tracklang v1.3.7: tracking language options

Nicola L. C. Talbot
http://www.dickimaw-books.com/

2019-08-31

Abstract

The tracklang package is provided for package developers who want a simple interface to find out which languages the user has requested through packages such as babel and polyglossia. *This package doesn't provide any translations.* Its purpose is simply to track which languages have been requested by the user. Generic TEX code is in `tracklang.tex` for non-LATEX users.

If the shell escape is enabled or `\directlua` is available, this package may also be used to query the `LC_ALL` or `LANG` environment variable (see Section 5). Windows users, who don't have the locale stored in environment variables, can use `texosquery` in combination with tracklang. (Similarly if `LC_ALL` or `LANG` don't contain sufficient information.) In order to use `texosquery` through the restricted shell escape, you must have at least Java 8 and set up `texosquery.cfg` appropriately. (See the `texosquery` manual for further details.)

The fundamental aim of this generic package is to be able to effectively say:

> The user (that is, the *document* author) wants to use dialects `xx-XX`, `yy-YY-Scrp`, etc in their document. Any packages used by their document that provide multilingual or region-dependent support should do whatever is required to activate the settings for those languages and regions (or warn the user that there's no support).

Naturally, this is only of use if the locale-sensitive packages use tracklang to pick up this information, which is entirely up to the package authors, but at the moment there's no standard method for packages to detect the required language and region. The aim of tracklang is to provide that method.

Related article: "Localisation of TEX documents: tracklang." TUGBoat, Volume 37 (2016), No. 3 (http://www.tug.org/TUGboat/tb37-3/tb117talbot.pdf).

Contents

List of Tables

1 Introduction

When I'm developing a package that provides multilingual support (for example, glossaries) it's cumbersome trying to work out if the user has requested translations for fixed text. This usually involves checking if babel or ngerman or translator or polyglossia has been loaded and, if so, what language settings have been used. The result can be a tangled mass of conditional code. The alternative is to tell users to add the language as a document class option, which they may or may not want to do, or to tell them to supply the language settings to every package they load that provides multilingual support, which users are even less likely to want to do.

The tracklang package tries to neaten this up by working out as much of this information as possible for you and providing a command that iterates through the loaded languages. This way, you can just iterate through the list of tracked languages and, for each language, either define the translations or warn the user that there's no translation for that language.

This package works best with ngerman or with recent versions of babel or when the language options are specified in the document class option list. It works fairly well with translator but will additionally assume the root language was also requested when a dialect is specified. So, for example,

```
\usepackage[british]{translator}
\usepackage{tracklang}
```

is equivalent to

```
\usepackage[british]{translator}
\usepackage[english,british]{tracklang}
```

This means that `\ForEachTrackedDialect` will iterate through the list "english,british" instead of just "british", which can result in some redundancy.

Unfortunately I can't work out how to pick up the language variant or script from polyglossia, so only the root languages are detected, which is suboptimal but at least provides some information. (polyglossia now provides `\xpg@loaded`, which tracklang uses to track the root languages, but the language variant command `\xpg@vloaded` only seems to be set when the language changes, which doesn't occur until the start of the document environment.)

If the ngerman package has been loaded, tracklang effectively does

```
\TrackPredefinedDialect{ngerman}
```

Similarly, if the german package has been loaded, tracklang effectively does

```
\TrackPredefinedDialect{german}
```

If any document class or package options are passed to tracklang, then tracklang won't bother checking for babel, translator, ngerman, german or polyglossia. So, if the above example is changed to:

```
\documentclass[british]{article}
\usepackage{translator}
\usepackage{tracklang}
```

then the dialect list will just consist of "british" rather than "english,british". This does, however, mean that if the user mixes class and package options, only the class options will be detected. For example:

```
\documentclass[british]{article}
\usepackage[french]{babel}
\usepackage{tracklang}
```

In this case, only the "british" option will be detected. The user can therefore use the document class option (or tracklang package option) to override the dialect and set the country code (where provided). For example:

```
\documentclass[es-MX]{article}
\usepackage[spanish]{babel}
\usepackage{tracklang}
```

This sets the dialect to "mexicanspanish" and the root language to "spanish".

Predefined dialects are listed in tables 1.1, 1.2 and 1.3. These may be passed in the document class options or used in `\TrackPredefinedDialect`, as illustrated above.

Section 2 provides brief examples of use for those who want a general overview before reading the more detailed sections. Section 3 describes generic commands for identifying the document languages. Section 4 is for package writers who want to add multilingual support to their package and need to know which settings the user has requested through language packages like babel. Section 5 is for developers of language definition packages who want to help other package writers to detect what languages have been requested.

Table 1.1: Predefined ISO Language-Region Dialects. (May be used as a package option or with `\TrackPredefinedDialect`.)

cy-GB	(GBwelsh)	de-AT	(austrian)
de-AT-1996	(naustrian)	de-BE	(belgiangerman)
de-CH	(swissgerman)	de-CH-1996	(nswissgerman)
de-DE	(germanDE)	de-DE-1996	(ngermanDE)
en-AU	(australian)	en-CA	(canadian)
en-GB	(british)	en-GG	(guernseyenglish)
en-IE	(IEenglish)	en-IM	(isleofmanenglish)
en-JE	(jerseyenglish)	en-MT	(maltaenglish)
en-NZ	(newzealand)	en-US	(american)
es-AR	(argentinespanish)	es-BO	(bolivianspanish)
es-CL	(chilianspanish)	es-CO	(columbianspanish)
es-CR	(costaricanspanish)	es-CU	(cubanspanish)
es-DO	(dominicanspanish)	es-EC	(ecudorianspanish)
es-ES	(spainspanish)	es-GT	(guatemalanspanish)
es-HN	(honduranspanish)	es-MX	(mexicanspanish)
es-NI	(nicaraguanspanish)	es-PA	(panamaspanish)
es-PE	(peruvianspanish)	es-PR	(puertoricospanish)
es-PY	(paraguayspanish)	es-SV	(elsalvadorspanish)
es-UY	(uruguayspanish)	es-VE	(venezuelanspanish)
fr-BE	(belgique)	fr-CA	(canadien)
fr-CH	(swissfrench)	fr-FR	(france)
fr-GG	(guernseyfrench)	fr-JE	(jerseyfrench)
ga-GB	(GBirish)	ga-IE	(IEirish)
gd-GB	(GBscottish)	hr-HR	(croatia)
hu-HU	(hungarian)	id-IN	(bahasa)
it-CH	(swissitalian)	it-HR	(istriacountyitalian)
it-IT	(italy)	it-SI	(sloveneistriaitalian)
it-SM	(sanmarino)	it-VA	(vatican)
ms-MY	(malay)	mt-MT	(maltamaltese)
nl-BE	(flemish)	nl-NL	(netherlands)
pt-BR	(brazilian)	pt-PT	(portugal)
rm-CH	(swissromansh)	sl-SI	(slovenia)

Other combinations need to be set with `\TrackLocale` or `\TrackLanguageTag`.

Table 1.2: Predefined Root Languages. (†Has an associated territory.) The corresponding language tag obtained with `\GetTrackedLanguageTag{`⟨*dialect*⟩`}` is shown in parentheses.

abkhaz (ab)	afar (aa)	afrikaans (af)
akan (ak)	albanian (sq)	amharic† (am-ET)
anglosaxon (ang)	apache (apa)	arabic (ar)
aragonese† (an-ES)	armenian (hy)	assamese (as)
asturian (ast)	avaric (av)	avestan (ae)
aymara (ay)	azerbaijani (az)	bahasai† (id-IN)
bahasam† (ms-MY)	bambara† (bm-ML)	bashkir (ba)
basque (eu)	belarusian (be)	bengali (bn)
berber (ber)	bihari (bh)	bislama† (bi-VU)
bokmal† (nb-NO)	bosnian (bs)	breton† (br-FR)
bulgarian (bg)	burmese (my)	catalan (ca)
chamorro (ch)	chechen (ce)	chichewa (ny)
chinese (zh)	churchslavonic (cu)	chuvash† (cv-RU)
coptic (cop)	cornish† (kw-GB)	corsican (co)
cree (cr)	croatian (hr)	czech (cs)
danish (da)	divehi† (dv-MV)	dutch (nl)
dzongkha† (dz-BT)	easternpunjabi† (pa-IN)	english (en)
esperanto (eo)	estonian (et)	ewe (ee)
faroese (fo)	farsi (fa)	fijian† (fj-FJ)
finnish (fi)	french (fr)	friulan† (fur-IT)
fula (ff)	galician (gl)	ganda† (lg-UG)
georgian (ka)	german (de)	greek (el)
guarani (gn)	gujarati (gu)	haitian† (ht-HT)
hausa (ha)	hebrew (he)	herero (hz)
hindi (hi)	hirimotu† (ho-PG)	icelandic† (is-IS)
ido (io)	igbo (ig)	interlingua (ia)
interlingue (ie)	inuktitut (iu)	inupiaq (ik)
irish (ga)	italian (it)	japanese (ja)
javanese (jv)	kalaallisut (kl)	kannada† (kn-IN)
kanuri (kr)	kashmiri† (ks-IN)	kazakh (kk)
khmer (km)	kikuyu (ki)	kinyarwanda (rw)
kirundi (rn)	komi† (kv-RU)	kongo (kg)
korean (ko)	kurdish (ku)	kwanyama (kj)
kyrgyz (ky)	lao (lo)	latin (la)
latvian (lv)	limburgish (li)	lingala (ln)
lithuanian (lt)	lsorbian† (dsb-DE)	lubakatanga† (lu-CD)
luxembourgish (lb)	macedonian (mk)	magyar (hu)
malagasy (mg)	malayalam† (ml-IN)	maltese (mt)
manx† (gv-IM)	maori† (mi-NZ)	marathi† (mr-IN)
marshallese† (mh-MH)	mongolian (mn)	nauruan† (na-NR)

Table 1.2: Predefined Root Languages Continued.

navajo† (nv-US)
nko (nqo)
northernsotho (nso)
occitan (oc)
oromo (om)
pashto (ps)
portuges (pt)
romansh† (rm-CH)
samoan (sm)
sardinian† (sc-IT)
shona (sn)
slovak (sk)
southernndebele† (nr-ZA)
sudanese (su)
swedish (sv)
tahitian† (ty-PF)
tamil (ta)
thai† (th-TH)
tonga† (to-TO)
turkish (tr)
ukrainian† (uk-UA)
uyghur† (ug-CN)
vietnamese (vi)
welsh (cy)
xhosa (xh)
zhuang† (za-CN)
ndonga (ng)
norsk (no)
nuosu† (ii-CN)
ojibwe (oj)
ossetian (os)
piedmontese† (pms-IT)
quechua (qu)
russian (ru)
sango (sg)
scottish (gd)
sindhi (sd)
slovene (sl)
southernsotho (st)
swahili (sw)
syriac (syr)
tai (tai)
tatar (tt)
tibetan (bo)
tsonga (ts)
turkmen (tk)
urdu (ur)
uzbek (uz)
volapuk (vo)
westernfrisian† (fy-NL)
yiddish (yi)
zulu (zu)
nepali (ne)
northernndebele (nd)
nynorsk† (nn-NO)
oriya (or)
pali (pi)
polish (pl)
romanian (ro)
samin (se)
sanskrit (sa)
serbian (sr)
sinhalese† (si-LK)
somali (so)
spanish (es)
swati (ss)
tagalog† (tl-PH)
tajik (tg)
telugu† (te-IN)
tigrinya (ti)
tswana (tn)
twi† (tw-GH)
usorbian† (hsb-DE)
venda† (ve-ZA)
walloon (wa)
wolof (wo)
yoruba (yo)

Table 1.3: Predefined Non-ISO Dialects. (†Has an associated territory.) The corresponding language tag obtained with `\GetTrackedLanguageTag{`⟨*dialect*⟩`}` is shown in parentheses. If the dialect has a corresponding mapping for the closest matching non-root language `\caption...` or `\date...`, this is also include after the tag following a slash.

acadian (fr)
argentinespanish† (es-AR)
austrian† (de-AT)
belgiangerman† (de-BE)
bolivianspanish† (es-BO)
brazilian† (pt-BR)
canadian† (en-CA)
chilianspanish† (es-CL)
costaricanspanish† (es-CR)
cubanspanish† (es-CU)
deutsch (de)
ecudorianspanish† (es-EC)
flemish† (nl-BE)
france† (fr-FR)
friulano† (fur-IT)
furlan† (fur-IT)
gaelic (gd)
GBirish† (ga-GB)
GBwelsh† (cy-GB)
germanDE† (de-DE)
guernseyenglish† (en-GG / british)
honduranspanish† (es-HN)
IEenglish† (en-IE / british)
indon† (id-IN)
isleofmanenglish† (en-IM / british)
istriacountyitalian† (it-HR)
jerseyenglish† (en-JE / british)
kurmanji (ku)
lowersorbian† (dsb-DE)
maltaenglish† (en-MT / british)
mexicanspanish† (es-MX)
naustrian† (de-AT-1996)
netherlands† (nl-NL)
ngerman (de-1996)
ngermanDE† (de-DE-1996 / ngerman)
norwegian† (no-NO)
panamaspanish† (es-PA)
persian (fa)
american† (en-US)
australian† (en-AU)
bahasa† (id-IN)
belgique† (fr-BE)
brazil† (pt-BR)
british† (en-GB)
canadien† (fr-CA)
columbianspanish† (es-CO)
croatia† (hr-HR)
cymraeg (cy)
dominicanspanish† (es-DO)
elsalvadorspanish† (es-SV)
francais (fr)
frenchb (fr)
friulian† (fur-IT)
gaeilge (ga)
galicien (gl)
GBscottish† (gd-GB)
germanb (de)
guatemalanspanish† (es-GT)
guernseyfrench† (fr-GG)
hungarian† (hu-HU)
IEirish† (ga-IE)
indonesian† (id-IN)
istriacountycroatian† (hr-HR)
italy† (it-IT)
jerseyfrench† (fr-JE)
latein (la)
malay† (ms-MY)
maltamaltese† (mt-MT)
meyalu† (ms-MY)
nbelgiangerman† (de-BE-1996 / ngerman)
newzealand† (en-NZ)
ngermanb (de-1996 / ngerman)
nicaraguanspanish† (es-NI)
nswissgerman† (de-CH-1996 / ngerman)
paraguayspanish† (es-PY)
peruvianspanish† (es-PE)

Table 1.3: Predefined Non-ISO Dialects Continued

piemonteis† (pms-IT)
polutonikogreek (el)
portuguese (pt)
romanche (rm)
rumantsch (rm)
sanmarino† (it-SM)
serbianl (sr-Latn)
sloveneistriaslovenian† (sl-SI / slovenian)
slovenian (sl)
swissfrench† (fr-CH)
swissitalian† (it-CH)
UKenglish† (en-GB)
ukraineb† (uk-UA)
uruguayspanish† (es-UY)
valencian (ca)
vatican† (it-VA)
polutoniko (el)
portugal† (pt-PT)
puertoricospanish† (es-PR)
romansch (rm)
russianb (ru)
serbianc (sr-Cyrl)
sloveneistriaitalian† (it-SI)
slovenia† (sl-SI / slovenian)
spainspanish† (es-ES)
swissgerman† (de-CH)
swissromansh† (rm-CH)
ukraine† (uk-UA)
uppersorbian† (hsb-DE)
USenglish† (en-US)
valencien (ca)
venezuelanspanish† (es-VE)

2 Summary of Use

There are three levels of use:

1. document level (code used by document authors);
2. locale-sensitive package level (code for package authors who need to know what languages or locale the document is using, such as glossaries to translate commands like `\descriptionname` or datetime2 to provide localised formats or time zone information);
3. language set-up level (code for packages that set up the document languages, such as babel or polyglossia).

2.1 Document Level

2.1.1 Generic TEX

Unix user wants the locale information picked up from the locale environment variable:

```
\input tracklang % v1.3
\TrackLangFromEnv
% load packages that use tracklang for localisation
```

Windows user wants the locale information picked up from the operating system:

```
\input texosquery
\input tracklang %v1.3
\TrackLangFromEnv
% load packages that use tracklang for localisation
```

Or (texosquery v1.2 currently pending)

```
\input texosquery % v1.2
\input tracklang % v1.3

\TeXOSQueryLangTag{\langtag}
\TrackLanguageTag{\langtag}
% load packages that use tracklang for localisation
```

Anticipate the release of texosquery v1.2:

```
\input texosquery
\input tracklang % v1.3

\ifx\TeXOSQueryLangTag\undefined
 \TrackLangFromEnv
\else
 \TeXOSQueryLangTag{\langtag}
 \TrackLanguageTag{\langtag}
\fi
% load packages that use tracklang for localisation
```

User is writing in Italy in Armenian with a Latin script and the arevela variant:

```
\input tracklang % v1.3
\TrackLanguageTag{hy-Latn-IT-arevela}
% load packages that use tracklang for localisation
```

User is writing in English in the UK:

```
\input tracklang
\TrackPredefinedDialect{british}
% load packages that use tracklang for localisation
```

Find out information about the current language (supplied in `\languagename`):

```
\SetCurrentTrackedDialect{\languagename}
Dialect: \CurrentTrackedDialect.
Language: \CurrentTrackedLanguage.
ISO Code: \CurrentTrackedIsoCode.
Region: \CurrentTrackedRegion.
Modifier: \CurrentTrackedDialectModifier.
Variant: \CurrentTrackedDialectVariant.
Script: \CurrentTrackedDialectScript.
Sub-Lang: \CurrentTrackedDialectSubLang.
Additional: \CurrentTrackedDialectAdditional.
Language Tag: \CurrentTrackedLanguageTag.
```

Additional information about the script can be obtained by also loading tracklang-scripts:

```
\input tracklang-scripts
```

The name, numeric code and direction can now be obtained:

```
Name: \TrackLangScriptAlphaToName{\CurrentTrackedDialectScript}.
Numeric: \TrackLangScriptAlphaToNumeric{\CurrentTrackedDialectScript}.
Dir: \TrackLangScriptAlphaToDir{\CurrentTrackedDialectScript}.
```

Test for a specific script:

```
Latin?
\ifx\CurrentTrackedDialectScript\TrackLangScriptLatn
 Yes
\else
 No
\fi
```

2.1.2 LaTeX

For babel users where the supplied babel dialect label is sufficient, there's no need to do anything special:

```
\documentclass[british,canadien]{article}
\usepackage[T1]{fontenc}
\usepackage{babel}
% load packages that use tracklang for localisation
```

If the region is important but there's no babel dialect that represents it, there are several options.

Use the class options recognised by tracklang and the root language labels when loading babel:

```
\documentclass[en-IE,ga-IE]{article}
\usepackage[english,irish]{babel}
% load packages that use tracklang for localisation
```

This method is needed for polyglossia where the regional information is required. For example

```
\documentclass[en-GB]{article}
\usepackage{polyglossia}
\setmainlanguage[variant=uk]{english}
% load packages that use tracklang for localisation
```

Another method with babel is to use `\TrackLanguageTag` and map the new dialect label to babel's nearest matching label:

```
\documentclass{article}

\usepackage{tracklang}% v1.3
\TrackLanguageTag{en-MT}
\SetTrackedDialectLabelMap{\TrackLangLastTrackedDialect}{UKenglish}

\usepackage[UKenglish]{babel}
% load packages that use tracklang for localisation
```

This ensures that the `\captionsUKenglish` hook is detected by the localisation packages. This mapping isn't needed for polyglossia as the caption hooks use the root language label. This mapping also isn't needed if `british` is used instead of `UKenglish` since the en-MT (maltaenglish) predefined dialect automatically sets up a mapping to `british`. (The default mappings are shown in table 1.3.)

2.2 Locale-Sensitive Packages

Let's suppose you are developing a package called `mypackage.sty` or `mypackage.tex` and you want to find out what languages the document author has requested.

Generic use:

```
\input tracklang
```

(Most of the commands used in this section require at least tracklang version 1.3.) Note that `tracklang.tex` has a check to determine if it's already been loaded, so you don't need to worry about that.

LaTeX use:

```
\RequirePackage{tracklang}[2016/10/07]% at least v1.3
```

This will picked up any language options supplied in the document class options and will also detect if babel or polyglossia have been loaded.

(LaTeX) If you want to allow the user to set the locale in the package options:

```
\DeclareOption*{\TrackLanguageTag{\CurrentOption}}
```

This means the user can do, say,

```
\usepackage[hy-Latn-IT-arevela]{mypackage}
```

The rest of the example package in this section uses generic code.

If you want to fetch the locale information from the operating system when the user hasn't requested a language:

```
\AnyTrackedLanguages
{}
{% fetch locale information from the operating system
  \ifx\TeXOSQueryLangTag\undefined
    % texosquery v1.2 not available
    \TrackLangFromEnv
  \else
    % texosquery v1.2 available
    \TeXOSQueryLangTag{\langtag}
    \TrackLanguageTag{\langtag}
  \fi
}
```

Set up the defaults if necessary:

```
\def\fooname{Foo}
\def\barname{Bar}
```

Now load the resource files:

```
\AnyTrackedLanguages
{%
  \ForEachTrackedDialect{\thisdialect}{%
    \TrackLangRequireDialect{mypackage}{\thisdialect}%
  }%
}
{}% no tracked languages, default already set up
```

Each resource file has the naming scheme ⟨*prefix*⟩-⟨*tag*⟩`.ldf`. In this example, the ⟨*prefix*⟩ is `mypackage`. The ⟨*tag*⟩ may be the language or dialect label (for example, `english` or `british`) or a combination of the ISO language and region codes (for example, `en-GB` or `en` or `GB`).

The simplest scheme is to use the root language label (not the dialect label) for the base language settings and use the ISO codes for regional support.

For example, the file `mypackage-english.ldf`:

```
\TrackLangProvidesResource{english}[2016/10/06 v1.0]% identify this file

\TrackLangAddToCaptions{%
  \def\fooname{Foo}%
  \def\barname{Bar}%
}
```

This sets up appropriate the `\captions...` hook (if it's found). For other hooks, such as `\date...`, use `\TrackLangAddToHook{`⟨*code*⟩`}{`⟨*hook type*⟩`}` instead.

Here's an example for a language with different writing systems. The resource file for Serbian `mypackage-serbian.ldf`:

```
\TrackLangProvidesResource{serbian}[2016/10/06 v1.0]% identify file

\TrackLangRequestResource{serbian-\CurrentTrackedScript}
{}% file not found, do something sensible here
```

The file `mypackage-serbian-Latn.ldf` sets up the Latin script:

```
\TrackLangProvidesResource{serbian-Latn}[2016/10/06 v1.0]

\TrackLangAddToCaptions{%
  \def\fooname{...}% provide appropriate Latin translations
  \def\barname{...}%
}
```

The file `mypackage-serbian-Cyrl.ldf` sets up the Cyrllic script:

```
\TrackLangProvidesResource{serbian-Cyrl}[2016/10/06 v1.0]

\TrackLangAddToCaptions{%
  \def\fooname{...}% provide appropriate Cyrllic translations
  \def\barname{...}%
}
```

2.3 Language Packages

Let's suppose now you're the developer of a package that sets up the language, hyphenation patterns and so on. It would be really helpful to the locale-sensitive packages in Section 2.2 to know what languages the document author has requested. You can use the tracklang package to identify this information by tracking the requested localisation, so that other packages can have a consistent way of querying it.

Generic use:

```
\input tracklang
```

Alternative LaTeX use:

```
\RequirePackage{tracklang}[2016/10/07] % v1.3
```

Unlike `\input`, `\RequirePackage` will allow tracklang to pick up the document class options, but using `\RequirePackage` will also trigger the tests for known language packages. (If you want to find out if tracklang has already been loaded and locales have already been tracked, you can use the same code as in the previous section.)

When a user requests a particular language through your package, the simplest way of letting tracklang know about it is to use `\TrackPredefinedDialect` or `\TrackLanguageTag`. For example, if the user requests `british`, that's a predefined dialect so you can just do:

```
\TrackPredefinedDialect{british}
```

Alternatively

```
\TrackLanguageTag{en-GB}
```

If your package uses caption hooks, then you can set up a mapping between tracklang's internal dialect label and your caption label. For example, let's suppose the closest match to English used in Malta (`en-MT`) is the dialect `UKenglish` (for example, the date format is similar between GB and MT):

```
\TrackLanguageTag{en-MT}
```

```
\SetTrackedDialectLabelMap{\TrackLangLastTrackedDialect}{UKenglish}
\def\captionsUKenglish{%
  \def\contentsname{Contents}%
  %...
}
```

(The predefined maltaenglish option provided by tracklang automatically sets the mapping to `british`, but the above method will change that mapping to `UKenglish`.)

This now means that `\TrackLangAddToCaptions` command used at the end of Section 2.2 above can find your caption hook. You don't need the map if your dialect label is the same as tracklang's root language label for that locale. For example:

```
\TrackLanguageTag{en-MT}
\def\captionsenglish{%
  \def\contentsname{Contents}%
  %...
}
```

When the user switches language through commands like `\selectlanguage` it would be useful to also use `\SetCurrentTrackedDialect{`⟨*dialect*⟩`}` to make it easier for the document author or locale-sensitive packages to pick up the current locale. The ⟨*dialect*⟩ argument may be tracklang's internal dialect label or the dialect label you assigned with `\SetTrackedDialectLabelMap`. It may also be the root language label, in which case tracklang will search for the last dialect to be tracked with that language. For example:

```
\def\selectlanguage#1{%
  % set up hyphenation patterns etc
  \SetCurrentTrackedDialect{#1}%
}
```

See the example in Section 2.1.

3 Generic Use

For plain TEX you can input `tracklang.tex`:

```
\input tracklang
```

or for TEX formats that have an argument form for `\input`:

```
\input{tracklang}
```

As from version 1.3, you don't need to change the category code of `@` before loading `tracklang.tex` as it will automatically be changed to 11 and switched back at the end (if required).

The LATEX package `tracklang.sty` inputs the generic TEX code in `tracklang.tex`, but before it does so it defines

```
\@tracklang@declareoption{⟨name⟩}
```

to

```
\DeclareOption{⟨name⟩}{\TrackPredefinedDialect{⟨name⟩}}
```

This means that all the predefined languages and dialects (tables 1.1, 1.2 and 1.3) automatically become package options, so the tracklang package can pick up document class options and add them to tracklang's internal list of tracked document languages.

If you're not using LATEX, this option isn't available (although you could redefine the internal command `\@tracklang@declareoption` to use something analogous to `\DeclareOption`). Instead, the document languages need to be explicitly identified (using any of the following commands) so that tracklang knows about them.

`\TrackPredefinedDialect`

```
\TrackPredefinedDialect{⟨dialect label⟩}
```

This will add the predefined dialect and its associated ISO codes to the list of tracked document languages. The ⟨*dialect label*⟩ may be any of those listed in tables 1.1, 1.2 and 1.3. (See also Section 6.2.5 and Section 6.2.6.)

For example:

```
\input tracklang
\TrackPredefinedDialect{british}
```

is the Plain TEX alternative to

```
\documentclass[british]{article}
\usepackage{tracklang}
```

Note that it's impractical to define every possible language and region combination as it would significantly slow the time taken to load tracklang so, after version 1.3, I don't intend adding any new predefined dialects. As from version 1.3, if you want to track a dialect that's not predefined by tracklang, then you can use:

\TrackLocale

\TrackLocale{⟨*locale*⟩}

If ⟨*locale*⟩ is a recognised dialect, this is equivalent to using \TrackPredefinedDialect, otherwise ⟨*locale*⟩ needs to be in one the following formats:

- ⟨*ISO lang*⟩
- ⟨*ISO lang*⟩@⟨*modifier*⟩
- ⟨*ISO lang*⟩-⟨*ISO country*⟩
- ⟨*ISO lang*⟩-⟨*ISO country*⟩@⟨*modifier*⟩

where ⟨*ISO lang*⟩ is the ISO 639-1 or 639-2 code identifying the language (lower case), ⟨*ISO country*⟩ is the 3166-1 ISO code identifying the territory (upper case) and ⟨*modifier*⟩ is the modifier or variant. The hyphen may be replaced by an underscore character. Code set information in the form .⟨*codeset*⟩ may optionally appear before the modifier. For example, de-DE.utf8@new (modifier is new) or en-GB.utf8 (modifier is missing). The codeset will be ignored if present, but it won't interfere with the parsing.

For example:

```
\TrackLocale{de-NA@new}
```

indicates German in Namibia using the new spelling.

If a language has different "T" and "B" ISO 639-2 codes, then the "T" form should be used. (So for the above example, deu may be used instead of de, but ger won't be recognised.)

Alternatively, you can use

\TrackLanguageTag

\TrackLanguageTag{⟨*tag*⟩}

where ⟨*tag*⟩ is a regular, well-formed language tag or a recognised dialect label. (Irregular grandfather tags aren't recognised.) This command will fully expand ⟨*tag*⟩. A warning is issued if the tag is empty. For example:

```
\TrackLanguageTag{hy-Latn-IT-arevela}
```

If ⟨*tag*⟩ contains a sub-language tag, this will be set as the 639-3 code for the *dialect* label. Note that this is different to the root language codes which are set using the language label. For example

```
\TrackLanguageTag{zh-cmn-Hans-CN}
```

creates a new dialect with the label zhcmnHansCN. The root language chinese has the ISO 639-1 code zh and the dialect zhcmnHansCN has the ISO 639-3 code cmn.

```
ISO 639-1: \TrackedIsoCodeFromLanguage{639-1}{chinese}.
ISO 639-3: \TrackedIsoCodeFromLanguage{639-3}{zhcmnHansCN}.
```

Version 1.2 (currently pending) of texosquery will have a new command \TeXOSQueryLangTag, which may be used to fetch the operating system's regional information as a language tag. These commands can be used as follows:

```
\input tracklang % v1.3
\input texosquery % v1.2

\TeXOSQueryLangTag{\langtag}
\TrackLanguageTag{\langtag}
```

(If the shell escape is disabled, \langtag will be empty, which will trigger a warning but no errors.)

