TrackLangEnv\undefined
  \TrackLangQueryEnv
\fi
\ifx\TrackLangEnv\empty
  \@tracklang@warn{\string\TrackLangFromEnv\space non-operational as \string\TrackLangEnv\space is empty}\
\else
  At this point \TrackLangEnv shouldn’t be undefined (if \TrackLangQueryEnv fails it should define \TrackLangEnv to be empty), but check in case something unexpected has happened.

  \ifx\TrackLangEnv\undefined
    \@tracklang@warn{\string\TrackLangFromEnv\space non-operational as \string\TrackLangEnv\space hasn’t been defined}\
  \else
    Parse and track.
    \@tracklang@parse@track@locale{\TrackLangEnv}\%
    \let\TrackLangEnvLang@TrackLangEnvLang
    \let\TrackLangEnvTerritory@TrackLangEnvTerritory
    \let\TrackLangEnvCodeSet@TrackLangEnvCodeSet
    \let\TrackLangEnvModifier@TrackLangEnvModifier
  \fi
\fi
}

\TrackLocale{⟨locale⟩}

Track the dialect identified by the given locale. The argument may either be a predefined language/dialect or in the same format as \TrackLangEnv.

\def\TrackLocale#1{%
  Is the argument a recognised dialect?
  \@tracklang@ifundef{@tracklang@add@#1}{%
    \@tracklang@parse@track@locale{#1}\%
  }{%
    \@tracklang@nameuse{@tracklang@add@#1}\%
  }%
}

\tracklang@parse@track@locale Parse localisation format and track.

\def\@tracklang@parse@track@locale#1{%
  \@tracklang@parse@locale{#1}\%
  \@tracklang@parse@track@locale{#1}\%
}
Is the language code known?
\TrackLangIfKnownLangFromIso{\@TrackLangEnvLang}
{%
  \edef\@tracklang@lang{\TrackLangGetKnownLangFromIso\@TrackLangEnvLang}
  \let\@tracklang@dialect\@TrackLangEnvLang
  \ifx\@TrackLangEnvSubLang\empty
  \edef\@tracklang@dialect{\@tracklang@dialect-\@TrackLangEnvSubLang}
  \fi
  \ifx\@TrackLangEnvScript\empty
  \edef\@tracklang@dialect{\@tracklang@dialect-\@TrackLangEnvScript}
  \fi
  \ifx\@TrackLangEnvTerritory\empty
  \edef\@tracklang@dialect{\@tracklang@dialect-\@TrackLangEnvTerritory}
  \fi
  \ifx\@TrackLangEnvModifier\empty
  \edef\@tracklang@dialect{\@tracklang@dialect-\@TrackLangEnvModifier}
  \fi
  \ifx\@TrackLangEnvVariant\empty
  \edef\@tracklang@dialect{\@tracklang@dialect-\@TrackLangEnvVariant}
  \fi
\} Language code is recognised. Is the dialect label recognised?
\@tracklang@ifundef{@tracklang@add@\@tracklang@dialect}{}
{%
  Not a recognised dialect. Form new dialect name (without hyphen).
  \edef\@tracklang@dialect{%
    \@TrackLangEnvLang
    \@TrackLangEnvSubLang
    \@TrackLangEnvScript
    \@TrackLangEnvTerritory
    \@TrackLangEnvModifier
    \@TrackLangEnvVariant}
\} Add this new dialect.
\AddTrackedDialect{\@tracklang@dialect}{\@tracklang@lang}%
\AddTrackedLanguageIsoCodes{\@tracklang@lang}%
Is there a sub-language tag?
\ifx\@TrackLangEnvFirstSubLang\empty
\else

53
Dialect is recognised.

```
\csname @tracklang@add@\@tracklang@dialect\endcsname
```

Unknown language code.

```
\@tracklang@warn{Unknown language code '\@TrackLangEnvLang'}%
\edef\@tracklang@dialect{\@TrackLangEnvLang\@TrackLangEnvSubLang\@TrackLangEnvScript\@TrackLangEnvTerritory\@TrackLangEnvModifier\@TrackLangEnvVariant}%
\AddTrackedDialect{\@tracklang@dialect}{\@TrackLangEnvLang}%
```

Determine if the language code is a two or three letter code.

```
\expandafter\@tracklang@hasthirdchar
\@TrackLangEnvLang\relax\relax\relax\@end@tracklang@hasthirdchar
```

639-2 code. Track it.

```
\AddTrackedIsoLanguage{639-2}{\@TrackLangEnvLang}{\@tracklang@lang}%
```

639-1 code. Track it.

```
\AddTrackedIsoLanguage{639-1}{\@TrackLangEnvLang}{\@tracklang@lang}%
```

Add the territory if provided. (The territory may not have been defined by the dialect option.)

```
\ifx\@TrackLangEnvTerritory\empty
\else
\AddTrackedIsoLanguage{3166-1}{\@TrackLangEnvTerritory}{\@tracklang@dialect}%
\fi
```

If a modifier was provided, add that.

```
\ifx\@TrackLangEnvModifier\empty
\else
\SetTrackedDialectModifier{\@tracklang@dialect}{\@TrackLangEnvModifier}%
\fi
```
If a variant was provided, add that.
\ifx\@TrackLangEnvVariant\empty
\else
  \SetTrackedDialectVariant{\@tracklang@dialect}{\@TrackLangEnvVariant}\fi

If a script was provided, add that.
\ifx\@TrackLangEnvScript\empty
\else
  \SetTrackedDialectScript{\@tracklang@dialect}{\@TrackLangEnvScript}\fi

If a language extension was provided, add that.
\ifx\@TrackLangEnvSubLang\empty
\else
  \SetTrackedDialectSubLang{\@tracklang@dialect}{\@TrackLangEnvSubLang}\fi

If additional information was provided, add that.
\ifx\@TrackLangEnvAdditional\empty
\else
  \SetTrackedDialectAdditional{\@tracklang@dialect}{\@TrackLangEnvAdditional}\fi

4.5 Predefined Root Languages

The ISO 639-1 and 639-2 codes are used to map the root language name to the
ISO language code. The 3166-1 codes are used to map the dialect/variant to
the ISO country code. The country code is omitted if ambiguous (for example,
the language is spoken in multiple countries). Languages that have a country
code may be spoken as a minority language in another region. In this case,
\TrackLocale should be used instead to set the country code as appropriate.
Some “dialects” are just synonyms for a language name, such as “français” or
“frenchb”. These are defined in §4.6. Some of the languages have two ISO 639-2
codes designated as “B” (bibliographic) or “T” (terminology). In these cases the
terminology code is used as the primary 639-2 code. The extra “B” and “T”
codes are only provided if they are different.

\@tracklang@declareoption Provide a hook to declare a predefined setting as a package option. This is
defined by tracklang.sty before loading tracklang.tex but if this file isn’t loaded
through tracklang.sty provide a definition that ignores its argument if not al-
ready defined.
\ifx\@tracklang@declareoption\undefined
  \def\@tracklang@declareoption#1{}
\fi

\@tracklang@declareoption Provide a hook to declare a predefined setting as a package option. This is
defined by tracklang.sty before loading tracklang.tex but if this file isn’t loaded
through tracklang.sty provide a definition that ignores its argument if not al-
ready defined.

\TrackLocale Provide a hook to set the country code as appropriate. This is
defined by tracklang.sty before loading tracklang.tex but if this file isn’t loaded
through tracklang.sty provide a definition that ignores its argument if not al-
ready defined.
\ifx\@TrackLocale\undefined
  \def\@TrackLocale#1{}
\fi

Set up a language label for use with \TrackPredefinedDialect. Note that this
doesn’t check if already defined.

55
Define a new root language that’s declared as an option. The language name must be expanded before use. The default script is the ISO 15924 alpha script code. (Some languages may be written in multiple scripts. Leave empty if not obvious default.)

\def\TrackLangDeclareLanguageOption#1#2#3#4#5#6#7{% 
   \@tracklang@ifundef{@tracklang@add@#1}{%
      \TrackLangNewLanguage{#1}{#2}{#3}{#4}{#5}{#6}{#7}%
      \TrackLangProvidePredefinedLanguage{#1}%
      \@tracklang@declareoption{#1}%
   }{%
      \@tracklang@err{language option ‘#1’ has already been defined}{%}
   }%
%
\}@tracklang@add@abkhaz
\TrackLangDeclareLanguageOption{abkhaz}{ab}{abk}{}{}{}{Cyrl}
\@tracklang@add@afar
\TrackLangDeclareLanguageOption{afar}{aa}{aar}{}{}{}{Latn}
\@tracklang@add@afrikaans
\TrackLangDeclareLanguageOption{afrikaans}{af}{afr}{}{}{}{Latn}
\@tracklang@add@akan
\TrackLangDeclareLanguageOption{akan}{ak}{aka}{}{}{}{Latn}
\@tracklang@add@albanian
\TrackLangDeclareLanguageOption{albanian}{sq}{sqi}{alb}{}{}{Latn}
\@tracklang@add@amharic
\TrackLangDeclareLanguageOption{amharic}{am}{amh}{}{}{ET}{Ethi}
The default script is dependent on the region, but this is a regionless definition so using Latin as the default here as Azerbaijani alphabet is a Latin alphabet. Other countries may be using a different script, such as Cyrillic in Russia.
\@tracklang@add@basque
\TrackLangDeclareLanguageOption{basque}{eu}{baq}{\{}{}{\{}{Latn}
\@tracklang@add@belarusian
\TrackLangDeclareLanguageOption{belarusian}{be}{bel}{\{}{}{\{}{Cyrl}
\@tracklang@add@bengali
\TrackLangDeclareLanguageOption{bengali}{bn}{ben}{\{}{}{\{}{Beng}
\@tracklang@add@berber
No default. Could be Tifinagh, Latin or Arabic.
\TrackLangDeclareLanguageOption{berber}{ber}{\{}{}{\}{}{}
\@tracklang@add@bihari
No clear default.
\TrackLangDeclareLanguageOption{bihari}{bh}{bih}{\}{}{\}{}
\@tracklang@add@bislama
\TrackLangDeclareLanguageOption{bislama}{bi}{bis}{\}{}{\{}{VU}{\{}{Latn}
\@tracklang@add@bokmal
\TrackLangDeclareLanguageOption{bokmal}{nb}{nob}{\}{}{\{}{NO}{\{}{Latn}
\@tracklang@add@bosnian
\TrackLangDeclareLanguageOption{bosnian}{bs}{bos}{\}{}{\}{}{Latn}
\@tracklang@add@breton
\TrackLangDeclareLanguageOption{breton}{br}{bre}{\}{}{\}{}{FR}{\}{}{Latn}
\@tracklang@add@bulgarian
\TrackLangDeclareLanguageOption{bulgarian}{bg}{bul}{\}{}{\}{}{Cyrl}
\@tracklang@add@burmese
\TrackLangDeclareLanguageOption{burmese}{my}{mya}{bur}{\}{}{\}{}{Mymr}
\@tracklang@add@catalan
\TrackLangDeclareLanguageOption{catalan}{ca}{cat}{\}{}{\}{}{Latn}
\@tracklang@add@chamorro
\TrackLangDeclareLanguageOption{chamorro}{ch}{cha}{\}{}{\}{}{Latn}
\@tracklang@add@chechen
\TrackLangDeclareLanguageOption{chechen}{ce}{che}{\}{}{\}{}{Cyrl}
\@tracklang@add@chichewa
\TrackLangDeclareLanguageOption{chichewa}{ny}{nya}{\}{}{\}{}{Latn}
\@tracklang@add@chinese
\TrackLangDeclareLanguageOption{chinese}{zh}{zho}{chi}{\}{}{\}{}{Hans}
\@tracklang@add@hausa
\TrackLangDeclareLanguageOption{hausa}{ha}{hau}{}{}{}{Latn}
\@tracklang@add@hebrew
\TrackLangDeclareLanguageOption{hebrew}{he}{heb}{}{}{}{Hebr}
\@tracklang@add@herero
\TrackLangDeclareLanguageOption{herero}{hz}{her}{}{}{}{Latn}
\@tracklang@add@hindi
\TrackLangDeclareLanguageOption{hindi}{hi}{hin}{}{}{}{Deva}
\@tracklang@add@hirimotu
\TrackLangDeclareLanguageOption{hirimotu}{ho}{hmo}{}{}{PG}{Latn}
\@tracklang@add@icelandic
\TrackLangDeclareLanguageOption{icelandic}{is}{isl}{ice}{}{}{}{IS}{Latn}
\@tracklang@add@ido
\TrackLangDeclareLanguageOption{ido}{io}{ido}{}{}{}{Latn}
\@tracklang@add@igbo
\TrackLangDeclareLanguageOption{igbo}{ig}{ibo}{}{}{}{Latn}
\@tracklang@add@interlingua
\TrackLangDeclareLanguageOption{interlingua}{ia}{ina}{}{}{}{Latn}
\@tracklang@add@interlingue
\TrackLangDeclareLanguageOption{interlingue}{ie}{ile}{}{}{}{Latn}
\@tracklang@add@inuktitut
\TrackLangDeclareLanguageOption{inuktitut}{iu}{iku}{}{}{}{Cans}
\@tracklang@add@inupiaq
\TrackLangDeclareLanguageOption{inupiaq}{ik}{ipk}{}{}{}{Latn}
\@tracklang@add@irish
\TrackLangDeclareLanguageOption{irish}{ga}{gle}{}{}{}{Latn}
\@tracklang@add@italian
\TrackLangDeclareLanguageOption{italian}{it}{ita}{}{}{}{Latn}
\@tracklang@add@japanese
\TrackLangDeclareLanguageOption{japanese}{ja}{jpn}{}{}{}{Hani}
\@tracklang@add@javanese
\TrackLangDeclareLanguageOption{javanese}{jv}{jav}{}{}{}{Latn}
\@tracklang@add@kalaallisut
\TrackLangDeclareLanguageOption{kalaallisut}{kl}{kal}{}{Latn}

\@tracklang@add@kannada
\TrackLangDeclareLanguageOption{kannada}{kn}{kan}{}{IN}{Knda}

\@tracklang@add@kanuri
\TrackLangDeclareLanguageOption{kanuri}{kr}{kau}{}{}{Latn}

\@tracklang@add@kashmiri No default script. Could be Arabic or Devanagari.
\TrackLangDeclareLanguageOption{kashmiri}{ks}{kas}{}{IN}{}

\@tracklang@add@kazakh Default script varies according to region.
\TrackLangDeclareLanguageOption{kazakh}{kk}{kaz}{}{IN}{}

\@tracklang@add@khmer
\TrackLangDeclareLanguageOption{khmer}{km}{khm}{}{}{Khm}

\@tracklang@add@kikuyu
\TrackLangDeclareLanguageOption{kikuyu}{ki}{kik}{}{Latn}

\@tracklang@add@kinyarwanda
\TrackLangDeclareLanguageOption{kinyarwanda}{rw}{kin}{}{Latn}

\@tracklang@add@kirundi
\TrackLangDeclareLanguageOption{kirundi}{rn}{run}{}{Latn}

