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 §6). 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 multi-lingual 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. In particular, the emphasis is on using ISO language and region codes rather than hard-coding the various language labels used by different language packages.

# List of Tables

1.1. Predefined ISO Language-Region Dialects ........................................ 4  
1.2. Predefined Root Languages ...................................................... 5  
1.3. Predefined Non-ISO Dialects .................................................... 7  

A.1. Region Mappings .............................................................. 84  
A.2. Script Mappings ............................................................... 88  


Part I.

User Guide
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 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 and german (since it's a simple test to determine if they have been loaded) and recent versions of polyglossia (which conveniently provides \xpg@bcp@loaded) 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 find any way of detecting a list of languages loaded through babel's new \b babel provide command. As far as I can tell, the only stored list is in \bb l@loaded which only contains the languages loaded through package options.

If the ngerman package has been loaded, tracklang effectively does:

\TrackPredefinedDialect{ngerman}

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

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 & 1.3. These may be passed in the document class options or used in \TrackPredefinedDialect, as illustrated above.

§2 provides brief examples of use for those who want a general overview before reading the more detailed sections. §3 describes generic commands for identifying the document languages. §5 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. §6 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. (ISO tag or dialect label may be used as a package option or with `\TrackPredefinedDialect`)

<table>
<thead>
<tr>
<th>ISO Tag</th>
<th>Dialect Label</th>
<th>ISO Tag</th>
<th>Dialect Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>cy-GB</td>
<td>GBwelsh</td>
<td>de-AT</td>
<td>austrian</td>
</tr>
<tr>
<td>de-AT-1996</td>
<td>naustrian</td>
<td>de-BE</td>
<td>belgiangerman</td>
</tr>
<tr>
<td>de-CH</td>
<td>swissgerman</td>
<td>de-CH-1996</td>
<td>nswissgerman</td>
</tr>
<tr>
<td>de-DE</td>
<td>germanDE</td>
<td>de-DE-1996</td>
<td>ngermanDE</td>
</tr>
<tr>
<td>en-AU</td>
<td>australian</td>
<td>en-CA</td>
<td>canadian</td>
</tr>
<tr>
<td>en-GB</td>
<td>british</td>
<td>en-GG</td>
<td>guernseyenglish</td>
</tr>
<tr>
<td>en-IE</td>
<td>IEenglish</td>
<td>en-IM</td>
<td>isleofmanenglish</td>
</tr>
<tr>
<td>en-JE</td>
<td>jerseyenglish</td>
<td>en-MT</td>
<td>maltaenglish</td>
</tr>
<tr>
<td>en-NZ</td>
<td>newzealand</td>
<td>en-US</td>
<td>american</td>
</tr>
<tr>
<td>es-AR</td>
<td>argentinespanish</td>
<td>es-BO</td>
<td>bolivianspanish</td>
</tr>
<tr>
<td>es-CL</td>
<td>chilianspanish</td>
<td>es-CO</td>
<td>columbianspanish</td>
</tr>
<tr>
<td>es-CR</td>
<td>costaricanspanish</td>
<td>es-CU</td>
<td>cubanspanish</td>
</tr>
<tr>
<td>es-DO</td>
<td>dominicanspanish</td>
<td>es-EC</td>
<td>ecudorianspanish</td>
</tr>
<tr>
<td>es-ES</td>
<td>spainspanish</td>
<td>es-GT</td>
<td>guatemalanspanish</td>
</tr>
<tr>
<td>es-HN</td>
<td>honduranspanish</td>
<td>es-MX</td>
<td>mexicanspanish</td>
</tr>
<tr>
<td>es-NI</td>
<td>nicaraguanspanish</td>
<td>es-PA</td>
<td>panamaspanish</td>
</tr>
</tbody>
</table>
| es-PE   | peruvianspanish | es-PR | puertoricospansh |}
| es-PS   | 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 tag obtained with `\GetTrackedLanguageTag{(dialect)}` is shown in parentheses.