Some of the predefined root language options listed in table 1.2 have an associated region (denoted by †). If \TrackLocale is used with just the language ISO code, no region is tracked for that language. For example

```
\TrackLocale{manx}
```

will track the "IM" ISO 3166-1 code but

```
\TrackLocale{gv}
```

won't track the region. Similarly for \TrackLanguageTag.

(New to version 1.3.) There's a similar command to \TrackLocale that doesn't take an argument:

\TrackLangFromEnv

\TrackLangFromEnv

If the shell escape has been enabled or \directlua is available, this will try to get the language information from the system environment variables LC_ALL or LANG and, if successful, track that.

Since tracklang is neither able to look up the POSIX locale tables nor interpret file locales, if the result is C or POSIX or starts with a forward slash / then the locale value is treated as empty.

Not all operating systems use environment variables for the system locale information. For example, Windows stores the locale information in the registry. In which case, consider using texosquery.

If the operating system locale can't be obtained from environment variables, then tracklang will use \TeXOSQueryLocale as a fallback if texosquery has been loaded. Since texosquery requires both the shell escape and the Java runtime environment, tracklang doesn't automatically load it.

Plain TEX example:

```
\input texosquery
\input tracklang
\TrackLangFromEnv
```

with `etex --shell-escape` ⟨*filename*⟩

LATEX example:

```
\usepackage{texosquery}
\usepackage{tracklang}
\TrackLangFromEnv
```

with `latex --shell-escape` ⟨*filename*⟩

If the locale can't be determined, there will be warning messages. These can be suppressed using

\TrackLangShowWarningsfalse

```
\TrackLangShowWarningsfalse
```

or switched back on again using

\TrackLangShowWarningstrue

```
\TrackLangShowWarningstrue
```

For example, I have the environment variable `LANG` set to `en_GB.utf8` on my Linux system so instead of

```
\TrackPredefinedDialect{british}
```

I can use

```
\TrackLangFromEnv
```

With LATEX documents I can do

```
\documentclass{article}
\usepackage{tracklang}
\TrackLangFromEnv
```

However, this only helps subsequently loaded packages that use tracklang to determine the required regional settings. For example:

```
\documentclass{article}
\usepackage{tracklang}
\TrackLangFromEnv
\usepackage[useregional]{datetime2}
```

In my case, with `LANG` set to `en_GB.utf8` and shell escape enabled, this automatically switches on the `en-GB` date style. Naturally this doesn't help locale-sensitive packages that don't use tracklang.

The `\TrackLangFromEnv` command also incidentally sets

\TrackLangEnv

```
\TrackLangEnv
```

to the value of the environment variable or empty if the query was unsuccessful (for example, the shell escape is unavailable).

If \TrackLangEnv is already defined before \TrackLangFromEnv is used, then the environment variable won't be queried and the value of \TrackLangEnv will be parsed instead.

The parser which splits the locale string into its component parts first tries splitting on the underscore _ with its usual category code 8, then tries splitting on a hyphen - with category code 12, and then tries splitting on the underscore _ with category code 12.

For example:

```
\def\TrackLangEnv{en-GB}
\TrackLangFromEnv
```

This doesn't perform a shell escape since \TrackLangEnv is already defined. In this case, you may just as well use

```
\TrackLocale{en-GB}
```

(unless you happen to additionally require the component commands that are set by \TrackLangFromEnv, see below.)

If the shell escape is unavailable (for example, your TEX installation prohibits it), you can set this value when you invoke TEX. For example, if the document file is called myDoc.tex (and it's in Plain TEX):

```
tex "\\def\TrackLangEnv{$LANG}\\input myDoc"
```

The \TrackLangFromEnv command also happens to store the component parts of the environment variable value in the following commands. (These aren't provided by \TrackLocale.)

The language code is stored in:

\TrackLangEnvLang — \TrackLangEnvLang

The territory (if present) is stored in:

\TrackLangEnvTerritory — \TrackLangEnvTerritory

(Defined to empty if not present.)

The codeset (if present) is stored in:

\TrackLangEnvCodeSet — \TrackLangEnvCodeSet

(Defined to empty if not present.)

The modifier (if present) is stored in:

\TrackLangEnvModifier — \TrackLangEnvModifier

(Defined to empty if not present.)

If you want to query the language environment, but don't want to track the result, you can just use:

\TrackLangQueryEnv

```
\TrackLangQueryEnv
```

This only tries to fetch the value of the language environment variable (and use texosquery as a fallback, if it has been loaded). It doesn't try to parse the result. The result is stored in \TrackLangEnv (empty if unsuccessful). Unlike \TrackLangFromEnv, this doesn't check if \TrackLangEnv already exists. A warning will occur if the shell escape is unavailable. For systems that store the locale information in environment variables, this is more efficient than using texosquery's \TeXOSQueryLocale command (which is what's used as the fallback).

The above queries LC_ALL and, if that is unsuccessful, then queries LANG (before optionally falling back on texosquery). If you want another environment variable tried after LC_ALL and before LANG, you can instead use:

\TrackLangQueryOtherEnv

```
\TrackLangQueryOtherEnv{⟨name⟩}
```

For example, to also query LC_MONETARY:

```
\TrackLangQueryOtherEnv{LC_MONETARY}
```

Since this sets \TrackLangEnv, you can use it before \TrackLangFromEnv. For example:

```
\TrackLangQueryOtherEnv{LC_MONETARY}
\TrackLangFromEnv
```

Remember that if you only want to do the shell escape if \TrackLangEnv hasn't already been defined, you can test for this first:

```
\ifx\TrackLangEnv\undefined
  \TrackLangQueryOtherEnv{LC_MONETARY}
\fi
\TrackLangFromEnv
```

It's also possible to just parse the value of \TrackLangEnv without tracking the result using:

\TrackLangParseFromEnv

```
\TrackLangParseFromEnv
```

This is like \TrackLangFromEnv but assumes that \TrackLangEnv has already been set and doesn't track the result. The component parts are stored as for \TrackLangFromEnv.

Example (Plain TEX):

```
\input tracklang

\def\TrackLangEnv{fr-BE.utf8@euro}

\TrackLangParseFromEnv

Language: \TrackLangEnvLang.
Territory: \TrackLangEnvTerritory.
Codeset: \TrackLangEnvCodeSet.
Modifier: \TrackLangEnvModifier.
Any tracked languages? \AnyTrackedLanguages{Yes}{No}.
```

This produces:

Language: fr. Territory: BE. Codeset: utf8. Modifier: euro. Any tracked languages? No.

Compare this with:

```
\input tracklang

\def\TrackLangEnv{fr-BE.utf8@euro}

\TrackLangFromEnv

Language: \TrackLangEnvLang.
Territory: \TrackLangEnvTerritory.
Codeset: \TrackLangEnvCodeSet.
Modifier: \TrackLangEnvModifier.
Any tracked languages? \AnyTrackedLanguages{Yes}{No}.
Tracked dialect(s):%
\ForEachTrackedDialect{\thisdialect}{\space\thisdialect}.
```

This produces:

Language: fr. Territory: BE. Codeset: utf8. Modifier: euro. Any tracked languages? Yes. Tracked dialect(s): belgique.

If `\TrackLangFromEnv` doesn't recognise the given language and territory combination, it will define a new dialect and add that.

For example, tracklang doesn't recognise `en-BE`, so the sample document below defines a new dialect labelled `enBEeuro`:

```
\input tracklang

\def\TrackLangEnv{en-BE.utf8@euro}

\TrackLangFromEnv

Language: \TrackLangEnvLang.
Territory: \TrackLangEnvTerritory.
Codeset: \TrackLangEnvCodeSet.
Modifier: \TrackLangEnvModifier.
Any tracked languages? \AnyTrackedLanguages{Yes}{No}.
Tracked dialect(s):%
\ForEachTrackedDialect{\thisdialect}{\space\thisdialect}.
```

This now produces:

Language: en. Territory: BE. Codeset: utf8. Modifier: euro. Any tracked languages? Yes. Tracked dialect(s): enBEeuro.

4 Detecting the User's Requested Languages

The tracklang package tries to track the loaded languages and the option names used to identify those languages. For want of a better term, the language option names are referred to as dialects even if they're only a synonym for the language rather than an actual dialect. For example, if the user has requested `british`, the *root language* label is `english` and the dialect is `british`, whereas if the user requested `UKenglish`, the root language label is `english` and the dialect is `UKenglish`. The exceptions to this are the tracklang package options that have been specified in the form ⟨*iso lang*⟩-⟨*iso country*⟩ (listed in table 1.2). For example, the package option `en-GB` behaves as though the user requested the package option `british`.

If `\TrackLocale` or `\TrackLangFromEnv` are used and the locale isn't recognised a new dialect is created with the label formed from the ISO codes (and modifier, if present). Similarly for `\TrackLanguageTag` a new dialect is created with a label that's essentially the language tag without the hyphen separators. For example,

```
\TrackLocale{xx-YY}
```

will add a new dialect with the label `xxYY`,

```
\TrackLocale{xx-YY@mod}
```

will add a new dialect with the label `xxYYmod` and

```
\TrackLanguageTag{xx-YY-Latn}
```

will add a new dialect with the label `xxYYLatn`.

If `\TrackLocale` or `\TrackLangFromEnv` find a modifier, the value will be sanitized to allow it to be used as a label. If the modifier is set explicitly using `\SetTrackedDialectModifier`, no sanitization is performed.

In addition to the root language label and the dialect identifier, many of the language options also have corresponding ISO codes. In most cases there is an ISO 639-1 or an ISO 639-2 code (or both), and in some cases there is an ISO 3166-1 code identifying the dialect region. Where a language has both a "T" and a "B" ISO 639-2 code, the "T" version is assumed.

When the tracklang LaTeX package is loaded, it first attempts to find the language options through the package options supplied to tracklang. This means that any languages that have been supplied in the document class options should get identified (provided that the document class has used the standard option declaration mechanism). If no languages have been

supplied in this way, tracklang then attempts to identify any babel language options and failing that it will try the translator language options. It will then check if ngerman or polyglossia have been loaded.

Each identified language and dialect is added to the *tracked language* and *tracked dialect* lists. Note that the tracked language and tracked dialect are labels rather than proper nouns. If a dialect label is identical to its root language label, the label will appear in both lists.

You can check whether or not any languages have been detected using:

\AnyTrackedLanguages

```
\AnyTrackedLanguages{⟨true part⟩}{⟨false part⟩}
```

This will do ⟨*true part*⟩ if one or more languages have been detected otherwise it will do ⟨*false part*⟩. (Each detected dialect will automatically have the root language label added to the tracked language list, if it's not already present.)

If you want to find out if any of the tracked dialects matches a particular language tag, you can use:

\GetTrackedDialectFromLanguageTag

```
\GetTrackedDialectFromLanguageTag{⟨tag⟩}{⟨cs⟩}
```

If successful, the supplied control sequence ⟨*cs*⟩ is set to the dialect label, otherwise ⟨*cs*⟩ is set to empty. The test is for an exact match on the root language, script, sub-language, variant and region. The control sequence ⟨*cs*⟩ will be empty if none of the tracked dialects matches all five of those elements. (If the script isn't given explicitly, the default for that language is assumed.) In the event that ⟨*cs*⟩ is empty, you can now (as from v1.3.6) get the closest match with

\TrackedDialectClosestSubMatch

```
\TrackedDialectClosestSubMatch
```

(which is set by `\GetTrackedDialectFromLanguageTag`). This will be empty if no tracked dialects match on the root language or if there's a tracked dialect label that exactly matches the label formed by concatenating the language code, sub-language, script, region, modifier and variant.

For example (Plain TEX):

```
\input tracklang
\TrackLanguageTag{en-826}
Has en-GB-Latn been tracked?
\GetTrackedDialectFromLanguageTag{en-GB-Latn}{\thisdialect}%
\ifx\thisdialect\empty
 No!
\else
 Yes! Dialect label: \thisdialect.
\fi
\bye
```

This matches because the territory code 826 is recognised as equivalent to the code GB, and the default script for `english` is `Latn`. In this case, the dialect label is `british`. Note that this doesn't require the use of `\TrackLanguageTag` to track the dialect. It also works if the dialect has been tracked using other commands, such as `\TrackLocale`.

Here's an example that doesn't have an exact match, but does have a partial match:

```
\input tracklang
\TrackLanguageTag{de-CH-1996}
Has de-DE-1996 been tracked?
\GetTrackedDialectFromLanguageTag{de-DE-1996}{\thisdialect}%
\ifx\thisdialect\empty
 No!
  \ifx\TrackedDialectClosestSubMatch\empty
    No match on root language.
  \else
    Closest match: \TrackedDialectClosestSubMatch.
  \fi
\else
 Yes! Dialect label: \thisdialect.
\fi
\bye
```

In this case the result is:

> Has de-DE-1996 been tracked? No! Closest match: nswissgerman.

You can iterate through each tracked dialect using:

\ForEachTrackedDialect

```
\ForEachTrackedDialect{⟨cs⟩}{⟨code⟩}
```

At the start of each iteration, this sets the control sequence ⟨*cs*⟩ to the tracked dialect and does ⟨*code*⟩.

You can iterate through each tracked language using:

\ForEachTrackedLanguage

```
\ForEachTrackedLanguage{⟨cs⟩}{⟨code⟩}
```

At the start of each iteration, this sets the control sequence ⟨*cs*⟩ to the tracked language and does ⟨*code*⟩.

The above for-loops use the same internal mechanism as LaTeX's `\@for` loop. The provided control sequence ⟨*cs*⟩ is updated at the start of each iteration to the current element. The loop is terminated when this control sequence is set to `\@nil`. This special control sequence should never been used as it's just a marker and isn't actually defined. If you get an error message stating that `\@nil` is undefined, then it's most likely due to a loop control sequence being used outside the loop. This can occur if the loop contains code that isn't expanded until later. For example, if the loop code includes `\AtBeginDocument`, you need to ensure that the loop control sequence is expanded before being added to the hook.

You can test if a root language has been detected using:

\IfTrackedLanguage

```
\IfTrackedLanguage{⟨label⟩}{⟨true part⟩}{⟨false part⟩}
```

where ⟨*label*⟩ is the language label. If true, this does ⟨*true part*⟩ otherwise it does ⟨*false part*⟩.

You can test if a particular dialect has been detected using:

\IfTrackedDialect

```
\IfTrackedDialect{⟨label⟩}{⟨true part⟩}{⟨false part⟩}
```

where ⟨*label*⟩ is the dialect label. If the root language was explicitly specified, then it will also be detected as a dialect.

For example:

```
\documentclass[british,dutch]{article}

\usepackage{tracklang}

\begin{document}
``english'' \IfTrackedDialect{english}{has}{hasn't} been specified.

``british'' \IfTrackedDialect{british}{has}{hasn't} been specified.

``flemish'' \IfTrackedDialect{flemish}{has}{hasn't} been specified.

``dutch'' \IfTrackedDialect{dutch}{has}{hasn't} been specified.

``english'' or an English variant
\IfTrackedLanguage{english}{has}{hasn't} been specified.

\end{document}
```

This produces:

> "english" hasn't been specified.
>
> "british" has been specified.
>
> "flemish" hasn't been specified.
>
> "dutch" has been specified.
>
> "english" or an English variant has been specified.

You can find the root language label for a given tracked dialect using:

\TrackedLanguageFromDialect

```
\TrackedLanguageFromDialect{⟨dialect⟩}
```

If ⟨*dialect*⟩ hasn't been defined this does nothing otherwise it expands to the root language label.

You can find the tracked dialects from a given root language using:

\TrackedDialectsFromLanguage

```
\TrackedDialectsFromLanguage{⟨root language label⟩}
```

This will expand to a comma-separated list of dialect labels if the root language label has been defined, otherwise it does nothing.

You can test if a language or dialect has a corresponding ISO code using:

\IfTrackedLanguageHasIsoCode

```
\IfTrackedLanguageHasIsoCode{⟨code type⟩}{⟨label⟩}{⟨true part⟩}{⟨false part⟩}
```

where ⟨*code type*⟩ is the type of ISO code (for example, `639-1` for root languages or `3166-1` for regional dialects), and ⟨*label*⟩ is the language or dialect label. Note that the `639-3` may be set for the dialect rather than root language for sub-languages parsed using `\TrackLanguageTag`.

Alternatively, you can test if a particular ISO code has been defined using:

\IfTrackedIsoCode

```
\IfTrackedIsoCode{⟨code type⟩}{⟨code⟩}{⟨true part⟩}{⟨false part⟩}
```

where ⟨*code type*⟩ is again the type of ISO code (for example, `639-1` or `3166-1`), and ⟨*code*⟩ is the particular code (for example, `en` for ISO 639-1 or `GB` for ISO 3166-1).

You can fetch the language (or dialect) label associated with a given ISO code using:

\TrackedLanguageFromIsoCode

```
\TrackedLanguageFromIsoCode{⟨code type⟩}{⟨code⟩}
```

This does nothing if the given ⟨*code*⟩ for the given ISO ⟨*code type*⟩ has not been defined, otherwise it expands a comma-separated list of language or dialect labels.

You can fetch the ISO code for a given code type using:

\TrackedIsoCodeFromLanguage

```
\TrackedIsoCodeFromLanguage{⟨code type⟩}{⟨label⟩}
```

where ⟨*label*⟩ is the language or dialect label and ⟨*code type*⟩ is the ISO code type (for example, `639-1` or `3166-1`). Unlike `\TrackedLanguageFromIsoCode`, this command only expands to a single label rather than a comma-separated list.

The above commands do nothing in the event of an unknown code or code type, so if you accidentally get the wrong code type, you won't get an error. If you're unsure of the code type, you can use the following commands:

\TwoLetterIsoCountryCode

```
\TwoLetterIsoCountryCode
```

This expands to 3166-1 and is used for the two-letter country codes.

\TwoLetterIsoLanguageCode

```
\TwoLetterIsoLanguageCode
```

This expands to 639-1 and is used for the two-letter root language codes.

\ThreeLetterIsoLanguageCode

```
\ThreeLetterIsoLanguageCode
```

This expands to 639-2 and is used for the three-letter root language codes.

\ThreeLetterExtIsoLanguageCode

```
\ThreeLetterExtIsoLanguageCode
```

(New to v1.3.) This expands to 639-3. This code is only used for a root language if there's no 639-1 or 639-2 code. It may also be used for a dialect if a sub-language part has been set in the language tag parsed by `\TrackLanguageTag`.

The `\Get...` commands below are designed to be expandable. If the supplied ⟨*dialect*⟩ is unrecognised they expand to empty. Remember that the dialect must first be identified as a tracked language for it to be recognised.

As from v1.3, the language tag for a given dialect can be obtained using:

\GetTrackedLanguageTag

\GetTrackedLanguageTag{⟨*dialect*⟩}

where ⟨*dialect*⟩ is the label identifying the dialect. Uses the `und` (undetermined) code for unknown languages.

As from v1.3, each tracked dialect may also have an associated modifier, which can be fetched using:

\GetTrackedDialectModifier

\GetTrackedDialectModifier{⟨*dialect*⟩}

where ⟨*dialect*⟩ is the label identifying the dialect. This value is typically obtained by parsing a POSIX locale identifier with `\TrackLocale` or `\TrackLangFromEnv` but may be set explicitly. (See Section 5 for setting this value. Likewise for the following commands.)

You can test if a dialect has an associated modifier using:

\IfHasTrackedDialectModifier

\IfHasTrackedDialectModifier{⟨*dialect*⟩}{⟨*true*⟩}{⟨*false*⟩}

If the dialect has an associated modifier this does ⟨*true*⟩ otherwise it does ⟨*false*⟩.

For example:

```
\documentclass[british,francais,american,canadian,canadien,dutch]{article}

\usepackage{tracklang}

\begin{document}

Languages: \ForEachTrackedLanguage{\ThisLanguage}{\ThisLanguage\space
(ISO \TwoLetterIsoLanguageCode:
``\TrackedIsoCodeFromLanguage{\TwoLetterIsoLanguageCode}{\ThisLanguage}''). }

Dialects: \ForEachTrackedDialect{\ThisDialect}{\ThisDialect\space
(\IfTrackedLanguageHasIsoCode{\TwoLetterIsoCountryCode}{\ThisDialect}%
 {ISO \TwoLetterIsoCountryCode:
  ``\TrackedIsoCodeFromLanguage{\TwoLetterIsoCountryCode}{\ThisDialect}''}%
 {no specific region};
root: \TrackedLanguageFromDialect{\ThisDialect}). }

Language for ISO \TwoLetterIsoCountryCode\ ``GB'':
\TrackedLanguageFromIsoCode{\TwoLetterIsoCountryCode}{GB}.

Language for ISO \TwoLetterIsoCountryCode\ ``CA'':
\TrackedLanguageFromIsoCode{\TwoLetterIsoCountryCode}{CA}.

Country ISO \TwoLetterIsoCountryCode\ code for ``canadian'':
\TrackedIsoCodeFromLanguage{\TwoLetterIsoCountryCode}{canadian}.

\end{document}
```

This produces:

> Languages: english (ISO 639-1: “en”). french (ISO 639-1: “fr”). dutch (ISO 639-1: “nl”).
>
> Dialects: american (ISO 3166-1: “US”; root: english). british (ISO 3166-1: “GB”; root: english). canadian (ISO 3166-1: “CA”; root: english). canadien (ISO 3166-1:

"CA"; root: french). dutch (no specific region; root: dutch). francais (no specific region; root: french).

Language for ISO 3166-1 "GB": british.

Language for ISO 3166-1 "CA": canadian,canadien.

Country ISO 3166-1 code for "canadian": CA.

As from v1.3, each tracked dialect may also have an associated variant, which can be fetched using:

\GetTrackedDialectVariant

```
\GetTrackedDialectVariant{⟨dialect⟩}
```

where ⟨*dialect*⟩ is the label identifying the dialect. This value is typically obtained by parsing a language tag with \TrackLanguageTag but may be set explicitly.

You can test if a dialect has an associated variant using:

\IfHasTrackedDialectVariant

```
\IfHasTrackedDialectVariant{⟨dialect⟩}{⟨true⟩}{⟨false⟩}
```

As from v1.3, each tracked dialect may also have an associated script, which can be fetched using:

\GetTrackedDialectScript

```
\GetTrackedDialectScript{⟨dialect⟩}
```

where ⟨*dialect*⟩ is the label identifying the dialect.

You can test if a dialect has an associated script using:

\IfHasTrackedDialectScript

```
\IfHasTrackedDialectScript{⟨dialect⟩}{⟨true⟩}{⟨false⟩}
```

If the dialect has an associated script this does ⟨*true*⟩ otherwise it does ⟨*false*⟩. This information is provided for language packages that need to know what script is required, but there's no guarantee that the script will actually be set in the document. Similarly for all the other attributes described here.

Note that the script should be a recognised four-letter ISO 15924 code, such as `Latn` or `Cyrl`. If a dialect doesn't have an associated script then the default for the root language should be assumed. For example, "Latn" for English dialects or "Cyrl" for Russian dialects. (The default script for known languages can be obtained using \TrackLangGetDefaultScript, see Section 6.2.2 for further details. Most root languages have a default script, but there are a few without one as it may depend on region, politics or ideology.)

There's a convenient expandable command for testing the script:

```
\IfTrackedDialectIsScriptCs{⟨dialect⟩}{⟨cs⟩}{⟨true⟩}{⟨false⟩}
```

This tests if the given tracked dialect has an associated script and compares the value with the replacement text of ⟨*cs*⟩. If the dialect hasn't been explicitly assigned a script, then test is performed against the default script for the root language.

The supplementary package tracklang-scripts provides some additional commands relating to writing systems, including commands in the form `\TrackLangScript`⟨*code*⟩ where ⟨*code*⟩ is the ISO 15924 four-letter code. If the dialect doesn't have an associated script, ⟨*false*⟩ is done. This package isn't loaded automatically, so you'll need to explicitly load it. The generic code is in `tracklang-scripts.tex`:

```
\input tracklang-scripts
```

There's a convenient LaTeX wrapper `tracklang-scripts.sty`:

```
\usepackage{tracklang-scripts}
```

See Section 6.5 for further details of that package.

For example, the following defines a command to check if the given dialect should use a Latin script:

```
\input tracklang-scripts
\def\islatin#1#2#3{%
  \IfTrackedDialectIsScriptCs{#1}{\TrackLangScriptLatn}{#2}{#3}%
}
```

Note that the script value doesn't mean that the document is actually using that script. It means that this is the user's *desired* script, but whether that script is actually set relies on the appropriate settings in the relevant language package (such as polyglossia's `script` key).

As from v1.3, each tracked dialect may also have a sub-language identifier (for example, `arevela`), which can be fetched using:

`\GetTrackedDialectSubLang`

```
\GetTrackedDialectSubLang{⟨dialect⟩}
```

where ⟨*dialect*⟩ is the label identifying the dialect.

You can test if a dialect has an associated sub-tag using:

`\IfHasTrackedDialectSubLang`

```
\IfHasTrackedDialectSubLang{⟨dialect⟩}{⟨true⟩}{⟨false⟩}
```

If the dialect has an associated sub-tag this does ⟨*true*⟩ otherwise it does ⟨*false*⟩.

As from v1.3, each tracked dialect may also have additional information, which can be fetched using:

`\GetTrackedDialectAdditional`

```
\GetTrackedDialectAdditional{⟨dialect⟩}
```

where ⟨*dialect*⟩ is the label identifying the dialect.

You can test if a dialect has additional information using:

`\IfHasTrackedDialectAdditional`

```
\IfHasTrackedDialectAdditional{⟨dialect⟩}{⟨true⟩}{⟨false⟩}
```

If the dialect has additional information this does ⟨*true*⟩ otherwise it does ⟨*false*⟩.

Most packages that implement multilingual support have a set of language definition files for each supported language or dialect. It may be that only the root language is needed, if there are no variations between that language's dialect (for the purposes of that package), or it may be that separate definition files are required for each dialect. However it can be awkward trying to map the requested dialect or language label to the file name. Should, say, the file containing the French code be called ⟨*prefix*⟩`-french-`⟨*suffix*⟩ or ⟨*prefix*⟩`-frenchb-`⟨*suffix*⟩ or ⟨*prefix*⟩`-francais-`⟨*suffix*⟩? Should, say, the file containing the British English code be called ⟨*prefix*⟩`-british-`⟨*suffix*⟩ or ⟨*prefix*⟩`-UKenglish-`⟨*suffix*⟩? If you want to modularise the language support for your package so that each language module has a different maintainer will the maintainers know what tag to use for their language?

To help with this, tracklang provides:

```
\IfTrackedLanguageFileExists{⟨dialect⟩}{⟨prefix⟩}{⟨suffix⟩}{⟨true part⟩}
{⟨false part⟩}
```

This attempts to find the file called ⟨*prefix*⟩⟨*tag*⟩⟨*suffix*⟩ where ⟨*tag*⟩ is determined from ⟨*dialect*⟩. If the file is found

```
\CurrentTrackedTag
```

is set to ⟨*tag*⟩ and ⟨*true part*⟩ is done, otherwise ⟨*false part*⟩ is done. If this command is empty, then the dialect hasn't been detected. If the dialect has been detected, but no file can be found, then `\CurrentTrackedTag` is set to the final attempt at determining ⟨*tag*⟩.

There's a convenient shortcut command new to version 1.3:

`\TrackLangRequireDialect`

```
\TrackLangRequireDialect[⟨load code⟩]{⟨pkgname⟩}{⟨dialect⟩}
```

which uses `\IfTrackedLanguageFileExists` to input the resource file if found. The prefix is given by ⟨*pkgname*⟩`-` and the suffix is `.ldf`. A warning is issued if no resource file is found. Note that while it makes sense for ⟨*pkgname*⟩ to be the same as the base name of the package that uses these resource files, they don't have to be the same. This command additionally defines

`\TrackLangRequireDialectPrefix`

```
\TrackLangRequireDialectPrefix
```

to ⟨*pkgname*⟩, which allows the prefix to be picked up by resource file commands, such as `\TrackLangProvidesResource` and `\TrackLangRequireResource`. (See below.)

The optional argument ⟨*load code*⟩ is the code that actually inputs the required file. This defaults to

```
\TrackLangRequireResource{\CurrentTrackedTag}
```

The `\IfTrackedLanguageFileExists` command sets up the current tracked dialect with `\SetCurrentTrackedDialect{`⟨*dialect*⟩`}`, which enables the following commands that may be used within ⟨*true part*⟩ or ⟨*false part*⟩:

```
\CurrentTrackedDialect
```

The dialect label.

```
\CurrentTrackedLanguage
```

If the dialect hasn't been detected, this command will be empty, otherwise it will expand to the root language label (which may be the same as the dialect label).

```
\CurrentTrackedRegion
```

If the dialect hasn't been detected, this command will be empty. If the dialect has been assigned an ISO 3166-1 code, `\CurrentTrackedRegion` will expand to that code, otherwise it will be empty.

```
\CurrentTrackedIsoCode
```

If the dialect hasn't been detected, this command will be empty. Otherwise it may be empty or it may expand to the ISO 639-1 or ISO 639-2 or ISO 639-3 code.