\@tracklang@add@komi
\TrackLangDeclareLanguageOption{komi}{kv}{kom}{RU}{Cyr}

\@tracklang@add@kongo
\TrackLangDeclareLanguageOption{kongo}{kg}{kon}{Latn}

\@tracklang@add@korean
\TrackLangDeclareLanguageOption{korean}{ko}{kor}{Hang}

\@tracklang@add@kurdish Script varies according to region.
\TrackLangDeclareLanguageOption{kurdish}{ku}{kur}{}

\@tracklang@add@kwanyama
\TrackLangDeclareLanguageOption{kwanyama}{kj}{kua}{Latn}

\@tracklang@add@kyrgyz
\TrackLangDeclareLanguageOption{kyrgyz}{ky}{kir}{Cyr}

\@tracklang@add@lao
\TrackLangDeclareLanguageOption{lao}{lo}{lao}{Lao}
\@tracklang@add@latin
\TrackLangDeclareLanguageOption{latin}{la}{lat}\{}\{}\{}{Latn}
\@tracklang@add@latvian
\TrackLangDeclareLanguageOption{latvian}{lv}{lav}\{}\{}\{}{Latn}
\@tracklang@add@limburgish
\TrackLangDeclareLanguageOption{limburgish}{li}{lim}\{}\{}\{}{Latn}
\@tracklang@add@lingala
\TrackLangDeclareLanguageOption{lingala}{ln}{lin}\{}\{}\{}{Latn}
\@tracklang@add@lithuanian
\TrackLangDeclareLanguageOption{lithuanian}{lt}{lit}\{}\{}\{}{Latn}
\@tracklang@add@lsorbian
\TrackLangDeclareLanguageOption{lsorbian}{dsb}\{}{DE}\{}{Latn}
\@tracklang@add@lubakatanga
\TrackLangDeclareLanguageOption{lubakatanga}{lu}{lub}\{}\{}{CD}\{}{Latn}
\@tracklang@add@luxembourgish
\TrackLangDeclareLanguageOption{luxembourgish}{lb}{ltz}\{}\{}\{}{Latn}
\@tracklang@add@macedonian
\TrackLangDeclareLanguageOption{macedonian}{mk}{mkd}{mac}\{}\{}{Cyril}
\@tracklang@add@magyar
\TrackLangDeclareLanguageOption{magyar}{hu}{hun}\{}\{}\{}{Latn}
\@tracklang@add@malagasy
\TrackLangDeclareLanguageOption{malagasy}{mg}{mlg}\{}\{}\{}{Latn}
\@tracklang@add@malayalam
\TrackLangDeclareLanguageOption{malayalam}{ml}{mal}\{}\{}{IN}\{}{Mlym}
\@tracklang@add@maltese
\TrackLangDeclareLanguageOption{maltese}{mt}{mlt}\{}\{}\{}{Latn}
\@tracklang@add@manx
\TrackLangDeclareLanguageOption{manx}{gv}{glv}\{}\{}{IM}\{}{Latn}
\@tracklang@add@maori
\TrackLangDeclareLanguageOption{maori}{mi}{mri}{mao}\{}{NZ}\{}{Latn}
\@tracklang@add@marathi
\TrackLangDeclareLanguageOption{marathi}{mr}{mar}\{}\{}{IN}\{}{Deva}
Undetermined language (ISO 636-2 code ‘und’) for unknown languages. There’s no ISO 639-1 language code for this language.

\TrackLangDeclareLanguageOption{undetermined}{und}{}{Latn}

\TrackLangDeclareLanguageOption{urdu}{ur}{urd}{Arab}

\TrackLangDeclareLanguageOption{usorbian}{hsb}{DE}{Latn}

\TrackLangDeclareLanguageOption{uyghur}{ug}{uig}{CN}{Arab}

\TrackLangDeclareLanguageOption{uzbek}{uz}{uzb}{Latn}

\TrackLangDeclareLanguageOption{venda}{ve}{ven}{ZA}{Latn}

\TrackLangDeclareLanguageOption{vietnamese}{vi}{vie}{Latn}

\TrackLangDeclareLanguageOption{volapuk}{vo}{vol}{Latn}

\TrackLangDeclareLanguageOption{walloon}{wa}{wln}{Latn}

\TrackLangDeclareLanguageOption{welsh}{cy}{cym}{wel}{Latn}

\TrackLangDeclareLanguageOption{westernfrisian}{fy}{fry}{NL}{Latn}

\TrackLangDeclareLanguageOption{wolof}{wo}{wol}{Latn}

\TrackLangDeclareLanguageOption{xhosa}{xh}{xho}{Latn}

\TrackLangDeclareLanguageOption{yiddish}{yi}{yid}{Hebr}

\TrackLangDeclareLanguageOption{yoruba}{yo}{yor}{Latn}
4.6 Predefined Dialects

Provide some predefined dialects.

This defines a predefined dialect label that can be used by \TrackPredefinedDialect. The arguments must be expanded before use. The final argument \langle map \rangle is the mapping from \langle dialect \rangle to the closest babel dialect label.

\def\TrackLangProvidePredefinedDialect#1#2#3#4#5#6#7{
  \ifx\relax#3\relax
    No region.
  \else
    No modifier.
    \ifx\relax#5\relax
      No variant.
      \ifx\relax#7\relax
        Make it easier for the parser to pick up the dialect label. Note that this should be the same as \TrackLangLastTrackedDialect but the parser references \@tracklang@dialect.
        \def\@tracklang@dialect{#1}
      \else
        Has variant but no modifier.
        \@tracklang@namedef{\@tracklang@add@#1}{%
          \AddTrackedDialect{#1}{#2}%
          \AddTrackedLanguageIsoCodes{#2}%
        }
      \fi
    \fi
  \else
    Has modifier.
    \@tracklang@namedef{\@tracklang@add@#1}{%
      \AddTrackedDialect{#1}{#2}%
      \AddTrackedLanguageIsoCodes{#2}%
      \SetTrackedDialectVariant{#1}{#5}%
    }
  \fi
}\relax

No region.
\relax
No modifier.
\relax
No variant.
\relax
Has variant but no modifier.
\relax
Has modifier.
\relax
No variant.
\@tracklang@namedef{@tracklang@add@#1}{% 
\AddTrackedDialect{#1}{#2}\
\AddTrackedLanguageIsoCodes{#2}\
\SetTrackedDialectModifier{#1}{#4}\
\def\@tracklang@dialect{#1}\
\}%
\else
\fi
\fi
\else

Variant and modifier.
\@tracklang@namedef{@tracklang@add@#1}{% 
\AddTrackedDialect{#1}{#2}\
\AddTrackedLanguageIsoCodes{#2}\
\SetTrackedDialectModifier{#1}{#4}\
\SetTrackedDialectVariant{#1}{#5}\
\def\@tracklang@dialect{#1}\
\}%
\fi
\fi
\else

Has a region.
\ifeq\relax#4\relax

No modifier.
\ifeq\relax#5\relax

No variant.
\@tracklang@namedef{@tracklang@add@#1}{% 
\AddTrackedDialect{#1}{#2}\
\AddTrackedLanguageIsoCodes{#2}\
\AddTrackedIsoLanguage{3166-1}{#3}{#1}\
\def\@tracklang@dialect{#1}\
\}%
\else

Variant no modifier.
\@tracklang@namedef{@tracklang@add@#1}{% 
\AddTrackedDialect{#1}{#2}\
\AddTrackedLanguageIsoCodes{#2}\
\AddTrackedIsoLanguage{3166-1}{#3}{#1}\
\SetTrackedDialectVariant{#1}{#5}\
\def\@tracklang@dialect{#1}\
\}%
\fi
\fi
\else

Has modifier.
\ifeq\relax#5\relax

No variant.
\@tracklang@namedef{@tracklang@add@#1}{%
Variants and modifiers.

```latex
\def\@tracklang@add@#1{\AddTrackedDialect{#1}{#2}\%
    \AddTrackedLanguageIsoCodes{#2}\%
    \AddTrackedIsoLanguage{3166-1}{#3}{#1}\%
    \SetTrackedDialectModifier{#1}{#4}\%
    \SetTrackedDialectVariant{#1}{#5}\%
    \def\@tracklang@dialect{#1}\%
}\fi
\fi
\fi
```

Add the mapping if provided.

```latex
\ifx\relax#6\relax
\else
\expandafter\let\expandafter\@tracklang@tmp\csname @tracklang@add@#1\endcsname
\expandafter\def\csname @tracklang@add@#1\expandafter\endcsname
\expandafter{\@tracklang@tmp\SetTrackedDialectLabelMap{#1}{#6}}\%
\fi
```

Add the script if provided.

```latex
\ifx\relax#7\relax
\else
\expandafter\let\expandafter\@tracklang@tmp\csname @tracklang@add@#1\endcsname
\expandafter\def\csname @tracklang@add@#1\expandafter\endcsname
\expandafter{\@tracklang@tmp\SetTrackedDialectScript{#1}{#7}}\%
\fi
```

The option name is the same as the dialect name. The arguments must be expanded before use. The final argument \texttt{(map)} is the mapping from \texttt{(dialect)} to the closest \texttt{babel} dialect label. May be empty if no relevant mapping.

```latex
\TrackLangDeclareDialectOption{(dialect)}{(root language)}{(3166-1 code)}{(modifier)}{(variant)}{(map)}{(script)}
```

\def\TrackLangDeclareDialectOption#1#2#3#4#5#6#7{\%

71
\@tracklang@ifundef{\@tracklang@add@#1}\
\% \TrackLangProvidePredefinedDialect{#1}{#2}{#3}{#4}{#5}{#6}{#7}\
\@tracklang@declareoption{#1}\
\%
\%  \@tracklang@err{dialect option ‘#1’ has already been defined}{%}
\%
\}
\@tracklang@add@acadian
\TrackLangDeclareDialectOption{acadian}{french}{}{}{}{}{}
\@tracklang@add@american
\TrackLangDeclareDialectOption{american}{english}{US}{}{}{}{}
\@tracklang@add@australian
\TrackLangDeclareDialectOption{australian}{english}{AU}{}{}{}{}
\@tracklang@add@austrian
\TrackLangDeclareDialectOption{austrian}{german}{AT}{}{}{}{}
\@tracklang@add@naustrian
\TrackLangDeclareDialectOption{naustrian}{german}{AT}{new}{1996}{}{}
\@tracklang@add@bahasa
\TrackLangDeclareDialectOption{bahasa}{bahasai}{IN}{}{}{}{}
\@tracklang@add@brazil
\TrackLangDeclareDialectOption{brazil}{portuges}{BR}{}{}{}{}
\@tracklang@add@brazilian
\TrackLangDeclareDialectOption{brazilian}{portuges}{BR}{}{}{}{}
\@tracklang@add@british
\TrackLangDeclareDialectOption{british}{english}{GB}{}{}{}{}
\@tracklang@add@canadian
\TrackLangDeclareDialectOption{canadian}{english}{CA}{}{}{}{}
\@tracklang@add@canadien
\TrackLangDeclareDialectOption{canadien}{french}{CA}{}{}{}{}
\@tracklang@add@croatia
\TrackLangDeclareDialectOption{croatia}{croatian}{HR}{}{}{}{}
\@tracklang@add@istriacountycroatian
\TrackLangDeclareDialectOption{istriacountycroatian}{croatian}{HR}{}{}{}{}

72
Allow it to hook to pick up `captionsbritish` as well as `captionsenglish` since the date format closely matches `british`.

```
\TrackLangDeclareDialectOption{jerseyenglish}{english}{JE}{}{}{british}{}
```

Allow it to hook to pick up `captionsbritish` as well as `captionsenglish` since the date format closely matches `british`.

```
\TrackLangDeclareDialectOption{guernseyenglish}{english}{GG}{}{}{british}{}
```

```
\TrackLangDeclareDialectOption{jerseyfrench}{french}{JE}{}{}{}{}
```

```
\TrackLangDeclareDialectOption{guernseyfrench}{french}{GG}{}{}{}{}
```

```
\TrackLangDeclareDialectOption{latein}{latin}{}{}{}{}{}
```

```
\TrackLangDeclareDialectOption{lowersorbian}{lsorbian}{DE}{}{}{}{}
```

```
\TrackLangDeclareDialectOption{malay}{bahasam}{MY}{}{}{}{}
```

```
\TrackLangDeclareDialectOption{meyalu}{bahasam}{MY}{}{}{}{}
```

```
\TrackLangDeclareDialectOption{maltamaltese}{maltese}{MT}{}{}{}{}
```

```
\TrackLangDeclareDialectOption{maltaenglish}{english}{MT}{}{}{british}{}
```

```
\TrackLangDeclareDialectOption{newzealand}{english}{NZ}{}{}{}{}
```

```
\TrackLangDeclareDialectOption{isleofmanenglish}{english}{IM}{}{}{}{}
```

```
\TrackLangDeclareDialectOption{norwegian}{norsk}{NO}{}{}{}{}
```

```
\TrackLangDeclareDialectOption{piemonteis}{piedmontese}{IT}{}{}{}{}
```

```
\TrackLangDeclareDialectOption{polutonikogreek}{greek}{}{polyton}{}{}{}{}
```
TrackLangDeclareDialectOption{bolivianspanish}{spanish}{BO}{}
TrackLangDeclareDialectOption{chilianspanish}{spanish}{CL}{}
TrackLangDeclareDialectOption{columbianspanish}{spanish}{CO}{}
TrackLangDeclareDialectOption{costaricanspanish}{spanish}{CR}{}
TrackLangDeclareDialectOption{cubanspanish}{spanish}{CU}{}
TrackLangDeclareDialectOption{dominicanspanish}{spanish}{DO}{}
TrackLangDeclareDialectOption{ecudorianspanish}{spanish}{EC}{}
TrackLangDeclareDialectOption{elsalvadorspanish}{spanish}{SV}{}
TrackLangDeclareDialectOption{guatemalanspanish}{spanish}{GT}{}
TrackLangDeclareDialectOption{honduranspanish}{spanish}{HN}{}
TrackLangDeclareDialectOption{mexicanspanish}{spanish}{MX}{}
TrackLangDeclareDialectOption{nicaraguanspanish}{spanish}{NI}{}
TrackLangDeclareDialectOption{panamaspanish}{spanish}{PA}{}
TrackLangDeclareDialectOption{paraguayspanish}{spanish}{PY}{}
TrackLangDeclareDialectOption{peruvianspanish}{spanish}{PE}{}
TrackLangDeclareDialectOption{puertoricospanish}{spanish}{PR}{}
Add babel’s nil as a dialect of undetermined language.
4.7 Dialect Option Synonyms

Add some dialect synonyms:

\LetTrackLangSynonym
\def\LetTrackLangSynonym#1#2{%
 \expandafter\let\csname @tracklang@add@#1\expandafter\endcsname
 \csname @tracklang@add@#2\endcsname
}

\LetTrackLangOption
\def\LetTrackLangOption#1#2{%
 \LetTrackLangSynonym{#1}{#2}%
 \@tracklang@declareoption{#1}%
}