<table>
<thead>
<tr>
<th>Language</th>
<th>Code</th>
<th>Tag</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>abkhaz (ab)</td>
<td>af</td>
<td>afrikaans</td>
<td>afrikaans (af)</td>
</tr>
<tr>
<td>akan (ak)</td>
<td>am</td>
<td>amharic</td>
<td>amharic (am-ET)</td>
</tr>
<tr>
<td>anglosaxon (ang)</td>
<td>ar</td>
<td>arabic</td>
<td>arabic (ar)</td>
</tr>
<tr>
<td>aragonese (an-ES)</td>
<td>az</td>
<td>azerbaijani</td>
<td>azerbaijani (az)</td>
</tr>
<tr>
<td>asturian (ast)</td>
<td>gu</td>
<td>avestan</td>
<td>avestan (ae)</td>
</tr>
<tr>
<td>aymara (ay)</td>
<td>km</td>
<td>bahasa</td>
<td>bahasa (id-IN)</td>
</tr>
<tr>
<td>bahasam (ms-MY)</td>
<td>bm</td>
<td>bashkir</td>
<td>bashkir (ba)</td>
</tr>
<tr>
<td>basque (eu)</td>
<td>be</td>
<td>bengali</td>
<td>bengali (bn)</td>
</tr>
<tr>
<td>berber (ber)</td>
<td>bh</td>
<td>bislami</td>
<td>bislami (bi-VU)</td>
</tr>
<tr>
<td>bokmal (nb-NL)</td>
<td>bs</td>
<td>breton</td>
<td>breton (br-FR)</td>
</tr>
<tr>
<td>bulgarian (bg)</td>
<td>bn</td>
<td>catalan</td>
<td>catalan (ca)</td>
</tr>
<tr>
<td>chamorro (ch)</td>
<td>ce</td>
<td>chichewa</td>
<td>chichewa (ny)</td>
</tr>
<tr>
<td>chinese (zh)</td>
<td>cu</td>
<td>chuvashtani</td>
<td>chuvashtani (cv-RU)</td>
</tr>
<tr>
<td>coptic (cop)</td>
<td>kw</td>
<td>corsican</td>
<td>corsican (co)</td>
</tr>
<tr>
<td>cree (cr)</td>
<td>hr</td>
<td>czech</td>
<td>czech (cs)</td>
</tr>
<tr>
<td>danish (da)</td>
<td>dv</td>
<td>dutch</td>
<td>dutch (nl)</td>
</tr>
<tr>
<td>dzongkha (dz-BT)</td>
<td>en</td>
<td>easternpunjabi</td>
<td>easternpunjabi (pa-IN)</td>
</tr>
<tr>
<td>esperanto (eo)</td>
<td>et</td>
<td>eve</td>
<td>eve (ee)</td>
</tr>
<tr>
<td>faroese (fo)</td>
<td>fa</td>
<td>fijian</td>
<td>fijian (fj-FJ)</td>
</tr>
<tr>
<td>finnish (fi)</td>
<td>fr</td>
<td>friulian</td>
<td>friulian (fur-IT)</td>
</tr>
<tr>
<td>fula (ff)</td>
<td>gl</td>
<td>ganda</td>
<td>ganda (lg-UG)</td>
</tr>
<tr>
<td>georgian (ka)</td>
<td>de</td>
<td>greek</td>
<td>greek (el)</td>
</tr>
<tr>
<td>guarani (gn)</td>
<td>gu</td>
<td>hatian</td>
<td>hatian (ht-HT)</td>
</tr>
<tr>
<td>hausa (ha)</td>
<td>he</td>
<td>herero</td>
<td>herero (hz)</td>
</tr>
<tr>
<td>hindi (hi)</td>
<td>ho</td>
<td>icelandic</td>
<td>icelandic (is-IS)</td>
</tr>
<tr>
<td>ido (io)</td>
<td>ig</td>
<td>interlingua</td>
<td>interlingua (ia)</td>
</tr>
<tr>
<td>interlingue (ie)</td>
<td>iu</td>
<td>inupiaq</td>
<td>inupiaq (ik)</td>
</tr>
<tr>
<td>irish (ga)</td>
<td>it</td>
<td>japanese</td>
<td>japanese (ja)</td>
</tr>
<tr>
<td>javanese (jv)</td>
<td>kl</td>
<td>kannada</td>
<td>kannada (kn-IN)</td>
</tr>
<tr>
<td>kanuri (kr)</td>
<td>ks</td>
<td>kazakh</td>
<td>kazakh (kk)</td>
</tr>
<tr>
<td>khmer (km)</td>
<td>kl</td>
<td>kinyarwanda</td>
<td>kinyarwanda (rw)</td>
</tr>
<tr>
<td>kirundi (rn)</td>
<td>kv</td>
<td>kongo</td>
<td>kongo (kg)</td>
</tr>
<tr>
<td>korean (ko)</td>
<td>ku</td>
<td>kwanyama</td>
<td>kwanyama (kj)</td>
</tr>
<tr>
<td>kyrgyz (ky)</td>
<td>lo</td>
<td>latin</td>
<td>latin (la)</td>
</tr>
<tr>
<td>latvian (lv)</td>
<td>li</td>
<td>lingala</td>
<td>lingala (ln)</td>
</tr>
<tr>
<td>lithuanian (lt)</td>
<td>lv</td>
<td>lubakatanga</td>
<td>lubakatanga (lu-CD)</td>
</tr>
<tr>
<td>luxembourgish (lb)</td>
<td>mk</td>
<td>magyar</td>
<td>magyar (hu)</td>
</tr>
<tr>
<td>malagasy (mg)</td>
<td>mn</td>
<td>maltese</td>
<td>maltese (mt)</td>
</tr>
<tr>
<td>manx (gv-IM)</td>
<td>mi</td>
<td>marathi</td>
<td>marathi (mr-IN)</td>
</tr>
<tr>
<td>marshallse (mh-MH)</td>
<td>mn</td>
<td>nauruan</td>
<td>nauruan (na-NR)</td>
</tr>
<tr>
<td>navajo (nv-US)</td>
<td>ng</td>
<td>nepali</td>
<td>nepali (ne)</td>
</tr>
</tbody>
</table>
Table 1.2.: Predefined Root Languages (Continued)