As from version 1.3, the following are also available, but don't contribute to the tag.

```
\CurrentTrackedDialectModifier
```

The dialect's modifier or empty if not set.

```
\CurrentTrackedDialectVariant
```

The dialect's variant or empty if not set.

```
\CurrentTrackedDialectSubTag
```

The dialect's sub-language code or empty if not set.

```
\CurrentTrackedDialectAdditional
```

The dialect's additional information or empty if not set.

```
\CurrentTrackedLanguageTag
```

The dialect's language tag. Take care not to confuse this with `\CurrentTrackedTag`.

```
\CurrentTrackedDialectScript
```

The dialect's script. If the dialect doesn't have the script set, the default script is used instead. For example, the file `foo-serbian.ldf` could test for the existence of the file

```
foo-serbian-\CurrentTrackedDialectScript.ldf
```

and load it if it exists. The most convenient way is to use `\TrackLangRequestResource` (described below):

```
\TrackLangRequestResource
 {serbian-\CurrentTrackedDialectScript}
 {}% not found, set default code here
```

The Cyrillic settings could then be placed in `foo-serbian-Cyrl.ldf` and the Latin settings in `foo-serbian-Latn.ldf`.

The ⟨*tag*⟩ is determined as follows:

1. If no dialect with the given label has been detected, the condition evaluates to *false* and `\CurrentTrackedTag` is empty.

2. If there is no language code (ISO 639-1 or 639-2 or 639-3):

   a) If there's also no ISO 3166-1 code (`\CurrentTrackedRegion` is empty), then ⟨*tag*⟩ (`\CurrentTrackedTag`) is set to the root language (`\CurrentTrackedLanguage`). If the file ⟨*prefix*⟩⟨*tag*⟩⟨*suffix*⟩ exists, the condition will evaluate to *true* otherwise it will evaluate to *false*.

   b) If there is an ISO 3166-1 code, then ⟨*tag*⟩ (`\CurrentTrackedTag`) will be set to `\CurrentTrackedRegion` and if the file ⟨*prefix*⟩⟨*tag*⟩⟨*suffix*⟩ exists, the condition will evaluate to *true* otherwise it will evaluate to *false*.

3. If `\CurrentTrackedRegion` is empty (no ISO 3166-1 territory code) then the ⟨*tag*⟩ (`\CurrentTrackedTag`) is set to just `\CurrentTrackedIsoCode`.

   If the file ⟨*prefix*⟩⟨*tag*⟩⟨*suffix*⟩ exists, the condition will evaluate to *true*. If the file doesn't exist and there are additional ISO language codes available (639-2 or 639-3) then the test will be repeated for the next code. If the test still fails, then the condition evaluates to *false*.

4. The ⟨*tag*⟩ (`\CurrentTrackedTag`) is then set to

   ```
   \CurrentTrackedIsoCode-\CurrentTrackedRegion
   ```

   and if the file ⟨*prefix*⟩⟨*tag*⟩⟨*suffix*⟩ exists, the condition evaluates to *true*.

5. The ⟨*tag*⟩ (`\CurrentTrackedTag`) is then set to just `\CurrentTrackedIsoCode`, and if the file ⟨*prefix*⟩⟨*tag*⟩⟨*suffix*⟩ exists, the condition evaluates to *true*. If the file doesn't exist and there are other ISO codes (639-2 or 639-3), then `\CurrentTrackedIsoCode` is set to the next available language code and step 4 is retried.

6. If there is an ISO 3166-1 region code, the ⟨*tag*⟩ (`\CurrentTrackedTag`) is then set to `\CurrentTrackedRegion`, and if the file ⟨*prefix*⟩⟨*tag*⟩⟨*suffix*⟩ exists, the condition evaluates to *true*.

7. Finally, the ⟨*tag*⟩ (\CurrentTrackedTag) is set to the root language label and if the file ⟨*prefix*⟩⟨*tag*⟩⟨*suffix*⟩ exist the condition evaluates to *true* otherwise it evaluates to *false*.

For example (pre v1.3):

```
\AnyTrackedLanguages
{%
  \ForEachTrackedDialect{\ThisDialect}%
  {% try to load the language file for this dialect
    \IfTrackedLanguageFileExists{\ThisDialect}%
    {mypackage-}% file prefix
    {.ldf}% file suffix
    {\input mypackage-\CurrentTrackedTag.ldf}% file found
    {% file not found
      \PackageWarning{mypackage}{No support for language
       `\ThisDialect'}%
    }%
  }%
}
{% no languages detected so use defaults
}
```

With version 1.3 onwards, this can be written more concisely as:

```
\AnyTrackedLanguages
{%
  \ForEachTrackedDialect{\ThisDialect}%
  {% try to load the language file for this dialect
    \TrackLangRequireDialect{mypackage}{\ThisDialect}%
  }%
}
{% no languages detected so use defaults
}
```

which additionally enables the tracklang version 1.3 commands described below, such as \TrackLangRequireResource.

If, for example, \ThisDialect is british, then the file search will be in the order:

1. mypackage-british.ldf
2. mypackage-en-GB.ldf
3. mypackage-eng-GB.ldf
4. mypackage-en.ldf
5. mypackage-eng.ldf
6. mypackage-GB.ldf
7. mypackage-english.ldf

If, for example, \ThisDialect is francais, then the file search will be in the order:

1. mypackage-francais.ldf
2. mypackage-fr.ldf

3. mypackage-fra.ldf

4. mypackage-french.ldf

This is because the predefined `francais` option has no region assigned to it. Be careful if the dialect label is the actual root language. For example, if `\ThisDialect` is `french`, then the file search will be in the order:

1. mypackage-french.ldf

2. mypackage-fr.ldf

3. mypackage-fra.ldf

4. mypackage-french.ldf

Note that the last try will always fail in this case since if the file exists, it will be found on the first try.

If you're only providing support for the root languages (pre v1.3):

```
\AnyTrackedLanguages
{%
  \ForEachTrackedLanguage{\ThisLanguage}%
  {% try to load the language file for this root language
    \IfTrackedLanguageFileExists{\ThisLanguage}%
    {mypackage-}% file prefix
    {.ldf}% file suffix
    {\input mypackage-\CurrentTrackedTag.ldf}% file found
    {% file not found
      \PackageWarning{mypackage}{No support for language
       `\ThisLanguage'}%
    }%
  }%
}
{% no languages detected so use defaults
}
```

With version 1.3 onwards, this can be written more concisely as:

```
\AnyTrackedLanguages
{%
  \ForEachTrackedLanguage{\ThisLanguage}%
  {% try to load the language file for this root language
    \TrackLangRequireDialect{mypackage}{\ThisLanguage}%
  }%
}
{% no languages detected so use defaults
}
```

which additionally enables the commands described below. Note that in this case, if more than one dialect for the same language has been tracked, only the hooks for the last dialect for that language will be adjusted, so it's usually best to iterate over the dialects.

The following `\TrackLang...Resource...` commands may only be used in resource files that are loaded using `\TrackLangRequireDialect`. An error will occur if the file is input through some other method.

Within the resource file ⟨*pkgname*⟩-⟨*tag*⟩.ldf, you can identify the file using (new to version 1.3):

\TrackLangProvidesResource

```
\TrackLangProvidesResource{⟨tag⟩}[⟨version info⟩]
```

If \ProvidesFile is defined (through the LaTeX kernel) this is used, otherwise a simplified generic alternative is used that's suitable for other TeX formats.

The resource file can load another resource file ⟨*pkgname*⟩-⟨*tag2*⟩.ldf, using (new to version 1.3):

\TrackLangRequireResource

```
\TrackLangRequireResource{⟨tag2⟩}
```

For example, the dialect file foo-en-GB.ldf might need to load the root language resource file foo-english.ldf:

```
\TrackLangProvidesResource{en-GB}
\TrackLangRequireResource{english}
```

If foo-english.ldf is also identified with \TrackLangProvidesResource, this will ensure that it's only loaded once.

If you require the resource file and want to perform ⟨*code1*⟩ if it's loaded at this point or ⟨*code2*⟩ if it's already been loaded then you can use (new to version 1.3):

\TrackLangRequireResourceOrDo

```
\TrackLangRequireResourceOrDo{⟨tag2⟩}{⟨code1⟩}{⟨code2⟩}
```

If you want to load a resource file if it exists (without an error if it doesn't exist), then you can use

\TrackLangRequestResource

```
\TrackLangRequestResource{⟨tag2⟩}{⟨not found code⟩}
```

If the file doesn't exist, ⟨*not found code*⟩ is done.

Note that these \...Resource commands are only permitted within the resource files. They are internally enabled through \TrackLangRequireDialect.

The above restriction on the resource files loaded through \TrackLangRequireDialect, and the fact that it internally uses \IfTrackedLanguageFileExists, means that commands like \CurrentTrackedLanguage or \CurrentTrackedDialect may be used in those files. This means that the name of the captions hook can be obtained through them. (Remember that the file foo-en-GB.ldf might have been loaded with, say, the british dialect or with the synonymous UKenglish dialect or with a dialect label that doesn't have a corresponding caption hook, such as enGBLatn.)

The polyglossia package has caption hooks in the form \captions⟨*language*⟩ whereas babel has captions hooks in the form \captions⟨*dialect*⟩. This leads to a rather cumbersome set of conditionals:

```
\ifcsundef{captions\CurrentTrackedLanguage}
{%
  \ifcsundef{captions\CurrentTrackedDialect}%
  {}%
  {%
    \csgappto{captions\CurrentTrackedDialect}{%
      % code to append to hook
    }%
  }%
}%
{%
  \csgappto{captions\CurrentTrackedLanguage}{%
    % code to append to hook
  }%
}
% do code now to initialise
```

Note that the above has been simplified through the use of etoolbox commands, which isn't suitable for generic use. It also doesn't query the mapping from tracklang's dialect label to the closest matching babel dialect label.

Instead (new to version 1.3), tracklang provides a command to perform this set of conditionals using generic code:

\TrackLangAddToHook

```
\TrackLangAddToHook{⟨code⟩}{⟨type⟩}
```

where ⟨*code*⟩ is the code to append to the ⟨*type*⟩ hook. This always performs ⟨*code*⟩ after testing for the hook in case the hook is undefined or has already been called (for example, ngerman uses `\captionsngerman` when the package is loaded, not at the start of the document).

Note that this command is enabled through `\TrackLangRequireDialect` so should only be used inside resource files.

Since `captions` is a commonly used hook type, there's a shortcut command provided:

\TrackLangAddToCaptions

```
\TrackLangAddToCaptions{⟨code⟩}
```

This is equivalent to `\TrackLangAddToHook{`⟨*code*⟩`}{captions}`.

4.1 Examples

The examples in this section illustrate the above commands.

4.1.1 animals.sty

This example is for a trivial package called animals.sty that defines three textual commands: `\catname`, `\dogname` and `\ladybirdname`. The default values are: "cat", "dog" and "bishy-barney-bee".[1]

The supported languages are defined in files with the prefix `animals-` and the suffix `.ldf`.

[1] Thass Broad Norfolk, my bewties :-P

Here's the code for `animals.sty`:

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{animals}

\RequirePackage{tracklang}[2016/10/07] %v1.3

% Any undeclared options are language settings:
\DeclareOption*{\TrackLanguageTag{\CurrentOption}}
\ProcessOptions

% Default definitions
\newcommand\catname{cat}
\newcommand\dogname{dog}
\newcommand\ladybirdname{bishy-barney-bee}

\AnyTrackedLanguages
{%
  \ForEachTrackedDialect{\this@dialect}{%
    \TrackLangRequireDialect{animals}{\this@dialect}%
  }%
}
{% no tracked languages, default already set up
}

\endinput
```

Here's a Plain TEX version that picks up the language from the locale environment variable:

```
\input tracklang

\TrackLangFromEnv

% Default definitions
\def\catname{cat}
\def\dogname{dog}
\def\ladybirdname{bishy-barney-bee}

\AnyTrackedLanguages
{%
  \ForEachTrackedDialect{\thisdialect}{%
    \TrackLangRequireDialect{animals}{\thisdialect}%
  }%
}
{% no tracked languages, default already set up
}
```

In the event that a user or supplementary package for some reason wants to load a resource file for a language that hasn't been tracked, it might be worth providing a command for this purpose:

```
\newcommand*{\RequireAnimalsDialect}[1]{%
  \TrackLangRequireDialect{animals}{#1}%
}
```

The loop can then be changed to:

```
  \ForEachTrackedDialect{\this@dialect}{%
    \RequireAnimalsDialect\this@dialect
  }%
```

The `animals-english.ldf` file valid for both the Plain TEX and LATEX formats contains:

```
\TrackLangProvidesResource{english}

\def\englishanimals{%
  \def\catname{cat}%
  \def\dogname{dog}%
  \def\ladybirdname{bishy-barney-bee}%
}

\TrackLangAddToCaptions{\englishanimals}
```

The `animals-en-GB.ldf` file contains:

```
\TrackLangProvidesResource{en-GB}
\TrackLangRequireResource{english}

\def\enGBanimals{%
  \englishanimals
  \def\ladybirdname{ladybird}%
}
\TrackLangAddToCaptions{\enGBanimals}
```

The `animals-en-US.ldf` file contains:

```
\TrackLangProvidesResource{en-US}
\TrackLangRequireResource{english}

\def\enUSanimals{%
  \englishanimals
  \def\ladybirdname{ladybug}%
}
\TrackLangAddToCaptions{\enUSanimals}
```

Here's a German version in the file `animals-german.ldf`:

```
\TrackLangProvidesResource{german}

\def\germananimals{%
  \def\catname{Katze}%
  \def\dogname{Hund}%
  \def\ladybirdname{Marienk\"afer}%
}

\TrackLangAddToCaptions{\germananimals}
```

This means that if babel or polyglossia are loaded, the redefinitions are automatically performed whenever the language is changed, but if there's no caption mechanism the user can switch the fixed names using the `\...animals` commands.

Here's an example LATEX document that doesn't have any caption hooks:

```
\documentclass[english,german,a4paper]{article}

\usepackage{animals}

\begin{document}
\englishanimals

\catname.
\dogname.
```

```
\ladybirdname.

\germananimals

\catname.
\dogname.
\ladybirdname.
\end{document}
```

Here's a babel example document:

```
\documentclass[american,german,british,a4paper]{article}

\usepackage{babel}
\usepackage{animals}

\begin{document}
\selectlanguage{american}

\catname.
\dogname.
\ladybirdname.

\selectlanguage{german}

\catname.
\dogname.
\ladybirdname.

\selectlanguage{british}

\catname.
\dogname.
\ladybirdname.

\end{document}
```

There is some redundancy with the above resource files. Consider the babel example above. The `american` dialect is the first option, so in that case `animals-en-US.ldf` is loaded followed by `animals-english.ldf`. This means that the `\captionsamerican` hook now includes

```
\englishanimals
\enUSanimals
```

Since `\enUSanimals` includes `\englishanimals`, there is redundant code. However, when the `british` dialect is processed, this loads the file `animals-en-GB.ldf` but not the file `animals-english.ldf` (since it's already been loaded). This means that `\captionsbritish` contains `\enGBanimals` but not `\englishanimals`.

If this redundancy is an issue (for example, there are so many redefinitions needed that it significantly slows the document build process), then it can be addressed with the following modifications. The `animals-en-GB.ldf` file is now:

```
\TrackLangProvidesResource{en-GB}

\def\enGBanimals{%
  \englishanimals
```

```
  \def\ladybirdname{ladybird}%
}

\TrackLangRequireResourceOrDo{english}%
{
  \TrackLangAddToCaptions{%
    \def\ladybirdname{ladybird}%
  }%
}
{
  \TrackLangAddToCaptions{\enGBanimals}
}
```

The `animals-en-US.ldf` file is now:

```
\TrackLangProvidesResource{en-US}

\providecommand*{\enUSanimals}{%
  \englishanimals
  \renewcommand*{\ladybirdname}{ladybug}%
}

\TrackLangRequireResourceOrDo{english}
{
  \TrackLangAddToCaptions{%
    \renewcommand*{\ladybirdname}{ladybird}%
  }%
}
{
  \TrackLangAddToCaptions{\enUSanimals}{captions}
}
```

This means that the document that has the dialects listed in the order `american`, `british` now has

```
\englishanimals
\def\ladybirdname{ladybird}
```

in the `\captionsbritish` hook and just `\enUSanimals` in the `\captionsamerican` hook, which has removed most of the redundancy.

Note that polyglossia has a `\captionsenglish` hook but not `\captionsamerican` or `\captionsbritish`, so this code doesn't allow for switching between variants of the same language with polyglossia.

4.1.2 regions.sty

Earlier, I mentioned the search order for `\IfTrackedLanguageFileExists` where if, for example, the dialect is `british`, the file search will be:

1. `mypackage-british.ldf`
2. `mypackage-en-GB.ldf`
3. `mypackage-eng-GB.ldf`
4. `mypackage-en.ldf`

5. `mypackage-eng.ldf`
6. `mypackage-GB.ldf`
7. `mypackage-english.ldf`

You may have wondered why `mypackage-GB.ldf` is included in the search given that some countries have multiple official languages, which means that the country code on its own may not indicate the language.

The reason for including just the country code as the ⟨*tag*⟩ in the file search is to allow for region rather than language dependent settings. For example, suppose I want to write a package that needs to know whether to use imperial or metric measurements in the document, but I also want to provide multilingual support. The language alone won't tell me whether to use imperial or metric (for example, the US uses imperial and the UK uses metric for most product attributes). I could provide `.ldf` files for every language and region combination, but this would result in a lot redundancy.

`\TrackLangRequireDialect` has an optional argument for adjusting the way the resource files are loaded. Suppose I have `regions-`⟨*tag*⟩`.ldf` resource files, then

```
\TrackLangRequireDialect{regions}{\this@dialect}
```

loads the resource file for the dialect given by `\this@dialect` using

```
\TrackLangRequireResource{\CurrentTrackedTag}
```

I can use the optional argument to also load the resource file for the root language as well:

```
\newcommand*{\RequireRegionsDialect}[1]{%
 \TrackLangRequireDialect
    [\TrackLangRequireResource{\CurrentTrackedTag}%
     \TrackLangRequireResource{\CurrentTrackedLanguage}%
    ]%
    {regions}{#1}%
}
```

Now the dialect `british` can load both `regions-GB.ldf` and `regions-english.ldf`.

The example package (`regions.sty`) below illustrates this.

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{regions}

\RequirePackage{tracklang}[2016/10/07]

\DeclareOption*{\TrackLanguageTag{\CurrentOption}}
\ProcessOptions

\newcommand*{\weightunit}{kg}
\newcommand*{\lengthunit}{mm}
\newcommand*{\currencyunit}{EUR}

\newcommand*{\unitname}{units}

\newcommand*{\RequireRegionsDialect}[1]{%
 \TrackLangRequireDialect
    [\TrackLangRequireResource{\CurrentTrackedTag}%
```

```
     \TrackLangRequireResource{\CurrentTrackedLanguage}%
    ]%
    {regions}{#1}%
}

\AnyTrackedLanguages
{%
  \ForEachTrackedDialect{\this@dialect}{%
    \RequireRegionsDialect\this@dialect
  }%
}
{% no tracked languages, default already set up
}

\endinput
```

There are separate `.ldf` files for region and language. First are the regions.

- `regions-BE.ldf` (Belgium):

  ```
  \TrackLangProvidesResource{BE}

  \providecommand*{\BEunits}{%
    \renewcommand*{\weightunit}{kg}%
    \renewcommand*{\lengthunit}{mm}%
    \renewcommand*{\currencyunit}{EUR}%
  }

  \TrackLangAddToCaptions{\BEunits}
  ```

- `regions-CA.ldf` (Canada):

  ```
  \TrackLangProvidesResource{CA}

  \providecommand*{\CAunits}{%
    \renewcommand*{\weightunit}{kg}%
    \renewcommand*{\lengthunit}{mm}%
    \renewcommand*{\currencyunit}{CAD}%
  }

  \TrackLangAddToCaptions{\CAunits}
  ```

- `regions-GB.ldf` (Great Britain):

  ```
  \TrackLangProvidesResource{GB}

  \providecommand*{\GBunits}{%
    \renewcommand*{\weightunit}{kg}%
    \renewcommand*{\lengthunit}{mm}%
    \renewcommand*{\currencyunit}{GBP}%
  }

  \TrackLangAddToCaptions{\GBunits}
  ```

- `regions-US.ldf` (USA):

  ```
  \TrackLangProvidesResource{US}
  ```

```
\providecommand*{\USunits}{%
  \renewcommand*{\weightunit}{lb}%
  \renewcommand*{\lengthunit}{in}%
  \renewcommand*{\currencyunit}{USD}%
}

\TrackLangAddToCaptions{\USunits}
```

Now the language files:

- `regions-dutch.ldf`:

```
\TrackLangProvidesResource{dutch}

\providecommand*{\dutchnames}{%
  \renewcommand*{\unitname}{meeteenheden}%
}

\TrackLangAddToCaptions{\dutchnames}
```

- `regions-english.ldf`:

```
\TrackLangProvidesResource{english}

\providecommand*{\englishnames}{%
  \renewcommand*{\unitname}{units}%
}

\TrackLangAddToCaptions{\englishnames}
```

- `regions-french.ldf`:

```
\TrackLangProvidesResource{french}

\providecommand*{\frenchnames}{%
  \renewcommand*{\unitname}{unit\'es}%
}

\TrackLangAddToCaptions{\frenchnames}
```

- `regions-german.ldf`:

```
\TrackLangProvidesResource{french}

\providecommand*{\germannames}{%
  \renewcommand*{\unitname}{Ma\ss einheiten}%
}

\TrackLangAddToCaptions{\germannames}
```

Here's an example document that uses this package:

```
\documentclass[canadien]{article}

\usepackage{regions}
```

```
\begin{document}

\unitname: \weightunit, \lengthunit, \currencyunit.

\end{document}
```

This works because the ⟨*tag*⟩ search looks for the country code before the root language label. However, this will fail if the dialect label is the same as a root language label that has an associated territory, marked with † in table 1.2, as then it will be picked up before the country code.

In the above example, `regions-CA.ldf` is matched rather than `regions-french.ldf`, so `regions-CA.ldf` is loaded by

```
\TrackLangRequireResource{\CurrentTrackedTag}
```

After this, the language file `regions-french.ldf` is then loaded:

```
\TrackLangRequireResource{\CurrentTrackedLanguage}
```

This assumes that there's a country code `.ldf` file available. This example needs a little modification to use default units in case the region is missing:

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{regions}

% Pass all options to tracklang:
\DeclareOption*{\PassOptionsToPackage{\CurrentOption}{tracklang}}
\ProcessOptions

\RequirePackage{tracklang}

\newcommand*{\weightunit}{kg}
\newcommand*{\lengthunit}{mm}
\newcommand*{\currencyunit}{EUR}

\newcommand*{\unitname}{units}

\newcommand*{\defaultunits}{%
  \renewcommand*{\weightunit}{kg}%
  \renewcommand*{\lengthunit}{mm}%
  \renewcommand*{\currencyunit}{EUR}%
}

\newcommand*{\RequireRegionsDialect}[1]{%
    \TrackLangRequireDialect
    [\TrackLangRequireResource{\CurrentTrackedTag}%
      \ifx\CurrentTrackedTag\CurrentTrackedLanguage
        \TrackLangAddToCaptions{\defaultunits}%
      \else
        \TrackLangRequireResource{\CurrentTrackedLanguage}%
      \fi
    ]%
    {regions}{#1}%
}

\AnyTrackedLanguages
{%
  \ForEachTrackedDialect{\this@dialect}{%
```

```
    \RequireRegionsDialect\this@dialect
  }%
}
{% no tracked languages, default already set up
}

\endinput
```

Note that we still have a problem for dialect labels that are identical to root language labels with an associated territory (such as manx). This case can be checked with the following adjustment:

```
\newcommand*{\RequireRegionsDialect}[1]{%
    \TrackLangRequireDialect
    [\TrackLangRequireResource{\CurrentTrackedTag}%
      \ifx\CurrentTrackedTag\CurrentTrackedLanguage
        \ifx\CurrentTrackedRegion\empty
          \TrackLangAddToCaptions{\defaultunits}%
        \else
          \TrackLangRequireResource{\CurrentTrackedRegion}%
        \fi
      \else
        \TrackLangRequireResource{\CurrentTrackedLanguage}%
      \fi
    ]%
    {regions}{#1}%
}
```

In the case where both the dialect and root language label are `manx` with the resource files `regions-manx.ldf` and `regions-IM.ldf`, then `\CurrentTrackedTag` will be `manx` (the dialect label) so `regions-manx.ldf` will be loaded with:

```
\TrackLangRequireResource{\CurrentTrackedTag}
```

In this case `\CurrentTrackedRegion` is `IM` (that is, it's not empty) so then `regions-IM.ldf` will be loaded with:

```
\TrackLangRequireResource{\CurrentTrackedRegion}
```

Here's another document that sets up dialects with tracklang labels that aren't recognised by babel. This means that there's no corresponding `\captions...` hook for either the dialect label or the root language label, so mappings need to be defined from the tracklang dialect label to the matching babel dialect label.

```
\documentclass{article}

\usepackage{tracklang}

\TrackLanguageTag{de-US-1996}
\SetTrackedDialectLabelMap{\TrackLangLastTrackedDialect}{ngerman}

\TrackLanguageTag{en-MT}
\SetTrackedDialectLabelMap{\TrackLangLastTrackedDialect}{UKenglish}

\usepackage[main=ngerman,UKenglish]{babel}
\usepackage{regions}
```

```
\begin{document}
\selectlanguage{ngerman}

\unitname: \weightunit, \lengthunit, \currencyunit.

\selectlanguage{UKenglish}

\unitname: \weightunit, \lengthunit, \currencyunit.

\end{document}
```

This produces:

> Maßeinheiten: lb, in, USD.
>
> units: kg, mm, EUR.

Compare this with:

```
\documentclass{article}

\usepackage[main=ngerman,UKenglish]{babel}
\usepackage{regions}

\begin{document}
\selectlanguage{ngerman}

\unitname: \weightunit, \lengthunit, \currencyunit.

\selectlanguage{UKenglish}

\unitname: \weightunit, \lengthunit, \currencyunit.

\end{document}
```

which produces:

> Maßeinheiten: kg, mm, EUR.
>
> units: kg, mm, GBP.

Note that these mappings aren't needed if babel is loaded with the root language labels instead. For example:

```
\documentclass{article}

\usepackage{tracklang}

\TrackLanguageTag{de-US-1996}
\SetTrackedDialectLabelMap{\TrackLangLastTrackedDialect}{ngerman}

\TrackLanguageTag{en-MT}

\usepackage[main=ngerman,english]{babel}
\usepackage{regions2}

\begin{document}
\selectlanguage{ngerman}

\unitname: \weightunit, \lengthunit, \currencyunit.
```

```
\selectlanguage{english}

\unitname: \weightunit, \lengthunit, \currencyunit.