\@tracklang@add@en-US
\LetTrackLangOption{en-US}{american}

\@tracklang@add@en-GB
\LetTrackLangOption{en-GB}{british}

\@tracklang@add@en-AU
\LetTrackLangOption{en-AU}{australian}

\@tracklang@add@en-NZ
\LetTrackLangOption{en-NZ}{newzealand}

\@tracklang@add@en-CA
\LetTrackLangOption{en-CA}{canadian}

\@tracklang@add@fr-CA
\LetTrackLangOption{fr-CA}{canadien}

\@tracklang@add@fr-BE
\LetTrackLangOption{fr-BE}{belgique}

\@tracklang@add@pt-BR
\LetTrackLangOption{pt-BR}{brazilian}

\@tracklang@add@it-HR
\LetTrackLangOption{it-HR}{istriacountyitalian}

\@tracklang@add@nl-BE
\LetTrackLangOption{nl-BE}{flemish}

\@tracklang@add@fr-FR
\LetTrackLangOption{fr-FR}{france}
\@tracklang@add@de-DE
\LetTrackLangOption{de-DE}{germanDE}
\@tracklang@add@de-BE
\LetTrackLangOption{de-BE}{belgiangerman}
\@tracklang@add@en-GG
\LetTrackLangOption{en-GG}{guernseyenglish}
\@tracklang@add@fr-GG
\LetTrackLangOption{fr-GG}{guernseyfrench}
\@tracklang@add@it-IT
\LetTrackLangOption{it-IT}{italy}
\@tracklang@add@mt-MT
\LetTrackLangOption{mt-MT}{maltamaltese}
\@tracklang@add@en-MT
\LetTrackLangOption{en-MT}{maltaenglish}
\@tracklang@add@en-IM
\LetTrackLangOption{en-IM}{isleofmanenglish}
\@tracklang@add@en-JE
\LetTrackLangOption{en-JE}{jerseyenglish}
\@tracklang@add@fr-JE
\LetTrackLangOption{fr-JE}{jerseyfrench}
\@tracklang@add@nl-NL
\LetTrackLangOption{nl-NL}{netherlands}
\@tracklang@add@pt-PT
\LetTrackLangOption{pt-PT}{portugal}
\@tracklang@add@it-SM
\LetTrackLangOption{it-SM}{sanmarino}
\@tracklang@add@sl-SI
\LetTrackLangOption{sl-SI}{slovenia}
\@tracklang@add@it-SI
\LetTrackLangOption{it-SI}{sloveneistriaitalian}
\@tracklang@add@es-ES
\LetTrackLangOption{es-ES}{spainspanish}
4.8 Conditionals and Loops

\texttt{\textbackslash IfTrackedLanguage\{language\}\{true part\}\{false part\}}

\texttt{\long\def\IfTrackedLanguage#1#2#3{%}
First find out if the language name is empty.
\texttt{\edef\@tracklang@element{#1}%}
\texttt{\ifx\@tracklang@element\empty}
Language is empty, so do false part.
\texttt{\else}
\texttt{\expandafter\@tracklang@ifinlist\expandafter{\@tracklang@element}\@tracklang@languages}
\texttt{%}
In list, so do true part.
\texttt{\else}
\texttt{\fi}
\texttt{}}%
\texttt{}}%
\texttt{}}%
Not in list, so do false part.
\texttt{}}%
\texttt{}}%
\texttt{}}%
\texttt{}}%
\texttt{}}

\texttt{\textbackslash IfTrackedDialect\{dialect\}\{true part\}\{false part\}}
\long\def\IfTrackedDialect#1#2#3{%
  \@tracklang@ifundef{\@tracklang@fromdialect@#1}{#3}{#2}%
}\IfTrackedDialect
\long\def\IfTrackedIsoCode#1#2#3#4{%
  \@tracklang@ifundef{\@tracklang@isotolang@#2}{#4}{#3}%
}\IfTrackedIsoCode
\long\def\IfTrackedLanguageHasIsoCode#1#2#3#4{%
  \@tracklang@ifundef{\@tracklang@isofromlang@#2}{#4}{#3}%
}\IfTrackedLanguageHasIsoCode
\ForEachTrackedLanguage\ForEachTrackedLanguage{
  \langle cs\rangle \langle body\rangle
}\ForEachTrackedLanguage
\ForEachTrackedDialect\ForEachTrackedDialect{
  \langle cs\rangle \langle body\rangle
}\ForEachTrackedDialect
\AnyTrackedLanguages\AnyTrackedLanguages{\%
  \ifx\@tracklang@languages\empty
    #2%
  \else
    #1%
  \fi
}\AnyTrackedLanguages
\IfTrackedLanguageFileExists{\langle\textit{dialect}\rangle\langle\textit{prefix}\rangle\langle\textit{suffix}\rangle}{\langle\textit{true part}\rangle}{\langle\textit{false part}\rangle}

Determines if the file \langle\textit{prefix}\rangle\langle\textit{tag}\rangle\langle\textit{suffix}\rangle exists, where \langle\textit{tag}\rangle is an ISO code or ISO codes or label identifying the language. If \langle\textit{dialect}\rangle hasn’t been identified as a tracked dialect, this just does \langle\textit{false part}\rangle, otherwise this tries various different combinations for \langle\textit{tag}\rangle. If the file \langle\textit{prefix}\rangle\langle\textit{tag}\rangle\langle\textit{suffix}\rangle exists, \CurrentTrackedTag is set to \langle\textit{tag}\rangle and \langle\textit{true part}\rangle is performed, otherwise \langle\textit{false part}\rangle is performed. The order of priority has changed with version 1.4 and now includes a check for variations that include the script. If the original ordering is specifically required, the old behaviour can be obtained with \IfTrackedLanguageFileExists@orig.

\begin{verbatim}
\long\def\IfTrackedLanguageFileExists#1#2#3#4#5{\%
    @tracklang@verbose{Finding file for dialect '#1'}\%
    Initialise.
    \def\CurrentTrackedTag{}\%
    Select this dialect.
    \SetCurrentTrackedDialect{#1}\%
    \IfTrackedDialect{#1}{\%
        Save the three types of language codes to for reference.
        \edef\CurrentTrackedIsoCodeI{\TrackedIsoCodeFromLanguage {639-1}{\CurrentTrackedLanguage}}\%
        \edef\CurrentTrackedIsoCodeII{\TrackedIsoCodeFromLanguage {639-2}{\CurrentTrackedLanguage}}\%
        \edef\CurrentTrackedIsoCodeIII{\TrackedIsoCodeFromLanguage {639-3}{\CurrentTrackedLanguage}}\%
        Earlier versions of this had too many nested conditionals, which was confusing and hard to follow. This is a rewrite that removes the nesting and adds extra checks for dialect scripts.
        @tracklang@test@if@file@exists@for@tag{#1}{#2}{#3}{#4}{#5}{%\%
            @tracklang@IfFileExists{#2\CurrentTrackedTag#3}{#5}%
        }%\%
        
        #5% unknown dialect
    }%\%
\}
\end{verbatim}

\texttt{@tracklang@test@if\texttt{file@exists@for\texttt{tag}}}
Test \( (639-1)-(sublang)-(script)-(region) \) combination.
No script or region set. If there’s a sub-language code then try again without it.

No region set. If there’s a sub-language code then try again without it.
Region is set but no script. If there's a sub-language code then try again without it.

\fi
else
\fi
\fi
\else
\ifx\CurrentTrackedDialectScript\empty
\ Region and script set. If there's a sub-language code then try again without it.
\fi
\else
\edef\CurrentTrackedTag{\CurrentTrackedIsoCode-\CurrentTrackedRegion}
\@tracklang@verbose{Testing tag '\CurrentTrackedTag'}
\@tracklang@IfFileExists{#2\CurrentTrackedTag#3}{%
\let\@tracklang@if@file@exists\@tracklang@firstoftwo
\let\@tracklang@test@if@file@exists@next\@tracklang@test@if@file@exists@noop
\%}
\else
\@tracklang@test@if@file@exists@for@langIIscriptregion
\fi
\else
\edef\CurrentTrackedTag{\CurrentTrackedIsoCode-\CurrentTrackedDialectScript-\CurrentTrackedRegion}
\@tracklang@verbose{Testing tag '\CurrentTrackedTag'}
\@tracklang@IfFileExists{#2\CurrentTrackedTag#3}{%
\let\@tracklang@if@file@exists\@tracklang@firstoftwo
\let\@tracklang@test@if@file@exists@next\@tracklang@test@if@file@exists@noop
\%}
\else
\fi
\fi
\@tracklang@test@if@file@exists@for@langIIscriptregion
\fi
\def\@tracklang@test@if@file@exists@for@langIIscriptregion#1#2#3{%
\let\CurrentTrackedIsoCode\CurrentTrackedIsoCodeII
\let\CurrentTrackedDialectScript\empty
\test\langle 639-2 \rangle -\langle sublang \rangle -\langle script \rangle -\langle region \rangle combination.
\def\@tracklang@test@if@file@exists@for@langIIscriptregion#1#2#3{
\let\CurrentTrackedIsoCode\CurrentTrackedIsoCodeII
\let\CurrentTrackedDialectScript\empty
\test\langle 639-2 \rangle -\langle sublang \rangle -\langle script \rangle -\langle region \rangle combination.
\"}
No script or region set. If there's a sub-language code then try again without it.

No region set. If there's a sub-language code then try again without it.
Region is set but no script. If there's a sub-language code then try again without it.

Both region and script are set. If there's a sub-language code then try again without it.
Test \( (639-3)-(\text{sublang})-(\text{script})-(\text{region}) \) combination.

\begin{verbatim}
def\@tracklang@test@if\if@file@exists@for@langIIIscriptregion#1#2#3{% 
  \let\CurrentTrackedIsoCode\CurrentTrackedIsoCodeIII
  \let\CurrentTrackedTag\CurrentTrackedIsoCode
  \ifx\CurrentTrackedDialectSubLang\empty
    \edef\CurrentTrackedTag{\CurrentTrackedTag-\CurrentTrackedDialectSubLang}%
  \fi
  \ifx\CurrentTrackedDialectScript\empty
    \edef\CurrentTrackedTag{\CurrentTrackedTag-\CurrentTrackedDialectScript}%
  \fi
  \ifx\CurrentTrackedRegion\empty
    \edef\CurrentTrackedTag{\CurrentTrackedTag-\CurrentTrackedRegion}%
  \fi
  \@tracklang@verbose{Testing tag '\CurrentTrackedTag'}%
  \@tracklang@IfFileExists{#2\CurrentTrackedTag#3}{%
    \let\@tracklang@if\if@file@exists\@tracklang@firstoftwo
    \let\@tracklang@test@if\if@file@exists\@tracklang@test@if@file@existsnoop
  }%
}
\end{verbatim}

No script or region set. If there's a sub-language code then try again without it.

\begin{verbatim}
\let\@tracklang@test@if\if@file@exists@for@langvariant
\else
  \let\CurrentTrackedTag\CurrentTrackedIsoCode
  \@tracklang@verbose{Testing tag '\CurrentTrackedTag'}%
  \@tracklang@IfFileExists{#2\CurrentTrackedTag#3}{%
    \let\@tracklang@if\if@file@exists\@tracklang@firstoftwo
    \let\@tracklang@test@if\if@file@exists\@tracklang@test@if@file@existsnoop
  }%
  \let\CurrentTrackedTag\CurrentTrackedIsoCode
  \@tracklang@verbose{Testing tag '\CurrentTrackedTag'}%
\fi
\end{verbatim}

No region set. If there's a sub-language code then try again without it.

\begin{verbatim}
\let\@tracklang@test@if\if@file@exists@for@langIIIskipscript
\else
  \edef\CurrentTrackedTag{\CurrentTrackedIsoCode-\CurrentTrackedDialectScript}%
  \@tracklang@verbose{Testing tag '\CurrentTrackedTag'}%
\fi
\end{verbatim}

91
Region is set but no script. If there's a sub-language code then try again without it.

Both region and script are set. If there's a sub-language code then try again without it.
No region. Has a script but already tested with 639-1 and script. Just test language code alone.

```latex
\def\@tracklang@test@if@file@exists@for@langIskipscript#1#2#3{%
\let\CurrentTrackedTag\CurrentTrackedIsoCode
\ifx\CurrentTrackedDialectSubLang\empty
\else
  \edef\CurrentTrackedTag{\CurrentTrackedTag-\CurrentTrackedDialectSubLang}%
\fi
\@tracklang@verbose{Testing tag '{\CurrentTrackedTag}'%}
\@tracklang@IfFileExists{#2\CurrentTrackedTag#3}{%
  \let\@tracklang@if@file@exists\@tracklang@firstoftwo
  \let\@tracklang@test@if@file@exists@next\@tracklang@test@if@file@exists@noop
}%
{%
  \ifx\CurrentTrackedDialectSubLang\empty
    \IfTrackedLanguageHasIsoCode
    \{639-2\}\{\CurrentTrackedLanguage\}
    {%
      \let\@tracklang@test@if@file@exists@for@langIskipscript\CurrentTrackedIsoCode
    }%
  \else
    Try without sub-language code.
    \let\CurrentTrackedTag\CurrentTrackedIsoCode
    \@tracklang@verbose{Testing tag '{\CurrentTrackedTag}'%}
    \@tracklang@IfFileExists{\#2\CurrentTrackedTag#3}{%
  }%
  \fi
  \IfTrackedLanguageHasIsoCode
  \{639-3\}\{\CurrentTrackedLanguage\}
  {%
    \let\@tracklang@test@if@file@exists@for@langIskipscript\CurrentTrackedIsoCode
  }%
  {%
    \let\@tracklang@test@if@file@exists@for@langvariant\CurrentTrackedIsoCode
  }%
}{%
  \IfTrackedLanguageHasIsoCode
  \{639-2\}\{\CurrentTrackedLanguage\}
  {%
    \let\@tracklang@test@if@file@exists@for@langIskipscript\CurrentTrackedIsoCode
  }%
}}
```

93
No region but has a script.

\edef\@tracklang@test0if0file@exists0next{#1}{#2}{#3}%
\@tracklang@verbose{Testing tag '\CurrentTrackedTag'}% 
\@tracklang@IfFileExists{#2\CurrentTrackedTag#3}{}{
\let\@tracklang@if@file@exists\@tracklang@firstoftwo
\let\@tracklang@test@if@file@exists@next\@tracklang@test@if@file@exists@noop

Try without sub-language.