<table>
<thead>
<tr>
<th>Language</th>
<th>Code</th>
<th>Language</th>
<th>Code</th>
<th>Language</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>nko (nqo)</td>
<td></td>
<td>norsk (no)</td>
<td></td>
<td>northernndebele (nd)</td>
<td></td>
</tr>
<tr>
<td>northernsotho (nso)</td>
<td></td>
<td>nuosu† (ii-CN)</td>
<td></td>
<td>nynorsk† (nn-NO)</td>
<td></td>
</tr>
<tr>
<td>occitan (oc)</td>
<td></td>
<td>ojibwe (oj)</td>
<td></td>
<td>oriya (or)</td>
<td></td>
</tr>
<tr>
<td>oromo (om)</td>
<td></td>
<td>ossetian (os)</td>
<td></td>
<td>pali (pi)</td>
<td></td>
</tr>
<tr>
<td>pashto (ps)</td>
<td></td>
<td>piedmontese† (pms-IT)</td>
<td></td>
<td>polish (p1)</td>
<td></td>
</tr>
<tr>
<td>portuges (pt)</td>
<td></td>
<td>quechua (qu)</td>
<td></td>
<td>romanian (ro)</td>
<td></td>
</tr>
<tr>
<td>romansh† (rm-CH)</td>
<td></td>
<td>russian (ru)</td>
<td></td>
<td>samin (se)</td>
<td></td>
</tr>
<tr>
<td>samoan (sm)</td>
<td></td>
<td>sango (sg)</td>
<td></td>
<td>sanskrit (sa)</td>
<td></td>
</tr>
<tr>
<td>sardinian† (sc-IT)</td>
<td></td>
<td>scottish (gd)</td>
<td></td>
<td>serbian (sr)</td>
<td></td>
</tr>
<tr>
<td>shona (sn)</td>
<td></td>
<td>sindhi (sd)</td>
<td></td>
<td>sinhalese† (si-LK)</td>
<td></td>
</tr>
<tr>
<td>slovak (sk)</td>
<td></td>
<td>slovene (sl)</td>
<td></td>
<td>somali (so)</td>
<td></td>
</tr>
<tr>
<td>southernndebele† (nr-ZA)</td>
<td></td>
<td>southernsotho (st)</td>
<td></td>
<td>spanish (es)</td>
<td></td>
</tr>
<tr>
<td>sudanese (su)</td>
<td></td>
<td>swahili (sw)</td>
<td></td>
<td>swati (ss)</td>
<td></td>
</tr>
<tr>
<td>swedish (sv)</td>
<td></td>
<td>syriac (syr)</td>
<td></td>
<td>tagalog† (tl-PH)</td>
<td></td>
</tr>
<tr>
<td>tahitian† (ty-PF)</td>
<td></td>
<td>tai (tai)</td>
<td></td>
<td>tajik (tg)</td>
<td></td>
</tr>
<tr>
<td>tamil (ta)</td>
<td></td>
<td>tatar (tt)</td>
<td></td>
<td>telugu† (te-IN)</td>
<td></td>
</tr>
<tr>
<td>thai† (th-TH)</td>
<td></td>
<td>tibetan (bo)</td>
<td></td>
<td>tigrinya (ti)</td>
<td></td>
</tr>
<tr>
<td>tonga† (to-TO)</td>
<td></td>
<td>tsonga (ts)</td>
<td></td>
<td>tswana (tn)</td>
<td></td>
</tr>
<tr>
<td>turkish (tr)</td>
<td></td>
<td>turkmen (tk)</td>
<td></td>
<td>twi† (tw-GH)</td>
<td></td>
</tr>
<tr>
<td>ukrainian† (uk-UA)</td>
<td></td>
<td>undetermined (und)</td>
<td></td>
<td>urdu (ur)</td>
<td></td>
</tr>
<tr>
<td>usorbian† (hsb-DE)</td>
<td></td>
<td>uyghur† (ug-CN)</td>
<td></td>
<td>uzbek (uz)</td>
<td></td>
</tr>
<tr>
<td>venda† (ve-ZA)</td>
<td></td>
<td>vietnamese (vi)</td>
<td></td>
<td>volapuk (vo)</td>
<td></td>
</tr>
<tr>
<td>walloon (wa)</td>
<td></td>
<td>welsh (cy)</td>
<td></td>
<td>westernfrisian† (fy-NL)</td>
<td></td>
</tr>
<tr>
<td>wolof (wo)</td>
<td></td>
<td>xhosa (xh)</td>
<td></td>
<td>yiddish (yi)</td>
<td></td>
</tr>
<tr>
<td>yoruba (yo)</td>
<td></td>
<td>zhuang† (za-CN)</td>
<td></td>
<td>zulu (zu)</td>
<td></td>
</tr>
</tbody>
</table>
Table 1.3.: Predefined Non-ISO Dialects. (†Has an associated territory.) The corresponding language tag obtained with $\text{GetTrackedLanguageTag}(\langle dialect \rangle)$ is shown in parentheses. If the dialect has a corresponding mapping for the closest matching non-root language $\text{captions}(\langle dialect \rangle)$ or $\text{date}(\langle dialect \rangle)$, this is also included after the tag following a slash.