\end{document}
```

No mapping is required for the `en-MT` locale as it can pick up `\captionsenglish` when `\TrackLangAddToHook` (used by `\TrackLangAddToCaptions`) queries the root language label after failing to find the language hook from the dialect label.

Some of the predefined tracklang dialects come with a mapping to the closest matching babel dialect label. For example, the option ngermanDE listed in table 1.3 automatically provides a mapping to `ngerman`. Since a tracklang dialect label can only map to one babel label, this can be problematic for synonymous labels such as `british`/`UKenglish` or `american`/`USenglish`. The default mappings used by tracklang are shown in table 1.3.

5 Adding Support for Language Tracking

If you are writing a package that *sets up* the document languages (rather than a package that provides multilingual support if the user has already loaded a language package) then you can load tracklang and use the commands below to help other packages track your provided languages.

The tracklang package can be loaded using

```
\input tracklang
```

or (LaTeX only)

```
\RequirePackage{tracklang}
```

When using LaTeX, there's a difference between the two. The first case prevents tracklang from picking up the document class options but skips the check for known language packages. This check is redundant since this is the language package, so the thing to decide is whether or not to allow the user to set up the localisation information through the document class options.

(If you just use `\input`, there's a test at the start of `tracklang.tex` to determine if it's already been loaded, so you don't need to worry if the user has already loaded it.)

When the language is set (using commands like `\selectlanguage`) it would be convenient to users and other packages if the following command is also used:

`\SetCurrentTrackedDialect`

```
\SetCurrentTrackedDialect{⟨dialect⟩}
```

where ⟨*dialect*⟩ may the tracklang dialect label, or the mapped label previously set through `\SetTrackedDialectLabelMap`, described below, or the language label (in which case the last dialect to be tracked with that root language will be assumed).

This will make the following commands available which may be of use to other packages:

- `\CurrentTrackedDialect` The dialect label recognised by tracklang (which may not be the same as ⟨*dialect*⟩).
- `\CurrentTrackedLanguage` The root language label used by tracklang.
- `\CurrentTrackedDialectModifier` The dialect modifier.
- `\CurrentTrackedDialectVariant` The dialect variant.
- `\CurrentTrackedDialectScript` The dialect script. Note that if tracklang-scripts is also loaded, this allows the script direction to be accessed using

  ```
  \TrackLangScriptAlphaToDir{\CurrentTrackedDialectScript}
  ```

See Section 6.5 for further details.

- `\CurrentTrackedDialectSubLang` The dialect sub-language code.
- `\GetTrackedDialectAdditional` The dialect's additional information.
- `\CurrentTrackedIsoCode` The dialect's root language ISO code. (The first found in the sequence 639-1, 639-2, 639-3.)
- `\CurrentTrackedRegion` The dialect's ISO 3166-1 region code.
- `\CurrentTrackedLanguageTag` The dialect's language tag.

(Without this automated use of `\SetCurrentTrackedDialect`, the same information can be picked up using commands like `\GetTrackedDialectScript`, but that's less convenient, especially if `\languagename` needs to be converted to ⟨*dialect*⟩. See the accompanying sample file `sample-setlang.tex` for an example.)

When the user requests a particular dialect through your language package, you can notify tracklang of this choice using

```
\TrackPredefinedDialect{⟨dialect⟩}
```

if the dialect label is recognised by tracklang (all those listed in tables 1.1, 1.2 and 1.3).

If there's no matching dialect predefined by tracklang, you can just use `\TrackLocale` or `\TrackLanguageTag` (described in Section 3) with the appropriate ISO codes *if you're not providing caption hooks*.

If you are providing a captions hook mechanism in the form `\captions`⟨*dialect*⟩, then if ⟨*dialect*⟩ doesn't match the corresponding tracklang dialect label, you can provide a mapping using `\SetTrackedDialectLabelMap`, described below.

For compatibility with pre version 1.3, if the dialect isn't predefined by tracklang, then you can use:

`\AddTrackedDialect`

```
\AddTrackedDialect{⟨dialect⟩}{⟨root language label⟩}
```

where ⟨*root language label*⟩ is the label for the dialect's root language (table 1.2) and ⟨*dialect*⟩ matches the captions hook. If the dialect is already in the tracked dialect list, it won't be added again. If the root language is already in the tracked language list, it won't be added again. As from version 1.3 this additionally defines:

`\TrackLangLastTrackedDialect`

```
\TrackLangLastTrackedDialect
```

to ⟨*dialect*⟩ for convenient reference if required. Note that `\AddTrackedDialect` is internally used by commands like `\TrackPredefinedDialect`, `\TrackLocale` and `\TrackLanguageTag`.

(New to version 1.3.) Many of the tracklang dialect labels don't have a corresponding match in various language packages. For example, tracklang provides `ngermanDE` but the closest

match in babel is ngerman. This means that the caption hook \captionsngerman can't be accessed through

```
\csname captions\CurrentTrackedDialect\endcsname
```

in the resource files. In this case, a mapping may be defined between the tracklang dialect label and the closest matching label used by the language hooks. This is done through

\SetTrackedDialectLabelMap

\SetTrackedDialectLabelMap{⟨*from*⟩}{⟨*to*⟩}

where ⟨*from*⟩ is the tracklang label and ⟨*to*⟩ is the language hook label. For example:

```
\TrackLanguageTag{de-AR-1996}
\SetTrackedDialectLabelMap{\TrackLangLastTrackedDialect}{ngerman}
```

Since \TrackLanguageTag internally uses \AddTrackedDialect the dialect label created by tracklang can be accessed using \TrackLangLastTrackedDialect. This means that \TrackLangAddToCaptions can now find the \captionsngerman hook even though the tracklang dialect label isn't ngerman.

(New to version 1.3.) If the root language label is recognised by tracklang, you can add the ISO codes using:

\AddTrackedLanguageIsoCodes

\AddTrackedLanguageIsoCodes{⟨*root language*⟩}

As from v1.3, you can also provide a modifier for a given dialect using:

\SetTrackedDialectModifier

\SetTrackedDialectModifier{⟨*dialect*⟩}{⟨*value*⟩}

where ⟨*dialect*⟩ is the dialect label and ⟨*value*⟩ is the modifier value. For example:

```
\AddTrackedDialect{oldgerman}{german}
\AddTrackedLanguageIsoCodes{german}
\SetTrackedDialectModifier{oldgerman}{old}
```

Note that no sanitization is performed on ⟨*value*⟩ when the modifier is set explicitly through \SetTrackedDialectModifier, since it's assumed that any package that specifically sets the modifier in this way is using a sensible labelling system. If the modifier is obtained through commands like \TrackLocale, then the modifier is sanitized as the value may have been obtained from the operating system and there's no guarantee that it won't contain problematic characters.

The modifier is typically obtained by parsing locale information in POSIX format.

⟨*language*⟩[_⟨*territory*⟩][.⟨*codeset*⟩][@⟨*modifier*⟩]

whereas the variant is typically obtained by parsing the language tag.

The information provided in the commands below (such as the script) are typically obtained by parsing the language tag. For example, with Serbian in the Latin alphabet the modifier would be latin whereas the script would be Latn:

```
\AddTrackedDialect{serbianlatin}{serbian}
\AddTrackedLanguageIsoCodes{serbian}
\SetTrackedDialectModifier{serbianlatin}{latin}
\SetTrackedDialectScript{serbianlatin}{Latn}
```

As from v1.3, you can provide a script (for example, “Latn” or “Cyrl”) using:

\SetTrackedDialectScript

```
\SetTrackedDialectScript{⟨dialect⟩}{⟨value⟩}
```

where ⟨*dialect*⟩ is the dialect label and ⟨*value*⟩ is the four letter script identifier. For example:

```
\AddTrackedDialect{serbiancyrl}{serbian}
\AddTrackedLanguageIsoCodes{serbian}
\SetTrackedDialectScript{serbiancyrl}{Cyrl}
```

As from v1.3, you can provide a variant for a given dialect using:

\SetTrackedDialectVariant

```
\SetTrackedDialectVariant{⟨dialect⟩}{⟨value⟩}
```

For example:

```
\AddTrackedDialect{german1901}{german}
\SetTrackedDialectVariant{german1901}{1901}
```

As from v1.3, you can also provide a sub-language using:

\SetTrackedDialectSubLang

```
\SetTrackedDialectSubLang{⟨dialect⟩}{⟨value⟩}
```

where ⟨*dialect*⟩ is the dialect label and ⟨*value*⟩ is the code. For example:

```
\AddTrackedDialect{mandarin}{chinese}
\AddTrackedLanguageIsoCodes{chinese}
\SetTrackedDialectSubLang{mandarin}{cmn}
\AddTrackedIsoLanguage{639-3}{cmn}{mandarin}
```

As from v1.3, you can also provide additional information using:

\SetTrackedDialectAdditional

```
\SetTrackedDialectAdditional{⟨dialect⟩}{⟨value⟩}
```

where ⟨*dialect*⟩ is the dialect label and ⟨*value*⟩ is the additional information.

(New to version 1.3.) If the root language isn't recognised by tracklang (not listed in table 1.2), then it can be defined (but not tracked at this point) using:

\TrackLangNewLanguage

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

where ⟨*language name*⟩ is the root language name, ⟨*639-1 code*⟩ is the ISO 639-1 code for that language (may be empty if there isn't one), ⟨*639-2 (T)*⟩ is the ISO 639-2 (T) code for that language (may be empty if there isn't one), ⟨*639-2 (B)*⟩ is the ISO 639-2 (B) code for that language (may be empty if it's the same as ⟨*639-2 (T)*⟩), ⟨*639-3*⟩ is the ISO 639-3 code for

that language (empty if the same as the 639-2 code), ⟨*3166-1*⟩ is the territory ISO code for languages that are only spoken in one territory (should be empty if the language is spoken in multiple territories), and ⟨*default script*⟩ is the default script (empty if disputed or varies according to region).

You can then track this language using `\AddTrackedDialect` for dialects or, if no regional variant is needed, you can instead use:

`\AddTrackedLanguage`

`\AddTrackedLanguage{`⟨*root language label*⟩`}`

This is equivalent to `\AddTrackedDialect{`⟨*root language label*⟩`}{`⟨*root language label*⟩`}`.

Suppose I want to create a language package `alien.sty` that defines the `martian` language with regional dialects `lowermartian` and `uppermartian`. First, let's suppose that tracklang recognises the root language `martian`:

```
\ProvidesPackage{alien}

\input{tracklang}% v1.3

\DeclareOption{martian}{%
  \TrackPredefinedDialect{martian}
}
\DeclareOption{lowermartian}{%
  \AddTrackedDialect{lowermartian}{martian}
  \AddTrackedIsoCodes{martian}
  \AddTrackedIsoLanguage{3166-1}{YY}{lowermartian}
  % other attributes such as
  % \SetTrackedDialectVariant{lowermartian}{...}
}
\DeclareOption{uppermartian}{%
  \AddTrackedDialect{uppermartian}{martian}
  \AddTrackedIsoCodes{martian}
  \AddTrackedIsoLanguage{3166-1}{XX}{uppermartian}
  % other attributes such as
  % \SetTrackedDialectVariant{uppermartian}{...}
}

\ProcessOptions

\newcommand*{\selectlanguage}[1]{%
  \def\languagename{#1}%
  % other stuff
  \SetCurrentTrackedDialect{#1}%
}

\AnyTrackedLanguages
{
  \ForEachTrackedDialect{\thisdialect}
  {%
    \TrackLangRequireDialect{alien}{\thisdialect}
  }
}
```

The caption commands and language set up are in the files `alien-`⟨*tag*⟩`.ldf` as in the examples from Section 4.1. This allows for the user having already loaded tracklang before alien and used `\TrackLangFromEnv` to pick up the locale from the operating system's environment variables. (For example, they may have `LANG` set to `xx_YY`.)

The resource files may need to set the mapping between the tracklang dialect label and the alien dialect label. For example, in `alien-xx-YY.ldf`:

```
\TrackLangProvidesResource{xx-YY}

\TrackLangRequireResource{martian}% load common elements

\newcommand{\captionslowermartian}{%
  \captionsmartian
  \def\contentsname{X'flurp}% regional variation
}

\SetTrackedDialectLabelMap{\CurrentTrackedDialect}{lowermartian}
```

Now let's consider the case where tracklang doesn't know about the `martian` language. In this case the user can't track the dialect until the root language has been defined, so the user can't use `\TrackLangFromEnv` before using the alien package.

With tracklang v1.3. The new root language can be defined with a minor adjustment to the above code:

```
\ProvidesPackage{alien}

\input{tracklang}% needs v1.3

\TrackLangIfKnownLang{martian}
{}% tracklang already knows about the martian language
{
  % tracklang doesn't known about the martian language, so define it
  % with ISO 639-1 (xx) and ISO 639-2 (xxx) codes:
  \TrackLangNewLanguage{martian}{xx}{xxx}{}{}{}{Latn}
}
```

The rest is as before.

Now other package writers who want to provide support for the Martian dialects can easily detect which language options the user requested through my package, *without needing to know anything about my alien package*.

6 The Code

6.1 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}[2019/08/31 v1.3.7 (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.

```
\ifx\@tracklang@languages\@empty
```

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

```
  \@ifundefined{bbl@loaded}%
  {%
```

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 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}}%
         }%
      }%
      {%
         \@for\this@language:=\xpg@loaded\do{%
           \@ifundefined{@tracklang@add@\this@language}%
           {%
              \PackageWarning{tracklang}%
                {Adding unknown polyglossia language `\this@language'}%

              \AddTrackedLanguage{\this@language}%
           }%
           {\@nameuse{@tracklang@add@\this@language}}%
         }%
      }%
    }%
    {%
```

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 babel has been loaded, then it's an older version that hasn't defined \bbl@loaded so check for this.

```
        \@ifpackageloaded{babel}
        {%
          \PackageInfo{tracklang}{babel loaded but
          \string\bbl@loaded\space not defined. Will attempt
          to track known languages.}%
          \@for\this@language:=\@tracklang@declaredoptions\do{%
            \@ifundefined{captions\this@language}%
            {}%
            {\@nameuse{@tracklang@add@\this@language}}%
          }%
        }%
        {}%
      }%
    }%
  }%
}%
{%
```

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

```
\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
\let\@tracklang@add@German\@tracklang@add@german
\@namedef{@tracklang@add@German1997}{\@tracklang@add@ngerman}
\let\@tracklang@add@Greek\@tracklang@add@greek
\let\@tracklang@add@Polutoniko\@tracklang@add@polutoniko
\let\@tracklang@add@Hebrew\@tracklang@add@hebrew
\let\@tracklang@add@Hungarian\@tracklang@add@hungarian
\let\@tracklang@add@Icelandic\@tracklang@add@icelandic
\let\@tracklang@add@Irish\@tracklang@add@irish
\let\@tracklang@add@Italian\@tracklang@add@italian
\let\@tracklang@add@Latin\@tracklang@add@latin
\let\@tracklang@add@LowerSorbian\@tracklang@add@lowersorbian
\let\@tracklang@add@Magyar\@tracklang@add@magyar
\let\@tracklang@add@Nynorsk\@tracklang@add@nynorsk
\let\@tracklang@add@Norsk\@tracklang@add@norsk
```

```
      \let\@tracklang@add@Polish\@tracklang@add@polish
      \let\@tracklang@add@Portuguese\@tracklang@add@portuguese
      \let\@tracklang@add@Romanian\@tracklang@add@romanian
      \let\@tracklang@add@Russian\@tracklang@add@russian
      \let\@tracklang@add@Scottish\@tracklang@add@scottish
      \let\@tracklang@add@Serbian\@tracklang@add@serbian
      \let\@tracklang@add@Slovak\@tracklang@add@slovak
      \let\@tracklang@add@Slovene\@tracklang@add@slovene
      \let\@tracklang@add@Spanish\@tracklang@add@spanish
      \let\@tracklang@add@Swedish\@tracklang@add@swedish
      \let\@tracklang@add@Turkish\@tracklang@add@turkish
      \let\@tracklang@add@Ukrainian\@tracklang@add@ukrainian
      \let\@tracklang@add@UpperSorbian\@tracklang@add@uppersorbian
      \let\@tracklang@add@Welsh\@tracklang@add@welsh
```

Now iterate through the declared languages:

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

Add from babel

```
    \@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 babel has been loaded with serbian, then the script needs to be set to `Latn`. (The Cyrillic script is provided with serbianc.)

```
    \ifx\captionsserbian\undefined
    \else
      \SetTrackedDialectScript{serbian}{Latn}%
    \fi
  }
```

End of check for language packages

```
  \fi
```

6.2 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{%
 2019/08/31 v1.3.7 (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
      #3%
    \fi
  \fi
}
```

`\ifcsname` is an eTEX primitive. Need to check if it's defined.

```
\ifx\ifcsname\undefined
```

Not using eTEX.

```
  \long\def\@tracklang@ifundef#1#2#3{%
    \expandafter\ifx\csname #1\endcsname\relax
      #2%
    \else
      #3%
    \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,\@nil,\@nil\@@ #1{#3}%
    \fi
  }
  \long\def\@tracklang@forloop#1,#2,#3\@@ #4#5{%
   \def #4{#1}%
   \ifx#4\@tracklang@nnil
   \else
     #5%
     \def #4{#2}%
     \ifx#4\@tracklang@nnil
     \else
       #5%
       \@tracklang@iforloop #3\@@ #4{#5}%
     \fi
   \fi
  }
  \long\def\@tracklang@iforloop#1,#2\@@ #3#4{%
    \def#3{#1}%
    \ifx#3\@tracklang@nnil
      \expandafter
        \@tracklang@fornoop
    \else
      #4\relax
      \expandafter\@tracklang@iforloop
    \fi
    #2\@@ #3{#4}%
  }
  \long\def\@tracklang@fornoop#1\@@ #2#3{}
\else
  \let\@tracklang@for\@for
\fi
```

\@tracklang@namedef

```
\ifx\@namedef\undefined
  \def\@tracklang@namedef#1{\expandafter\def\csname#1\endcsname}
\else
  \let\@tracklang@namedef\@namedef
\fi
```

\@tracklang@enamedef

```
\def\@tracklang@enamedef#1{\expandafter\edef\csname#1\endcsname}
```

\@tracklang@nameuse

```
\def\@tracklang@nameuse#1{%
  \@tracklang@ifundef{#1}{}{\csname#1\endcsname}%
}
```

\@tracklang@sanitize

```
\ifx\@onelevel@sanitize\undefined
  \def\@tracklang@sanitize#1{%
    \edef#1{\expandafter\@tracklang@strip@prefix\meaning#1}%
  }
  \def\@tracklang@strip@prefix#1>{}
\else
  \let\@tracklang@sanitize\@onelevel@sanitize
\fi
```

\@tracklang@firstoftwo

```
\def\@tracklang@firstoftwo#1#2{#1}
```

\@tracklang@secondoftwo

```
\def\@tracklang@secondoftwo#1#2{#2}
```

\@tracklang@err

```
\ifx\PackageError\undefined
  \def\@tracklang@err#1#2{%
    \errhelp{#2}%
    \errmessage{tracklang: #1}}
\else
  \def\@tracklang@err#1#2{\PackageError{tracklang}{#1}{#2}}
\fi
```

\ifTrackLangShowWarnings Allow user to switch warnings on or off.

```
\newif\ifTrackLangShowWarnings
\TrackLangShowWarningstrue
```

\@tracklang@pkgwarn Provided for related packages such as texosquery.

```
\ifx\PackageWarning\undefined
  \def\@tracklang@pkgwarn#1#2{%
    \ifTrackLangShowWarnings
      {%
        \newlinechar=`\^^J
        \def\MessageBreak{^^J}%
        \message{^^J#1 Warning: #2 on line \the\inputlineno.^^J}%
      }%
    \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
```

\@tracklang@IfFileExists

```
\ifx\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
```

Provide a way to query the environment variables LC_ALL or LANG to determine the region and language. The result is stored in \TrackLangEnv if it can be obtained. If it can't be obtained, \TrackLangEnv is set to empty. Also define \TrackLangQueryOtherEnv{⟨*name*⟩} to query LC_ALL, ⟨*name*⟩, 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{%
  \ifx\TrackLangEnv\empty
  \else
    \ifx\TrackLangEnv\@tracklang@locale@posix
      \def\TrackLangEnv{}%
```

```
    \else
      \ifx\TrackLangEnv\@tracklang@locale@c
        \def\TrackLangEnv{}%
      \else
        \expandafter\@@tracklang@checklocale
           \TrackLangEnv\empty\relax
      \fi
    \fi
  \fi
}
```

`\@@tracklang@checklocale` Check for leading slash.

```
\def\@@tracklang@checklocale#1#2\relax{%
  \ifx#1/\relax
    \def\TrackLangEnv{}%
  \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\TrackLangQueryEnv{%
     \@tracklang@warn{\string\TrackLangQueryEnv\space
     non-operational as shell escape has been disabled}%
     \def\TrackLangEnv{}%
   }%
   \def\TrackLangQueryOtherEnv##1{%
     \@tracklang@warn{\string\TrackLangQueryOtherEnv{##1}\space
     non-operational as shell escape has been disabled}%
     \def\TrackLangEnv{}%
   }%
  }%

  \ifx\pdfshellescape\undefined
    \ifx\shellescape\undefined
```

Can't determine if the shell escape has been enabled.

```
      \def\@tracklang@tryshellescape#1{%
        \def\TrackLangQueryEnv{%
          \@tracklang@warn{\string\TrackLangQueryEnv\space
          non-operational as can't determine if the
          shell escape has been enabled. (Consider using
          eTeX or pdfTeX.)}%
          \def\TrackLangEnv{}%
        }%
        \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{}%
          }%
        }%
      \else
```

\shellescape is defined. Check no one's been messing around with it and set it to \relax.

```
        \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
    {%
```

\TrackLangQueryEnv

```
      \def\TrackLangQueryEnv{%
        \begingroup\endlinechar=-1\relax
        \everyeof{\noexpand}%
        \edef\x{\endgroup\def\noexpand\TrackLangEnv{%
          \@tracklang@input|"kpsewhich --var-value LC_ALL" }}\x
        \@tracklang@checklocale
        \ifx\TrackLangEnv\empty
          \begingroup\endlinechar=-1\relax
          \everyeof{\noexpand}%
          \edef\x{\endgroup\def\noexpand\TrackLangEnv{%
            \@tracklang@input|"kpsewhich --var-value LANG" }}\x
```

Not sure if a path is likely to occur with `kpsewhich` but check for it just in case.

```
      \@tracklang@checklocale
      \ifx\TrackLangEnv\empty
```

Try texosquery if available.

```
        \ifx\TeXOSQueryLocale\undefined
          \@tracklang@warn{Locale environment variables
           unavailable (tried LC\string_ALL and LANG)}%
        \else
          \@tracklang@info{Using texosquery to find locale}%
          \TeXOSQueryLocale\TrackLangEnv
          \ifx\TrackLangEnv\empty
            \@tracklang@warn{Locale can't be found
            (tried querying LC\string_ALL and LANG variables and
            tried using texosquery)}%
          \fi
        \fi
      \fi
    \fi
  }%
```

`\TrackLangQueryOtherEnv`

```
  \def\TrackLangQueryOtherEnv#1{%
    \begingroup\endlinechar=-1\relax
    \everyeof{\noexpand}%
    \edef\x{\endgroup\def\noexpand\TrackLangEnv{%
      \@tracklang@input|"kpsewhich --var-value LC_ALL" }}\x
    \@tracklang@checklocale
    \ifx\TrackLangEnv\empty
      \begingroup\endlinechar=-1\relax
      \everyeof{\noexpand}%
      \edef\x{\endgroup\def\noexpand\TrackLangEnv{%
        \@tracklang@input|"kpsewhich --var-value #1" }}\x
      \@tracklang@checklocale
      \ifx\TrackLangEnv\empty
        \begingroup\endlinechar=-1\relax
        \everyeof{\noexpand}%
        \edef\x{\endgroup\def\noexpand\TrackLangEnv{%
          \@tracklang@input|"kpsewhich --var-value LANG"}}\x
        \@tracklang@checklocale
        \ifx\TrackLangEnv\empty
```

Try texosquery if available.

```
          \ifx\TeXOSQueryLocale\undefined
          \@tracklang@warn{Locale environment variables unavailable
           (tried LC\string_ALL, #1 and LANG)}%
          \else
            \@tracklang@info{Using texosquery to find locale}%
            \TeXOSQueryLocale\TrackLangEnv
            \ifx\TrackLangEnv\empty
              \@tracklang@warn{Locale can't be found
              (tried querying LC\string_ALL, #1 and LANG variables and
              tried using texosquery)}%
            \fi
          \fi
        \fi
      \fi
    \fi
  }%

}%
```

```
  \else
```

\directlua is defined, so we can query it through Lua:

\TrackLangQueryEnv

```
\def\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.

```
      \ifx\TeXOSQueryLocale\undefined
      \@tracklang@warn{Locale can't be found through Lua
       (tried querying LC\string_ALL 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 and LANG variables and
          os.setlocale(nil) and tried using texosquery)}%
        \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
    end
    tex.print(l)}}%
    \ifx\TrackLangEnv\empty
```

Try texosquery if available.

```
          \ifx\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
    }

\fi
```

Allowed formats for the localisation environment variables are

⟨*iso-lang*⟩[_⟨*iso-territory*⟩][.⟨*encoding*⟩][@⟨*modifier*⟩]

(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\@TrackLangEnvVariant{}%
    \def\@TrackLangEnvModifier{}%
    \def\@TrackLangEnvScript{}%
    \def\@TrackLangEnvAdditional{}%
```

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
```

Parse language and territory.

```
      \ifx\@tracklang@result\empty
      \else
        \expandafter\@tracklang@split@underscoreorhyp\expandafter
         {\@tracklang@result}%
        \let\@TrackLangEnvLang\@tracklang@split@pre
        \let\@TrackLangEnvTerritory\@tracklang@split@post
      \fi
    \fi
  }
```

`@tracklang@split@underscoreorhyp` Split on either an underscore or a hyphen and store the results 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@p`
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
    }
    \gdef\@@tracklang@split@underscore#1_#2_#3\@tracklang@end@split@underscore{%
      \def\@tracklang@split@pre{#1}%
      \ifx\relax#3\relax
        \def\@tracklang@split@post{#2}%
```

```
    \else
      \@tracklang@split@underscore@remainder#2_#3%
    \fi
  }
  \gdef\@tracklang@split@underscore@remainder#1__\relax{%
    \def\@tracklang@split@post{#1}%
  }
}
```

@tracklang@split@otherunderscore As above but where underscore has catcode 12.

```
{
  \catcode`\_12\relax
  \gdef\@tracklang@split@otherunderscore#1{%
    \@@tracklang@split@otherunderscore#1__\relax\@tracklang@end@split@underscore
  }
  \gdef\@@tracklang@split@otherunderscore#1_#2_#3\@tracklang@end@split@underscore{%
    \def\@tracklang@split@pre{#1}%
    \ifx\relax#3\relax
      \def\@tracklang@split@post{#2}%
    \else
      \@tracklang@split@otherunderscore@remainder#2_#3%
    \fi
  }
  \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#1{%
    \@@tracklang@split@hyphen#1--\relax\@tracklang@end@split@hyphen
  }
  \gdef\@@tracklang@split@hyphen#1-#2-#3\@tracklang@end@split@hyphen{%
    \def\@tracklang@split@pre{#1}%
    \ifx\relax#3\relax
      \def\@tracklang@split@post{#2}%
    \else
      \@tracklang@split@hyphen@remainder#2-#3%
    \fi
  }
  \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}%
```

Need to use \csname here as can't use internal commands.

```
    \expandafter\def\csname @TrackLangEnvModifier\endcsname{#2}%
```

Sanitize in case it contains any special characters.

```
    \csname @tracklang@sanitize\expandafter\endcsname
      \csname @TrackLangEnvModifier\endcsname
  }
}
```

6.2.1 Internal Lists

\@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 `\@tracklang@ifinlist{⟨item⟩}{⟨list⟩}{⟨true part⟩}{⟨false part⟩}`

Checks if ⟨*item*⟩ is in ⟨*list*⟩. (Performs a one-level expansion on ⟨*list*⟩ but no expansion on ⟨*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

\@tracklang@add{⟨*item*⟩}{⟨*list cs*⟩}

Adds an item to the list given by ⟨*list cs*⟩. Does nothing if ⟨*item*⟩ is empty or is already in the list. The ⟨*item*⟩ is fully expanded before being added.

```
\def\@tracklang@add#1#2{%
```

First find out if the item is empty.

```
  \edef\@tracklang@element{#1}%
  \ifx\@tracklang@element\empty
```

Item is empty, so do nothing.

```
  \else
    \expandafter\@tracklang@ifinlist\expandafter{\@tracklang@element}#2%
    {%
```

Already in list, so do nothing.

```
    }%
    {%
```

Not in list, so add.

```
      \ifx\empty#2\relax
        \let#2\@tracklang@element
      \else
        \edef#2{#2,\@tracklang@element}%
      \fi
    }%
  \fi
}
```

\AddTrackedDialect

\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}%
 {\@tracklang@enamedef{@tracklang@todialect@#2}{#1}}%
 {%
   \def\@tracklang@lang{#1}%
   \expandafter\@tracklang@add\expandafter\@tracklang@lang
     \csname @tracklang@todialect@#2\endcsname
 }%
```

Provide a convenient way of referencing the last dialect to be tracked.

```
  \edef\TrackLangLastTrackedDialect{#1}%
}
```

\AddTrackedLanguage

\AddTrackedLanguage{⟨*language name*⟩}

Add a dialect.

```
\def\AddTrackedLanguage#1{%
  \AddTrackedDialect{#1}{#1}%
}
```

6.2.2 Known Languages

\@tracklang@known@langs List of known (root) languages (that may or may not be tracked).

```
\def\@tracklang@known@langs{}
```

\TrackLangNewLanguage `\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}%
 \ifx\@tracklang@tmp\empty
 \else
   \@tracklang@enamedef{@tracklang@knownisolang@#2}{#1}%
   \@tracklang@enamedef{@tracklang@knowniso@639@1@#1}{#2}%
 \fi
 \edef\@tracklang@tmp{#3}%
 \ifx\@tracklang@tmp\empty
 \else
   \@tracklang@enamedef{@tracklang@knownisolang@#3}{#1}%
   \@tracklang@enamedef{@tracklang@knowniso@639@2@#1}{#3}%
 \fi
 \edef\@tracklang@tmp{#4}%
 \ifx\@tracklang@tmp\empty
 \else
   \@tracklang@enamedef{@tracklang@knowniso@639@2B@#1}{#4}%
 \fi
 \edef\@tracklang@tmp{#5}%
 \ifx\@tracklang@tmp\empty
 \else
   \@tracklang@enamedef{@tracklang@knownisolang@#5}{#1}%
   \@tracklang@enamedef{@tracklang@knowniso@639@3@#1}{#5}%
 \fi
 \edef\@tracklang@tmp{#6}%
 \ifx\@tracklang@tmp\empty
 \else
   \@tracklang@enamedef{@tracklang@knowniso@3166@#1}{#6}%
 \fi
 \edef\@tracklang@tmp{#7}%
 \ifx\@tracklang@tmp\empty
 \else
   \@tracklang@enamedef{@tracklang@knowniso@script@#1}{#7}%
 \fi
}
```

\TrackLangIfKnownLang `\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

\TrackLangIfKnownIsoTwoLetterLang{⟨*language*⟩}{⟨*true*⟩}{⟨*false*⟩}

Checks if the given language has an ISO 639-1 language code (but is not necessarily tracked).

```
\def\TrackLangIfKnownIsoTwoLetterLang#1#2#3{%
  \@tracklang@ifundef{@tracklang@knowniso@639@1@#1}%
  {#3}%
  {#2}%
}
```

TrackLangGetKnownIsoTwoLetterLang

\TrackLangGetKnownIsoTwoLetterLang{⟨*language*⟩}

Gets the ISO 639-1 language code for the given language.