\edef\@tracklang@test0if0file@exists0next{#1}{#2}{#3}%
\@tracklang@verbose{Testing tag '\CurrentTrackedTag'}% 
\@tracklang@IfFileExists{#2\CurrentTrackedTag#3}{}{
\let\@tracklang@if@file@exists\@tracklang@firstoftwo
\let\@tracklang@test@if@file@exists@next\@tracklang@test@if@file@exists@noop

94
\@tracklang@test@if@file@exists@for@langIIIskipscript No region. Has a script but already checked.
\def\@tracklang@test@if@file@exists@for@langIIIskipscript#1#2#3{\%
 \let\CurrentTrackedTag\CurrentTrackedIsoCode
 \ifx\CurrentTrackedDialectSubLang\empty
 \else
 \edef\CurrentTrackedTag{\CurrentTrackedTag-\CurrentTrackedDialectSubLang}\%
 \fi
 \@tracklang@verbose{Testing tag ‘\CurrentTrackedTag’}\%
 \@tracklang@IfFileExists{#2\CurrentTrackedTag#3}{\%
 \let\@tracklang@if@file@exists\@tracklang@firstoftwo
 \let\@tracklang@test@if@file@exists@next\@tracklang@test@if@file@exists@noop
 }\%
 \ifx\CurrentTrackedDialectSubLang\empty
 \IfTrackedLanguageHasIsoCode {639-3}{\CurrentTrackedLanguage}\%
 \let\@tracklang@test@if@file@exists@next\@tracklang@test@if@file@exists@for@langIIIscript
 \%
 \let\@tracklang@test@if@file@exists@next\@tracklang@test@if@file@exists@for@langvariant
 \%
 \else
 Try without sub-language.
 \let\CurrentTrackedTag\CurrentTrackedIsoCode
 \@tracklang@verbose{Testing tag ‘\CurrentTrackedTag’}\%
 \@tracklang@IfFileExists{#2\CurrentTrackedTag#3}{\%
 \let\@tracklang@if@file@exists\@tracklang@firstoftwo
 \let\@tracklang@test@if@file@exists@next\@tracklang@test@if@file@exists@noop
 }\%
 \IfTrackedLanguageHasIsoCode {639-3}{\CurrentTrackedLanguage}\%
 \let\@tracklang@test@if@file@exists@next\@tracklang@test@if@file@exists@for@langIIIscript
 \%
 \let\@tracklang@test@if@file@exists@next\@tracklang@test@if@file@exists@for@langvariant
 \%
 \fi
 \@tracklang@test@if@file@exists@for@langIIIskipscript%
 \}
file@exists@for@langIIIscript No region but has a script.
\def\@tracklang@test@if@file@exists@for@langIIIscript#1#2#3{%
  \let\CurrentTrackedIsoCode\CurrentTrackedIsoCodeIII
  \let\CurrentTrackedTag\CurrentTrackedIsoCode
  \ifx\CurrentTrackedDialectSubLang\empty
  \else
    \edef\CurrentTrackedTag{\CurrentTrackedTag-\CurrentTrackedDialectSubLang}\%
    \fi
  \edef\CurrentTrackedTag{\CurrentTrackedTag-\CurrentTrackedDialectScript}\%
  \@tracklang@verbose{Testing tag ‘\CurrentTrackedTag’}\%
  \@tracklang@IfFileExists{#2\CurrentTrackedTag#3}{%
    %
    \let@tracklang@if@file@exists\@tracklang@firstoftwo
    \let@tracklang@test@if@file@exists\@tracklang@test@if@file@exists@noop
  }%
  {%
    \ifx\CurrentTrackedDialectSubLang\empty
      \let@tracklang@test@if@file@exists\@tracklang@test@if@file@exists@for@langIIIskipscript
    \else
      Try without sub-language.
      \edef\CurrentTrackedTag{\CurrentTrackedIsoCode-\CurrentTrackedDialectScript}\%
      \@tracklang@verbose{Testing tag ‘\CurrentTrackedTag’}\%
      \@tracklang@IfFileExists{#2\CurrentTrackedTag#3}{%
        %
        \let@tracklang@if@file@exists\@tracklang@firstoftwo
        \let@tracklang@test@if@file@exists\@tracklang@test@if@file@exists@noop
      }%
      {%
        \let@tracklang@test@if@file@exists@next\@tracklang@test@if@file@exists@for@langIIIskipscript
      }%
      \fi
  }%
  \@tracklang@test@if@file@exists@next{#1}{#2}{#3}%
}\@tracklang@test@if@file@exists@for@langIIIskipscript No region. Has a script but already checked.
\def\@tracklang@test@if@file@exists@for@langIIIskipscript#1#2#3{%
  \let\CurrentTrackedIsoCode\CurrentTrackedIsoCode
  \ifx\CurrentTrackedDialectSubLang\empty
  \else
    \edef\CurrentTrackedTag{\CurrentTrackedTag-\CurrentTrackedDialectSubLang}\%
    \fi
  \edef\CurrentTrackedTag{\CurrentTrackedTag-\CurrentTrackedDialectScript}\%
  \@tracklang@verbose{Testing tag ‘\CurrentTrackedTag’}\%
  \@tracklang@IfFileExists{#2\CurrentTrackedTag#3}{%
    %
    \let@tracklang@if@file@exists\@tracklang@firstoftwo
    \let@tracklang@test@if@file@exists\@tracklang@test@if@file@exists@noop
  }%
  {%
    \let@tracklang@test@if@file@exists@next\@tracklang@test@if@file@exists@for@langIIIskipscript
  }%
}\@tracklang@test@if@file@exists@for@langIIIskipscript No region. Has a script but already checked.
\def\@tracklang@test@if@file@exists@for@langIIIskipscript#1#2#3{%
  \let\CurrentTrackedIsoCode\CurrentTrackedIsoCode
  \ifx\CurrentTrackedDialectSubLang\empty
  \else
    \edef\CurrentTrackedTag{\CurrentTrackedTag-\CurrentTrackedDialectSubLang}\%
    \fi
  \edef\CurrentTrackedTag{\CurrentTrackedTag-\CurrentTrackedDialectScript}\%
  \@tracklang@verbose{Testing tag ‘\CurrentTrackedTag’}\%
  \@tracklang@IfFileExists{#2\CurrentTrackedTag#3}{%
    %
    \let@tracklang@if@file@exists\@tracklang@firstoftwo
    \let@tracklang@test@if@file@exists\@tracklang@test@if@file@exists@noop
  }%
}
Try without sub-language code.

\let\CurrentTrackedIsoCode\CurrentTrackedLanguage
\@tracklang@verbose{Testing tag '\CurrentTrackedTag'}%
\@tracklang@ifFileExists{#2\CurrentTrackedTag#3}
{%
 \let\@tracklang@if@file@exists\@tracklang@firstoftwo
 \let\@tracklang@test@if@file@exists@next\@tracklang@test@if@file@exists@noop
%}
{%
 \IfTrackedLanguageHasIsoCode
 {639-2}{\CurrentTrackedLanguage}
 {%
  \let\@tracklang@test@if@file@exists@next\@tracklang@test@if@file@exists@for@langIIregion
%}
{%
 \IfTrackedLanguageHasIsoCode
 {639-3}{\CurrentTrackedLanguage}
 {%
  \let\@tracklang@test@if@file@exists@next\@tracklang@test@if@file@exists@for@langIIregion
%}
{%
  \let\@tracklang@test@if@file@exists@next\@tracklang@test@if@file@exists@for@langvariant
%}
%
\fi
%\@tracklang@test@if@file@exists@next{#1}{#2}{#3}%

\IfFileExists{\CurrentTrackedback#1#2#3}{%
 \let\CurrentTrackedback\CurrentTrackedback#1#2#3
 \let\CurrentTrackedback\CurrentTrackedbackII
 %\@tracklang@verbose{Testing tag '\CurrentTrackedback'}%
 %\@tracklang@ifFileExists{\CurrentTrackedback\CurrentTrackedback#3}
 {%
  \let\@tracklang@if@file@exists\@tracklang@firstoftwo
  \let\@tracklang@test@if@file@exists@next\@tracklang@test@if@file@exists@noop
%}
%}
%\@tracklang@test@if@file@exists@next{#1}{#2}{#3}%

file@exists@for@langIIregion No script but has a region.

\edef\@tracklang@test@if@file@exists@for@langIIregion#1#2#3{%
 %\@tracklang@verbose{Testing tag '\CurrentTrackedback'}%
 %\@tracklang@ifFileExists{#2\CurrentTrackedback#3}
 {%
  \let\@tracklang@if@file@exists\@tracklang@firstoftwo
  \let\@tracklang@test@if@file@exists@next\@tracklang@test@if@file@exists@noop
%}
%\@tracklang@test@if@file@exists@next{#1}{#2}{#3}%

\ife\CurrentTrackedback\CurrentTrackedback\CurrentTrackedback\empty
 else
  \edef\CurrentTrackedback{\CurrentTrackedback\CurrentTrackedback\empty}
\fi
\edef\CurrentTrackedback{\CurrentTrackedback\CurrentTrackedback\empty}
\@tracklang@verbose{Testing tag '\CurrentTrackedback'}%
\@tracklang@ifFileExists{#2\CurrentTrackedback#3}
{%
 \let\@tracklang@if@file@exists\@tracklang@firstoftwo
 \let\@tracklang@test@if@file@exists@next\@tracklang@test@if@file@exists@noop
%}
%\@tracklang@test@if@file@exists@next{#1}{#2}{#3}%

98
Try without sub-language.

\texttt{\def\CurrentTrackedTag{\CurrentTrackedIsoCode-\CurrentTrackedDialectScript}\
  \@tracklang@verbose{Testing tag \texttt{\CurrentTrackedTag}}\
  \@tracklang@IfFileExists{#2\CurrentTrackedTag#3}\
  {\let\@tracklang@if@file@exists\@tracklang@firstoftwo\
    \let\@tracklang@test@if@file@exists@next\@tracklang@test@if@file@exists@noop}\
  {\let\@tracklang@test@if@file@exists@next\@tracklang@test@if@file@exists@for@langIIskipregion}}

No script. Has a region but already tested with 639-2 and region. Just test language code alone.

\texttt{\def\CurrentTrackedTag{\CurrentTrackedTag-\CurrentTrackedDialectSubLang}\
  \@tracklang@verbose{Testing tag \texttt{\CurrentTrackedTag}}\
  \@tracklang@IfFileExists{#2\CurrentTrackedTag#3}\
  {\let\@tracklang@if@file@exists\@tracklang@firstoftwo\
    \let\@tracklang@test@if@file@exists@next\@tracklang@test@if@file@exists@for@langIIskipregion}}

Try without sub-language code.

\texttt{\let\CurrentTrackedTag\CurrentTrackedIsoCode\
  \@tracklang@verbose{Testing tag \texttt{\CurrentTrackedTag}}\
  \@tracklang@IfFileExists{#2\CurrentTrackedTag#3}\
  {99}
No script but has a region.

Try without sub-language.
Try without sub-language.

\edef\CurrentTrackedTag{\CurrentTrackedIsoCode-\CurrentTrackedDialectScript}\n\@tracklang@verbose{\texttt{Testing tag \texttt{\CurrentTrackedTag}}}\n\@tracklang@IfFileExists{#2\CurrentTrackedTag#3}{% \let\@tracklang@if@file@exists\@tracklang@firstoftwo \let\@tracklang@test@if@file@exists@next\@tracklang@test@if@file@exists@for@langIIskipregion \fi \@tracklang@test@if@file@exists@next{#1}{#2}{#3} %

\texttt{for@langIIskipscriptregion} Already tested script and region. Test without script if the script is the default then without region.
\def\@tracklang@test@if@file@exists@for@langIIskipscriptregion#1#2#3{% 
Is the script the default for this dialect?
\@tracklang@ifcseqcsname\CurrentTrackedDialectScript 
{\@tracklang@knowniso@script\empty\CurrentTrackedLanguage} %
{\}
Try just language code and region.
\let\CurrentTrackedTag\CurrentTrackedIsoCode 
\ifdef\CurrentTrackedTag\CurrentTrackedDialectSubLang\empty \else \edef\CurrentTrackedTag{\CurrentTrackedTag-\CurrentTrackedDialectSubLang}\fi \ifdef\CurrentTrackedTag\CurrentTrackedRegion\CurrentTrackedTag\CurrentTrackedDialectSubLang\empty \else \edef\CurrentTrackedTag{\CurrentTrackedTag-\CurrentTrackedRegion}\fi \@tracklang@verbose{\texttt{Testing tag \texttt{\CurrentTrackedTag}}}\n\@tracklang@IfFileExists{#2\CurrentTrackedTag#3}{% \let\@tracklang@if@file@exists\@tracklang@firstoftwo \let\@tracklang@test@if@file@exists@next\@tracklang@test@if@file@exists@for@langIIskipregion \fi \@tracklang@test@if@file@exists@next{#1}{#2}{#3} %