<table>
<thead>
<tr>
<th>Dialect</th>
<th>Language Tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>acadian (fr)</td>
<td>american (en-US)</td>
</tr>
<tr>
<td>argentinespanish (es-AR)</td>
<td>american (en-US)</td>
</tr>
<tr>
<td>austrian (de-AT)</td>
<td>bahasa (id-IN)</td>
</tr>
<tr>
<td>belgiangerman (de-BE)</td>
<td>belgique (fr-BE)</td>
</tr>
<tr>
<td>bolivianspanish (es-BO)</td>
<td>brazil (pt-BR)</td>
</tr>
<tr>
<td>brazilian (pt-BR)</td>
<td>british (en-GB)</td>
</tr>
<tr>
<td>canadian (en-CA)</td>
<td>canadien (fr-CA)</td>
</tr>
<tr>
<td>chilianspanish (es-CL)</td>
<td>colombianspanish (es-CO)</td>
</tr>
<tr>
<td>costaricanspanish (es-CR)</td>
<td>croatia (hr-HR)</td>
</tr>
<tr>
<td>cubanspanish (es-CU)</td>
<td>cymraeg (cy)</td>
</tr>
<tr>
<td>deutsch (de)</td>
<td>dominicanspanish (es-DO)</td>
</tr>
<tr>
<td>ecudorianspanish (es-EC)</td>
<td>elsalvadorspanish (es-SV)</td>
</tr>
<tr>
<td>flemish (nl-BE)</td>
<td>francais (fr)</td>
</tr>
<tr>
<td>france (fr)</td>
<td>frenchb (fr)</td>
</tr>
<tr>
<td>friulano (fur-IT)</td>
<td>friulian (fur-IT)</td>
</tr>
<tr>
<td>furlan (fur-IT)</td>
<td>gaeilge (ga)</td>
</tr>
<tr>
<td>gaelic (gd)</td>
<td>galicien (gl)</td>
</tr>
<tr>
<td>GBirish (ga-GB)</td>
<td>GBscottish (gd-GB)</td>
</tr>
<tr>
<td>G BWelsh (cy-GB)</td>
<td>germanb (de)</td>
</tr>
<tr>
<td>germanDE (de-DE)</td>
<td>guatemalanspanish (es-GT)</td>
</tr>
<tr>
<td>guernseyenglish (en-GG / british)</td>
<td>guernseyfrench (fr-GG)</td>
</tr>
<tr>
<td>honduranspanish (es-HN)</td>
<td>hungarian (hu-HU)</td>
</tr>
<tr>
<td>IE English (en-IE / british)</td>
<td>IEIrish (ga-IE)</td>
</tr>
<tr>
<td>indon (id-IN)</td>
<td>indonesian (id-IN)</td>
</tr>
<tr>
<td>isleofmanenglish (en-IM / british)</td>
<td>istriacounty (it-HR)</td>
</tr>
<tr>
<td>istriacounty (it-HR)</td>
<td>italy (it-IT)</td>
</tr>
<tr>
<td>jerseyenglish (en-JE / british)</td>
<td>jerseyfrench (fr-JE)</td>
</tr>
<tr>
<td>kurmanji (ku)</td>
<td>latein (la)</td>
</tr>
<tr>
<td>lowersorbian (dsb-DE)</td>
<td>malay (ms-MY)</td>
</tr>
<tr>
<td>maltaenglish (en-MT / british)</td>
<td>maltamaltese (mt-MT)</td>
</tr>
<tr>
<td>mexicanspanish (es-MX)</td>
<td>meyalu (ms-MY)</td>
</tr>
<tr>
<td>naustrian (de-AT-1996)</td>
<td>nbelgiangerman (de-BE-1996 / ngerman)</td>
</tr>
<tr>
<td>netherlands (nl-NL)</td>
<td>newzealand (en-NZ)</td>
</tr>
<tr>
<td>ngerman (de-1996)</td>
<td>ngerman (de-1996 / ngerman)</td>
</tr>
<tr>
<td>ngermanDE (de-DE-1996 / ngerman)</td>
<td>nicaraguanspanish (es-NI)</td>
</tr>
<tr>
<td>nil (und)</td>
<td>norwegian (no-NO)</td>
</tr>
<tr>
<td>nswissgerman (de-CH-1996 / ngerman)</td>
<td>panamaspanish (es-PA)</td>
</tr>
</tbody>
</table>
1. Introduction

Table 1.3.: Predefined Non-ISO Dialects (Continued)

<table>
<thead>
<tr>
<th>Dialect</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>paraguayspanish† (es-PY)</td>
<td></td>
</tr>
<tr>
<td>peruvianspanish† (es-PE)</td>
<td></td>
</tr>
<tr>
<td>polutoniko (el)</td>
<td></td>
</tr>
<tr>
<td>portugal† (pt-PT)</td>
<td></td>
</tr>
<tr>
<td>puertoricospansh† (es-PR)</td>
<td></td>
</tr>
<tr>
<td>romansch (rm-CH)</td>
<td></td>
</tr>
<tr>
<td>russianb (ru)</td>
<td></td>
</tr>
<tr>
<td>serbianc (sr-Cyrl)</td>
<td></td>
</tr>
<tr>
<td>sloveneistria italian† (it-SI)</td>
<td></td>
</tr>
<tr>
<td>slovenia† (sl-SI / slovenian)</td>
<td></td>
</tr>
<tr>
<td>spainspanish† (es-ES)</td>
<td></td>
</tr>
<tr>
<td>swissgerman† (de-CH)</td>
<td></td>
</tr>
<tr>
<td>swissromansh† (rm-CH)</td>
<td></td>
</tr>
<tr>
<td>ukraine† (uk-UA)</td>
<td></td>
</tr>
<tr>
<td>uppersorbian† (hsb-DE)</td>
<td></td>
</tr>
<tr>
<td>USenglish† (en-US)</td>
<td></td>
</tr>
<tr>
<td>valencien (ca)</td>
<td></td>
</tr>
<tr>
<td>venezuelanspanish† (es-VE)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dialect</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>persian (fa)</td>
<td></td>
</tr>
<tr>
<td>piemonteis† (pms-IT)</td>
<td></td>
</tr>
<tr>
<td>polutonikogreek (el)</td>
<td></td>
</tr>
<tr>
<td>portuguese (pt)</td>
<td></td>
</tr>
<tr>
<td>romanche (rm-CH)</td>
<td></td>
</tr>
<tr>
<td>rumantsch (rm-CH)</td>
<td></td>
</tr>
<tr>
<td>sanmarino† (it-SM)</td>
<td></td>
</tr>
<tr>
<td>serbianl (sr-Latn)</td>
<td></td>
</tr>
<tr>
<td>sloveneistaslovenian† (sl-SI / slovenian)</td>
<td></td>
</tr>
<tr>
<td>ukraineb† (uk-UA)</td>
<td></td>
</tr>
<tr>
<td>uruguayspanish† (es-UY)</td>
<td></td>
</tr>
<tr>
<td>valencian (ca)</td>
<td></td>
</tr>
<tr>
<td>vatican† (it-VA)</td>
<td></td>
</tr>
</tbody>
</table>
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

Document level use can be divided into generic \TeX{} use (§2.1.1) and \LaTeX{}-specific use (§2.1.2).