```
\def\TrackLangGetKnownIsoTwoLetterLang#1{%
  \@tracklang@nameuse{@tracklang@knowniso@639@1@#1}%
}
```

rackLangIfKnownIsoThreeLetterLang

\TrackLangIfKnownIsoThreeLetterLang{⟨*language*⟩}{⟨*true*⟩}{⟨*false*⟩}

Checks if the given language has an ISO 639-2 language code (but is not necessarily tracked).

```
\def\TrackLangIfKnownIsoThreeLetterLang#1#2#3{%
  \@tracklang@ifundef{@tracklang@knowniso@639@2@#1}%
  {#3}%
  {#2}%
}
```

ackLangGetKnownIsoThreeLetterLang

\TrackLangGetKnownIsoThreeLetterLang{⟨*language*⟩}

Gets the ISO 639-2 language code.

```
\def\TrackLangGetKnownIsoThreeLetterLang#1{%
  \@tracklang@nameuse{@tracklang@knowniso@639@2@#1}%
}
```

ackLangIfKnownIsoThreeLetterLangB

\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@knowniso@639@2B@#1}%
  {#3}%
  {#2}%
}
```

ckLangGetKnownIsoThreeLetterLangB

\TrackLangGetKnownIsoThreeLetterLangB{⟨*language*⟩}

Gets the ISO 639-2 (B) language code.

```
\def\TrackLangGetKnownIsoThreeLetterLangB#1{%
  \@tracklang@nameuse{@tracklang@knowniso@639@2B@#1}%
}
```

\TrackLangIfKnownLangFromIso

\TrackLangIfKnownLangFromIso{⟨*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\TrackLangIfKnownLangFromIso#1#2#3{%
  \@tracklang@ifundef{@tracklang@knownisolang@#1}%
  {#3}%
  {#2}%
}
```

\TrackLangGetKnownLangFromIso

\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

\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@knowniso@3166@#1}%
  {#3}%
  {#2}%
}
```

\TrackLangGetKnownCountry

\TrackLangGetKnownCountry{⟨*language*⟩}

Fetches the ISO 3166-1 country code for the given language.

```
\def\TrackLangGetKnownCountry#1{%
  \@tracklang@nameuse{@tracklang@knowniso@3166@#1}%
}
```

\TrackLangGetDefaultScript

\TrackLangGetDefaultScript{⟨*language*⟩}

Gets the default script for the given root language label.

```
\def\TrackLangGetDefaultScript#1{%
  \@tracklang@nameuse{@tracklang@knowniso@script@#1}%
}
```

\TrackLangIfHasDefaultScript

\TrackLangIfHasDefaultScript{⟨*language*⟩}{⟨*true*⟩}{⟨*false*⟩}

If there's a default script for ⟨*language*⟩, do ⟨*true*⟩ otherwise do ⟨*false*⟩.

```
\def\TrackLangIfHasDefaultScript#1#2#3{%
  \@tracklang@ifundef{@tracklang@knowniso@script@#1}{#3}{#2}%
}
```

6.2.3 Mappings

\AddTrackedIsoLanguage

\AddTrackedIsoLanguage{⟨*code type*⟩}{⟨*code*⟩}{⟨*language*⟩}

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.)

```
\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

\TrackedLanguageFromIsoCode{⟨*code type*⟩}{⟨*code*⟩}

Fetches the language label (or labels) associated with the given code.

```
\def\TrackedLanguageFromIsoCode#1#2{%
  \@tracklang@nameuse{@tracklang@#1@isotolang@#2}%
}
```

\TrackedIsoCodeFromLanguage

\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

\TrackedLanguageFromDialect{⟨*dialect*⟩}

Fetches the language name from the given dialect.

```
\def\TrackedLanguageFromDialect#1{%
 \@tracklang@nameuse{@tracklang@fromdialect@#1}%
}
```

\TrackedDialectsFromLanguage

\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

\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

\GetTrackedDialectModifier{⟨*dialect*⟩}

Get the modifier for ⟨*dialect*⟩.

```
\def\GetTrackedDialectModifier#1{%
  \@tracklang@nameuse{@tracklang@modifier@#1}%
}
```

\IfHasTrackedDialectModifier

\IfHasTrackedDialectModifier{⟨*dialect*⟩}{⟨*true*⟩}{⟨*false*⟩}

If there's a modifier for ⟨*dialect*⟩, do ⟨*true*⟩ otherwise do ⟨*false*⟩.

```
\def\IfHasTrackedDialectModifier#1#2#3{%
  \@tracklang@ifundef{@tracklang@modifier@#1}{#3}{#2}%
}
```

\SetTrackedDialectScript

\SetTrackedDialectScript{⟨*dialect*⟩}{⟨*value*⟩}

Set the script for ⟨*dialect*⟩. (For example, Latn or Cyrl.) Arguments are expanded.

```
\def\SetTrackedDialectScript#1#2{%
  \@tracklang@enamedef{@tracklang@script@#1}{#2}%
}
```

\GetTrackedDialectScript

\GetTrackedDialectScript{⟨*dialect*⟩}

Get the script for ⟨*dialect*⟩.

```
\def\GetTrackedDialectScript#1{%
  \@tracklang@nameuse{@tracklang@script@#1}%
}
```

\IfHasTrackedDialectScript

\IfHasTrackedDialectScript{⟨*dialect*⟩}{⟨*true*⟩}{⟨*false*⟩}

If there's a script for ⟨*dialect*⟩, do ⟨*true*⟩ otherwise do ⟨*false*⟩.

```
\def\IfHasTrackedDialectScript#1#2#3{%
  \@tracklang@ifundef{@tracklang@script@#1}{#3}{#2}%
}
```

\IfTrackedDialectIsScriptCs

\IfTrackedDialectIsScriptCs{⟨*dialect*⟩}{⟨*cs*⟩}{⟨*true*⟩}{⟨*false*⟩}

If the given tracked dialect has an associated script and that script code matches the replacement text for the control sequence ⟨*cs*⟩ then do ⟨*true*⟩ otherwise to ⟨*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 `\ifx` for the test, which means that this command can expand. The supplementary package tracklang-script provides control sequences for known ISO 15924 codes.

```
\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}%
  }%
}
```

\SetTrackedDialectVariant

\SetTrackedDialectVariant{⟨*dialect*⟩}{⟨*value*⟩}

Set the modifier for ⟨*dialect*⟩. (For example, old or new.) Arguments are expanded.

```
\def\SetTrackedDialectVariant#1#2{%
```

```
  \@tracklang@enamedef{@tracklang@variant@#1}{#2}%
}
```

\GetTrackedDialectVariant

\GetTrackedDialectVariant{⟨*dialect*⟩}

Get the modifier for ⟨*dialect*⟩.

```
\def\GetTrackedDialectVariant#1{%
  \@tracklang@nameuse{@tracklang@variant@#1}%
}
```

\IfHasTrackedDialectVariant

\IfHasTrackedDialectVariant{⟨*dialect*⟩}{⟨*true*⟩}{⟨*false*⟩}

If there's a modifier for ⟨*dialect*⟩, do ⟨*true*⟩ otherwise do ⟨*false*⟩.

```
\def\IfHasTrackedDialectVariant#1#2#3{%
  \@tracklang@ifundef{@tracklang@variant@#1}{#3}{#2}%
}
```

\SetTrackedDialectSubLang

\SetTrackedDialectSubLang{⟨*dialect*⟩}{⟨*value*⟩}

Set the sublang for ⟨*dialect*⟩. Arguments are expanded.

```
\def\SetTrackedDialectSubLang#1#2{%
  \@tracklang@enamedef{@tracklang@sublang@#1}{#2}%
}
```

\GetTrackedDialectSubLang

\GetTrackedDialectSubLang{⟨*dialect*⟩}

Get the sublang for ⟨*dialect*⟩.

```
\def\GetTrackedDialectSubLang#1{%
  \@tracklang@nameuse{@tracklang@sublang@#1}%
}
```

\IfHasTrackedDialectSubLang

\IfHasTrackedDialectSubLang{⟨*dialect*⟩}{⟨*true*⟩}{⟨*false*⟩}

If there's a sublang for ⟨*dialect*⟩, do ⟨*true*⟩ otherwise do ⟨*false*⟩.

```
\def\IfHasTrackedDialectSubLang#1#2#3{%
  \@tracklang@ifundef{@tracklang@sublang@#1}{#3}{#2}%
}
```

\SetTrackedDialectAdditional

\SetTrackedDialectAdditional{⟨*dialect*⟩}{⟨*value*⟩}

Set the extra for ⟨*dialect*⟩. Arguments are expanded.

```
\def\SetTrackedDialectAdditional#1#2{%
  \@tracklang@enamedef{@tracklang@extra@#1}{#2}%
}
```

\GetTrackedDialectAdditional

\GetTrackedDialectAdditional{⟨*dialect*⟩}

Get the extra for ⟨*dialect*⟩.

```
\def\GetTrackedDialectAdditional#1{%
  \@tracklang@nameuse{@tracklang@extra@#1}%
}
```

\IfHasTrackedDialectAdditional

\IfHasTrackedDialectAdditional{⟨*dialect*⟩}{⟨*true*⟩}{⟨*false*⟩}

If there's extra info for ⟨*dialect*⟩, do ⟨*true*⟩ otherwise do ⟨*false*⟩.

```
\def\IfHasTrackedDialectAdditional#1#2#3{%
  \@tracklang@ifundef{@tracklang@extra@#1}{#3}{#2}%
}
```

\GetTrackedLanguageTag

\GetTrackedLanguageTag{⟨*dialect*⟩}

Get the language tag for ⟨*dialect*⟩.

```
\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

\SetCurrentTrackedDialect{⟨*dialect*⟩}

Provided for use by language hooks to establish the current tracked dialect. This command doesn't change \languagename or hyphenation patterns etc. It just provides convenient

commands that can be accessed. The argument may be a tracklang dialect label or the language hook label from which a tracklang dialect label can be obtained or the root language label.

```
\def\SetCurrentTrackedDialect#1{%
  \edef\CurrentTrackedDialect{#1}%
  \IfTrackedDialect{\CurrentTrackedDialect}%
  {}%
  {%
```

Has a mapping from this dialect to a tracklang dialect been supplied?

```
    \IfHookHasMappingFromTrackedDialect{\CurrentTrackedDialect}%
    {%
      \IfTrackedDialect{\GetTrackedDialectFromMapping\CurrentTrackedDialect}%
      {%
        \edef\CurrentTrackedDialect{\GetTrackedDialectFromMapping
          {\CurrentTrackedDialect}}%
      }%
      {%
```

Has the root language name been supplied?

```
        \IfTrackedLanguage{#1}%
        {%
```

Get the last dialect to be tracked with this language.

```
          \edef\@tracklang@dialects{\TrackedDialectsFromLanguage{#1}}%
          \@tracklang@for\@tracklang@dialect:=\@tracklang@dialects\do{%
            \let\CurrentTrackedDialect\@tracklang@dialect
          }%
        }%
        {}%
      }%
    }%
    {%
```

Has the root language name been supplied?

```
      \IfTrackedLanguage{#1}%
      {%
```

Get the last dialect to be tracked with this language.

```
        \edef\@tracklang@dialects{\TrackedDialectsFromLanguage{#1}}%
        \@tracklang@for\@tracklang@dialect:=\@tracklang@dialects\do{%
          \let\CurrentTrackedDialect\@tracklang@dialect
        }%
      }%
      {}%
    }%
  }%
  \IfTrackedDialect{\CurrentTrackedDialect}%
  {%
    \edef\CurrentTrackedLanguage{%
      \TrackedLanguageFromDialect{\CurrentTrackedDialect}}%
    \edef\CurrentTrackedDialectModifier{%
      \GetTrackedDialectModifier{\CurrentTrackedDialect}}%
    \edef\CurrentTrackedDialectVariant{%
      \GetTrackedDialectVariant{\CurrentTrackedDialect}}%
```

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}}%
  }%
  {\def\CurrentTrackedRegion{}}%
```

Language code.

```
  \IfTrackedLanguageHasIsoCode{639-1}{\CurrentTrackedLanguage}%
  {%
    \edef\CurrentTrackedIsoCode{%
     \TrackedIsoCodeFromLanguage{639-1}{\CurrentTrackedLanguage}}%
  }%
  {%
    \IfTrackedLanguageHasIsoCode{639-2}{\CurrentTrackedLanguage}%
    {%
      \edef\CurrentTrackedIsoCode{%
       \TrackedIsoCodeFromLanguage{639-2}{\CurrentTrackedLanguage}}%
    }%
    {%
      \IfTrackedLanguageHasIsoCode{639-3}{\CurrentTrackedLanguage}%
      {%
        \edef\CurrentTrackedIsoCode{%
         \TrackedIsoCodeFromLanguage{639-3}{\CurrentTrackedLanguage}}%
      }%
      {%
        \def\CurrentTrackedIsoCode{}%
      }%
    }%
  }%
  {%
    \@tracklang@warn{Unknown dialect label `#1' passed to
      \string\SetCurrentTrackedDialect}%
    \edef\CurrentTrackedLanguage{\languagename}%
    \def\CurrentTrackedDialectModifier{}%
    \def\CurrentTrackedDialectVariant{}%
    \def\CurrentTrackedDialectScript{}%
    \def\CurrentTrackedDialectSubLang{}%
    \def\CurrentTrackedDialectAdditional{}%
    \def\CurrentTrackedIsoCode{}%
    \def\CurrentTrackedRegion{}%
    \def\CurrentTrackedLanguageTag{und}%
  }%
}
```

\AddTrackedLanguageIsoCodes

\AddTrackedLanguageIsoCodes{⟨*language*⟩}

Adds the ISO 639-1 and 639-2 ISO codes for the given language, which must have previously been declared using \TrackLangNewLanguage.

```
\def\AddTrackedLanguageIsoCodes#1{%
  \@tracklang@ifundef{@tracklang@knowniso@639@1@#1}%
  {}%
  {%
    \AddTrackedIsoLanguage\TwoLetterIsoLanguageCode
      {\csname @tracklang@knowniso@639@1@#1\endcsname}{#1}%
  }%
  \@tracklang@ifundef{@tracklang@knowniso@639@2@#1}%
  {}%
  {%
    \AddTrackedIsoLanguage\ThreeLetterIsoLanguageCode
       {\csname @tracklang@knowniso@639@2@#1\endcsname}{#1}%
```

Does it have a different 639-2 (B) code?

```
    \@tracklang@ifundef{@tracklang@knowniso@639@2B@#1}%
    {}%
    {%
      \AddTrackedIsoLanguage{\ThreeLetterIsoLanguageCode-T}%
         {\csname @tracklang@knowniso@639@2@#1\endcsname}{#1}%
      \AddTrackedIsoLanguage{\ThreeLetterIsoLanguageCode-B}%
         {\csname @tracklang@knowniso@639@2B@#1\endcsname}{#1}%
    }%
  }%
  \@tracklang@ifundef{@tracklang@knowniso@639@3@#1}%
  {}%
  {%
    \AddTrackedIsoLanguage\ThreeLetterExtIsoLanguageCode
      {\csname @tracklang@knowniso@639@3@#1\endcsname}{#1}%
  }%
}
```

\AddTrackedCountryIsoCode

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.

```
\def\AddTrackedCountryIsoCode#1{%
  \@tracklang@ifundef{@tracklang@knowniso@3166@#1}%
  {}%
  {%
    \AddTrackedIsoLanguage{3166-1}%
      {\csname @tracklang@knowniso@3166@#1\endcsname}{#1}%
  }%
}
```

6.2.4 Tracking Languages and Dialects

The commands here are provided to indicate that a language or dialect is active (tracked) in the document.

\TrackPredefinedDialect

\TrackPredefinedDialect{⟨*dialect label*⟩}

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).

```
\def\@tracklang@hasninthchar#1#2#3#4#5#6#7#8#9\@end@tracklang@hasninthchar{%
  \ifx\relax#9\relax
    \expandafter\@tracklang@secondoftwo
  \else
    \expandafter\@tracklang@firstoftwo
  \fi
}
```

\@tracklang@ifalpha Check if argument a, . . . , z or A, . . . , Z.

```
\def\@tracklang@ifalpha#1#2#3{%
  \ifx\relax#1\relax
```

First argument empty or \relax.

```
     #3%
   \else
    \ifnum\lccode‘#1<‘a\relax
      #3%
    \else
      \ifnum\lccode‘#1>‘z\relax
       #3%
      \else
```

Is alpha.

```
        #2%
      \fi
    \fi
  \fi
}
```

\@tracklang@ifdigit Check if argument is digit (0,. . . ,9).

```
\def\@tracklang@ifdigit#1#2#3{%
  \ifx\relax#1\relax
```

First argument empty or \relax.

```
     #3%
   \else
    \ifnum‘#1<‘0\relax
      #3%
    \else
      \ifnum‘#1>‘9\relax
       #3%
      \else
```

Is digit.

```
        #2%
      \fi
    \fi
  \fi
}
```

\@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@ifalldigits
```

```
    }%
    {%
      \def\@tracklang@next##1\@tracklang@nnil{%
        \expandafter\@tracklang@secondoftwo}%
    }%
  \fi
  \@tracklang@next
}
```

\@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}%
  {%
    \@tracklang@ifdigit{#1}{#2}{#3}%
  }%
}
```

\TrackLangIfAlphaNumericChar `\TrackLangIfAlphaNumericChar{⟨tag⟩}{⟨true⟩}{⟨false⟩}`

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}%
    {\expandafter\@tracklang@ifalphanumeric#1{#2}{#3}}%
  \fi
}
```

\TrackLangIfLanguageTag `\TrackLangIfLanguageTag{⟨tag⟩}{⟨true⟩}{⟨false⟩}`

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}%
      }%
      {#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
        \ifnum\lccode‘#3=‘#3\relax
          #4%
        \else
          #5%
        \fi
      \else
        #5%
      \fi
    \else
      #5%
    \fi
  }
```

\TrackLangIfRegionTag

\TrackLangIfRegionTag{⟨*tag*⟩}{⟨*true*⟩}{⟨*false*⟩}

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}%
    }%
  }
```

\@tracklang@ifregion@ii@tag

```
  \def\@tracklang@ifregion@ii@tag#1#2#3#4{%
    \ifnum\uccode‘#1=‘#1\relax
      \ifnum\uccode‘#2=‘#2\relax
        #3%
      \else
        #4%
      \fi
    \else
      #4%
    \fi
  }
```

\@tracklang@ifregion@iii@tag

```
  \def\@tracklang@ifregion@iii@tag#1#2#3#4#5{%
    \ifnum\uccode‘#1=‘#1\relax
      \ifnum\uccode‘#2=‘#2\relax
        \ifnum\uccode‘#3=‘#3\relax
          #4%
        \else
          #5%
        \fi
      \else
        #5%
      \fi
    \else
      #5%
    \fi
  }
```

\TrackLangIfScriptTag

\TrackLangIfScriptTag{⟨*tag*⟩}{⟨*true*⟩}{⟨*false*⟩}

Check if the argument is a script tag (four letter title case).

```
  \def\TrackLangIfScriptTag#1#2#3{%
    \expandafter\@tracklang@hasfifthchar#1\relax\relax\relax\relax\relax
      \@end@tracklang@hasfifthchar
    {#3}%
    {%
```

Has less than 5 characters.

```
      \expandafter\@tracklang@hasfourthchar#1\relax\relax\relax\relax
        \@end@tracklang@hasfourthchar
      {%
```

Has four characters. Are they title case? (First letter upper case, others lower case.)

```
       \expandafter\@tracklang@ifscripttag#1{#2}{#3}%
      }%
      {#3}%
    }%
  }
```

\@tracklang@ifscripttag

```
  \def\@tracklang@ifscripttag#1#2#3#4#5#6{%
    \ifnum\uccode‘#1=‘#1\relax
      \ifnum\lccode‘#2=‘#2\relax
        \ifnum\lccode‘#3=‘#3\relax
          \ifnum\lccode‘#4=‘#4\relax
            #5%
          \else
            #6%
          \fi
        \else
          #6%
        \fi
      \else
        #6%
      \fi
    \else
      #6%
    \fi
  }
```

\TrackLangIfVariantTag

\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\relax\relax\relax
        \@end@tracklang@hasninthchar
      {#3}%
      {#2}%
    }%
    {%
```

Less than 5 characters.

```
      \expandafter\@tracklang@hasfourthchar#1\relax\relax\relax\relax
        \@end@tracklang@hasfourthchar
      {%
```

Has 4 characters.

```
        \expandafter\@tracklang@ifvariant@iv@tag#1{#2}{#3}%
```

```
    }%
    {#3}%
  }%
}
```

`\@tracklang@ifvariant@iv@tag` 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
```

Split again if there's more.

```
     \ifx\@tracklang@split@post\empty
     \else
       \expandafter\@tracklang@split@underscoreorhyp\expandafter
         {\@tracklang@split@post}%
       \ifx\@tracklang@split@pre\empty
       \else
         \@tracklang@parse@extlang
       \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}%
       \ifx\@tracklang@split@pre\empty
       \else
         \@tracklang@parse@variant
       \fi
```

```
      \fi
    }%
    {}%
}
```

\TrackLanguageTag

\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{%
  \edef\@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{%
```

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}%
  }%
}
```

\@tracklang@parse@langtag

```
\def\@tracklang@parselangtag#1{%
```

Initialise.

```
  \def\@TrackLangEnvLang{}%
  \def\@TrackLangEnvSubLang{}%
  \def\@TrackLangEnvFirstSubLang{}%
  \def\@TrackLangEnvTerritory{}%
  \def\@TrackLangEnvCodeSet{}%
  \def\@TrackLangEnvVariant{}%
  \def\@TrackLangEnvModifier{}%
  \def\@TrackLangEnvScript{}%
  \def\@TrackLangEnvAdditional{}%
```

First split to determine language code.

```
  \@tracklang@split@underscoreorhyp{#1}%
```

Save the result.

```
  \let\@TrackLangEnvLang\@tracklang@split@pre
```

Is there anything else?

```
\ifx\@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.

```
    \ifx\@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 `tracklang-region-codes.tex` been loaded?

```
      \ifx\TrackLangIfKnownNumericRegion\undefined
        \@tracklang@input tracklang-region-codes.tex
      \fi
      \TrackLangIfKnownNumericRegion{\@tracklang@split@pre}%
      {%
        \edef\@TrackLangEnvTerritory{%
          \TrackLangNumericToAlphaIIRegion{\@tracklang@split@pre}%
        }%
      }%
      {%
        \let\@TrackLangEnvTerritory\@tracklang@split@pre
        \@tracklang@warn{Unrecognised numeric region code
          `\@tracklang@split@pre'}%
      }%
    }%
    {%
```

Is two letter alpha code.

```
      \let\@TrackLangEnvTerritory\@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
  }%
```

\GetTrackedDialectFromLanguageTag

\GetTrackedDialectFromLanguageTag{⟨*tag*⟩}{⟨*cs*⟩}

Find the tracked dialect that matches the given language tag and stores the dialect label in ⟨*cs*⟩. If no match found, ⟨*cs*⟩ will be empty. Just tests the root language, script, variant, sub-language and region. Doesn't check the additional information. As from v1.3.6, this sets \TrackedDialectClosestSubMatch to the closest sub-match.

```
  \def\GetTrackedDialectFromLanguageTag#1#2{%
```

Initialise default values (in case of no match).

```
    \def#2{}%
    \def\TrackedDialectClosestSubMatch{}%
    \@tracklang@parselangtag{#1}%
    \edef\@tracklang@dialect{%
     \@TrackLangEnvLang
     \@TrackLangEnvSubLang
     \@TrackLangEnvScript
     \@TrackLangEnvTerritory
     \@TrackLangEnvModifier
     \@TrackLangEnvVariant}%
```

Has this dialect label been tracked?

```
    \IfTrackedDialect{\@tracklang@dialect}%
    {%
```

Found it. All done.

```
      \let#2\@tracklang@dialect
    }%
    {%
```

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
\fi
```

Does the sub-language match?

```
\edef\@tracklang@tmp{%
  \GetTrackedDialectSubLang{\@tracklang@dialect}}%
\ifx\@tracklang@tmp\@TrackLangEnvSubLang
```

Sub-language matches.

```
  \edef\@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
```

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{}%
```

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

\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@track@locale
}
```

\@tracklang@track@locale

```
\def\@tracklang@track@locale{%
```

Is the language code known?

```
  \TrackLangIfKnownLangFromIso{\@TrackLangEnvLang}
  {%
    \edef\@tracklang@lang{\TrackLangGetKnownLangFromIso\@TrackLangEnvLang}%
    \let\@tracklang@dialect\@TrackLangEnvLang
    \ifx\@TrackLangEnvSubLang\empty
    \else
      \edef\@tracklang@dialect{\@tracklang@dialect-\@TrackLangEnvSubLang}%
    \fi
    \ifx\@TrackLangEnvScript\empty
    \else
      \edef\@tracklang@dialect{\@tracklang@dialect-\@TrackLangEnvScript}%
    \fi
    \ifx\@TrackLangEnvTerritory\empty
    \else
      \edef\@tracklang@dialect{\@tracklang@dialect-\@TrackLangEnvTerritory}%
    \fi
    \ifx\@TrackLangEnvModifier\empty
    \else
      \edef\@tracklang@dialect{\@tracklang@dialect-\@TrackLangEnvModifier}%
    \fi
    \ifx\@TrackLangEnvVariant\empty
    \else
      \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
      \expandafter\AddTrackedIsoLanguage
        \expandafter\ThreeLetterExtIsoLanguageCode
        \expandafter{\@TrackLangEnvFirstSubLang}%
        {\@tracklang@dialect}%
    \fi
  }%
  {%
```

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
}
```

6.2.5 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 "francais" or "frenchb". These are defined in Section 6.2.6. 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 already defined.

```
\ifx\@tracklang@declareoption\undefined
  \def\@tracklang@declareoption#1{}
\fi
```

`\TrackLangDeclareLanguageOption`

`\TrackLangDeclareLanguageOption{⟨language name⟩}{⟨639-1 code⟩}{⟨639-2 (T)⟩}{⟨639-2 (B)⟩}{⟨639-3⟩}{⟨3166-1⟩}{⟨default script⟩}`

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}%
    \@tracklang@namedef{@tracklang@add@#1}{%
      \AddTrackedLanguage{#1}%
      \AddTrackedLanguageIsoCodes{#1}%
      \AddTrackedCountryIsoCode{#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}

\@tracklang@add@anglosaxon
\TrackLangDeclareLanguageOption{anglosaxon}{}{ang}{}{}{}{Runr}

\@tracklang@add@apache
\TrackLangDeclareLanguageOption{apache}{}{apa}{}{}{}{Latn}

\@tracklang@add@arabic
\TrackLangDeclareLanguageOption{arabic}{ar}{ara}{}{}{}{Arab}

\@tracklang@add@aragonese
\TrackLangDeclareLanguageOption{aragonese}{an}{arg}{}{}{ES}{Latn}

\@tracklang@add@armenian
\TrackLangDeclareLanguageOption{armenian}{hy}{hye}{arm}{}{}{Armn}
```

```
\@tracklang@add@assamese
                        \TrackLangDeclareLanguageOption{assamese}{as}{asm}{}{}{}{Beng}
\@tracklang@add@asturian
                        \TrackLangDeclareLanguageOption{asturian}{}{ast}{}{}{}{Latn}
 \@tracklang@add@avaric
                        \TrackLangDeclareLanguageOption{avaric}{av}{ava}{}{}{}{Cyrl}
 \@tracklang@add@avestan
                        \TrackLangDeclareLanguageOption{avestan}{ae}{ave}{}{}{}{Avst}
 \@tracklang@add@aymara
                        \TrackLangDeclareLanguageOption{aymara}{ay}{aym}{}{}{}{Latn}
```

`\@tracklang@add@azerbaijani` 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.