\texttt{for@langIIskipscriptregion} Try just language code and script.
Try without sub-language.
\edef\CurrentTrackedTag{\CurrentTrackedIsoCode-\CurrentTrackedRegion}\
@tracklang@verbose{Testing tag ‘\CurrentTrackedTag’}\%
@tracklang@ifdef{\CurrentTrackedTag}{\CurrentTrackedTag}\%
\let@tracklang@if@file@exists@next@tracklang@firstoftwo
\let@tracklang@test@if@file@exists@next@tracklang@noop
\}
\%
\let@tracklang@ifdef@next@tracklang@firstoftwo
\let@tracklang@test@if@file@exists@next@tracklang@noop
\fi
\}
%
%
Script isn’t the default.
\let\tracklang@ifdef\tracklang@firstoftwo
\let\tracklang@test@if@file@exists\tracklang@noop
Try just language code.
\let@tracklang@if@file@exists@next\tracklang@firstoftwo
\let@tracklang@test@if@file@exists\tracklang@noop
Try without sub-language.
\edef\CurrentTrackedTag{\CurrentTrackedIsoCode-\CurrentTrackedDialectScript}\
@tracklang@verbose{Testing tag ‘\CurrentTrackedTag’}\%
@tracklang@ifdef{\CurrentTrackedTag}{\CurrentTrackedTag}\%
Try other language codes with script and region.
\IfLanguageHasIsoCode \{639-2\}{\CurrentLanguage}
{%
\let\CurrentTag\test\if\exists\for\lang\is\skip\script\skip\region\#1\#2\#3{%%
\let\CurrentIsoCode\CurrentLanguage
\let\CurrentTag\test\if\exists\for\lang\is\skip\script\skip\region
\let\CurrentIsoCode\CurrentLanguage
\let\CurrentTag\test\if\exists\for\lang\is\script\skip\region
\let\CurrentIsoCode\CurrentLanguage
\let\CurrentTag\test\if\exists\for\lang\is\script\region
\let\CurrentIsoCode\CurrentLanguage
\let\CurrentTag\test\if\exists\for\lang\is\region
\let\CurrentIsoCode\CurrentLanguage
\else
Try just language code.
\let\CurrentTag\CurrentIsoCode
\verb\test\exists\for\lang\is\region
\verb\testing\tag\CurrentTag
Try other language codes with script and region.
\IfTrackedLanguageHasIsoCode
{639-2}{\CurrentTrackedLanguage}
{%
\let\@tracklang@test@if@file@exists@next\@tracklang@test@if@file@exists@for@langIIscriptregion
{%
\IfTrackedLanguageHasIsoCode
{639-3}{\CurrentTrackedLanguage}
{%
\let\CurrentTrackedIsoCode\CurrentTrackedIsoCodeIII
\let\@tracklang@test@if@file@exists@next\@tracklang@test@if@file@exists@for@langIIiscriptregion
{%
\IfTrackedLanguageHasIsoCode
{639-3}{\CurrentTrackedLanguage}
{%
\let\CurrentTrackedIsoCode\CurrentTrackedIsoCodeIII
\let\@tracklang@test@if@file@exists@next\@tracklang@test@if@file@exists@for@langIIiscriptregion
{%
\let\CurrentTrackedTag\CurrentTrackedIsoCode
\ifx\CurrentTrackedDialectSubLang\empty
\else
\edef\CurrentTrackedTag{\CurrentTrackedTag-\CurrentTrackedDialectSubLang}
\fi
\edef\CurrentTrackedTag{\CurrentTrackedTag:\CurrentTrackedRegion}
\@tracklang@IfFileExists{#2\CurrentTrackedTag#3}%
\fi
\@tracklang@test@if@file@exists@for@langIIiscriptregion
Already tested script and region. Test without script if the script is the default then without region.
\def\@tracklang@test@if@file@exists@for@langIIiscriptregion#1#2#3{%
Is the script the default for this dialect?
\@tracklang@ifcseqcsname\CurrentTrackedDialectScript
{\@tracklang@knowniso@script@\CurrentTrackedLanguage}%
{%
Try just language code and region.
\let\CurrentTrackedTag\CurrentTrackedIsoCode
\if\CurrentTrackedDialectSubLang\empty
\else
\edef\CurrentTrackedTag{\CurrentTrackedTag-\CurrentTrackedDialectSubLang}
\fi
\edef\CurrentTrackedTag{\CurrentTrackedTag:\CurrentTrackedRegion}
\@tracklang@verbose{Testing tag ‘\CurrentTrackedTag’}%
\@tracklang@IfFileExists{#2\CurrentTrackedTag#3}%
{%
\fi
\edef\CurrentTrackedTag{\CurrentTrackedTag-\CurrentTrackedRegion}
\@tracklang@IfFileExists{#2\CurrentTrackedTag#3}%
{%
\fi
\edef\CurrentTrackedTag{\CurrentTrackedTag:\CurrentTrackedRegion}
\@tracklang@IfFileExists{#2\CurrentTrackedTag#3}%
Try just language code and script.
\let\@tracklang@test@if@file@exists@next\@tracklang@test@if@file@exists@for@langIIscriptskipregion
\else
Try without sub-language.
\edef\CurrentTrackedTag{\CurrentTrackedIsoCode-\CurrentTrackedRegion}%
\@tracklang@IfFileExists{\CurrentTrackedTag}{%}
{\let\@tracklang@if@file@exists\@tracklang@firstoftwo
\let\@tracklang@test@if@file@exists@next\@tracklang@test@if@file@exists@for@langIIscriptskipregion
\else
Script isn’t the default.
\let\@tracklang@test@if@file@exists@next\@tracklang@test@if@file@exists@for@langIIscriptskipregion
\@tracklang@test@if@file@exists@next{\CurrentTrackedIsoCode}\CurrentTrackedRegion\CurrentTrackedTag\CurrentTrackedRegion%}
{\let\@tracklang@if@file@exists\@tracklang@firstoftwo
\let\@tracklang@test@if@file@exists@next\@tracklang@test@if@file@exists@for@langIIscriptskipregion
\else
Test with script and without region.
\def\@tracklang@test@if@file@exists@for@langIIscriptskipregion#1#2#3{%\CurrentTrackedTag\CurrentTrackedIsoCode\CurrentTrackedDialectSubLang\CurrentTrackedDialectScript\CurrentTrackedRegion#1#2#3%}
\@tracklang@test@if@file@exists@next{\CurrentTrackedIsoCode}\CurrentTrackedRegion%}
{\let\@tracklang@if@file@exists\@tracklang@firstoftwo
\let\@tracklang@test@if@file@exists@next\@tracklang@test@if@file@exists@for@langIIscriptskipregion
\else
Try just language code.
\let\@tracklang@test@if@file@exists@next\@tracklang@test@if@file@exists@for@langIIscriptskipregion
\else
106
Try without sub-language.
\edef\CurrentTrackedTag{\CurrentTrackedIsoCode-\CurrentTrackedDialectScript}
\@tracklang@verbose{Testing tag ‘\CurrentTrackedTag’}
\@tracklang@IfFileExists{\CurrentTrackedTag#3}{%
 \let\@tracklang@if@file@exists\@tracklang@firstoftwo
 \let\@tracklang@test@if@file@exists@next\@tracklang@test@if@file@exists@noop
}%
\@tracklang@test@if@file@exists@next{\CurrentTrackedTag#3}%
\fi
\@tracklang@test@if@file@exists@for@langIIskipscriptskipregion
Test without script and without region.
\def\@tracklang@test@if@file@exists@for@langIIskipscriptskipregion#1#2#3{%
 \let\CurrentTrackedTag\CurrentTrackedIsoCode
 \ifx\CurrentTrackedDialectSubLang\empty
 \else
 \edef\CurrentTrackedTag{\CurrentTrackedTag-\CurrentTrackedDialectSubLang}
 \fi
 \@tracklang@verbose{Testing tag ‘\CurrentTrackedTag’}%
 \@tracklang@IfFileExists{\CurrentTrackedTag#3}{%
 \let\@tracklang@if@file@exists\@tracklang@firstoftwo
 \let\@tracklang@test@if@file@exists@next\@tracklang@test@if@file@exists@noop
}%
\fi
\@tracklang@test@if@file@exists@for@langIIskipscriptskipregion
Try other language code with script and region.
\IfTrackedLanguageHasIsoCode
{639-3}{\CurrentTrackedLanguage}{%
 \let\CurrentTrackedIsoCode\CurrentTrackedIsoCodeIII
 \let\@tracklang@test@if@file@exists@next\@tracklang@test@if@file@exists@for@langIIIscriptregion
}%
}%
No other language codes. Try just the region.
\let\@tracklang@test@if@file@exists@next\@tracklang@test@if@file@exists@for@region
}%
\else
Try just language code.
\let\CurrentTrackedTag\CurrentTrackedIsoCode
\@tracklang@verbose{Testing tag ‘\CurrentTrackedTag’}%
\@tracklang@IfFileExists{\CurrentTrackedTag#3}{%
Try without sub-language.

```latex
\edef\CurrentTrackedTag{\CurrentTrackedIsoCode-\CurrentTrackedRegion}
\@tracklang@verbose{Testing tag `\CurrentTrackedTag'}
\@tracklang@ifFileExists{#2\CurrentTrackedTag#3}
{\let@tracklang@if@file@exists@next\@tracklang@firstoftwo
\let@tracklang@test@if@file@exists@next\@tracklang@test@if@file@exists@noop}
{\let@tracklang@test@if@file@exists@next\@tracklang@test@if@file@exists@for@langIIIscriiptskipregion}
\fi
\%}
\%}
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
```

Script isn’t the default.

```latex
\let@tracklang@test@if@file@exists@next\@tracklang@test@if@file@exists@for@langIIIscriiptskipregion
\@tracklang@test@if@file@exists@next{#1}{#2}{#3}
```

Test with script and without region.

```latex
\def@tracklang@test@if@file@exists@for@langIIIscriiptskipregion#1#2#3{%
\let\CurrentTrackedTag\CurrentTrackedIsoCode
\ifx\CurrentTrackedDialectSubLang\empty
\else
  \edef\CurrentTrackedTag{\CurrentTrackedIsoCode-\CurrentTrackedDialectSubLang}
\fi
\edef\CurrentTrackedTag{\CurrentTrackedTag-\CurrentTrackedDialectScript}
\@tracklang@verbose{Testing tag `\CurrentTrackedTag'}
\@tracklang@ifFileExists{#2\CurrentTrackedTag#3}
{\let@tracklang@if@file@exists@next\@tracklang@firstoftwo
\let@tracklang@test@if@file@exists@next\@tracklang@test@if@file@exists@noop}
{\let@tracklang@test@if@file@exists@next\@tracklang@test@if@file@exists@for@langIIIscriiptskipregion}
\fi
```

Try just language code.

```latex
\let@tracklang@test@if@file@exists@next\@tracklang@test@if@file@exists@for@langIIIscriiptskipregion
\else
```

Try without sub-language.

```latex
\edef\CurrentTrackedTag{\CurrentTrackedIsoCode-\CurrentTrackedDialectScript}
\@tracklang@verbose{Testing tag `\CurrentTrackedTag'}
\@tracklang@ifFileExists{#2\CurrentTrackedTag#3}
{\let@tracklang@if@file@exists@next\@tracklang@firstoftwo
\let@tracklang@test@if@file@exists@next\@tracklang@test@if@file@exists@noop}
```
Test without script and without region.

```
\def\@tracklang@test@if@file@exists@for@langII{#1#2#3}{% 
  \let\CurrentTrackedTag\CurrentTrackedIsoCode 
  \ifx\CurrentTrackedDialectSubLang\empty 
    \edef\CurrentTrackedTag{\CurrentTrackedTag-\CurrentTrackedDialectSubLang} \fi 
  \@tracklang@verbose{Testing tag ‘\CurrentTrackedTag’} 
  \@tracklang@IfFileExists{#2\CurrentTrackedTag#3}{% 
    \let\@tracklang@if@file@exists\@tracklang@firstoftwo 
    \let\@tracklang@test@if@file@exists@next\@tracklang@test@if@file@exists@noop 
  }{% 
    \ifx\CurrentTrackedDialectSubLang\empty 
      No other language codes. Try just the region.
      \let\@tracklang@test@if@file@exists@next\@tracklang@test@if@file@exists@for@region 
    \else 
      Try just language code.
      \let\CurrentTrackedTag\CurrentTrackedIsoCode 
      \@tracklang@verbose{Testing tag ‘\CurrentTrackedTag’} 
      \@tracklang@IfFileExists{#2\CurrentTrackedTag#3}{% 
        \let\@tracklang@if@file@exists\@tracklang@firstoftwo 
        \let\@tracklang@test@if@file@exists@next\@tracklang@test@if@file@exists@noop 
      }{% 
        No other language codes. Try just the region.
        \let\@tracklang@test@if@file@exists@next\@tracklang@test@if@file@exists@for@region 
    } 
  } 
\@tracklang@test@if@file@exists@next{#1}{#2}{#3} 
```


```
\def\@tracklang@test@if@file@exists@for@langIIvariant#1#2#3{\
  \let\CurrentTrackedIsoCode\CurrentTrackedIsoCodeII\
  \let\CurrentTrackedTag\CurrentTrackedIsoCode\
  \ifx\CurrentTrackedDialectSubLang\empty\
  \else\
  \edef\CurrentTrackedTag{\CurrentTrackedTag-\CurrentTrackedDialectSubLang}\
  \fi\
  \edef\CurrentTrackedTag{\CurrentTrackedTag-\CurrentTrackedDialectVariant}\
  \@tracklang@verbose{Testing tag '\CurrentTrackedTag'}\
  \@tracklang@IfFileExists{#2\CurrentTrackedTag#3}\
  {\let\@tracklang@if@file@exists\@tracklang@firstoftwo\
   \let\@tracklang@test@if@file@exists@next\@tracklang@test@if@file@exists@noop}\
  {\
   \IfTrackedLanguageHasIsoCode{639-3}{\CurrentTrackedLanguage}\
   {\let\@tracklang@test@if@file@exists@next\@tracklang@test@if@file@exists@for@langIIIvariant}\
   {\let\@tracklang@test@if@file@exists@next\@tracklang@test@if@file@exists@for@rootlang}\
  }\
  \@tracklang@test@if@file@exists@next{#1}{#2}{#3}\
}
```

```
\def\@tracklang@test@if@file@exists@for@langIIIvariant#1#2#3{\
  \let\CurrentTrackedIsoCode\CurrentTrackedIsoCodeIII\
  \let\CurrentTrackedTag\CurrentTrackedIsoCode\
  \ifx\CurrentTrackedDialectSubLang\empty\
  \else\
  \edef\CurrentTrackedTag{\CurrentTrackedTag-\CurrentTrackedDialectSubLang}\
  \fi\
  \edef\CurrentTrackedTag{\CurrentTrackedTag-\CurrentTrackedDialectVariant}\
  \@tracklang@verbose{Testing tag '\CurrentTrackedTag'}\
  \@tracklang@IfFileExists{#2\CurrentTrackedTag#3}\
  {\let\@tracklang@if@file@exists\@tracklang@firstoftwo\
   \let\@tracklang@test@if@file@exists@next\@tracklang@test@if@file@exists@noop}\
  {\
   \IfTrackedLanguageHasIsoCode{639-3}{\CurrentTrackedLanguage}\
   {\let\@tracklang@test@if@file@exists@next\@tracklang@test@if@file@exists@for@langIIIvariant}\
   {\let\@tracklang@test@if@file@exists@next\@tracklang@test@if@file@exists@for@rootlang}\
  }\
  \@tracklang@test@if@file@exists@next{#1}{#2}{#3}\
```

```
\def\@tracklang@test@if@file@exists@for@rootlang#1#2#3{\
  \let\CurrentTrackedIsoCode\CurrentTrackedIsoCode\rootlang\
  \let\CurrentTrackedTag\CurrentTrackedIsoCode\
  \ifx\CurrentTrackedDialectSubLang\empty\
  \else\
  \edef\CurrentTrackedTag{\CurrentTrackedTag-\CurrentTrackedDialectSubLang}\
  \fi\
  \edef\CurrentTrackedTag{\CurrentTrackedTag-\CurrentTrackedDialectVariant}\
  \@tracklang@verbose{Testing tag '\CurrentTrackedTag'}\
  \@tracklang@IfFileExists{#2\CurrentTrackedTag#3}\
  {\let\@tracklang@if@file@exists\@tracklang@firstoftwo\
   \let\@tracklang@test@if@file@exists@next\@tracklang@test@if@file@exists@noop}\
  {\
   \IfTrackedLanguageHasIsoCode{639-3}{\CurrentTrackedLanguage}\
   {\let\@tracklang@test@if@file@exists@next\@tracklang@test@if@file@exists@for@langIIIvariant}\
   {\let\@tracklang@test@if@file@exists@next\@tracklang@test@if@file@exists@for@rootlang}\
  }\
  \@tracklang@test@if@file@exists@next{#1}{#2}{#3}\
```

Test 639-2 code and variant

Test 639-3 code and variant

Test the root language label. (Final test.)