2.1.1. Generic \TeX{}

This section is for generic \TeX{} use. The `tracklang` files are loaded with `\input`. See §2.1.2 for \LaTeX{} use.

A Unix-like user wants the locale information picked up from the locale environment variable (the `.tex` extension may be omitted):

\begin{verbatim}
\input tracklang.tex % v1.3
\TrackLangFromEnv
% load packages that use tracklang for localisation
\end{verbatim}

A Windows user wants the locale information picked up from the operating system (again the `.tex` extension may be omitted):

\begin{verbatim}
\input texosquery.tex
\input tracklang.tex % v1.3
\TrackLangFromEnv
% load packages that use tracklang for localisation
\end{verbatim}

2. Summary of Use

Or (texasquery v1.2)

\input texosquery.tex % v1.2
\input tracklang.tex % v1.3
\TeXOSQueryLangTag{\langtag}
\TrackLanguageTag{\langtag}
% load packages that use tracklang for localisation

A Unix-like user who may or may not have texosquery setup to run in the shell escape:

\input texosquery.tex
\input tracklang.tex % v1.3
\ifx\TeXOSQueryLangTag\undefined
  \TrackLangFromEnv
\else
  \TeXOSQueryLangTag{\langtag}
  \TrackLanguageTag{\langtag}
\fi
% load packages that use tracklang for localisation

A user is writing in Italy in Armenian with a Latin script (Latn) and the arevela variant:

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

A user is writing in English in the UK:

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

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

\SetCurrentTrackedDialect{\languagename}
Dialect: \CurrentTrackedDialect.
2. Summary of Use

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.tex

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

Name: \TrackLangScriptAlphaToName{\CurrentTrackedDialectScript}.
Numeric: \TrackLangScriptAlphaToNumeric{\CurrentTrackedDialectScript}.
Direction: \TrackLangScriptAlphaToDir{\CurrentTrackedDialectScript}.

Test for a specific script (in this case Latn):

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

2.1.2. \LaTeX

This section is for \LaTeX use. See §2.1.1 for generic \TeX use.

With newer versions of polyglossia, where \xpg@bcp@loaded is defined, you just need to make sure the languages are set before tracklang is loaded:
2. Summary of Use

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

For older versions of polyglossia where the regional information is required, use recognised class options:

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

For babel users where the supplied babel dialect label is sufficient, and is passed either through the document class or package options, there’s no need to do anything special:

\documentclass[british,canadien]{article}
\usepackage[T1]{fontenc}
\usepackage[babel]{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. The first method is to 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

Another method with babel is to use \TrackLanguageTag and map the new dialect label to the nearest matching \captions\(\text{dialect}\):

\documentclass{article}
\usepackage[tracklang]{tracklang}% v1.3
\TrackLanguageTag{en-MT}
\SetTrackedDialectLabelMap{\TrackLangLastTrackedDialect}{UKenglish}
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 \texttt{british} is used instead of \texttt{UKenglish} since the en-MT (maltaenglish) predefined dialect automatically sets up a mapping to \texttt{british}. (The default mappings are shown in Table 1.3 on page 7.)

There’s no support for \texttt{babelprovide}. If you are using \texttt{babelprovide}, you will need to use the class option or \texttt{\TrackLanguageTag} as above.

## 2.2. Locale-Sensitive Packages

Let’s suppose you are developing a package called \texttt{mypackage.sty} or \texttt{mypackage.tex} and you want to find out what languages the document author has requested. (See also: Using tracklang.tex in Packages with Localisation Features.¹)

**Generic \LaTeX use (the \texttt{tex} extension may be omitted):**

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

(\texttt{\LaTeX} use:)

\begin{verbatim}
\input tracklang.tex
\end{verbatim}

\begin{verbatim}
\RequirePackage{tracklang}[2019/11/30]\% at least v1.4
\end{verbatim}

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

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

\begin{verbatim}
\DeclareOption*{\TrackLanguageTag{CurrentOption}}
\end{verbatim}

This means the user can do, say,

\begin{verbatim}
\usepackage[hy-Latn-IT-arevela]{mypackage}
\end{verbatim}

¹\url{dickimaw-books.com/latex/tracklang/otherpkg.shtml}
With at least version 1.4, it’s better to use \TrackIfKnownLanguage:

\DeclareOption*{%
\TrackIfKnownLanguage{\CurrentOption}%%
{\% successful
  \PackageInfo{mypackage}{Tracking language `\CurrentOption'}%}
}%%
{\% failed
  \PackageError{mypackage}%%
  {Unknown language specification `\CurrentOption'}%
  {You need to supply either a known dialect label
   or a valid language tag}%
}%%
}

The rest of the example package in this section uses generic code. If you are using \LaTeX, it’s better to replace \def and \ifx with more appropriate \LaTeX commands.

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{\textup{Foo}}
\def\barname{\textup{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⟩-⟨localeid⟩.ldf. In this example, the ⟨prefix⟩ is mypackage. The ⟨localeid⟩ part 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). As from version 1.4, ⟨localeid⟩ may also include the script or variant. (See the definition of \IfTrackedLanguageFileExists on page 44 for further details.)

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

This sets up appropriate the \captions⟨dialect⟩ hook (if it’s found). For other hooks, such as \date⟨dialect⟩, use \TrackLangAddToHook or \TrackLangRedefHook instead.