```
                        \TrackLangDeclareLanguageOption{azerbaijani}{az}{aze}{}{}{}{Latn}
 \@tracklang@add@bahasai
                        \TrackLangDeclareLanguageOption{bahasai}{id}{ind}{}{}{IN}{Latn}
 \@tracklang@add@bahasam
                        \TrackLangDeclareLanguageOption{bahasam}{ms}{msa}{may}{}{MY}{Latn}
 \@tracklang@add@bambara
                        \TrackLangDeclareLanguageOption{bambara}{bm}{bam}{}{}{ML}{Latn}
 \@tracklang@add@bashkir
                        \TrackLangDeclareLanguageOption{bashkir}{ba}{bak}{}{}{}{Cyrl}
  \@tracklang@add@basque
                        \TrackLangDeclareLanguageOption{basque}{eu}{eus}{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@churchslavonic
                        \TrackLangDeclareLanguageOption{churchslavonic}{cu}{chu}{}{}{}{Glag}
 \@tracklang@add@chuvash
                        \TrackLangDeclareLanguageOption{chuvash}{cv}{chv}{}{}{RU}{Cyrl}
  \@tracklang@add@coptic
                        \TrackLangDeclareLanguageOption{coptic}{}{cop}{}{}{}{Copt}
 \@tracklang@add@cornish
                        \TrackLangDeclareLanguageOption{cornish}{kw}{cor}{}{}{GB}{Latn}
\@tracklang@add@corsican
                        \TrackLangDeclareLanguageOption{corsican}{co}{cos}{}{}{}{Latn}
```

```
\@tracklang@add@cree
                        \TrackLangDeclareLanguageOption{cree}{cr}{cre}{}{}{}{Cans}
\@tracklang@add@croatian
                        \TrackLangDeclareLanguageOption{croatian}{hr}{hrv}{}{}{}{Latn}
\@tracklang@add@czech
                        \TrackLangDeclareLanguageOption{czech}{cs}{ces}{cze}{}{}{Latn}
\@tracklang@add@danish
                        \TrackLangDeclareLanguageOption{danish}{da}{dan}{}{}{}{Latn}
\@tracklang@add@divehi
                        \TrackLangDeclareLanguageOption{divehi}{dv}{div}{}{}{MV}{Thaa}
\@tracklang@add@dutch
                        \TrackLangDeclareLanguageOption{dutch}{nl}{nld}{dut}{}{}{Latn}
\@tracklang@add@dzongkha
                        \TrackLangDeclareLanguageOption{dzongkha}{dz}{dzo}{}{}{BT}{Tibt}
\@tracklang@add@easternpunjabi
                        \TrackLangDeclareLanguageOption{easternpunjabi}{pa}{pan}{}{}{IN}{Guru}
\@tracklang@add@english
                        \TrackLangDeclareLanguageOption{english}{en}{eng}{}{}{}{Latn}
\@tracklang@add@esperanto
                        \TrackLangDeclareLanguageOption{esperanto}{eo}{epo}{}{}{}{Latn}
\@tracklang@add@estonian
                        \TrackLangDeclareLanguageOption{estonian}{et}{est}{}{}{}{Latn}
\@tracklang@add@ewe
                        \TrackLangDeclareLanguageOption{ewe}{ee}{ewe}{}{}{}{Latn}
\@tracklang@add@faroese
                        \TrackLangDeclareLanguageOption{faroese}{fo}{fao}{}{}{}{Latn}
\@tracklang@add@farsi
                        \TrackLangDeclareLanguageOption{farsi}{fa}{fas}{per}{}{}{Arab}
\@tracklang@add@fijian
                        \TrackLangDeclareLanguageOption{fijian}{fj}{fij}{}{}{FJ}{Latn}
\@tracklang@add@finnish
                        \TrackLangDeclareLanguageOption{finnish}{fi}{fin}{}{}{}{Latn}
```

```
\@tracklang@add@french
                        \TrackLangDeclareLanguageOption{french}{fr}{fra}{fre}{}{}{Latn}
\@tracklang@add@friulan
                        \TrackLangDeclareLanguageOption{friulan}{}{fur}{}{}{IT}{Latn}
```

`\@tracklang@add@fula` No default. Could be Latin or Arabic.

```
                        \TrackLangDeclareLanguageOption{fula}{ff}{ful}{}{}{}{}
\@tracklang@add@galician
                        \TrackLangDeclareLanguageOption{galician}{gl}{glg}{}{}{}{Latn}
\@tracklang@add@ganda
                        \TrackLangDeclareLanguageOption{ganda}{lg}{lug}{}{}{UG}{Latn}
\@tracklang@add@georgian
                        \TrackLangDeclareLanguageOption{georgian}{ka}{kat}{geo}{}{}{Geor}
\@tracklang@add@german
                        \TrackLangDeclareLanguageOption{german}{de}{deu}{ger}{}{}{Latn}
\@tracklang@add@greek
                        \TrackLangDeclareLanguageOption{greek}{el}{ell}{gre}{}{}{Grek}
\@tracklang@add@guarani
                        \TrackLangDeclareLanguageOption{guarani}{gn}{grn}{}{}{}{Latn}
\@tracklang@add@guiarati
                        \TrackLangDeclareLanguageOption{gujarati}{gu}{guj}{}{}{}{Gujr}
\@tracklang@add@haitian
                        \TrackLangDeclareLanguageOption{haitian}{ht}{hat}{}{}{HT}{Latn}
\@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}{}{}{}{}
```

```
\@tracklang@add@khmer
                \TrackLangDeclareLanguageOption{khmer}{km}{khm}{}{}{}{Khmr}
\@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}{Cyrl}
\@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}{}{}{}{Cyrl}
\@tracklang@add@lao
                \TrackLangDeclareLanguageOption{lao}{lo}{lao}{}{}{}{Laoo}
\@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}{}{}{Cyrl}
\@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}
\@tracklang@add@marshallese
                \TrackLangDeclareLanguageOption{marshallese}{mh}{mah}{}{}{MH}{Latn}
\@tracklang@add@mongolian
                \TrackLangDeclareLanguageOption{mongolian}{mn}{mon}{}{}{}{Mong}
\@tracklang@add@nauruan
                \TrackLangDeclareLanguageOption{nauruan}{na}{nau}{}{}{NR}{Latn}
\@tracklang@add@navajo
                \TrackLangDeclareLanguageOption{navajo}{nv}{nav}{}{}{US}{Latn}
\@tracklang@add@ndonga
                \TrackLangDeclareLanguageOption{ndonga}{ng}{ndo}{}{}{}{Latn}
```

```
\@tracklang@add@nepali
                        \TrackLangDeclareLanguageOption{nepali}{ne}{nep}{}{}{}{Deva}
\@tracklang@add@nko
                        \TrackLangDeclareLanguageOption{nko}{}{nqo}{}{}{}{Nkoo}
\@tracklang@add@northernndebele
                        \TrackLangDeclareLanguageOption{northernndebele}{nd}{nde}{}{}{}{Latn}
\@tracklang@add@nynorsk
                        \TrackLangDeclareLanguageOption{nynorsk}{nn}{nno}{}{}{NO}{Latn}
\@tracklang@add@norsk
                        \TrackLangDeclareLanguageOption{norsk}{no}{nor}{}{}{}{Latn}
\@tracklang@add@northernsotho
                        \TrackLangDeclareLanguageOption{northernsotho}{}{nso}{}{}{}{Latn}
\@tracklang@add@nuosu
                        \TrackLangDeclareLanguageOption{nuosu}{ii}{iii}{}{}{CN}{Yiii}
\@tracklang@add@occitan
                        \TrackLangDeclareLanguageOption{occitan}{oc}{oci}{}{}{}{Latn}
\@tracklang@add@ojibwe
                        \TrackLangDeclareLanguageOption{ojibwe}{oj}{oji}{}{}{}{Latn}
\@tracklang@add@oromo
                        \TrackLangDeclareLanguageOption{oromo}{om}{orm}{}{}{}{Latn}
\@tracklang@add@oriya
                        \TrackLangDeclareLanguageOption{oriya}{or}{ori}{}{}{}{Orya}
\@tracklang@add@ossetian
                        \TrackLangDeclareLanguageOption{ossetian}{os}{oss}{}{}{}{Cyrl}
\@tracklang@add@pali
                        \TrackLangDeclareLanguageOption{pali}{pi}{pli}{}{}{}{Brah}
\@tracklang@add@pashto
                        \TrackLangDeclareLanguageOption{pashto}{ps}{pus}{}{}{}{Arab}
\@tracklang@add@piedmontese
                        \TrackLangDeclareLanguageOption{piedmontese}{}{}{}{pms}{IT}{Latn}
\@tracklang@add@polish
                        \TrackLangDeclareLanguageOption{polish}{pl}{pol}{}{}{}{Latn}
```

```
\@tracklang@add@portuges
                        \TrackLangDeclareLanguageOption{portuges}{pt}{por}{}{}{}{Latn}
 \@tracklang@add@quechua
                        \TrackLangDeclareLanguageOption{quechua}{qu}{que}{}{}{}{Latn}
\@tracklang@add@romanian
                        \TrackLangDeclareLanguageOption{romanian}{ro}{ron}{rum}{}{}{Latn}
 \@tracklang@add@romansh
                        \TrackLangDeclareLanguageOption{romansh}{rm}{roh}{}{}{CH}{Latn}
 \@tracklang@add@russian
                        \TrackLangDeclareLanguageOption{russian}{ru}{rus}{}{}{}{Cyrl}
   \@tracklang@add@samin
                        \TrackLangDeclareLanguageOption{samin}{se}{sme}{}{}{}{Latn}
\@tracklang@add@sanskrit
                        \TrackLangDeclareLanguageOption{sanskrit}{sa}{san}{}{}{}{}
  \@tracklang@add@samoan
                        \TrackLangDeclareLanguageOption{samoan}{sm}{smo}{}{}{}{Latn}
   \@tracklang@add@sango
                        \TrackLangDeclareLanguageOption{sango}{sg}{sag}{}{}{}{Latn}
\@tracklang@add@sardinian
                        \TrackLangDeclareLanguageOption{sardinian}{sc}{srd}{}{}{IT}{Latn}
 \@tracklang@add@scottish
```

Also spoken in Canada, so no region.

```
                        \TrackLangDeclareLanguageOption{scottish}{gd}{gla}{}{}{}{Latn}
 \@tracklang@add@serbian
                        \TrackLangDeclareLanguageOption{serbian}{sr}{srp}{}{}{}{Cyrl}
   \@tracklang@add@shona
                        \TrackLangDeclareLanguageOption{shona}{sn}{sna}{}{}{}{Latn}
  \@tracklang@add@sindhi
                        \TrackLangDeclareLanguageOption{sindhi}{sd}{snd}{}{}{}{Sind}
\@tracklang@add@sinhalese
                        \TrackLangDeclareLanguageOption{sinhalese}{si}{sin}{}{}{LK}{Sinh}
  \@tracklang@add@slovak
                        \TrackLangDeclareLanguageOption{slovak}{sk}{slk}{slo}{}{}{Latn}
```

```
        \@tracklang@add@slovene
                                \TrackLangDeclareLanguageOption{slovene}{sl}{slv}{}{}{}{Latn}
         \@tracklang@add@somali
                                \TrackLangDeclareLanguageOption{somali}{so}{som}{}{}{}{Latn}
\@tracklang@add@southernndebele
                                \TrackLangDeclareLanguageOption{southernndebele}{nr}{nbl}{}{}{ZA}{Latn}
  \@tracklang@add@southernsotho
                                \TrackLangDeclareLanguageOption{southernsotho}{st}{sot}{}{}{}{Latn}
        \@tracklang@add@spanish
                                \TrackLangDeclareLanguageOption{spanish}{es}{spa}{}{}{}{Latn}
       \@tracklang@add@sudanese
                                \TrackLangDeclareLanguageOption{sudanese}{su}{sun}{}{}{}{Sund}
        \@tracklang@add@swahili
                                \TrackLangDeclareLanguageOption{swahili}{sw}{swa}{}{}{}{}
          \@tracklang@add@swati
                                \TrackLangDeclareLanguageOption{swati}{ss}{ssw}{}{}{}{Latn}
        \@tracklang@add@swedish
                                \TrackLangDeclareLanguageOption{swedish}{sv}{swe}{}{}{}{Latn}
         \@tracklang@add@syriac
                                \TrackLangDeclareLanguageOption{syriac}{}{syr}{}{}{}{Syrc}
        \@tracklang@add@tagalog
                                \TrackLangDeclareLanguageOption{tagalog}{tl}{tgl}{}{}{PH}{Latn}
       \@tracklang@add@tahitian
                                \TrackLangDeclareLanguageOption{tahitian}{ty}{tah}{}{}{PF}{Latn}
            \@tracklang@add@tai
                                \TrackLangDeclareLanguageOption{tai}{}{tai}{}{}{}{}
          \@tracklang@add@tajik
                                \TrackLangDeclareLanguageOption{tajik}{tg}{tgk}{}{}{}{Cyrl}
          \@tracklang@add@tamil
                                \TrackLangDeclareLanguageOption{tamil}{ta}{tam}{}{}{}{Taml}
          \@tracklang@add@tatar
                                \TrackLangDeclareLanguageOption{tatar}{tt}{tat}{}{}{}{Cyrl}
```

```
\@tracklang@add@telugu
                        \TrackLangDeclareLanguageOption{telugu}{te}{tel}{}{}{IN}{Telu}
  \@tracklang@add@thai
                        \TrackLangDeclareLanguageOption{thai}{th}{tha}{}{}{TH}{Thai}
\@tracklang@add@tibetan
                        \TrackLangDeclareLanguageOption{tibetan}{bo}{bod}{tib}{}{}{Tibt}
\@tracklang@add@tigrinya
                        \TrackLangDeclareLanguageOption{tigrinya}{ti}{tir}{}{}{}{Ethi}
  \@tracklang@add@tonga
                        \TrackLangDeclareLanguageOption{tonga}{to}{ton}{}{}{TO}{Latn}
  \@tracklang@add@tsonga
                        \TrackLangDeclareLanguageOption{tsonga}{ts}{tso}{}{}{}{Latn}
  \@tracklang@add@tswana
                        \TrackLangDeclareLanguageOption{tswana}{tn}{tsn}{}{}{}{Latn}
 \@tracklang@add@turkish
                        \TrackLangDeclareLanguageOption{turkish}{tr}{tur}{}{}{}{Latn}
 \@tracklang@add@turkmen
                        \TrackLangDeclareLanguageOption{turkmen}{tk}{tuk}{}{}{}{Latn}
    \@tracklang@add@twi
                        \TrackLangDeclareLanguageOption{twi}{tw}{twi}{}{}{GH}{Latn}
\@tracklang@add@ukrainian
                        \TrackLangDeclareLanguageOption{ukrainian}{uk}{ukr}{}{}{UA}{Cyrl}
    \@tracklang@add@urdu
                        \TrackLangDeclareLanguageOption{urdu}{ur}{urd}{}{}{}{Arab}
\@tracklang@add@usorbian
                        \TrackLangDeclareLanguageOption{usorbian}{}{hsb}{}{}{DE}{Latn}
  \@tracklang@add@uyghur
                        \TrackLangDeclareLanguageOption{uyghur}{ug}{uig}{}{}{CN}{Arab}
   \@tracklang@add@uzbek
                        \TrackLangDeclareLanguageOption{uzbek}{uz}{uzb}{}{}{}{Latn}
   \@tracklang@add@venda
                        \TrackLangDeclareLanguageOption{venda}{ve}{ven}{}{}{ZA}{Latn}
```

\@tracklang@add@vietnamese
```
\TrackLangDeclareLanguageOption{vietnamese}{vi}{vie}{}{}{}{Latn}
```

\@tracklang@add@volapuk
```
\TrackLangDeclareLanguageOption{volapuk}{vo}{vol}{}{}{}{Latn}
```

\@tracklang@add@walloon
No country code as Walloon is spoken in multiple regions (Wallonia in Belgium, some villages in Northern France and north-east of Wisconsin.) Not the same as Belgian French.
```
\TrackLangDeclareLanguageOption{walloon}{wa}{wln}{}{}{}{Latn}
```

\@tracklang@add@welsh
Also spoken in Argentina, so no region.
```
\TrackLangDeclareLanguageOption{welsh}{cy}{cym}{wel}{}{}{Latn}
```

\@tracklang@add@westernfrisian
```
\TrackLangDeclareLanguageOption{westernfrisian}{fy}{fry}{}{}{NL}{Latn}
```

\@tracklang@add@wolof
```
\TrackLangDeclareLanguageOption{wolof}{wo}{wol}{}{}{}{Latn}
```

\@tracklang@add@xhosa
```
\TrackLangDeclareLanguageOption{xhosa}{xh}{xho}{}{}{}{Latn}
```

\@tracklang@add@yiddish
```
\TrackLangDeclareLanguageOption{yiddish}{yi}{yid}{}{}{}{Hebr}
```

\@tracklang@add@yoruba
```
\TrackLangDeclareLanguageOption{yoruba}{yo}{yor}{}{}{}{Latn}
```

\@tracklang@add@zhuang
```
\TrackLangDeclareLanguageOption{zhuang}{za}{zha}{}{}{CN}{Hani}
```

\@tracklang@add@zulu
```
\TrackLangDeclareLanguageOption{zulu}{zu}{zul}{}{}{}{Latn}
```

6.2.6 Predefined Dialects

Provide some predefined dialects.

\TrackLangDeclareDialectOption
\TrackLangDeclareDialectOption{⟨*dialect*⟩}{⟨*root language*⟩}{⟨*3166-1 code*⟩}{⟨*modifier*⟩}{⟨*variant*⟩}{⟨*map*⟩}{⟨*script*⟩}

The option name is the same as the dialect name. The arguments must be expanded before use. The final argument ⟨*map*⟩ is the mapping from ⟨*dialect*⟩ to the closest babel dialect label. May be empty if no relevant mapping.
```
\def\TrackLangDeclareDialectOption#1#2#3#4#5#6#7{%
  \@tracklang@ifundef{@tracklang@add@#1}%
  {%
    \ifx\relax#3\relax
```

No region.

```
\ifx\relax#4\relax
```

No modifier.

```
\ifx\relax#5\relax
```

No variant.

```
\@tracklang@namedef{@tracklang@add@#1}{%
  \AddTrackedDialect{#1}{#2}%
  \AddTrackedLanguageIsoCodes{#2}%
```

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}%
   \SetTrackedDialectVariant{#1}{#5}%
   \def\@tracklang@dialect{#1}%
  }%
 \fi
\else
```

Has modifier.

```
 \ifx\relax#5\relax
```

No variant.

```
  \@tracklang@namedef{@tracklang@add@#1}{%
    \AddTrackedDialect{#1}{#2}%
    \AddTrackedLanguageIsoCodes{#2}%
    \SetTrackedDialectModifier{#1}{#4}%
    \def\@tracklang@dialect{#1}%
   }%
 \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.

```
\ifx\relax#4\relax
```

No modifier.

```
 \ifx\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
\else
```

Has modifier.

```
  \ifx\relax#5\relax
```

No variant.

```
    \@tracklang@namedef{@tracklang@add@#1}{%
      \AddTrackedDialect{#1}{#2}%
      \AddTrackedLanguageIsoCodes{#2}%
      \AddTrackedIsoLanguage{3166-1}{#3}{#1}%
      \SetTrackedDialectModifier{#1}{#4}%
      \def\@tracklang@dialect{#1}%
    }%
  \else
```

Variant and modifier.

```
    \@tracklang@namedef{@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.

```
\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.

```
\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
\@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}{}{}{}{}

acklang@add@istriacountycroatian
                    \TrackLangDeclareDialectOption{istriacountycroatian}{croatian}{HR}{}{}{}{}

racklang@add@istriacountyitalian
                    \TrackLangDeclareDialectOption{istriacountyitalian}{italian}{HR}{}{}{}{}
```

```
\@tracklang@add@netherlands
                        \TrackLangDeclareDialectOption{netherlands}{dutch}{NL}{}{}{}{}
    \@tracklang@add@persian
                        \TrackLangDeclareDialectOption{persian}{farsi}{}{}{}{}{}
    \@tracklang@add@flemish
                        \TrackLangDeclareDialectOption{flemish}{dutch}{BE}{}{}{}{}
   \@tracklang@add@francais
                        \TrackLangDeclareDialectOption{francais}{french}{}{}{}{}{}
    \@tracklang@add@frenchb
                        \TrackLangDeclareDialectOption{frenchb}{french}{}{}{}{}{}
     \@tracklang@add@france
                        \TrackLangDeclareDialectOption{france}{french}{FR}{}{}{}{}
   \@tracklang@add@belgique
                        \TrackLangDeclareDialectOption{belgique}{french}{BE}{}{}{}{}
\@tracklang@add@belgiangerman
                        \TrackLangDeclareDialectOption{belgiangerman}{german}{BE}{}{}{}{}
\@tracklang@add@nbelgiangerman
                        \TrackLangDeclareDialectOption{nbelgiangerman}{german}{BE}{new}{1996}{ngerman}{}
   \@tracklang@add@friulian
                        \TrackLangDeclareDialectOption{friulian}{friulan}{IT}{}{}{}{}
   \@tracklang@add@friulano
                        \TrackLangDeclareDialectOption{friulano}{friulan}{IT}{}{}{}{}
     \@tracklang@add@furlan  Added since it's a babel alias for friulan.
                        \TrackLangDeclareDialectOption{furlan}{friulan}{IT}{}{}{}{}
   \@tracklang@add@kurmanji  Added since it's a babel label.
                        \TrackLangDeclareDialectOption{kurmanji}{kurdish}{}{}{}{}{}
   \@tracklang@add@galicien
                        \TrackLangDeclareDialectOption{galicien}{galician}{}{}{}{}{}
    \@tracklang@add@deutsch
                        \TrackLangDeclareDialectOption{deutsch}{german}{}{}{}{}{}
    \@tracklang@add@ngerman
                        \TrackLangDeclareDialectOption{ngerman}{german}{}{new}{1996}{}{}
```

\@tracklang@add@ngermanb
```
\TrackLangDeclareDialectOption{ngermanb}{german}{}{new}{1996}{ngerman}{}
```

\@tracklang@add@germanb
```
\TrackLangDeclareDialectOption{germanb}{german}{}{}{}{}{}
```

\@tracklang@add@ngermanDE
```
\TrackLangDeclareDialectOption{ngermanDE}{german}{DE}{new}{1996}{ngerman}{}
```

\@tracklang@add@germanDE
```
\TrackLangDeclareDialectOption{germanDE}{german}{DE}{}{}{}{}
```

\@tracklang@add@hungarian
```
\TrackLangDeclareDialectOption{hungarian}{magyar}{HU}{}{}{}{}
```

\@tracklang@add@indon
```
\TrackLangDeclareDialectOption{indon}{bahasai}{IN}{}{}{}{}
```

\@tracklang@add@indonesian
```
\TrackLangDeclareDialectOption{indonesian}{bahasai}{IN}{}{}{}{}
```

\@tracklang@add@gaeilge
```
\TrackLangDeclareDialectOption{gaeilge}{irish}{}{}{}{}{}
```

\@tracklang@add@IEirish Irish spoken in Republic of Ireland
```
\TrackLangDeclareDialectOption{IEirish}{irish}{IE}{}{}{}{}
```

\@tracklang@add@GBirish Irish spoken in the United Kingdom of Great Britain and Northern Ireland
```
\TrackLangDeclareDialectOption{GBirish}{irish}{GB}{}{}{}{}
```

\@tracklang@add@IEenglish English spoken in the Republic of Ireland.
```
\TrackLangDeclareDialectOption{IEenglish}{english}{IE}{}{}{british}{}
```

\@tracklang@add@italy
```
\TrackLangDeclareDialectOption{italy}{italian}{IT}{}{}{}{}
```

\@tracklang@add@vatican
```
\TrackLangDeclareDialectOption{vatican}{italian}{VA}{}{}{}{}
```

\@tracklang@add@sanmarino
```
\TrackLangDeclareDialectOption{sanmarino}{italian}{SM}{}{}{}{}
```

acklang@add@sloveneistriaitalian
```
\TrackLangDeclareDialectOption{sloveneistriaitalian}{italian}{SI}{}{}{}{}
```

\@tracklang@add@jerseyenglish Allow it to hook to pick up `\captionsbritish` as well as `\captionsenglish` since the date format closely matches `british`.
```
\TrackLangDeclareDialectOption{jerseyenglish}{english}{JE}{}{}{british}{}
```

\@tracklang@add@jerseyfrench

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

\@tracklang@add@guernseyenglish Allow it to hook to pick up \captionsbritish as well as \captionsenglish since the date format closely matches british.

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

\@tracklang@add@guernseyfrench

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

\@tracklang@add@latein

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

\@tracklang@add@lowersorbian

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

\@tracklang@add@malay

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

\@tracklang@add@meyalu

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

\@tracklang@add@maltamaltese

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

\@tracklang@add@maltaenglish Allow it to hook to pick up \captionsbritish as well as \captionsenglish since the date format closely matches british.

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

\@tracklang@add@newzealand

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

\@tracklang@add@isleofmanenglish Allow it to hook to pick up \captionsbritish as well as \captionsenglish since the date format closely matches british.

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

\@tracklang@add@norwegian

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

\@tracklang@add@piemonteis

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

\@tracklang@add@polutonikogreek

```
\TrackLangDeclareDialectOption{polutonikogreek}{greek}{}{polyton}{}{}{}
```

\@tracklang@add@polutoniko

```
\TrackLangDeclareDialectOption{polutoniko}{greek}{}{polyton}{}{}{}
```

```
\@tracklang@add@portuguese
                        \TrackLangDeclareDialectOption{portuguese}{portuges}{}{}{}{}{}
 \@tracklang@add@portugal
                        \TrackLangDeclareDialectOption{portugal}{portuges}{PT}{}{}{}{}
 \@tracklang@add@romansch
                        \TrackLangDeclareDialectOption{romansch}{romansh}{}{}{}{}{}
\@tracklang@add@rumantsch
                        \TrackLangDeclareDialectOption{rumantsch}{romansh}{}{}{}{}{}
 \@tracklang@add@romanche
                        \TrackLangDeclareDialectOption{romanche}{romansh}{}{}{}{}{}
 \@tracklang@add@russianb
                        \TrackLangDeclareDialectOption{russianb}{russian}{}{}{}{}{}
   \@tracklang@add@gaelic
                        \TrackLangDeclareDialectOption{gaelic}{scottish}{}{}{}{}{}
\@tracklang@add@GBscottish
                        \TrackLangDeclareDialectOption{GBscottish}{scottish}{GB}{}{}{}{}
 \@tracklang@add@serbianc
                        \TrackLangDeclareDialectOption{serbianc}{serbian}{}{}{}{}{Cyrl}
 \@tracklang@add@serbianl
                        \TrackLangDeclareDialectOption{serbianl}{serbian}{}{}{}{}{Latn}
\@tracklang@add@slovenian
                        \TrackLangDeclareDialectOption{slovenian}{slovene}{}{}{}{}{}
 \@tracklang@add@slovenia
                        \TrackLangDeclareDialectOption{slovenia}{slovene}{SI}{}{}{slovenian}{}
klang@add@sloveneistriaslovenian
                        \TrackLangDeclareDialectOption{sloveneistriaslovenian}{slovene}{SI}{}{}{slovenian}{}
   \@tracklang@add@spainspanish
                        \TrackLangDeclareDialectOption{spainspanish}{spanish}{ES}{}{}{}{}
\@tracklang@add@argentinespanish
                        \TrackLangDeclareDialectOption{argentinespanish}{spanish}{AR}{}{}{}{}
 \@tracklang@add@bolivianspanish
                        \TrackLangDeclareDialectOption{bolivianspanish}{spanish}{BO}{}{}{}{}
```

```
  \@tracklang@add@chilianspanish
                                \TrackLangDeclareDialectOption{chilianspanish}{spanish}{CL}{}{}{}{}
\@tracklang@add@columbianspanish
                                \TrackLangDeclareDialectOption{columbianspanish}{spanish}{CO}{}{}{}{}
@tracklang@add@costaricanspanish
                                \TrackLangDeclareDialectOption{costaricanspanish}{spanish}{CR}{}{}{}{}
    \@tracklang@add@cubanspanish
                                \TrackLangDeclareDialectOption{cubanspanish}{spanish}{CU}{}{}{}{}
\@tracklang@add@dominicanspanish
                                \TrackLangDeclareDialectOption{dominicanspanish}{spanish}{DO}{}{}{}{}
\@tracklang@add@ecudorianspanish
                                \TrackLangDeclareDialectOption{ecudorianspanish}{spanish}{EC}{}{}{}{}
@tracklang@add@elsalvadorspanish
                                \TrackLangDeclareDialectOption{elsalvadorspanish}{spanish}{SV}{}{}{}{}
@tracklang@add@guatemalanspanish
                                \TrackLangDeclareDialectOption{guatemalanspanish}{spanish}{GT}{}{}{}{}
 \@tracklang@add@honduranspanish
                                \TrackLangDeclareDialectOption{honduranspanish}{spanish}{HN}{}{}{}{}
  \@tracklang@add@mexicanspanish
                                \TrackLangDeclareDialectOption{mexicanspanish}{spanish}{MX}{}{}{}{}
@tracklang@add@nicaraguanspanish
                                \TrackLangDeclareDialectOption{nicaraguanspanish}{spanish}{NI}{}{}{}{}
   \@tracklang@add@panamaspanish
                                \TrackLangDeclareDialectOption{panamaspanish}{spanish}{PA}{}{}{}{}
 \@tracklang@add@paraguayspanish
                                \TrackLangDeclareDialectOption{paraguayspanish}{spanish}{PY}{}{}{}{}
 \@tracklang@add@peruvianspanish
                                \TrackLangDeclareDialectOption{peruvianspanish}{spanish}{PE}{}{}{}{}
@tracklang@add@puertoricospanish
                                \TrackLangDeclareDialectOption{puertoricospanish}{spanish}{PR}{}{}{}{}
  \@tracklang@add@uruguayspanish
                                \TrackLangDeclareDialectOption{uruguayspanish}{spanish}{UY}{}{}{}{}
```

```
@tracklang@add@venezuelanspanish
                                \TrackLangDeclareDialectOption{venezuelanspanish}{spanish}{VE}{}{}{}{}

     \@tracklang@add@swissgerman
                                \TrackLangDeclareDialectOption{swissgerman}{german}{CH}{}{}{}{}

   \@tracklang@add@nswissgerman
                                \TrackLangDeclareDialectOption{nswissgerman}{german}{CH}{new}{1996}{ngerman}{}

     \@tracklang@add@swissfrench
                                \TrackLangDeclareDialectOption{swissfrench}{french}{CH}{}{}{}{}

   \@tracklang@add@swissitalian
                                \TrackLangDeclareDialectOption{swissitalian}{italian}{CH}{}{}{}{}

   \@tracklang@add@swissromansh
                                \TrackLangDeclareDialectOption{swissromansh}{romansh}{CH}{}{}{}{}

       \@tracklang@add@UKenglish
                                \TrackLangDeclareDialectOption{UKenglish}{english}{GB}{}{}{}{}