114
The new definition of `\IfTrackedLanguageFileExists` has a different order of priority. If the original behaviour is specifically required, this has the pre version 1.4 definition.
\let\CurrentTrackedTag\CurrentTrackedRegion
\@tracklang@verbose{Testing tag ‘\CurrentTrackedTag’}\%
\@tracklang@ifFileExists{#2\CurrentTrackedTag#3}\%
{#4}\%
\let\CurrentTrackedTag\CurrentTrackedLanguage
\@tracklang@verbose{Testing tag ‘\CurrentTrackedTag’}\%
\@tracklang@ifFileExists{#2\CurrentTrackedTag#3}{#4}{#5}\%
\fi
\else
\edef\CurrentTrackedTag{%
\CurrentTrackedIsoCode-\CurrentTrackedRegion\%
\@tracklang@verbose{Testing tag ‘\CurrentTrackedTag’}\%
\@tracklang@ifFileExists{#2\CurrentTrackedTag#3}\%
{#4}\%
\let\CurrentTrackedTag\CurrentTrackedLanguage
\@tracklang@verbose{Testing tag ‘\CurrentTrackedTag’}\%
\@tracklang@ifFileExists{#2\CurrentTrackedTag#3}{#4}{#5}\%
\let\CurrentTrackedTag\CurrentTrackedRegion
\@tracklang@verbose{Testing tag ‘\CurrentTrackedTag’}\%
\@tracklang@ifFileExists{#2\CurrentTrackedTag#3}\%
{#4}\%
\let\CurrentTrackedTag\CurrentTrackedLanguage
\@tracklang@verbose{Testing tag ‘\CurrentTrackedTag’}\%
\@tracklang@ifFileExists{#2\CurrentTrackedTag#3}{#4}{#5}\%
}\%
\fi
{#5}% unknown dialect
4.9 Resources

Provide some commands to make it easier for package authors to integrate the package code with \tracklang. In the command definition describes below, \langle \pkgname \rangle indicates the initial part of the resource files that follow the naming convention, \langle \pkgname \rangle-\langle \tag \rangle.ldf. Typically this will match the base name of the package that uses those resource files, but this isn’t compulsory. The argument \langle \tag \rangle is the current tracked tag obtained from \IfTrackedLanguageFileExists.

\TrackLangProvidesResource{\langle \tag \rangle}{\langle \version \rangle}

If \ProvidesFile exists, we can use that, otherwise we need to provide a generic version.

\ifx\ProvidesFile\undefined
\Generic code uses simplistic method to grab the version details in the final optional argument. Since we’re not using \LaTeX we don’t have commands like \@ifnextchar available.
\endif
\ifx\undefined\undefined
\TrackLangProvidesResource\TrackLangProvidesResource\TrackLangProvidesResource\TrackLangProvidesResource\TrackLangProvidesResource\TrackLangProvidesResource
\@tracklang@err{Resources files using \TrackLangProvidesResource must be loaded with \string\TrackLangRequireDialect}\%
\else
\expandafter\@tracklang@providesresource\TrackLangRequireDialectPrefix-\#1.ldf\#2%
\fi
\relax
\expandafter\@tracklang@providesresource\TrackLangRequireDialectPrefix-\#1.ldf\#2%
\else
\expandafter%\csname ver@\TrackLangRequireDialectPrefix-\#1.ldf\endcsname
{%
\newlinechar='\^^J
\def\MessageBreak{\^^J}%
\message{\^^JFile: \TrackLangRequireDialectPrefix-\#1.ldf\^^J}%
\long\@tracklang@next{#2}%
\fi
\@tracklang@next
\expandafter%\csname ver@\TrackLangRequireDialectPrefix-\#1.ldf\endcsname
{%
\newlinechar='\^^J
\expandafter%\csname ver@\#1\endcsname
{%
\newlinechar='\^^J

119
LaTeX code can simply use \ProvidesFile.  
def\TrackLangProvidesResource#1{%  \ifx\TrackLangRequireDialectPrefix\undefined  \@tracklang@err{Resources files using \string\TrackLangProvidesResource\space must be loaded with \string\TrackLangRequireDialect}%  \fi  \ProvidesFile{\TrackLangRequireDialectPrefix-#1.ldf}%  \fi
}

Within the resource files, a check is required for the language hook, where the hook type is given by \langle \textit{type} \rangle. For example, if \langle \textit{type} \rangle is captions, the for babel, this is \caption{\textit{dialect}} (dialect obtained through \texttt{\CurrentTrackedDialect} and for polyglossia, this is \caption{\textit{language}} (language obtained through \texttt{\CurrentTrackedLanguage}). This command is not permitted outside resource files.

def\TrackLangAddToHook\noop@TrackLangAddToHook

\@TrackLangAddToHook
\def\noop@TrackLangAddToHook#1#2{%  \@tracklang@ifundef{#2\CurrentTrackedDialect}%  {Does the dialect label have a mapping?  \IfTrackedDialectHasMapping{\CurrentTrackedDialect}%  {Try the mapping label next.  \edef\@tracklang@tmp{%  \csname @tracklang@dialectmap@tohook@\CurrentTrackedDialect\endcsname}\@tracklang@ifundef{#2\@tracklang@tmp}%  {}}}}
No captions hook. Try polyglossia.
\@tracklang@ifundef{#2\CurrentTrackedLanguage}\
{% No captions hook. Do nothing (it’s possible no language package has been
loaded).
\@tracklang@verbose{no hook found for \CurrentTrackedDialect}\
}\
{\@tracklang@addtohook{#2}{\CurrentTrackedLanguage}{#1}\
}\
{\@tracklang@addtohook{#2}{\@tracklang@tmp}{#1}\
}\
{polyglossia check next.
\@tracklang@ifundef{#2\CurrentTrackedLanguage}\
{% No captions hook.
\@tracklang@verbose{no hook found for \CurrentTrackedDialect}\
}\
{\@tracklang@addtohook{#2}{\CurrentTrackedLanguage}{#1}\
}\
{\@tracklang@addtohook{#2}{\CurrentTrackedDialect}{#1}\
}\
Do the code now. (This is needed for cases such as the ngerman which defines
\captionsngerman but calls it immediately rather than at the start of the
document.)
#1\
}\@tracklang@addtohook{\langle\type\rangle}{\langle\label\rangle}{\langle\code\rangle}
\def\@tracklang@addtohook#1#2#3{%\
\expandafter\let\expandafter\@tracklang@hook\csname #1#2\endcsname\
\expandafter\gdef\csname #1#2\endcsname\expandafter\endcsname\expandafter\\expandafter{\@tracklang@hook#3}\
}\
Since the captions hook is the most common, provide a shortcut.

121
\TrackLangAddToCaptions{⟨code⟩}
\def\TrackLangAddToCaptions#1{\TrackLangAddToHook#1{captions}}

\TrackLangRedefHook{⟨code⟩}{⟨type⟩}
Similar to \TrackLangAddToHook but redefines the hook rather than appending to it. This command is not permitted outside resource files.
\def\TrackLangRedefHook{\noop@TrackLangRedefHook}

\noop@TrackLangRedefHook
\def\noop@TrackLangRedefHook#1#2{%
\@tracklang@err{\string\TrackLangRedefHook\space only permitted within resource files}
}

\@TrackLangRedefHook
\def\@TrackLangRedefHook#1#2{%
babel check first.
\@tracklang@ifundef{#2\CurrentTrackedDialect} {%
Does the dialect label have a mapping?
\IfTrackedDialectHasMapping{\CurrentTrackedDialect} {%
Try the mapping label next.
\edef\@tracklang@tmp{%
\csname @tracklang@dialectmap@tohook@\CurrentTrackedDialect\endcsname}%
\@tracklang@ifundef{#2\@tracklang@tmp} {%
No captions hook. Try polyglossia.
\@tracklang@ifundef{#2\CurrentTrackedLanguage} {%
No captions hook. Do nothing (it’s possible no language package has been loaded).
\@tracklang@verbose{no hook found for \CurrentTrackedDialect}%
}%
\@tracklang@redefhook{#2}{\CurrentTrackedLanguage}{#1}%
}%
}%
\@tracklang@redefhook{#2}{\@tracklang@tmp}{#1}%
}%
}%
}
Define a mapping between a tracklang dialect label and the corresponding label used by the language hook. For example, ngermanDE is a recognised tracklang dialect label, but the closest babel equivalent is ngerman, so ngermanDE would need to be mapped to ngerman for the language hooks. The arguments are ⟨tracklang-label⟩ (the tracklang dialect label) and ⟨hook-label⟩ (the babel, polyglossia etc label).
\def\SetTrackedDialectLabelMap#1#2{% Store the mapping that can obtain the hook label from the tracklang label (tracklang to hook).
   \@tracklang@enamedef{@tracklang@dialectmap@tohook@#1}{#2}%
Store the mapping that can obtain the tracklang label from the hook label.
   \@tracklang@enamedef{@tracklang@dialectmap@fromhook@#2}{#1}%
}\IfTrackedDialectHasMapping⟨tracklang label⟩⟨true⟩
{⟨false⟩}
Test if the tracklang dialect label has been assigned a mapping to a language hook.
\def\IfTrackedDialectHasMapping#1#2#3{%\@tracklang@ifundef{@tracklang@dialectmap@tohook@#1}{#3}{#2}%}

IfHookHasMappingFromTrackedDialect
Tests if the language hook label has been assigned a mapping from a tracklang dialect label.
\def\IfHookHasMappingFromTrackedDialect#1#2#3{%\@tracklang@ifundef{@tracklang@dialectmap@fromhook@#1}{#3}{#2}%}

GetTrackedDialectToMapping
Gets the mapping for the given tracklang dialect label to a language hook label or the \textit{label} itself if no mapping has been defined.
\def\GetTrackedDialectToMapping#1{%\@tracklang@ifundef{@tracklang@dialectmap@tohook@#1}{#1}{{\csname @tracklang@dialectmap@tohook@#1\endcsname}}%}

GetTrackedDialectFromMapping
Gets the reverse mapping from the given language hook to the tracklang label.
\def\GetTrackedDialectFromMapping#1{%\@tracklang@ifundef{@tracklang@dialectmap@fromhook@#1}{#1}{{\csname @tracklang@dialectmap@fromhook@#1\endcsname}}%}

TrackLangRequireResource
\def\TrackLangRequireResource\noop@TrackLangRequireResource
\def\noop@TrackLangRequireResource#1{%\@tracklang@err{\string\TrackLangRequireResource\space only permitted within resource files}}
\@TrackLangRequireResource \Actual behaviour.
\def\@TrackLangRequireResource#1{%\@tracklang@ifundef{ver\@\TrackLangRequireDialectPrefix-#1.ldf}{%\@tracklang@IfFileExists{\TrackLangRequireDialectPrefix-#1.ldf}{%\input \TrackLangRequireDialectPrefix-#1.ldf}\%\@tracklang@warn{No ‘\TrackLangRequireDialectPrefix’ support for language/region ‘#1’}MessageBreak\resource file ‘\TrackLangRequireDialectPrefix-#1.ldf’ not found)%}{%}{%}}%

\TrackLangRequireResourceOrDo\langle resource loaded code\rangle\langle resource already loaded code\rangle

Like \TrackLangRequireResource but also does \langle resource loaded code\rangle if the resource file is loaded or \langle resource already loaded code\rangle if the resource file has already been loaded.
\def\TrackLangRequireResourceOrDo{%\noop@TrackLangRequireResourceOrDo%
}

\noop@TrackLangRequireResourceOrDo

Default behaviour outside of resources files: generate an error and ignore arguments.
\def\noop@TrackLangRequireResourceOrDo#1#2#3{%\@tracklang@err{\string\TrackLangRequireResourceOrDo\space only permitted within resource files}%}

\TrackLangRequireResourceOrDo \Actual behaviour.
\def\@TrackLangRequireResourceOrDo#1#2#3{%\@tracklang@ifundef{ver\@\TrackLangRequireDialectPrefix-#1.ldf}{%\@tracklang@IfFileExists{\TrackLangRequireDialectPrefix-#1.ldf}{%\input \TrackLangRequireDialectPrefix-#1.ldf #2\%\@tracklang@warn{No ‘\TrackLangRequireDialectPrefix’ support for language/region ‘#1’}MessageBreak\resource file ‘\TrackLangRequireDialectPrefix-#1.ldf’ not found)%}{%}{%}}%
\TrackLangRequestResource\{\langle tag\rangle\}\{\langle not found code\rangle\}

Like \TrackLangRequireResource but does \textit{\langle not found code\rangle} if the file doesn't exist.

\def\TrackLangRequestResource{\noop\TrackLangRequestResource}

\noop\TrackLangRequestResource

Default behaviour outside of resources files: generate an error and ignore arguments.

\def\noop\TrackLangRequestResource\#1\#2{\%
\@tracklang@ifundef{ver\TrackLangRequireDialectPrefix-#1.ldf}{\%
\@tracklang@IfFileExists{\TrackLangRequireDialectPrefix-#1.ldf}{\%
\input \TrackLangRequireDialectPrefix-#1.ldf\%
\}{#2}\%
}{%}
%
}

\TrackLangRequireDialect\{\langle load code\rangle\}\{\langle pkgname\rangle\}\{\langle dialect\rangle\}

Actual behaviour.

\def\@TrackLangRequireDialect\#1\#2{\%
\@tracklang@ifundef{ver\TrackLangRequireDialectPrefix-#1.ldf}{\%
\@tracklang@IfFileExists{\TrackLangRequireDialectPrefix-#1.ldf}{\%
\input \TrackLangRequireDialectPrefix-#1.ldf\%
\}{#2}\%
}{%}
%
}

\TrackLangRequireDialect\noop\TrackLangRequireDialect

No-op code.

\def\noop\TrackLangRequireDialect\#1{\%
\@tracklang@err{\string\TrackLangRequestResource\space only permitted within resource files}\%
}

\TrackLangRequireDialect\noop\TrackLangRequireDialect\#1\#2\#3{\%
\@tracklang@err{\string\TrackLangRequestResource\space only permitted within resource files}\%
}

126
only permitted within resource files}
}
\@TrackLangRequireDialect Actual code.
\def\@TrackLangRequireDialect#1{%\def\@tracklang@next{\@TrackLangRequireDialect[#1]}%\ifx[#1]\relax\def\@tracklang@next{\@@TrackLangRequireDialect%\TrackLangRequireResource{\CurrentTrackedTag}}{#1}}%\fi\@tracklang@next}
\def\@@TrackLangRequireDialect[#1]#2#3{%\def\TrackLangRequireDialectPrefix{#2}\IfTrackedLanguageFileExists{#3}{%Enable \TrackLangRequireResource etc so that they can only be used in resource files.
\let\TrackLangRequireResource\@TrackLangRequireResource\let\TrackLangRequireResourceOrDo\@TrackLangRequireResourceOrDo\let\TrackLangRequestResource\@TrackLangRequestResource
\disable\TrackLangRequireDialect so that it can’t be used in resource files.
\let\TrackLangRequireDialect\noop\@TrackLangRequireDialect
\enable\TrackLangAddToHook.
\let\TrackLangAddToHook\@TrackLangAddToHook
Load resource file using the code provided in the first argument.
\#1%
\disable\TrackLangRequireResource etc.
\let\TrackLangRequireResource\noop\@TrackLangRequireResource\let\TrackLangRequireResourceOrDo\noop\@TrackLangRequireResourceOrDo\let\TrackLangRequestResource\noop\@TrackLangRequestResource
\enable\TrackLangRequireDialect.
\let\TrackLangRequireDialect\@TrackLangRequireDialect
\disable\TrackLangAddToHook.
\let\TrackLangAddToHook\noop\@TrackLangAddToHook
}%
% \tracklang@warn{No ‘#2’ support for dialect ‘#3’}%
}%
\restore category code for @ if necessary.
\@tracklang@restore@at
5 Regions Generic Code (tracklang-region-codes.tex)

This is only loaded if a mapping is required between numeric and alphabetic region codes. (It would slow down the package loading to automatically load if not required.) Since this is loaded on the fly, we need to be careful about spurious spaces.