With pre-v1.4 versions of tracklang, the script isn’t included in the file search. If it’s needed then either require at least v1.4 or have a base ldf file that tries to load a version for the particular script (which can be accessed with \CurrentTrackedDialectScript). Here’s an example for a language with different writing systems. The resource file for Serbian mypackage-serbian.1df:

```
\TrackLangProvidesResource{serbian}[2016/10/06 v1.0]
\TrackLangRequestResource{serbian-\CurrentTrackedDialectScript}{%
% file not found, do something sensible here
```
2. **Summary of Use**

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

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

\[
\text{\textbackslash TrackLangAddToCaptions}\{\%
\text{\textbackslash def\ fooname\{\ldots\}\% provide appropriate Latin translations}
\text{\textbackslash def\ barname\{\ldots\}\%
\}
\]

The file `mypackage-serbian-Cyrl.ldf` sets up the Cyrillic script (Cyrl):

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

\[
\text{\textbackslash TrackLangAddToCaptions}\{\%
\text{\textbackslash def\ fooname\{\ldots\}\% provide appropriate Cyrillic translations}
\text{\textbackslash def\ barname\{\ldots\}\%
\}
\]

With v1.4+ you just need `mypackage-sr-Latn.ldf` and `mypackage-sr-Cyrl.ldf` for the regionless versions.

### 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 §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. (See also: Integrating tracklang.tex into Language Packages.²)

Generic use:

\[
\text{\textbackslash input\ tracklang}
\]

Alternative \LaTeX{} use:

\[
\text{\textbackslash RequirePackage}\{\text{tracklang}\}[2019/11/30]\% \ v1.4
\]

² [dickimaw-books.com/latex/tracklang/langpkg.shtml](dickimaw-books.com/latex/tracklang/langpkg.shtml)
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:

\begin{verbatim}
\TrackPredefinedDialect{british}
\end{verbatim}

Alternatively

\begin{verbatim}
\TrackLanguageTag{en-GB}
\end{verbatim}

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):

\begin{verbatim}
\TrackLanguageTag{en-MT}
\SetTrackedDialectLabelMap{\TrackLangLastTrackedDialect}{UKenglish}
\def\captionsUKenglish{%
    \def\contentsname{Contents}%
    % ...
}
\end{verbatim}

(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 `\TrackLangAddToHook` and `\TrackLangRedefHook` commands can find your language hooks. You don’t need the map if your dialect label is the same as tracklang’s root language label for that locale. For example:

\begin{verbatim}
\TrackLanguageTag{en-MT}
\def\captionsenglish{%
    \def\contentsname{Contents}%
    % ...
}
\end{verbatim}

When the user switches language through commands like `\selectlanguage` it would be
2. Summary of Use

useful to also use \SetCurrentTrackedDialect{\langle dialect \rangle} to make it easier for the document author or locale-sensitive packages to pick up the current locale. The \langle dialect \rangle 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:

\begin{verbatim}
def\selectlanguage#1{\%
  \SetCurrentTrackedDialect{#1}\
  \% set up hyphenation patterns etc
}
\end{verbatim}

See the example in §2.1 or the example in Integrating tracklang.tex into Language Packages.³

³dickimaw-books.com/latex/tracklang/langpkg.shtml
3. Generic Use

For plain \TeX you can input `tracklang.tex`:

\begin{verbatim}
\input tracklang
\end{verbatim}

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

\begin{verbatim}
\input{tracklang}
\end{verbatim}

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

\begin{verbatim}
\@tracklang@declareoption{⟨dialect⟩}
\end{verbatim}

to

\begin{verbatim}
\DeclareOption{⟨dialect⟩}{\TrackPredefinedDialect{⟨dialect⟩}}
\end{verbatim}

If `\@tracklang@declareoption` isn’t defined when `tracklang.tex` is input, it will be defined to ignore its argument.

This means that all the predefined languages and dialects (tables 1.1, 1.2 & 1.3) automatically become package options, so the `tracklang.sty` 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 can redefine `\@tracklang@declareoption` to use something analogous to `\DeclareOption`, if appropriate. Otherwise, the document languages need to be explicitly identified (using any of the following commands) so that `tracklang` knows about them.

\begin{verbatim}
\TrackPredefinedDialect{⟨dialect label⟩}
\end{verbatim}

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 & 1.3.
3. Generic Use

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{⟨locale⟩}

If ⟨locale⟩ is a recognised dialect, this is equivalent to using \TrackPredefinedDialect, otherwise ⟨locale⟩ needs to be in one of 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 code-set 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 ISO 639-2 (T) and 639-2 (B) 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{⟨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.

If you want to first check that ⟨tag⟩ includes a valid language code, then you can instead use:

```
\TrackIfKnownLanguage{⟨tag⟩}{⟨success code⟩}{⟨fail code⟩}
```

This will only track ⟨tag⟩ (and then do ⟨success code⟩) if ⟨tag⟩ starts with a valid language code (or is a predefined dialect) otherwise it will do ⟨fail code⟩. Both \TrackLanguageTag{} and \TrackIfKnownLanguage{} will check if ⟨tag⟩ is a predefined option. (This saves parsing the tag if it’s recognised.)

For example:

```
\TrackLanguageTag{hy-Latn-IT-arevela}
\TrackIfKnownLanguage{Latn-ME}{success}{fail}.
\TrackIfKnownLanguage{brazilian}{success}{fail}.
```

This will track hy-Latn-IT-arevela and brazilian (pt-BR) but not Latn-ME (because it doesn’t contain a valid language code) even though it’s a valid script and country code. The above is just for illustrative purposes. Typically the language tracking isn’t performed within the document text.

The datetime2 package assumes that any unknown package option is a language identifier. It could simply do:

```
\TrackLanguageTag{\CurrentOption}
```

but users can make mistakes sometimes and this won’t provide any helpful information if they, for example, misspelt a package option or forgot the “⟨key⟩=” part of a ⟨key⟩=⟨value⟩ setting. Instead (as from v1.5.5) datetime2 now does:
This will now give the user some guidance.