      \@tracklang@add@ukraineb
                                \TrackLangDeclareDialectOption{ukraineb}{ukrainian}{UA}{}{}{}{}

       \@tracklang@add@ukraine
                                \TrackLangDeclareDialectOption{ukraine}{ukrainian}{UA}{}{}{}{}

   \@tracklang@add@uppersorbian
                                \TrackLangDeclareDialectOption{uppersorbian}{usorbian}{DE}{}{}{}{}

       \@tracklang@add@USenglish
                                \TrackLangDeclareDialectOption{USenglish}{english}{US}{}{}{}{}

     \@tracklang@add@valencian
                                \TrackLangDeclareDialectOption{valencian}{catalan}{}{}{}{}{}

     \@tracklang@add@valencien
                                \TrackLangDeclareDialectOption{valencien}{catalan}{}{}{}{}{}

       \@tracklang@add@cymraeg
                                \TrackLangDeclareDialectOption{cymraeg}{welsh}{}{}{}{}{}

       \@tracklang@add@GBwelsh
                                \TrackLangDeclareDialectOption{GBwelsh}{welsh}{GB}{}{}{}{}
```

6.2.7 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}

\@tracklang@add@es-AR
                        \LetTrackLangOption{es-AR}{argentinespanish}
```

```
\@tracklang@add@es-BO
                        \LetTrackLangOption{es-BO}{bolivianspanish}

\@tracklang@add@es-CL
                        \LetTrackLangOption{es-CL}{chilianspanish}

\@tracklang@add@es-CO
                        \LetTrackLangOption{es-CO}{columbianspanish}

\@tracklang@add@es-CR
                        \LetTrackLangOption{es-CR}{costaricanspanish}

\@tracklang@add@es-CU
                        \LetTrackLangOption{es-CU}{cubanspanish}

\@tracklang@add@es-DO
                        \LetTrackLangOption{es-DO}{dominicanspanish}

\@tracklang@add@es-EC
                        \LetTrackLangOption{es-EC}{ecudorianspanish}

\@tracklang@add@es-SV
                        \LetTrackLangOption{es-SV}{elsalvadorspanish}

\@tracklang@add@es-GT
                        \LetTrackLangOption{es-GT}{guatemalanspanish}

\@tracklang@add@es-HN
                        \LetTrackLangOption{es-HN}{honduranspanish}

\@tracklang@add@es-MX
                        \LetTrackLangOption{es-MX}{mexicanspanish}

\@tracklang@add@es-NI
                        \LetTrackLangOption{es-NI}{nicaraguanspanish}

\@tracklang@add@es-PA
                        \LetTrackLangOption{es-PA}{panamaspanish}

\@tracklang@add@es-PY
                        \LetTrackLangOption{es-PY}{paraguayspanish}

\@tracklang@add@es-PE
                        \LetTrackLangOption{es-PE}{peruvianspanish}

\@tracklang@add@es-PR
                        \LetTrackLangOption{es-PR}{puertoricospanish}
```

```
\@tracklang@add@es-UY
                        \LetTrackLangOption{es-UY}{uruguayspanish}
\@tracklang@add@es-VE
                        \LetTrackLangOption{es-VE}{venezuelanspanish}
\@tracklang@add@de-CH
                        \LetTrackLangOption{de-CH}{swissgerman}
\@tracklang@add@fr-CH
                        \LetTrackLangOption{fr-CH}{swissfrench}
\@tracklang@add@it-CH
                        \LetTrackLangOption{it-CH}{swissitalian}
\@tracklang@add@rm-CH
                        \LetTrackLangOption{rm-CH}{swissromansh}
\@tracklang@add@it-VA
                        \LetTrackLangOption{it-VA}{vatican}
```

`\@tracklang@add@ga-IE` Irish Gaelic in Republic of Ireland.

```
                        \LetTrackLangOption{ga-IE}{IEirish}
```

`\@tracklang@add@ga-GB` Irish Gaelic in Northern Ireland.

```
                        \LetTrackLangOption{ga-GB}{GBirish}
```

`\@tracklang@add@en-IE` English spoken in Republic of Ireland.

```
                        \LetTrackLangOption{en-IE}{IEenglish}
\@tracklang@add@de-AT-1996
                        \LetTrackLangOption{de-AT-1996}{naustrian}
\@tracklang@add@de-AT
                        \LetTrackLangOption{de-AT}{austrian}
\@tracklang@add@id-IN
                        \LetTrackLangOption{id-IN}{bahasa}
\@tracklang@add@ms-MY
                        \LetTrackLangOption{ms-MY}{malay}
\@tracklang@add@hr-HR
                        \LetTrackLangOption{hr-HR}{croatia}
\@tracklang@add@de-DE-1996
                        \LetTrackLangOption{de-DE-1996}{ngermanDE}
```

```
\@tracklang@add@de-CH-1996
                        \LetTrackLangOption{de-CH-1996}{nswissgerman}
    \@tracklang@add@hu-HU
                        \LetTrackLangOption{hu-HU}{hungarian}
    \@tracklang@add@gd-GB
                        \LetTrackLangOption{gd-GB}{GBscottish}
    \@tracklang@add@cy-GB
                        \LetTrackLangOption{cy-GB}{GBwelsh}
```

6.2.8 Conditionals and Loops

\IfTrackedLanguage `\IfTrackedLanguage{⟨language⟩}{⟨true part⟩}{⟨false part⟩}`

```
  \long\def\IfTrackedLanguage#1#2#3{%
```

First find out if the language name is empty.

```
    \edef\@tracklang@element{#1}%
    \ifx\@tracklang@element\empty
```

Language is empty, so do false part.

```
      #3%
    \else
      \expandafter\@tracklang@ifinlist\expandafter{\@tracklang@element}%
        \@tracklang@languages
      {%
```

In list, so do true part.

```
        #2%
      }%
      {%
```

Not in list, so do false part.

```
        #3%
      }%
    \fi
  }
```

\IfTrackedDialect `\IfTrackedDialect{⟨dialect⟩}{⟨true part⟩}{⟨false part⟩}`

```
  \long\def\IfTrackedDialect#1#2#3{%
    \@tracklang@ifundef{@tracklang@fromdialect@#1}{#3}{#2}%
  }
```

\IfTrackedIsoCode `\IfTrackedIsoCode{⟨code type⟩}{⟨code⟩}{⟨true part⟩}{⟨false part⟩}`

```
  \long\def\IfTrackedIsoCode#1#2#3#4{%
    \@tracklang@ifundef{@tracklang@#1@isotolang@#2}{#4}{#3}%
  }
```

\IfTrackedLanguageHasIsoCode

\IfTrackedLanguageHasIsoCode{⟨*code type*⟩}{⟨*language*⟩}{⟨*true part*⟩}{⟨*false part*⟩}

```
\long\def\IfTrackedLanguageHasIsoCode#1#2#3#4{%
  \@tracklang@ifundef{@tracklang@#1@isofromlang@#2}{#4}{#3}%
}
```

\ForEachTrackedLanguage

\ForEachTrackedLanguage{⟨*cs*⟩}{⟨*body*⟩}

Iterates through the list of tracked languages. On each iteration ⟨*cs*⟩ is set to the language tag and ⟨*body*⟩ is performed.

```
\long\def\ForEachTrackedLanguage#1#2{%
  \@tracklang@for#1:=\@tracklang@languages\do{#2}%
}
```

\ForEachTrackedDialect

\ForEachTrackedDialect{⟨*cs*⟩}{⟨*body*⟩}

Iterates through the list of tracked dialects. On each iteration ⟨*cs*⟩ is set to the dialect tag and ⟨*body*⟩ is performed.

```
\long\def\ForEachTrackedDialect#1#2{%
  \@tracklang@for#1:=\@tracklang@dialects\do{#2}%
}
```

\AnyTrackedLanguages

```
\long\def\AnyTrackedLanguages#1#2{%
  \ifx\@tracklang@languages\empty
    #2%
  \else
    #1%
  \fi
}
```

\IfTrackedLanguageFileExists

\IfTrackedLanguageFileExists{⟨*dialect*⟩}{⟨*prefix*⟩}{⟨*suffix*⟩}{⟨*true part*⟩}{⟨*false part*⟩}

Determines if the file ⟨*prefix*⟩⟨*tag*⟩⟨*suffix*⟩ exists, where ⟨*tag*⟩ is an ISO code or ISO codes identifying the language. If ⟨*dialect*⟩ hasn't been identified as a tracked dialect, this just does ⟨*false part*⟩, otherwise this first tries with ⟨*tag*⟩ set to ⟨*dialect*⟩, then tries with ⟨*tag*⟩ set to the root language label for ⟨*dialect*⟩, then tries with ⟨*tag*⟩ set to ⟨*ISO 639-1 code*⟩-⟨*ISO 3166-1 code*⟩, then tries with ⟨*tag*⟩ set to ⟨*ISO 639-1 code*⟩, then tries with ⟨*tag*⟩ set to ⟨*ISO 639-2 code*⟩-⟨*ISO 3166-1 code*⟩, then tries with ⟨*tag*⟩ set to ⟨*ISO 639-2 code*⟩. If the file ⟨*prefix*⟩⟨*tag*⟩⟨*suffix*⟩ exists, \CurrentTrackedTag is set to ⟨*tag*⟩ and ⟨*true part*⟩ is performed, otherwise ⟨*false part*⟩ is performed.

```
\long\def\IfTrackedLanguageFileExists#1#2#3#4#5{%
```

Initialise.

```
  \def\CurrentTrackedTag{}%
```

Select this dialect.

```
\SetCurrentTrackedDialect{#1}%
\IfTrackedDialect{#1}%
{%
```

Try just the dialect label.

```
  \edef\CurrentTrackedTag{#1}%
  \@tracklang@IfFileExists{#2\CurrentTrackedTag#3}%
  {#4}%
  {%
```

No file found for dialect label, try ⟨*ISO 639-1*⟩-⟨*ISO 3166-1*⟩ next.

```
    \IfTrackedLanguageHasIsoCode
      {639-1}{\CurrentTrackedLanguage}
    {%
       \edef\CurrentTrackedIsoCode{%
          \TrackedIsoCodeFromLanguage
            {639-1}{\CurrentTrackedLanguage}}%
       \ifx\CurrentTrackedRegion\empty
```

No region, just try ISO 639-1 code

```
        \let\CurrentTrackedTag\CurrentTrackedIsoCode
        \@tracklang@IfFileExists{#2\CurrentTrackedTag#3}%
        {#4}
        {%
```

No region and file for just ⟨*ISO 639-1 code*⟩. Try ISO 639-2 code. (No check for 639-3 since it has 639-1 code.)

```
          \IfTrackedLanguageHasIsoCode
            {639-2}{\CurrentTrackedLanguage}
          {%
             \edef\CurrentTrackedIsoCode{%
                \TrackedIsoCodeFromLanguage
                  {639-2}{\CurrentTrackedLanguage}}%
             \let\CurrentTrackedTag\CurrentTrackedIsoCode
             \@tracklang@IfFileExists{#2\CurrentTrackedTag#3}%
             {#4}%
             {%
```

Try root language.

```
                \let\CurrentTrackedTag\CurrentTrackedLanguage
                \@tracklang@IfFileExists{#2\CurrentTrackedTag#3}{#4}{#5}%
             }%
          }%
          {%
```

No region, no ISO 639-1 code and no ISO 639-2 code. No file found for dialect label, try root language label next.

```
           \let\CurrentTrackedTag\CurrentTrackedLanguage
           \@tracklang@IfFileExists{#2\CurrentTrackedTag#3}{#4}{#5}%
          }%
       }%
    \else
```

Has a region so try ⟨*ISO 639-1 code*⟩-⟨*region*⟩

```
        \edef\CurrentTrackedTag{%
          \CurrentTrackedIsoCode-\CurrentTrackedRegion}%
        \@tracklang@IfFileExists{#2\CurrentTrackedTag#3}%
        {#4}
        {%
```

Has a region but ⟨*ISO 639-1 code*⟩-⟨*region*⟩ not found, so try ⟨*ISO 639-2 code*⟩-⟨*region*⟩

```
\IfTrackedLanguageHasIsoCode
{639-2}{\CurrentTrackedLanguage}
{%
  \let\org@currenttrackedisocode\CurrentTrackedIsoCode
  \edef\CurrentTrackedIsoCode{%
    \TrackedIsoCodeFromLanguage
    {639-2}{\CurrentTrackedLanguage}}%
  \edef\CurrentTrackedTag{%
    \CurrentTrackedIsoCode-\CurrentTrackedRegion}%
  \@tracklang@IfFileExists{#2\CurrentTrackedTag#3}%
  {#4}%
  {%
```

Has a region but ⟨*ISO 639-1 code*⟩-⟨*region*⟩ not found and ⟨*ISO 639-2 code*⟩-⟨*region*⟩ not found, so try just ⟨*ISO 639-1 code*⟩

```
    \let\CurrentTrackedTag\org@currenttrackedisocode
    \let\org@currenttrackedisocode\CurrentTrackedIsoCode
    \let\CurrentTrackedIsoCode\CurrentTrackedTag
    \@tracklang@IfFileExists{#2\CurrentTrackedTag#3}%
    {#4}
    {%
```

Try just the ISO 639-2 code.

```
      \let\CurrentTrackedIsoCode\org@currenttrackedisocode
      \let\CurrentTrackedTag\CurrentTrackedIsoCode
      \@tracklang@IfFileExists{#2\CurrentTrackedTag#3}%
      {#4}%
      {%
```

Try just the region.

```
        \let\CurrentTrackedTag\CurrentTrackedRegion
        \@tracklang@IfFileExists{#2\CurrentTrackedTag#3}%
        {#4}%
        {%
```

Try the root language.

```
          \let\CurrentTrackedTag\CurrentTrackedLanguage
          \@tracklang@IfFileExists{#2\CurrentTrackedTag#3}{#4}{#5}%
        }%
      }%
    }%
  }%
}%
{%
```

Has a region but ⟨*ISO 639-1 code*⟩-⟨*region*⟩ not found, and no ISO 639-2 code so just try ⟨*ISO 639-1 code*⟩

```
  \let\CurrentTrackedTag\CurrentTrackedIsoCode
  \@tracklang@IfFileExists{#2\CurrentTrackedTag#3}%
  {#4}%
  {%
```

Just try the region

```
    \let\CurrentTrackedTag\CurrentTrackedRegion
    \@tracklang@IfFileExists{#2\CurrentTrackedTag#3}%
    {#4}%
    {%
```

Try the root language.

```
                \let\CurrentTrackedTag\CurrentTrackedLanguage
                \@tracklang@IfFileExists{#2\CurrentTrackedTag#3}{#4}{#5}%
              }%
            }%
          }%
        }%
      \fi
    }%
    {%
```

No ISO 639-1 code. Try ISO 639-2 code.

```
      \IfTrackedLanguageHasIsoCode
        {639-2}{\CurrentTrackedLanguage}
      {%
         \edef\CurrentTrackedIsoCode{%
            \TrackedIsoCodeFromLanguage
            {639-2}{\CurrentTrackedLanguage}}%
      }%
      {%
```

No ISO 639-1 code or ISO 639-2 code. Try 639-3 code.

```
        \IfTrackedLanguageHasIsoCode
          {639-3}{\CurrentTrackedLanguage}
        {%
           \edef\CurrentTrackedIsoCode{%
              \TrackedIsoCodeFromLanguage
              {639-3}{\CurrentTrackedLanguage}}%
        }%
        {%
           \let\CurrentTrackedIsoCode\empty
        }%
      }%
      \ifx\CurrentTrackedIsoCode\empty
```

No ISO 639-1 code or ISO 639-2 code. Try just the region.

```
        \ifx\CurrentTrackedRegion\empty
```

No region. Try the root language.

```
          \let\CurrentTrackedTag\CurrentTrackedLanguage
          \@tracklang@IfFileExists{#2\CurrentTrackedTag#3}{#4}{#5}%
        \else
```

Try the region.

```
          \let\CurrentTrackedTag\CurrentTrackedRegion
          \@tracklang@IfFileExists{#2\CurrentTrackedTag#3}%
          {#4}%
          {%
```

Try the root language.

```
            \let\CurrentTrackedTag\CurrentTrackedLanguage
            \@tracklang@IfFileExists{#2\CurrentTrackedTag#3}{#4}{#5}%
          }%
        \fi
      \else
```

Has ISO 639-2 or 639-3 code.

```
        \ifx\CurrentTrackedRegion\empty
```

Has ISO 639-2 or 639-3 code but no region.

```
          \let\CurrentTrackedTag\CurrentTrackedIsoCode
          \@tracklang@IfFileExists{#2\CurrentTrackedTag#3}%
```

```
      {#4}%
      {%
```

Try the root language.

```
        \let\CurrentTrackedTag\CurrentTrackedLanguage
        \@tracklang@IfFileExists{#2\CurrentTrackedTag#3}{#4}{#5}%
      }%
    \else
```

Has ISO 639-2 or 639-3 code and a region. Try ⟨*ISO 639-2 or 639-3*⟩-⟨*region*⟩ first.

```
      \edef\CurrentTrackedTag{%
        \CurrentTrackedIsoCode-\CurrentTrackedRegion}%
      \@tracklang@IfFileExists{#2\CurrentTrackedTag#3}%
      {#4}%
      {%
```

Has ISO 639-2 or 639-3 code and a region but ⟨*ISO 639-2 or 639-3*⟩-⟨*region*⟩ doesn't exist. Try just ⟨*ISO 639-2 or 639-3*⟩

```
        \let\CurrentTrackedTag\CurrentTrackedIsoCode
        \@tracklang@IfFileExists{#2\CurrentTrackedTag#3}%
        {#4}%
        {%
```

Has ISO 639-2 or 639-3 code and a region but ⟨*ISO 639-2 or 639-3*⟩ doesn't exist. Try just ⟨*region*⟩

```
          \let\CurrentTrackedTag\CurrentTrackedRegion
          \@tracklang@IfFileExists{#2\CurrentTrackedTag#3}%
          {#4}%
          {%
```

Try the root language.

```
            \let\CurrentTrackedTag\CurrentTrackedLanguage
            \@tracklang@IfFileExists{#2\CurrentTrackedTag#3}{#4}{#5}%
          }%
        }%
      }%
    \fi
  \fi
 }%
}%
{#5}% unknown dialect
}
```

6.2.9 Resources

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

`\TrackLangProvidesResource`

`\TrackLangProvidesResource{`⟨*tag*⟩`}[`⟨*version*⟩`]`

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.

```
  \long\def\TrackLangProvidesResource#1#2{%
    \ifx\TrackLangRequireDialectPrefix\undefined
      \@tracklang@err{Resources files using
        \string\TrackLangProvidesResource\space
        must be loaded with \string\TrackLangRequireDialect}%
    \fi
    \ifx#2[\relax
      \def\@tracklang@next{%
        \@tracklang@providesresource{\TrackLangRequireDialectPrefix-#1.ldf}#2%
      }
    \else
      \expandafter\xdef\csname ver@\TrackLangRequireDialectPrefix
        -#1.ldf\endcsname{}%
      {%
        \newlinechar=`\^^J
        \def\MessageBreak{^^J}%
        \message{^^JFile: \TrackLangRequireDialectPrefix-#1.ldf^^J}%
      }%
      \long\def\@tracklang@next{#2}%
    \fi
    \@tracklang@next
  }
  \def\@tracklang@providesresource#1[#2]{%
    \expandafter\xdef\csname ver@#1\endcsname{#2}%
     {%
       \newlinechar=`\^^J
       \def\MessageBreak{^^J}%
       \message{^^JFile: #1 #2^^J}%
     }%
  }
\else
```

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
```

\TrackLangAddToHook

\TrackLangAddToHook{⟨*code*⟩}{⟨*type*⟩}

Within the resource files, a check is required for the language hook, where the hook type is given by ⟨*type*⟩. For example, if ⟨*type*⟩ is captions, the for babel, this is \captions⟨*dialect*⟩ (dialect obtained through \CurrentTrackedDialect) and for polyglossia, this is \captions⟨*languag* (language obtained through \CurrentTrackedLanguage). This command is not permitted outside resource files.

```
\def\TrackLangAddToHook{\noop@TrackLangAddToHook}
```

\noop@TrackLangAddToHook

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

\@TrackLangAddToHook

```
\def\@TrackLangAddToHook#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 the code now.

```
          #1%
        }%
        {%
          \@tracklang@addtohook{#2}{\CurrentTrackedLanguage}{#1}%
        }%
      }%
      {%
        \@tracklang@addtohook{#2}{\@tracklang@tmp}{#1}%
      }%
    }%
    {%
```

polyglossia check next.

```
      \@tracklang@ifundef{#2\CurrentTrackedLanguage}%
      {%
```

No captions hook.

```
      }%
      {%
        \@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

\@tracklang@addtohook{⟨*type*⟩}{⟨*label*⟩}{⟨*code*⟩}

```
\def\@tracklang@addtohook#1#2#3{%
 \expandafter\let\expandafter\@tracklang@hook\csname #1#2\endcsname
 \expandafter
  \gdef\csname#1#2\expandafter\endcsname\expandafter{\@tracklang@hook#3}%
}
```

Since the captions hook is the most common, provide a shortcut.

\TrackLangAddToCaptions

\TrackLangAddToCaptions{⟨*code*⟩}

```
\def\TrackLangAddToCaptions#1{\TrackLangAddToHook{#1}{captions}}
```

\SetTrackedDialectLabelMap

\SetTrackedDialectLabelMap{⟨*tracklang-label*⟩}{⟨*hook-label*⟩}

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

\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}%
}
```

fHookHasMappingFromTrackedDialect

\IfHookHasMappingFromTrackedDialect{⟨*hook label*⟩}{⟨*true*⟩}{⟨*false*⟩}

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

\GetTrackedDialectToMapping{⟨*tracklang label*⟩}

Gets the mapping for the given tracklang dialect label to a language hook label or the ⟨*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

\GetTrackedDialectFromMapping{⟨*language hook*⟩}

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

\TrackLangRequireResource{⟨*tag*⟩}

```
\def\TrackLangRequireResource{\noop@TrackLangRequireResource}
```

\noop@TrackLangRequireResource

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

```
\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

\TrackLangRequireResourceOrDo{⟨*tag*⟩}{⟨*resource loaded code*⟩}{⟨*resource already loaded code*⟩}

Like \TrackLangRequireResource but also does ⟨*resource loaded code*⟩ if the resource file is loaded or ⟨*resource already loaded code*⟩ if the resource file has already been loaded.

```
\def\TrackLangRequireResourceOrDo{%
  \noop@TrackLangRequireResourceOrDo
}
```

oop@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)}%
    }%
  }%
  {#3}%
}
```

\TrackLangRequestResource `\TrackLangRequestResource{⟨tag⟩}{⟨not found code⟩}`

Like \TrackLangRequireResource but does ⟨*not found code*⟩ 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@err{\string\TrackLangRequestResource\space
  only permitted within resource files}
}
```

\@TrackLangRequestResource Actual behaviour.

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

\TrackLangRequireDialect `\TrackLangRequireDialect[⟨load code⟩]{⟨pkgname⟩}{⟨dialect⟩}`

```
\def\TrackLangRequireDialect{\@TrackLangRequireDialect}
```

\noop@TrackLangRequireDialect No-op code.

```
\def\noop@TrackLangRequireDialect#1{%
  \ifx[#1\relax
    \def\@tracklang@next{\@noop@TrackLangRequireDialect[]%
  \else
    \def\@tracklang@next{\@noop@TrackLangRequireDialect[]{#1}}%
  \fi
  \@tracklang@next
}
\def\@noop@TrackLangRequireDialect[#1]#2#3{%
  \@tracklang@err{\string\TrackLangRequireDialect\space
  only permitted within resource files}
}
```

\@TrackLangRequireDialect Actual code.

```
\def\@TrackLangRequireDialect#1{%
  \ifx[#1\relax
    \def\@tracklang@next{\@@TrackLangRequireDialect[]%
  \else
    \def\@tracklang@next{%
      \@@TrackLangRequireDialect
        [\TrackLangRequireResource{\CurrentTrackedTag}]{#1}}%
  \fi
  \@tracklang@next
}
\def\@@TrackLangRequireDialect[#1]#2#3{%
   \def\TrackLangRequireDialectPrefix{#2}%
   \IfTrackedLanguageFileExists{#3}%
   {#2-}% prefix
   {.ldf}% suffix
   {%
```

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
```

6.3 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\relax
\fi
```

Check if this file has already been loaded:

```
\ifx\TrackLangRegionMap\undefined
\else
  \@tracklang@regions@restore@at
  \expandafter\endinput
\fi
```

Version info.

```
\expandafter\def\csname ver@tracklang-region-codes.tex\endcsname{%
 2019/08/31 v1.3.7 (NLCT) Track Languages Regions}%
```

\TrackLangRegionMap `\TrackLangRegionMap{⟨numeric code⟩}{⟨alpha-2 code⟩}{⟨alpha-3 code⟩}`

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 `\TrackLangAlphaIIToNumericRegion{⟨alpha-2 code⟩}`

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

\TrackLangNumericToAlphaIIRegion

\TrackLangNumericToAlphaIIRegion{⟨*numeric code*⟩}

```
\def\TrackLangNumericToAlphaIIRegion#1{%
  \@tracklang@nameuse{@tracklang@region@numtoalphaii@#1}%
}%
```

\TrackLangIfKnownAlphaIIRegion

\TrackLangIfKnownAlphaIIRegion{⟨*alpha-2 code*⟩}{⟨*true*⟩}{⟨*false*⟩}

```
\def\TrackLangIfKnownAlphaIIRegion#1#2#3{%
  \@tracklang@ifundef{@tracklang@region@alphaiitonum@#1}%
  {#3}%
  {#2}%
}%
```

\TrackLangIfKnownNumericRegion

\TrackLangIfKnownNumericRegion{⟨*numeric code*⟩}{⟨*true*⟩}{⟨*false*⟩}

```
\def\TrackLangIfKnownNumericRegion#1#2#3{%
  \@tracklang@ifundef{@tracklang@region@numtoalphaii@#1}%
  {#3}%
  {#2}%
}%
```

\TrackLangAlphaIIIToNumericRegion

\TrackLangAlphaIIIToNumericRegion{⟨*alpha-3 code*⟩}

```
\def\TrackLangAlphaIIIToNumericRegion#1{%
  \@tracklang@nameuse{@tracklang@region@alphaiiitonum@#1}%
}%
```

\TrackLangNumericToAlphaIIIRegion

\TrackLangNumericToAlphaIIIRegion{⟨*numeric code*⟩}

```
\def\TrackLangNumericToAlphaIIIRegion#1{%
  \@tracklang@nameuse{@tracklang@region@numtoalphaiii@#1}%
}%
```

\TrackLangIfKnownAlphaIIIRegion

\TrackLangIfKnownAlphaIIIRegion{⟨*alpha-3 code*⟩}{⟨*true*⟩}{⟨*false*⟩}

```
\def\TrackLangIfKnownAlphaIIIRegion#1#2#3{%
  \@tracklang@ifundef{@tracklang@region@alphaiiitonum@#1}%
  {#3}%
  {#2}%
}%
```

Define mappings.