\ifnum\catcode'\@=11\relax
  \def\@tracklang@regions@restore@at{}%
\else
  \expandafter\edef\csname @tracklang@regions@restore@at\endcsname{
    \noexpand\catcode'\noexpand\@=\number\catcode'\@\relax
  }%
  \catcode'\@=11elax
\fi

Check if this file has already been loaded:
\ifx\TrackLangRegionMap\undefined
  \@tracklang@regions@restore@at
  \expandafter\endinput
\fi

Version info.
\expandafter\def\csname ver@tracklang-region-codes.tex\endcsname{2022/12/13 v1.6.1 (NLCT) Track Languages Regions}%

\TrackLangRegionMap\langle numeric code\rangle\{\langle alpha-2 code\rangle\}{\langle alpha-3 code\rangle}

Define mapping.
\def\TrackLangRegionMap#1#2#3{% 
  \@tracklang@enamedef{@tracklang@region@numtoalphaii@#1}{#2}%
  \@tracklang@enamedef{@tracklang@region@numtoalphaiii@#1}{#3}%
  \@tracklang@enamedef{@tracklang@region@alphaiitonum@#2}{#1}%
  \@tracklang@enamedef{@tracklang@region@alphaiiitonum@#3}{#1}%
  \@tracklang@enamedef{@tracklang@region@alphaiitoalphaiii@#2}{#3}%
  \@tracklang@enamedef{@tracklang@region@alphaiiitoalphaii@#3}{#2}%
}%

\TrackLangAlphaIIToNumericRegion\langle alpha-2 code\rangle

\def\TrackLangAlphaIIToNumericRegion#1{% 
  \@tracklang@nameuse{@tracklang@region@alphaiitonum@#1}%
}%

128
Define mappings.
\TrackLangRegionMap{004}{AF}{AFG}
\TrackLangRegionMap{248}{AX}{ALA}
\TrackLangRegionMap{008}{AL}{ALB}
\TrackLangRegionMap{012}{SZ}{DZA}
\TrackLangRegionMap{016}{AS}{ASM}
\TrackLangRegionMap{020}{AD}{AND}
\TrackLangRegionMap{024}{AO}{AGO}
\TrackLangRegionMap{660}{AI}{AIA}
\TrackLangRegionMap{010}{AQ}{ATA}
\TrackLangRegionMap{028}{AG}{ATG}
\TrackLangRegionMap{032}{AR}{ARG}
\TrackLangRegionMap{051}{AM}{ARM}
\TrackLangRegionMap{533}{AW}{ABW}
\TrackLangRegionMap{036}{AU}{AUS}
\TrackLangRegionMap{040}{AT}{AUT}
\TrackLangRegionMap{031}{AZ}{AZE}
\TrackLangRegionMap{044}{BS}{BHS}
\TrackLangRegionMap{048}{BH}{BHR}
\TrackLangRegionMap{050}{BD}{BGD}
\TrackLangRegionMap{052}{BB}{BRB}
\TrackLangRegionMap{112}{BY}{BLR}
\TrackLangRegionMap{056}{BE}{BEL}
\TrackLangRegionMap{084}{BZ}{BLZ}
\TrackLangRegionMap{204}{BJ}{BEN}
\TrackLangRegionMap{060}{BM}{BHU}
\TrackLangRegionMap{064}{BT}{BTN}
\TrackLangRegionMap{068}{BO}{BOL}
\TrackLangRegionMap{553}{BQ}{BES}
\TrackLangRegionMap{070}{BA}{BIH}
\TrackLangRegionMap{072}{BW}{BWA}
\TrackLangRegionMap{074}{BY}{BVT}
\TrackLangRegionMap{076}{BR}{BRA}
\TrackLangRegionMap{086}{IO}{IOT}
\TrackLangRegionMap{096}{BN}{BRN}
\TrackLangRegionMap{100}{BG}{BGR}
\TrackLangRegionMap{854}{BF}{BFA}
\TrackLangRegionMap{108}{BI}{BDI}
6 ISO 15924 Scripts \LaTeX Package (tracklang-scripts.sty)

This is just a \LaTeX package wrapper for the generic code in tracklang-scripts.tex.

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{tracklang-scripts}[2022/12/13  v1.6.1 (NLCT) Track Language Scripts (LaTeX)]
\RequirePackage{tracklang}
\input{tracklang-scripts}

7 ISO 15924 Scripts Generic Code (tracklang-scripts.tex)

Provides information about ISO 15924 scripts. Not automatically loaded.
\ifnum\catcode\'@=11\relax
  \def\@tracklang@scripts@restore@at{}
\else
  \expandafter\edef\csname @tracklang@scripts@restore@at\endcsname{\noexpand\catcode\'\@=%\number\catcode\'\relax}
\fi
\catcode\'\@=11\relax
\fi

Check if this file has already been loaded:
\ifx\TrackLangScriptMap\undefined
  \else
    \@tracklang@scripts@restore@at
    \expandafter\endinput
  \fi
Define mapping. To avoid problems with encodings, only use ASCII characters in the arguments. The first argument is the four-letter ISO 15924 code. The second argument is the numeric code. The third argument is just intended for informational purposes. The fourth argument indicates the direction. This may be LR (left-to-right), RL (right-to-left), TB (top-to-bottom), varies or inherited. The ⟨parent⟩ argument is for the parent writing system, which may be left blank. (Currently, this is blank for all the mappings provided here, but the syntax has five arguments in case of future development.)

\def\TrackLangScriptMap#1#2#3#4#5{\%
This user command is provided to make it easier to test the script using \iffx.
   \@tracklang@enamedef{TrackLangScript#1}{#1}\%
   \@tracklang@enamedef{@tracklang@script@numtoalpha@#2}{#1}\%
   \@tracklang@enamedef{@tracklang@script@alphatonum@#1}{#2}\%
   \@tracklang@enamedef{@tracklang@script@alphatodir@#1}{#3}\%
   \@tracklang@enamedef{\ifx\relax#5\relax}{\else\%\fi\relax\%\else\%\fi\relax\%\}
%
}
I wasn’t sure whether or not to implement a parent, but it’s here if required. Unlike the other elements above, there’s also a command to set this field.
Define mappings. The parent information is currently missing.

\TrackLangScriptMap{Adlm}{166}{Adlam}{RL}{%}
\TrackLangScriptMap{Afak}{439}{Afaka}{varies}{%}
\TrackLangScriptMap{Aghb}{239}{Caucasian Albanian}{LR}{%}
\TrackLangScriptMap{Ahom}{338}{Ahom, Tai Ahom}{LR}{%}
\TrackLangScriptMap{Arab}{160}{Arabic}{RL}{%}
\TrackLangScriptMap{Aran}{161}{Arabic (Nastaliq variant)}{RL}{%}
\TrackLangScriptMap{Armi}{124}{Imperial Aramaic}{RL}{%}
\TrackLangScriptMap{Avst}{134}{Avestan}{RL}{%}
\TrackLangScriptMap{Bali}{360}{Balinese}{LR}{%}
\TrackLangScriptMap{Bamu}{435}{Bamum}{LR}{%}
\TrackLangScriptMap{Bass}{259}{Bassa Vah}{LR}{%}
\TrackLangScriptMap{Batk}{365}{Batak}{LR}{%}
\TrackLangScriptMap{Beng}{334}{Bhaiksuki}{LR}{%}
\TrackLangScriptMap{Blis}{550}{Blissymbols}{varies}{%}
\TrackLangScriptMap{Bopo}{285}{Bopomofo}{LR}{%}
\TrackLangScriptMap{Brah}{300}{Brahmi}{LR}{%}
\TrackLangScriptMap{Brai}{570}{Braille}{LR}{%}
\TrackLangScriptMap{Bugi}{367}{Buginese}{LR}{%}
\TrackLangScriptMap{Buhd}{372}{Buhid}{LR}{%}
\TrackLangScriptMap{Cakm}{349}{Chakma}{LR}{%}
\TrackLangScriptMap{Cans}{440}{Unified Canadian Aboriginal Syllabics}{LR}{%}
\TrackLangScriptMap{Cari}{201}{Carian}{LR}{%}
\TrackLangScriptMap{Cham}{358}{Cham}{LR}{%}
\TrackLangScriptMap{Cher}{445}{Cherokee}{LR}{%}
\TrackLangScriptMap{Cirt}{291}{Cirth}{varies}{%}
\TrackLangScriptMap{Copt}{204}{Coptic}{LR}{%}
\TrackLangScriptMap{Cprt}{403}{Cypriot}{RL}{%}
\TrackLangScriptMap{Cyril}{220}{Cyrillic}{LR}{%}

138
Load additional script files.
\@tracklang\@input\@extra\@script\@files
    Restore category code of \@.
\@tracklang\@scripts\@restore\@at
Change History

1.0 – 2014-09-29
General: Initial release          2

1.1 – 2014-11-21
  
  \@tracklang@add@argentinespanish:  
  new ........................................... 76
  \@tracklang@add@bolivianspanish:  
  new ........................................... 77
  \@tracklang@add@chilianspanish:  
  new ........................................... 77
  \@tracklang@add@columbianspanish:  
  new ........................................... 77
  \@tracklang@add@costaricanspanish:  
  new ........................................... 77
  \@tracklang@add@cubanspanish:  
  new ........................................... 77
  \@tracklang@add@de-BE:  
  new ........................................... 80
  \@tracklang@add@de-CH:  
  new ........................................... 82
  \@tracklang@add@de-DE:  
  new ........................................... 80
  \@tracklang@add@dominicanspanish:  
  new ........................................... 77
  \@tracklang@add@ecudorianspanish:  
  new ........................................... 77
  \@tracklang@add@elsalvadorspanish:  
  new ........................................... 77
  \@tracklang@add@en-AU:  
  new ........................................... 79
  \@tracklang@add@en-CA:  
  new ........................................... 79
  \@tracklang@add@en-GB:  
  new ........................................... 79
  \@tracklang@add@en-GG:  
  new ........................................... 80
  \@tracklang@add@en-JE:  
  new ........................................... 80
  \@tracklang@add@en-MT:  
  new ........................................... 80
  \@tracklang@add@en-NZ:  
  new ........................................... 79
  \@tracklang@add@en-US:  
  new ........................................... 79
  \@tracklang@add@en-AR:  
  new ........................................... 81
  \@tracklang@add@es-BO:  
  new ........................................... 81
  \@tracklang@add@es-CL:  
  new ........................................... 81
  \@tracklang@add@es-CO:  
  new ........................................... 81
  \@tracklang@add@es-CR:  
  new ........................................... 81
  \@tracklang@add@es-CU:  
  new ........................................... 81
  \@tracklang@add@es-D0:  
  new ........................................... 81
  \@tracklang@add@es-EC:  
  new ........................................... 81
  \@tracklang@add@es-ES:  
  new ........................................... 80
  \@tracklang@add@es-GT:  
  new ........................................... 81
  \@tracklang@add@es-HN:  
  new ........................................... 81
  \@tracklang@add@es-MX:  
  new ........................................... 81
  \@tracklang@add@es-NI:  
  new ........................................... 81
  \@tracklang@add@es-PA:  
  new ........................................... 81
  \@tracklang@add@es-PY:  
  new ........................................... 81
  \@tracklang@add@es-SV:  
  new ........................................... 81
  \@tracklang@add@es-UY:  
  new ........................................... 82
  \@tracklang@add@es-VE:  
  new ........................................... 82
  \@tracklang@add@fr-BE:  
  new ........................................... 79
  \@tracklang@add@fr-CA:  
  new ........................................... 79
  \@tracklang@add@fr-CH:  
  new ........................................... 82
  \@tracklang@add@fr-FR:  
  new ........................................... 79
  \@tracklang@add@fr-GG:  
  new ........................................... 80
  \@tracklang@add@fr-JE:  
  new ........................................... 80
  \@tracklang@add@france:  
  new ........................................... 73
  \@tracklang@add@guatemalanspanish:  
  new ........................................... 77
  \@tracklang@add@guernseyenglish:  
  new ........................................... 75
  \@tracklang@add@guernseyfrench:  
  new ........................................... 75
  \@tracklang@add@honduranspanish:  
  new ........................................... 77
  \@tracklang@add@isleofmanenglish:  
  new ........................................... 75
  \@tracklang@add@jerseyenglish:  
  new ........................................... 75
  \@tracklang@add@jerseyfrench:  
  new ........................................... 75
  \@tracklang@add@maltaenglish:  
  new ........................................... 75
  \@tracklang@add@maltaenglish:  
  new ........................................... 75
  \@tracklang@add@maltaenglish:  
  new ........................................... 75
  \@tracklang@add@maltaenglish:  
  new ........................................... 75
  \@tracklang@add@manx:  
  new ........................................... 74
  \@tracklang@add@maltamaltese:  
  new ........................................... 63
  \@tracklang@add@maltese:  
  new ........................................... 63
  \@tracklang@add@mexicanspanish:  
  new ........................................... 77
  \@tracklang@add@ermanDE:  
  new ........................................... 74

143
\tracklang@add@nicaraguanspanish: new ............... 77
\tracklang@add@nl-NE: new . 79
\tracklang@add@nl-NL: new . 80
\tracklang@add@panamaspanish: new ............... 77
\tracklang@add@paraguayspanish: new ............... 77
\tracklang@add@peruvianspanish: new ............... 77
\tracklang@add@portugal: new 76
\tracklang@add@pt-BR: new . 79
\tracklang@add@pt-PT: new . 80
\tracklang@add@puertoricospanish: new ............... 77
\tracklang@add@rm-CH: new . 82
\tracklang@add@sl-SI: new . 80
\tracklang@add@spainspanish: new ............... 76
\tracklang@add@swissfrench: new ............... 78
\tracklang@add@swissgerman: new ............... 78
\tracklang@add@swissitalian: new ............... 78
\tracklang@add@swissromanian: new ............... 78
\tracklang@add@uruguayspanish: new ............... 78
\tracklang@add@venezuelanspanish: new ............... 78
\tracklang@add@declareoption: new ............... 55
\LetTrackLangOption: new ... 79

1.2 – 2015-03-23
\tracklang@add@GBirish: new ... 74
\tracklang@add@IEenglish: new ............... 74
\tracklang@add@IEIrish: new ... 74
\tracklang@add@GBenglish: new ... 82
\tracklang@add@GBIrish: new ... 82