If \texttt{tag} contains a sub-language tag, this will be set as the 639-3 code for the \textit{dialect} label. Note that this is different to the root language codes which are set using the language label. For example:

\begin{verbatim}
\TrackLanguageTag{zh-cmn-Hans-CN}
\end{verbatim}

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

\begin{verbatim}
ISO 639-1: \TrackedIsoCodeFromLanguage{639-1}{chinese}.
ISO 639-3: \TrackedIsoCodeFromLanguage{639-3}{zhcmnHansCN}.
\end{verbatim}

Version 1.2 of \texttt{texosquery} provides the command \texttt{\TeXOSQueryLangTag}, which may be used to fetch the operating system’s regional information as a language tag. These commands can be used as follows:

\begin{verbatim}
\input tracklang % v1.3
\input texosquery % v1.2
\TeXOSQueryLangTag{\langtag}
\TrackLanguageTag{\langtag}
\end{verbatim}

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

Some of the predefined root language options listed in Table 1.2 on page 5 have an associated region (denoted by \texttt{†}). If \texttt{\TrackLocale} is used with just the language ISO code, no region is tracked for that language. For example

\begin{verbatim}
\TrackLocale{manx}
\end{verbatim}

will track the IM (Isle of Man) ISO 3166-1 code but

\begin{verbatim}
\TrackLocale{gv}
\end{verbatim}
3. Generic Use

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

If the shell escape has been enabled or \directlua is available, this will try to get the language information from the system environment variables \texttt{LC\_ALL} or \texttt{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 \texttt{C} or \texttt{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 \texttt{texosquery}.

If the operating system locale can’t be obtained from environment variables, then tracklang will use \TeXOSShutdown 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:

\begin{verbatim}
\input texosquery
\input tracklang
\TrackLangFromEnv
\end{verbatim}

Document build:

etex \texttt{--shell-escape} \texttt{(filename)}

\TeX\ example:

\begin{verbatim}
\usepackage{texosquery}
\usepackage{tracklang}
\TrackLangFromEnv
\end{verbatim}

Document build:
3. Generic Use

```latex
pdflatex --shell-escape ⟨filename⟩
```

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

```latex
\TrackLangShowWarningsfalse
```
or switched back on again using

```latex
\TrackLangShowWarningstrue
```

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

```latex
\TrackPredefinedDialect{british}
```
I can use

```latex
\TrackLangFromEnv
```

With \LaTeX\ documents I can do

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

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

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

In my case, with the `LANG` environment variable set to `en_GB.utf8` and the 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`. 
3. Generic Use

The \TrackLangFromEnv command also incidentally sets \TrackLangEnv to the value of the environment variable or empty if the query was unsuccessful (for example, the shell escape is unavailable).

If the command:

\begin{verbatim}
\TrackLangEnv
\end{verbatim} user defined

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:

\begin{verbatim}
\def\TrackLangEnv{en-GB}
\TrackLangFromEnv
\end{verbatim}

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

\begin{verbatim}
\TrackLocale{en-GB}
\end{verbatim}

(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):

\begin{verbatim}
tex "\def\TrackLangEnv{\$LANG}\input myDoc"
\end{verbatim}

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.) If the information is unavailable, the relevant commands will be set to empty.
3. Generic Use

The language code is stored in:

\TrackLangEnvLang

The territory (if present) is stored in:

\TrackLangEnvTerritory

The code-set (if present) is stored in:

\TrackLangEnvCodeSet

The modifier (if present) is stored in:

\TrackLangEnvModifier

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

\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{(env-name)}

For example, to also query LC_MONETARY:

\TrackLangQueryOtherEnv{LC_MONETARY}
3. Generic Use

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

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:

3. Generic Use

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:


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:
3. Generic Use

4. Supplementary Packages

In addition to the main `tracklang.tex` file and `tracklang.sty` \LaTeX{} wrapper, the `tracklang` package also provides supplementary files for region and script mappings.

`tracklang-region-codes.tex`

This file is only loaded if a mapping is required between numeric and alphabetic region codes. If \TrackLanguageTag{} encounters a numeric region code, it will automatically input `tracklang-region-codes.tex`, if it hasn’t already been input. This file provides the following commands.

\begin{itemize}
  \item \TrackLangAlphaIIIToNumericRegion\{⟨alpha-2 code⟩\}
    Expands to the numeric code corresponding to the given alpha-2 code or empty if no mapping has been supplied.
  \item \TrackLangNumericToAlphaIIRegion\{⟨numeric code⟩\}
    Expands to the alpha-2 code corresponding to the given numeric code or empty if no mapping has been supplied.
  \item \TrackLangIfKnownAlphaIIRegion\{⟨alpha-2 code⟩\}{⟨true⟩}{⟨false⟩}
    Expands to ⟨true⟩ if there’s an alpha-2 to numeric region code mapping, otherwise expands to ⟨false⟩.
  \item \TrackLangIfKnownNumericRegion\{⟨numeric code⟩\}{⟨true⟩}{⟨false⟩}
    Expands to ⟨true⟩ if there’s a numeric to alpha-2 region code mapping, otherwise expands to ⟨false⟩.
  \item \TrackLangAlphaIIIToNumericRegion\{⟨alpha-3 code⟩\}
\end{itemize}
4. Supplementary Packages

Expands to the numeric code corresponding to the given alpha-3 code or empty if no mapping has been supplied.