```
\TrackLangRegionMap{004}{AF}{AFG}%
\TrackLangRegionMap{248}{AX}{ALA}%
\TrackLangRegionMap{008}{AL}{ALB}%
\TrackLangRegionMap{012}{DZ}{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}{BMU}%
\TrackLangRegionMap{064}{BT}{BTN}%
\TrackLangRegionMap{068}{BO}{BOL}%
\TrackLangRegionMap{535}{BQ}{BES}%
\TrackLangRegionMap{070}{BA}{BIH}%
\TrackLangRegionMap{072}{BW}{BWA}%
\TrackLangRegionMap{074}{BV}{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}%
\TrackLangRegionMap{132}{CV}{CPV}%
\TrackLangRegionMap{116}{KH}{KHM}%
\TrackLangRegionMap{120}{CM}{CMR}%
\TrackLangRegionMap{124}{CA}{CAN}%
\TrackLangRegionMap{136}{KY}{CYM}%
\TrackLangRegionMap{140}{CF}{CAF}%
\TrackLangRegionMap{148}{TD}{TCD}%
\TrackLangRegionMap{152}{CL}{CHL}%
\TrackLangRegionMap{156}{CN}{CHN}%
\TrackLangRegionMap{162}{CX}{CXR}%
\TrackLangRegionMap{166}{CC}{CCK}%
\TrackLangRegionMap{170}{CO}{COL}%
\TrackLangRegionMap{174}{KM}{COM}%
\TrackLangRegionMap{180}{CD}{COD}%
\TrackLangRegionMap{178}{CG}{COG}%
\TrackLangRegionMap{184}{CK}{COK}%
\TrackLangRegionMap{188}{CR}{CRI}%
\TrackLangRegionMap{384}{CI}{CIV}%
\TrackLangRegionMap{191}{HR}{HRV}%
\TrackLangRegionMap{192}{CU}{CUB}%
\TrackLangRegionMap{531}{CW}{CUW}%
\TrackLangRegionMap{196}{CY}{CYP}%
\TrackLangRegionMap{203}{CZ}{CZE}%
```

```
\TrackLangRegionMap{208}{DK}{DNK}%
\TrackLangRegionMap{262}{DJ}{DJI}%
\TrackLangRegionMap{212}{DM}{DMA}%
\TrackLangRegionMap{214}{DO}{DOM}%
\TrackLangRegionMap{218}{EC}{ECU}%
\TrackLangRegionMap{818}{EG}{EGY}%
\TrackLangRegionMap{222}{SV}{SLV}%
\TrackLangRegionMap{226}{GQ}{GNQ}%
\TrackLangRegionMap{232}{ER}{ERI}%
\TrackLangRegionMap{233}{EE}{EST}%
\TrackLangRegionMap{231}{ET}{ETH}%
\TrackLangRegionMap{238}{FK}{FLK}%
\TrackLangRegionMap{234}{FO}{FRO}%
\TrackLangRegionMap{242}{FJ}{FJI}%
\TrackLangRegionMap{246}{FI}{FIN}%
\TrackLangRegionMap{250}{FR}{FRA}%
\TrackLangRegionMap{254}{GF}{GUF}%
\TrackLangRegionMap{258}{PF}{PYF}%
\TrackLangRegionMap{260}{TF}{ATF}%
\TrackLangRegionMap{266}{GA}{GAB}%
\TrackLangRegionMap{270}{GM}{GMB}%
\TrackLangRegionMap{268}{GE}{GEO}%
\TrackLangRegionMap{276}{DE}{DEU}%
\TrackLangRegionMap{288}{GH}{GHA}%
\TrackLangRegionMap{292}{GI}{GIB}%
\TrackLangRegionMap{300}{GR}{GRC}%
\TrackLangRegionMap{304}{GL}{GRL}%
\TrackLangRegionMap{308}{GD}{GRD}%
\TrackLangRegionMap{312}{GP}{GLP}%
\TrackLangRegionMap{316}{GU}{GUM}%
\TrackLangRegionMap{320}{GT}{GTM}%
\TrackLangRegionMap{831}{GG}{GGY}%
\TrackLangRegionMap{324}{GN}{GIN}%
\TrackLangRegionMap{624}{GW}{GNB}%
\TrackLangRegionMap{328}{GY}{GUY}%
\TrackLangRegionMap{332}{HT}{HTI}%
\TrackLangRegionMap{334}{HM}{HMD}%
\TrackLangRegionMap{336}{VA}{VAT}%
\TrackLangRegionMap{340}{HN}{HND}%
\TrackLangRegionMap{344}{HK}{HKG}%
\TrackLangRegionMap{348}{HU}{HUN}%
\TrackLangRegionMap{352}{IS}{ISL}%
\TrackLangRegionMap{356}{IN}{IND}%
\TrackLangRegionMap{360}{ID}{IDN}%
\TrackLangRegionMap{364}{IR}{IRN}%
\TrackLangRegionMap{368}{IQ}{IRQ}%
\TrackLangRegionMap{372}{IE}{IRL}%
\TrackLangRegionMap{833}{IM}{IMN}%
\TrackLangRegionMap{376}{IL}{ISR}%
\TrackLangRegionMap{380}{IT}{ITA}%
\TrackLangRegionMap{388}{JM}{JAM}%
\TrackLangRegionMap{392}{JP}{JPN}%
\TrackLangRegionMap{832}{JE}{JEY}%
\TrackLangRegionMap{400}{JO}{JOR}%
\TrackLangRegionMap{398}{KZ}{KAZ}%
\TrackLangRegionMap{404}{KE}{KEN}%
\TrackLangRegionMap{296}{KI}{KIR}%
\TrackLangRegionMap{408}{KP}{PRK}%
\TrackLangRegionMap{410}{KR}{KOR}%
\TrackLangRegionMap{414}{KW}{KWT}%
\TrackLangRegionMap{417}{KG}{KGZ}%
\TrackLangRegionMap{418}{LA}{LAO}%
```

```
\TrackLangRegionMap{428}{LV}{LVA}%
\TrackLangRegionMap{422}{LB}{LBN}%
\TrackLangRegionMap{426}{LS}{LSO}%
\TrackLangRegionMap{430}{LR}{LBR}%
\TrackLangRegionMap{434}{LY}{LBY}%
\TrackLangRegionMap{438}{LI}{LIE}%
\TrackLangRegionMap{440}{LT}{LTU}%
\TrackLangRegionMap{442}{LU}{LUX}%
\TrackLangRegionMap{446}{MO}{MAC}%
\TrackLangRegionMap{807}{MK}{MKD}%
\TrackLangRegionMap{450}{MG}{MDG}%
\TrackLangRegionMap{454}{MW}{MWI}%
\TrackLangRegionMap{458}{MY}{MYS}%
\TrackLangRegionMap{462}{MV}{MDV}%
\TrackLangRegionMap{466}{ML}{MLI}%
\TrackLangRegionMap{470}{MT}{MLT}%
\TrackLangRegionMap{584}{MH}{MHL}%
\TrackLangRegionMap{474}{MQ}{MTQ}%
\TrackLangRegionMap{478}{MR}{MRT}%
\TrackLangRegionMap{480}{MU}{MUS}%
\TrackLangRegionMap{175}{YT}{MYT}%
\TrackLangRegionMap{484}{MX}{MEX}%
\TrackLangRegionMap{583}{FM}{FSM}%
\TrackLangRegionMap{498}{MD}{MDA}%
\TrackLangRegionMap{492}{MC}{MCO}%
\TrackLangRegionMap{496}{MN}{MNG}%
\TrackLangRegionMap{499}{ME}{MNE}%
\TrackLangRegionMap{500}{MS}{MSR}%
\TrackLangRegionMap{504}{MA}{MAR}%
\TrackLangRegionMap{508}{MZ}{MOZ}%
\TrackLangRegionMap{104}{MM}{MMR}%
\TrackLangRegionMap{516}{NA}{NAM}%
\TrackLangRegionMap{520}{NR}{NRU}%
\TrackLangRegionMap{524}{NP}{NPL}%
\TrackLangRegionMap{528}{NL}{NLD}%
\TrackLangRegionMap{540}{NC}{NCL}%
\TrackLangRegionMap{554}{NZ}{NZL}%
\TrackLangRegionMap{558}{NI}{NIC}%
\TrackLangRegionMap{562}{NE}{NER}%
\TrackLangRegionMap{566}{NG}{NGA}%
\TrackLangRegionMap{570}{NU}{NIU}%
\TrackLangRegionMap{574}{NF}{NFK}%
\TrackLangRegionMap{580}{MP}{MNP}%
\TrackLangRegionMap{578}{NO}{NOR}%
\TrackLangRegionMap{512}{OM}{OMN}%
\TrackLangRegionMap{586}{PK}{PAK}%
\TrackLangRegionMap{585}{PW}{PLW}%
\TrackLangRegionMap{275}{PS}{PSE}%
\TrackLangRegionMap{591}{PA}{PAN}%
\TrackLangRegionMap{598}{PG}{PNG}%
\TrackLangRegionMap{600}{PY}{PRY}%
\TrackLangRegionMap{604}{PE}{PER}%
\TrackLangRegionMap{608}{PH}{PHL}%
\TrackLangRegionMap{612}{PN}{PCN}%
\TrackLangRegionMap{616}{PL}{POL}%
\TrackLangRegionMap{620}{PT}{PRT}%
\TrackLangRegionMap{630}{PR}{PRI}%
\TrackLangRegionMap{634}{QA}{QAT}%
\TrackLangRegionMap{638}{RE}{REU}%
\TrackLangRegionMap{642}{RO}{ROU}%
\TrackLangRegionMap{643}{RU}{RUS}%
\TrackLangRegionMap{646}{RW}{RWA}%
```

```
\TrackLangRegionMap{652}{BL}{BLM}%
\TrackLangRegionMap{654}{SH}{SHN}%
\TrackLangRegionMap{659}{KN}{KNA}%
\TrackLangRegionMap{662}{LC}{LCA}%
\TrackLangRegionMap{663}{MF}{MAF}%
\TrackLangRegionMap{666}{PM}{SPM}%
\TrackLangRegionMap{670}{VC}{VCT}%
\TrackLangRegionMap{882}{WS}{WSM}%
\TrackLangRegionMap{674}{SM}{SMR}%
\TrackLangRegionMap{678}{ST}{STP}%
\TrackLangRegionMap{682}{SA}{SAU}%
\TrackLangRegionMap{686}{SN}{SEN}%
\TrackLangRegionMap{688}{RS}{SRB}%
\TrackLangRegionMap{690}{SC}{SYC}%
\TrackLangRegionMap{694}{SL}{SLE}%
\TrackLangRegionMap{702}{SG}{SGP}%
\TrackLangRegionMap{534}{SX}{SXM}%
\TrackLangRegionMap{703}{SK}{SVK}%
\TrackLangRegionMap{705}{SI}{SVN}%
\TrackLangRegionMap{090}{SB}{SLB}%
\TrackLangRegionMap{706}{SO}{SOM}%
\TrackLangRegionMap{710}{ZA}{ZAF}%
\TrackLangRegionMap{239}{GS}{SGS}%
\TrackLangRegionMap{728}{SS}{SSD}%
\TrackLangRegionMap{724}{ES}{ESP}%
\TrackLangRegionMap{144}{LK}{LKA}%
\TrackLangRegionMap{729}{SD}{SDN}%
\TrackLangRegionMap{740}{SR}{SUR}%
\TrackLangRegionMap{744}{SJ}{SJM}%
\TrackLangRegionMap{748}{SZ}{SWZ}%
\TrackLangRegionMap{752}{SE}{SWE}%
\TrackLangRegionMap{756}{CH}{CHE}%
\TrackLangRegionMap{760}{SY}{SYR}%
\TrackLangRegionMap{158}{TW}{TWN}%
\TrackLangRegionMap{762}{TJ}{TJK}%
\TrackLangRegionMap{834}{TZ}{TZA}%
\TrackLangRegionMap{764}{TH}{THA}%
\TrackLangRegionMap{626}{TL}{TLS}%
\TrackLangRegionMap{768}{TG}{TGO}%
\TrackLangRegionMap{772}{TK}{TKL}%
\TrackLangRegionMap{776}{TO}{TON}%
\TrackLangRegionMap{780}{TT}{TTO}%
\TrackLangRegionMap{788}{TN}{TUN}%
\TrackLangRegionMap{792}{TR}{TUR}%
\TrackLangRegionMap{795}{TM}{TKM}%
\TrackLangRegionMap{796}{TC}{TCA}%
\TrackLangRegionMap{798}{TV}{TUV}%
\TrackLangRegionMap{800}{UG}{UGA}%
\TrackLangRegionMap{804}{UA}{UKR}%
\TrackLangRegionMap{784}{AE}{ARE}%
\TrackLangRegionMap{826}{GB}{GBR}%
\TrackLangRegionMap{581}{UM}{UMI}%
\TrackLangRegionMap{840}{US}{USA}%
\TrackLangRegionMap{858}{UY}{URY}%
\TrackLangRegionMap{860}{UZ}{UZB}%
\TrackLangRegionMap{548}{VU}{VUT}%
\TrackLangRegionMap{862}{VE}{VEN}%
\TrackLangRegionMap{704}{VN}{VNM}%
\TrackLangRegionMap{092}{VG}{VGB}%
\TrackLangRegionMap{850}{VI}{VIR}%
\TrackLangRegionMap{876}{WF}{WLF}%
\TrackLangRegionMap{732}{EH}{ESH}%
```

```
\TrackLangRegionMap{887}{YE}{YEM}%
\TrackLangRegionMap{894}{ZM}{ZMB}%
\TrackLangRegionMap{716}{ZW}{ZWE}%
```

Restore category code of `@`.

```
\@tracklang@regions@restore@at
```

6.4 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}[2019/08/31 v1.3.7 (NLCT) Track
Language Scripts (LaTeX)]
\RequirePackage{tracklang}
\input{tracklang-scripts}
```

6.5 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`\noexpand\@=\number\catcode`\@\relax
  }%
 \catcode`\@=11\relax
\fi
```

Check if this file has already been loaded:

```
\ifx\TrackLangScriptMap\undefined
\else
  \@tracklang@scripts@restore@at
  \expandafter\endinput
\fi
```

Version info.

```
\expandafter\def\csname ver@tracklang-scripts.tex\endcsname{%
 2019/08/31 v1.3.7 (NLCT) Track Languages Scripts (Generic)}%
```

\TrackLangScriptsMap

\TrackLangScriptMap{⟨*letter code*⟩}{⟨*number*⟩}{⟨*name*⟩}{⟨*direction*⟩}{⟨*parent*⟩}

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 `\ifx`.

```
  \@tracklang@enamedef{TrackLangScript#1}{#1}%
  \@tracklang@enamedef{@tracklang@script@numtoalpha@#2}{#1}%
  \@tracklang@enamedef{@tracklang@script@alphatonum@#1}{#2}%
  \@tracklang@enamedef{@tracklang@script@alphatoname@#1}{#3}%
  \@tracklang@enamedef{@tracklang@script@alphatodir@#1}{#4}%
  \ifx\relax#5\relax
  \else
    \@tracklang@enamedef{@tracklang@script@parent@#1}{#5}%
  \fi
}
```

\TrackLangScriptAlphaToNumeric

\TrackLangScriptAlphaToNumeric{⟨*alpha code*⟩}

```
\def\TrackLangScriptAlphaToNumeric#1{%
  \@tracklang@nameuse{@tracklang@script@alphatonum@#1}%
}%
```

\TrackLangScriptIfKnownAlpha

\TrackLangScriptIfKnownAlpha{⟨*alpha code*⟩}{⟨*true*⟩}{⟨*false*⟩}

```
\def\TrackLangScriptIfKnownAlpha#1#2#3{%
  \@tracklang@ifundef{@tracklang@script@alphatonum@#1}%
  {#3}%
  {#2}%
}%
```

\TrackLangScriptNumericToAlpha

\TrackLangScriptNumericToAlpha{⟨*numeric code*⟩}

```
\def\TrackLangScriptNumericToAlpha#1{%
  \@tracklang@nameuse{@tracklang@script@numtoalpha@#1}%
}%
```

\TrackLangScriptIfKnownNumeric

\TrackLangScriptIfKnownNumeric{⟨*numeric code*⟩}{⟨*true*⟩}{⟨*false*⟩}

```
\def\TrackLangScriptIfKnownNumeric#1#2#3{%
  \@tracklang@ifundef{@tracklang@script@numtoalpha@#1}%
  {#3}%
  {#2}%
}%
```

\TrackLangScriptAlphaToName

\TrackLangScriptAlphaToName{⟨*alpha code*⟩}

```
\def\TrackLangScriptAlphaToName#1{%
  \@tracklang@nameuse{@tracklang@script@alphatoname@#1}%
}%
```

\TrackLangScriptAlphaToDir `\TrackLangScriptAlphaToDir{⟨alpha code⟩}`

```
\def\TrackLangScriptAlphaToDir#1{%
  \@tracklang@nameuse{@tracklang@script@alphatodir@#1}%
}%
```

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.

\TrackLangScriptSetParent `\TrackLangScriptSetParent{⟨alpha code⟩}{⟨parent alpha code⟩}`

```
\def\TrackLangScriptSetParent#1#2{%
  \@tracklang@enamedef{@tracklang@script@parent@#1}{#2}%
}%
```

\TrackLangScriptGetParent `\TrackLangScriptGetParent{⟨alpha code⟩}`

```
\def\TrackLangScriptGetParent#1{%
  \@tracklang@nameuse{@tracklang@script@parent@#1}%
}%
```

\TrackLangScriptIfHasParent `\TrackLangScriptIfHasParent{⟨alpha code⟩}{⟨true⟩}{⟨false⟩}`

```
\def\TrackLangScriptIfHasParent#1#2#3{%
  \@tracklang@ifundef{@tracklang@script@parent@#1}%
  {#3}%
  {#2}%
}%
```

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{Armn}{230}{Armenian}{LR}{}
\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{Cyrl}{220}{Cyrillic}{LR}{}
\TrackLangScriptMap{Cyrs}{221}{Cyrillic (Old Church Slavonic{}
variant)}{varies}{}
\TrackLangScriptMap{Deva}{315}{Devanagari (Nagari)}{LR}{}
\TrackLangScriptMap{Dsrt}{250}{Deseret (Mormon)}{LR}{}
\TrackLangScriptMap{Dupl}{755}{Duployan shorthand, Duployan{}
stenography}{LR}{}
\TrackLangScriptMap{Egyd}{070}{Egyptian demotic}{RL}{}
\TrackLangScriptMap{Egyh}{060}{Egyptian hieratic}{RL}{}
\TrackLangScriptMap{Egyp}{050}{Egyptian hieroglyphs}{LR}{}
\TrackLangScriptMap{Elba}{226}{Elbasan}{LR}{}
\TrackLangScriptMap{Ethi}{430}{Ethiopic (Ge'ez)}{LR}{}
\TrackLangScriptMap{Geok}{241}{Khutsuri (Asomtavruli and{}
Nuskhuri)}{LR}{}
\TrackLangScriptMap{Geor}{240}{Georgian (Mkhedruli)}{LR}{}
\TrackLangScriptMap{Glag}{225}{Glagolitic}{LR}{}
\TrackLangScriptMap{Goth}{206}{Gothic}{LR}{}
\TrackLangScriptMap{Gran}{343}{Grantha}{LR}{}
\TrackLangScriptMap{Grek}{200}{Greek}{LR}{}
\TrackLangScriptMap{Gujr}{320}{Gujarati}{LR}{}
\TrackLangScriptMap{Guru}{310}{Gurmukhi}{LR}{}
\TrackLangScriptMap{Hanb}{503}{Han with Bopomofo (alias for Han +{}
Bopomofo)}{LR}{}
\TrackLangScriptMap{Hang}{286}{Hangul}{LR}{}
\TrackLangScriptMap{Hani}{500}{Han (Hanzi, Kanji, Hanja)}{LR}{}
\TrackLangScriptMap{Hano}{371}{Hanunoo}{LR}{}
\TrackLangScriptMap{Hans}{501}{Han (Simplified variant)}{varies}{}
\TrackLangScriptMap{Hant}{502}{Han (Traditional variant)}{varies}{}
\TrackLangScriptMap{Hatr}{127}{Hatran}{RL}{}
\TrackLangScriptMap{Hebr}{125}{Hebrew}{RL}{}
\TrackLangScriptMap{Hira}{410}{Hiragana}{LR}{}
\TrackLangScriptMap{Hluw}{080}{Anatolian Hieroglyphs (Luwian{}
Hieroglyphs, Hittite Hieroglyphs)}{LR}{}
\TrackLangScriptMap{Hmng}{450}{Pahawh Hmong}{LR}{}
\TrackLangScriptMap{Hrkt}{412}{Japanese syllabaries (alias for{}
Hiragana + Katakana)}{varies}{}
\TrackLangScriptMap{Hung}{176}{Old Hungarian (Hungarian Runic)}{RL}{}
\TrackLangScriptMap{Inds}{610}{Indus (Harappan)}{RL}{}
\TrackLangScriptMap{Ital}{210}{Old Italic (Etruscan, Oscan, etc.)}{LR}{}
\TrackLangScriptMap{Jamo}{284}{Jamo (alias for Jamo subset of{}
Hangul)}{LR}{}
\TrackLangScriptMap{Java}{361}{Javanese}{LR}{}
\TrackLangScriptMap{Jpan}{413}{Japanese (alias for Han + Hiragana +{}
Katakana)}{varies}{}
\TrackLangScriptMap{Jurc}{510}{Jurchen}{LR}{}
```

```
\TrackLangScriptMap{Kali}{357}{Kayah Li}{LR}{}
\TrackLangScriptMap{Kana}{411}{Katakana}{LR}{}
\TrackLangScriptMap{Khar}{305}{Kharoshthi}{RL}{}
\TrackLangScriptMap{Khmr}{355}{Khmer}{LR}{}
\TrackLangScriptMap{Khoj}{322}{Khojki}{LR}{}
\TrackLangScriptMap{Kitl}{505}{Khitan large script}{LR}{}
\TrackLangScriptMap{Kits}{288}{Khitan small script}{TB}{}
\TrackLangScriptMap{Knda}{345}{Kannada}{LR}{}
\TrackLangScriptMap{Kore}{287}{Korean (alias for Hangul + Han)}{LR}{}
\TrackLangScriptMap{Kpel}{436}{Kpelle}{LR}{}
\TrackLangScriptMap{Kthi}{317}{Kaithi}{LR}{}
\TrackLangScriptMap{Lana}{351}{Tai Tham (Lanna)}{LR}{}
\TrackLangScriptMap{Laoo}{356}{Lao}{LR}{}
\TrackLangScriptMap{Latf}{217}{Latin (Fraktur variant)}{varies}{}
\TrackLangScriptMap{Latg}{216}{Latin (Gaelic variant)}{LR}{}
\TrackLangScriptMap{Latn}{215}{Latin}{LR}{}
\TrackLangScriptMap{Leke}{364}{Leke}{LR}{}
\TrackLangScriptMap{Lepc}{335}{Lepcha}{LR}{}
\TrackLangScriptMap{Limb}{336}{Limbu}{LR}{}
\TrackLangScriptMap{Lina}{400}{Linear A}{LR}{}
\TrackLangScriptMap{Linb}{401}{Linear B}{LR}{}
\TrackLangScriptMap{Lisu}{399}{Lisu (Fraser)}{LR}{}
\TrackLangScriptMap{Loma}{437}{Loma}{LR}{}
\TrackLangScriptMap{Lyci}{202}{Lycian}{LR}{}
\TrackLangScriptMap{Lydi}{116}{Lydian}{RL}{}
\TrackLangScriptMap{Mahj}{314}{Mahajani}{LR}{}
\TrackLangScriptMap{Mand}{140}{Mandaic, Mandaean}{RL}{}
\TrackLangScriptMap{Mani}{139}{Manichaean}{RL}{}
\TrackLangScriptMap{Marc}{332}{Marchen}{LR}{}
\TrackLangScriptMap{Maya}{090}{Mayan hieroglyphs}{varies}{}
\TrackLangScriptMap{Mend}{438}{Mende Kikakui}{RL}{}
\TrackLangScriptMap{Merc}{101}{Meroitic Cursive}{RL}{}
\TrackLangScriptMap{Mero}{100}{Meroitic Hieroglyphs}{RL}{}
\TrackLangScriptMap{Mlym}{347}{Malayalam}{LR}{}
\TrackLangScriptMap{Modi}{324}{Modi}{LR}{}
\TrackLangScriptMap{Mong}{145}{Mongolian}{TB}{}
\TrackLangScriptMap{Moon}{218}{Moon (Moon code, Moon script, Moon{}
type)}{varies}{}
\TrackLangScriptMap{Mroo}{199}{Mro, Mru}{LR}{}
\TrackLangScriptMap{Mtei}{337}{Meitei Mayek (Meithei, Meetei)}{LR}{}
\TrackLangScriptMap{Mult}{323}{Multani}{LR}{}
\TrackLangScriptMap{Mymr}{350}{Myanmar (Burmese)}{LR}{}
\TrackLangScriptMap{Narb}{106}{Old North Arabian (Ancient North{}
Arabian)}{RL}{}
\TrackLangScriptMap{Nbat}{159}{Nabataean}{RL}{}
\TrackLangScriptMap{Newa}{333}{Newa, Newar, Newari}{LR}{}
\TrackLangScriptMap{Nkgb}{420}{Nakhi Geba}{LR}{}
\TrackLangScriptMap{Nkoo}{165}{N'Ko}{RL}{}
\TrackLangScriptMap{Nshu}{499}{Nushu}{LR}{}
\TrackLangScriptMap{Ogam}{212}{Ogham}{varies}{}
\TrackLangScriptMap{Olck}{261}{Ol Chiki}{LR}{}
\TrackLangScriptMap{Orkh}{175}{Old Turkic, Orkhon Runic}{RL}{}
\TrackLangScriptMap{Orya}{327}{Oriya}{LR}{}
\TrackLangScriptMap{Osge}{219}{Osage}{LR}{}
\TrackLangScriptMap{Osma}{260}{Osmanya}{LR}{}
\TrackLangScriptMap{Palm}{126}{Palmyrene}{RL}{}
\TrackLangScriptMap{Pauc}{263}{Pau Cin Hau}{LR}{}
\TrackLangScriptMap{Perm}{227}{Old Permic}{LR}{}
\TrackLangScriptMap{Phag}{331}{Phags-pa}{TB}{}
\TrackLangScriptMap{Phli}{131}{Inscriptional Pahlavi}{RL}{}
\TrackLangScriptMap{Phlp}{132}{Psalter Pahlavi}{RL}{}
\TrackLangScriptMap{Phlv}{133}{Book Pahlavi}{RL}{}
```

```
\TrackLangScriptMap{Phnx}{115}{Phoenician}{RL}{}
\TrackLangScriptMap{Piqd}{293}{Klingon (KLI plqaD)}{LR}{}
\TrackLangScriptMap{Plrd}{282}{Miao (Pollard)}{LR}{}
\TrackLangScriptMap{Prti}{130}{Inscriptional Parthian}{RL}{}
\TrackLangScriptMap{Qaaa}{900}{Reserved for private use{}
(start)}{varies}{}
\TrackLangScriptMap{Qaai}{908}{Private use}{varies}{}
\TrackLangScriptMap{Qabx}{949}{Reserved for private use{}
(end)}{varies}{}
\TrackLangScriptMap{Rjng}{363}{Rejang (Redjang, Kaganga)}{LR}{}
\TrackLangScriptMap{Roro}{620}{Rongorongo}{varies}{}
\TrackLangScriptMap{Runr}{211}{Runic}{LR}{}
\TrackLangScriptMap{Samr}{123}{Samaritan}{RL}{}
\TrackLangScriptMap{Sara}{292}{Sarati}{varies}{}
\TrackLangScriptMap{Sarb}{105}{Old South Arabian}{RL}{}
\TrackLangScriptMap{Saur}{344}{Saurashtra}{LR}{}
\TrackLangScriptMap{Sgnw}{095}{SignWriting}{TB}{}
\TrackLangScriptMap{Shaw}{281}{Shavian (Shaw)}{LR}{}
\TrackLangScriptMap{Shrd}{319}{Sharada}{LR}{}
\TrackLangScriptMap{Sidd}{302}{Siddham}{LR}{}
\TrackLangScriptMap{Sind}{318}{Khudawadi, Sindhi}{LR}{}
\TrackLangScriptMap{Sinh}{348}{Sinhala}{LR}{}
\TrackLangScriptMap{Sora}{398}{Sora Sompeng}{LR}{}
\TrackLangScriptMap{Sund}{362}{Sundanese}{LR}{}
\TrackLangScriptMap{Sylo}{316}{Syloti Nagri}{LR}{}
\TrackLangScriptMap{Syrc}{135}{Syriac}{RL}{}
\TrackLangScriptMap{Syre}{138}{Syriac (Estrangelo variant)}{RL}{}
\TrackLangScriptMap{Syrj}{137}{Syriac (Western variant)}{RL}{}
\TrackLangScriptMap{Syrn}{136}{Syriac (Eastern variant)}{RL}{}
\TrackLangScriptMap{Tagb}{373}{Tagbanwa}{LR}{}
\TrackLangScriptMap{Takr}{321}{Takri}{LR}{}
\TrackLangScriptMap{Tale}{353}{Tai Le}{LR}{}
\TrackLangScriptMap{Talu}{354}{New Tai Lue}{LR}{}
\TrackLangScriptMap{Taml}{346}{Tamil}{LR}{}
\TrackLangScriptMap{Taml}{346}{Tamil}{LR}{}
\TrackLangScriptMap{Tang}{520}{Tangut}{LR}{}
\TrackLangScriptMap{Tavt}{359}{Tai Viet}{LR}{}
\TrackLangScriptMap{Telu}{340}{Telugu}{LR}{}
\TrackLangScriptMap{Teng}{290}{Tengwar}{LR}{}
\TrackLangScriptMap{Tfng}{120}{Tifinagh (Berber)}{LR}{}
\TrackLangScriptMap{Tglg}{370}{Tagalog (Baybayin, Alibata)}{LR}{}
\TrackLangScriptMap{Thaa}{170}{Thaana}{RL}{}
\TrackLangScriptMap{Thai}{352}{Thai}{LR}{}
\TrackLangScriptMap{Tibt}{330}{Tibetan}{LR}{}
\TrackLangScriptMap{Tirh}{326}{Tirhuta}{LR}{}
\TrackLangScriptMap{Ugar}{040}{Ugaritic}{LR}{}
\TrackLangScriptMap{Vaii}{470}{Vai}{LR}{}
\TrackLangScriptMap{Visp}{280}{Visible Speech}{LR}{}
\TrackLangScriptMap{Wara}{262}{Warang Citi (Varang Kshiti)}{LR}{}
\TrackLangScriptMap{Wole}{480}{Woleai}{RL}{}
\TrackLangScriptMap{Xpeo}{030}{Old Persian}{LR}{}
\TrackLangScriptMap{Xsux}{020}{Cuneiform, Sumero-Akkadian}{LR}{}
\TrackLangScriptMap{Yiii}{460}{Yi}{LR}{}
\TrackLangScriptMap{Zinh}{994}{Inherited script}{inherited}{}
\TrackLangScriptMap{Zmth}{995}{Mathematical notation}{LR}{}
\TrackLangScriptMap{Zsym}{996}{Symbols}{varies}{}
\TrackLangScriptMap{Zsye}{993}{Symbols (emoji variant)}{varies}{}
\TrackLangScriptMap{Zxxx}{997}{Unwritten documents}{varies}{}
\TrackLangScriptMap{Zyyy}{998}{Undetermined script}{varies}{}
\TrackLangScriptMap{Zzzz}{999}{Uncoded script}{varies}{}
```

Restore category code of `@`.

`\@tracklang@scripts@restore@at`

Main Index

Code Index

Symbols

A

B

C

Change History