1.3 – 2016-10-07
General: added check for @
category code .............. 7
added test for german.sty ... 3
added tracklang-region-codes.tex ... 128
added tracklang-scripts.sty ... 135
added tracklang-scripts.tex ... 135

removed hard-coded
polyglossia language list ... 3
\tracklang@checklocale: new 13
\TrackLangRequestResource: new ............... 126
\tracklang@requireResource: new ............... 127
\tracklang@requireResourceOrDo: new ............... 125
\tracklang@add@GBscottish: new ............... 76
\tracklang@add@GBwelsh: new .... 78
\tracklang@add@abkhaz: new ... 56
\tracklang@add@afar: new ... 56
\tracklang@add@akan: new ... 56
\tracklang@add@aragonese: new ............... 57
\tracklang@add@assamese: new ... 57
\tracklang@add@avaric: new ... 57
\tracklang@add@avestan: new ... 57
\tracklang@add@aymara: new ... 57
\tracklang@add@azerbaijani: new ............... 57
\tracklang@add@bambara: new ... 57
\tracklang@add@bashkir: new ... 57
\tracklang@add@belarusan: new ............... 58
\tracklang@add@berber: new ... 58
\tracklang@add@bihari: new ... 58
\tracklang@add@bislama: new ... 58
\tracklang@add@bokmal: new ... 58
\tracklang@add@bosnian: new ... 58
\tracklang@add@burmese: new ... 58
\tracklang@add@chamorro: new ... 58
\tracklang@add@chечен: new ... 58
\tracklang@add@chichewa: new ... 58
\tracklang@add@chinese: new ... 58
\tracklang@add@churchslavonic: new ............... 59
\tracklang@add@chuvash: new ... 59
\tracklang@add@cornish: new ... 59
\tracklang@add@corsican: new ... 59
\tracklang@add@cree: new ... 59
\tracklang@add@cy-GB: new ... 83
\tracklang@add@de-AT: new ... 82
\tracklang@add@de-AT-1996: new ............... 82

144
\@tracklang\@add@sardinian: new 65  
\@tracklang\@add@serbian: new 76  
\@tracklang\@add@serbianl: new 76  
\@tracklang\@add@shona: new 66  
\@tracklang\@add@sindhi: new 66  
\@tracklang\@add@sinhalese: new 66  
\@tracklang\@add@sloveneistriaslovenian: fixed root language name new 76  
\@tracklang\@add@somali: new 66  
\@tracklang\@add@southernndebele: new 66  
\@tracklang\@add@southernsotho: new 66  
\@tracklang\@add@sudanese: new 66  
\@tracklang\@add@swahili: new 66  
\@tracklang\@add@swati: new 66  
\@tracklang\@add@tagalog: new 66  
\@tracklang\@add@tahtian: new 67  
\@tracklang\@add@tajik: new 67  
\@tracklang\@add@tatar: new 67  
\@tracklang\@add@tigrinya: new 67  
\@tracklang\@add@tonga: new 67  
\@tracklang\@add@tsonga: new 67  
\@tracklang\@add@tswananew 67  
\@tracklang\@add@twi: new 67  
\@tracklang\@add@uyghur: new 67  
\@tracklang\@add@uzbek: new 68  
\@tracklang\@add@venda: new 68  
\@tracklang\@add@volapuk: new 68  
\@tracklang\@add@westernfrisian: corrected ISO 639-1 code new 68  
\@tracklang\@add@woolf: new 68  
\@tracklang\@add@xhosa: new 68  
\@tracklang\@add@yiddish: new 68  
\@tracklang\@add@yoruba: new 68  
\@tracklang\@add@zhuang: new 69  
\@tracklang\@add@zulu: new 69  
\@tracklang\@add@zhook: new 121  
\@tracklang\@@hasfifthchar: new 37  
\@tracklang\@@hasfourthchar: new 37  
\@tracklang\@@hassecondchar: new 37
\GetTrackedDialectFromMapping: new 124
\GetTrackedDialectModifier: new 29
\GetTrackedDialectScript: new 29
\GetTrackedDialectSubLang: new 31
\GetTrackedDialectToMapping: new 124
\GetTrackedDialectVariant: new 124
\GetTrackedLanguageTag: new 30
\IfHasTrackedDialectAdditional: new 32
\IfHasTrackedDialectModifier: new 29
\IfHasTrackedDialectScript: new 29
\IfHasTrackedDialectSubLang: new 31
\IfHasTrackedDialectVariant: new 30
\IfTrackedDialectHasMapping: new 123
\IfTrackedDialectIsScriptCs: new 27
\IfTrackLangShowInfo: new 11
\IfTrackLangShowWarnings: new 10
\LetTrackLangSynonym: new 79

noop@TrackLangRequestResource: new 126
noop@TrackLangRequireDialect: new 126
noop@TrackLangRequireResource: new 124
noop@TrackLangRequireResourceOrDo: new 125

\SetCurrentTrackedDialect: new 32
\SetTrackedDialectAdditional: new 31
\SetTrackedDialectLabelMap: new 123
\SetTrackedDialectModifier: new 28
\SetTrackedDialectScript: new 29
\SetTrackedDialectSubLang: new 31
\SetTrackedDialectVariant: new 30

\ThreeLetterExtIsoLanguageCode: new 28
\TrackLangAddToCaptions: new 122
\TrackLangAddToHook: new 120
\TrackLangAlphaIIIToNumericRegion: new 129
\TrackLangAlphaIIIToNumericRegionB: new 128
\TrackLangDeclareDialectOption: new 71
\TrackLangDeclareLanguageOption: new 56
\TrackLangFromEnv: new 51
\TrackLangGetDefaultScript: new 27
\TrackLangGetKnownCountry: new 26
\TrackLangGetKnownIsoThreeLetterLang: new 25
\TrackLangGetKnownIsoThreeLetterLangB: new 26
\TrackLangGetKnownIsoTwoLetterLang: new 25
\TrackLangGetKnownLangFromIso: new 26
\TrackLangIfAlphaNumericChar: new 39
\TrackLangIfHasDefaultScript: new 27
\TrackLangIfHasKnownCountry: new 26
\TrackLangIfKnownAlphaIIIRegion: new 130
\TrackLangIfKnownAlphaIIRegion: new 129
\TrackLangIfKnownIsoThreeLetterLang: new 25
\TrackLangIfKnownIsoThreeLetterLangB: new 25
\TrackLangIfKnownIsoTwoLetterLang: new 24
\TrackLangIfKnownLangFromIso: new 26
\TrackLangIfKnownNumericRegion: new 129
\TrackLangIfLanguageTag: new 40
\TrackLangIfRegionTag: new 41
\TrackLangIfScriptTag: new 42
\TrackLangIfVariantTag: new 43

147
\TrackLangNewLanguage: new . 23
\TrackLangNumericToAlphaIIIRegion: 
new .......................... 129
\TrackLangNumericToAlphaIIRegion: 
new .......................... 129
\TrackLangParseFromEnv: new . 18
\TrackLangProvidesResource: 
new .......................... 119
\TrackLangQueryEnv: new . 15, 16
\TrackLangQueryOtherEnv: 
new .......................... 15, 17
\TrackLangRegionMap: new . 128
\TrackLangRequestResource: 
new .......................... 126
\TrackLangRequireDialect: 
new .......................... 126
\TrackLangRequireResource: 
new .......................... 124
\TrackLangRequireResourceOrDo: 
new .......................... 125
\TrackLangScriptAlphaToDir: 
new .......................... 137
\TrackLangScriptAlphaToName: 
new .......................... 137
\TrackLangScriptAlphaToNumeric: 
new .......................... 136
\TrackLangScriptGetParent: 
new .......................... 138
\TrackLangScriptIfHasParent: 
new .......................... 138
\TrackLangScriptIfKnownAlpha: 
new .......................... 136
\TrackLangScriptIfKnownNumeric: 
new .......................... 137
\TrackLangScriptMap: new . 136
\TrackLangScriptNumericToAlpha: 
new .......................... 137
\TrackLangScriptSetParent: 
new .......................... 137
\TrackLanguageTag: new ...... 45
\TrackLocale: new ............. 52
1.3.2 – 2016-10-11
\TrackLangTryShellEscape: 
added check if \shellescape 
has been set to \relax ....... 13
1.3.3 – 2016-11-03
\IfHookHasMappingFromTrackedDialect: 
new .......................... 124
\SetCurrentTrackedDialect: 
fixed mapping ................. 33
\TrackLangProvidePredefinedDialect: 
fixed mapping order .......... 71
1.3.4 – 2017-03-25
\@tracklang@add@furlan: new . 73
\@tracklang@add@kurmanji: new 73
\@tracklang@err: fixed typo in 
\errhelp command name ...... 10
\@tracklang@pkgwarn: new ... 10
1.3.5 – 2018-02-21
General: check for \xpgloaded . 3
1.3.6 – 2018-05-13
\@tracklang@fullmatch: new . 51
\GetTrackedDialectFromLanguageTag: 
added
\TrackedDialectClosestSubMatch 
................................. 49
1.3.7 – 2019-08-31
General: corrected misspelt 
command ........................ 4, 6
1.3.8 – 2019-10-06
\@tracklang@add@nil: new ... 78
\@tracklang@add@undetermined: 
new ............................ 68
\@tracklang@prelangpkgcheck@hook: 
new ............................ 2
1.3.9 – 2019-11-11
\@TrackIfKnownLanguage: new . 46
\TrackIfKnownLanguage: new . 46
1.4 – 2019-11-30
General: input extra regions if 
provided ........................ 135
new ............................ 142
removed some spurious empty 
braces and commented out 
EOL ............................. 138
\@tracklang@firstoftwo: made 
long ............................. 10
\@tracklang@ifcsescaname: new 9
\@tracklang@input@extra@region@filelist: 
new ............................ 48
\@tracklang@input@extra@region@files: 
new ............................ 48
\@tracklang@input@extra@script@filelist: 
new ............................ 48
\@tracklang@input@extra@script@files: 
new ............................ 49
\@tracklang@redefhook: new ... 123
\@tracklang@secondoftwo: made 
long ............................. 10
Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the definition; numbers in roman refer to the pages where the entry is used.

Symbols
\- ........................................ 20
\@ .................................. 7, 21, 128, 135
\@@ .................................. 8, 9
\@TrackLangRequireDialect ....... 127
\@input .................................. 8
\@tracklang@checklocale ....... 12, 13
\@tracklang@ifalldigits .......... 39, 39
\@tracklang@split@hyphen ....... 20
\@tracklang@split@otherunderscore ........ 19, 20
\@tracklang@split@underscore  ........
\@TrackIfKnownLanguage ........ 46, 46
\@TrackLangAddToHook ........ 120, 127
\@TrackLangEnvAdditional ........
.................................. 19, 46, 48, 55
\@TrackLangEnvCodeSet ........
.................................. 18, 19, 21, 24, 52
\@TrackLangEnvFirstSubLang ....
.................................. 18, 44, 46, 53, 54
\@TrackLangEnvLang ........
.................................. 18, 19, 46, 47, 49, 50, 52–54
\@TrackLangEnvModifier ........
.................................. 18, 19, 46, 49, 52–54
\@TrackLangEnvScript ........
.................................. 19, 46, 47, 49, 50, 53–55
\@TrackLangEnvSubLang ........
.................................. 18, 44, 46, 49, 50, 53–55
\@TrackLangEnvTerritory ....
.................................. 18, 19, 46–49, 51–54
\@TrackLangEnvVariant ........
.................................. 19, 44, 46, 49, 50, 53–55
\@TrackLangRedefHook ........
.................................. 122
\@TrackLangRequestResource ..... 126, 127
\@TrackLangRequireDialect ....
.................................. 126, 127
\@TrackLangRequireResource ....
.................................. 125, 127
\@TrackLangRequireResourceOrDo  
.................................. 125, 127
\@TrackLanguageTag ........
.................................. 45, 45
\@after .................................. 22
\@before .................................. 22
\@declaredoptions ........
.................................. 2
\@empty .................................. 2
\@end@tracklang@hasfifthchar ...
\@end@tracklang@hasfourthchar...
\@end@tracklang@hasninthchar 
\@end@tracklang@hassecondchar
\@end@tracklang@hashthirdchar
\@filef@und ........ 37, 40, 41, 47, 54
\@for .................................. 3–6, 8, 9
\@fortmp ................................ 8
\@gobble ................................ 2
\@ifpackageloaded ........ 3, 5
\@ifundefined ........ 3–6
\@namedef .......................... 5, 6, 9
\@nameuse .................................. 3–6
\@nil ................................ 8, 22
\@nnil .................................. 8
\@noop@TrackLangRequireDialect 
\@org@tracklang@addlatex ........
\@tracklang@IfFileExists ........
.................................. 12
.................................. 85–118, 125, 126
\@tracklang@add ........ 22, 23, 24, 27, 48, 49
\@tracklang@add@Acadian ........
\@tracklang@add@Afrikaans ........
\@tracklang@add@AmericanEnglish ....
\@tracklang@add@Austrian ........
\@tracklang@add@Bahasa ........
\@tracklang@add@Basque ........
\@tracklang@add@Brazilian ........
\@tracklang@add@Breton ........
\@tracklang@add@BritishEnglish ....
\@tracklang@add@Bulgarian ........
\@tracklang@add@Canadien........
\@tracklang@add@Catalan ........
\@tracklang@add@Croatian ........
\@tracklang@add@Czech ........
\@tracklang@add@Danish ........
\@tracklang@add@Dutch ........
\texttt{@tracklang@test@if@file@exists@for@region} 104, 105, 107, 108, 110, 112
\texttt{@tracklang@test@if@file@exists@for@rootlang} 112–114
\texttt{@tracklang@test@if@file@exists@for@scriptregion} 86
\texttt{@tracklang@test@if@file@exists@for@tag} 85, 85
\texttt{@tracklang@test@if@file@exists@next} 86–114
\texttt{@tracklang@test@if@file@exists@noop} 85
\texttt{\@tracklang@tmp} 12, 24, 50, 51, 71, 120–122
\texttt{\@tracklang@track@locale} 45, 46, 53, 54
\texttt{\@tracklang@tryshellescape} 13
\texttt{\@tracklang@verbose} 11, 33–35, 85–118, 121–123
\texttt{\@tracklang@warn} 13–18, 35, 45, 48, 52, 54, 125, 127
\texttt{\@tracklang@x} 15, 16
\texttt{\_} 19, 20
\texttt{\^} 10, 11, 119
\texttt{\_} 19, 20

\texttt{\texttt{A}}
\texttt{\AddTrackedCountryIsoCode} ... 36, 56
\texttt{\AddTrackedDialect} ......... 23, 23, 53, 54, 69–71
\texttt{\AddTrackedIsoLanguage} ......... 27, 36, 54, 70, 71
\texttt{\AddTrackedLanguage} ......... 4, 6, 23, 56
\texttt{\AddTrackedLanguageIsoCodes} ......... 33, 53, 56, 69–71
\texttt{\AnyTrackedLanguages} ......... 84

\texttt{\texttt{B}}
\texttt{\bbll@loaded} ......... 5, 6
\texttt{\begingroup} ......... 15

\texttt{\texttt{C}}
\texttt{\captionsserbian} ......... 7
\texttt{\catcode} ......... 7, 19–21, 128, 135
\texttt{\closein} ......... 12
\texttt{\CurrentTrackedDialect} ......... 33–35, 120–123
\texttt{\CurrentTrackedDialectAdditional} ......... 34, 35

\texttt{\texttt{D}}
\texttt{\DeclareOption} ......... 2
\texttt{\directlua} ......... 13, 16, 17
\texttt{\do} ......... 3–6, 8, 33, 48–50, 84

\texttt{\texttt{E}}
\texttt{\end@tracklang@doifinlist} ......... 22
\texttt{\endcsname} ......... 2
\texttt{\endgroup} ......... 15, 16