\TrackLangNumericToAlphaIIIRegion{⟨numeric code⟩}

Expands to the alpha-3 code corresponding to the given numeric code or empty if no mapping has been supplied.

\TrackLangIfKnownAlphaIIIRegion{⟨alpha-3 code⟩}{⟨true⟩}{⟨false⟩}

Expands to ⟨true⟩ if there’s an alpha-3 to numeric region code mapping, otherwise expands to ⟨false⟩.

Mappings are established with:

\TrackLangRegionMap{⟨numeric⟩}{⟨alpha-2⟩}{⟨alpha-3⟩}

Predefined mappings are listed in Table A.1 on page 84.

When tracklang-region-codes.tex is input, it can load additional files that provide supplementary mappings.

\TrackLangAddExtraRegionFile{⟨file⟩}

This command adds the supplied ⟨file⟩ to the list of extra region code files that should be input by tracklang-region-codes.tex, unless tracklang-region-codes.tex has already been input, in which case ⟨file⟩ will be input straight away.

tracklang-scripts.tex

The tracklang-scripts package provides information about ISO 15924 scripts. The file isn’t automatically loaded. If you want to use any of the commands provided in it you need to input it.

Plain \TeX:

\input tracklang-scripts

There’s a simple wrapper package tracklang-scripts.sty for \LaTeX users:

\usepackage{tracklang-scripts}
4. Supplementary Packages

\TrackLangScriptMap{⟨letter code⟩}{⟨numeric code⟩}{⟨script name⟩}{⟨direction⟩}{⟨parent script⟩}

Defines a mapping. The first argument is the four letter alpha code, such as Latn or Cyr1. The second argument is the numeric code. The third argument is the script’s name, for example “Imperial Aramaic”. The fourth argument is the direction, which may be one of: 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 unsupported).

This command defines:

\TrackLangScript{⟨Code⟩}

which expands to ⟨Code⟩ for use with \IfTrackedDialectIsScriptCs.

See Table A.2 on page 88 for a summary of all the mappings that are provided by the file tracklang-scripts.tex.

\TrackLangScriptAlphaToNumeric{⟨alpha code⟩}

Expands to the numeric code corresponding to the given alpha code or empty if no mapping.

\TrackLangScriptIfKnownAlpha{⟨alpha code⟩}{⟨true⟩}{⟨false⟩}

Expands to ⟨true⟩ if there is a known alpha to numeric mapping or ⟨false⟩ otherwise.

\TrackLangScriptNumericToAlpha{⟨numeric code⟩}

Expands to the alpha code corresponding to the given numeric code or empty if no mapping.

\TrackLangScriptIfKnownNumeric{⟨numeric code⟩}{⟨true⟩}{⟨false⟩}

Expands to ⟨true⟩ if there is a known numeric to alpha mapping or ⟨false⟩ otherwise.

\TrackLangScriptAlphaToName{⟨alpha code⟩}
4. Supplementary Packages

Expands to the name corresponding to the given alpha code or empty if no mapping.

\TrackLangScriptAlphaToDir{⟨alpha code⟩}

Expands to the direction corresponding to the given alpha code or empty if no mapping.

\TrackLangScriptSetParent{⟨alpha code⟩}{⟨parent alpha code⟩}

Sets the parent for the given alpha code.

\TrackLangScriptGetParent{⟨alpha code⟩}

Expands to the parent for the given alpha code or empty if no mapping.

\TrackLangScriptIfHasParent{⟨alpha code⟩}{⟨true⟩}{⟨false⟩}

Expands to ⟨true⟩ if the given alpha code has a parent or to ⟨false⟩ otherwise. Note that if a parent is explicitly set to empty with \TrackLangScriptSetParent then it will be considered defined, but if the ⟨parent⟩ argument was empty in \TrackLangScriptMap, then it will be undefined.

\TrackLangAddExtraScriptFile{⟨file⟩}

This command adds ⟨file⟩ to the list of extra script files that should be input by tracklang-scripts.tex, unless tracklang-scripts.tex has already been input, in which case ⟨file⟩ will be input straight away.
5. 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 on page 5). 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-Latn-YY}
```

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

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
5. Detecting the User’s Requested Languages

639-2 code (or both), and in some cases there is an ISO 3166-1 code identifying the dialect region. Where a language has different ISO 639-2 (T) and 639-2 (B) codes, the “T” version is assumed.

When the \texttt{tracklang.sty} \LaTeX package is loaded, it first attempts to find the language options through the package options supplied to \texttt{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, \texttt{tracklang.sty} then attempts to identify language settings in the following order:

1. if \texttt{\bbl@loaded} is defined (babel), \texttt{tracklang} will iterate over each label in that command definition;
2. if \texttt{\trans@languages} is defined (translator), \texttt{tracklang} will iterate over each label in that command definition;
3. if \texttt{ngerman} has been loaded, the \texttt{ngerman} dialect will be tracked;
4. if \texttt{german} has been loaded, the \texttt{german} root language will be tracked;
5. if \texttt{polyglossia} has been loaded:
   a) if \texttt{\xpg@bcp@loaded} has been defined, \texttt{tracklang} will iterate over the BCP 47 tags in that command definition;
   b) if \texttt{\xpg@loaded} has been defined, \texttt{tracklang} will iterate over each language label in that command definition;
   c) \texttt{tracklang} will iterate over all \texttt{tracklang} options and test if the root language has been loaded.

Note that this references internal commands provided by other packages. Of these, only the \texttt{polyglossia} commands are documented in the package manual, and so are the only ones that can be relied on.

Each identified language and dialect is added to the \textit{tracked language} and \textit{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:

\begin{verbatim}
\AnyTrackedLanguages{⟨true⟩}{⟨false⟩}
\end{verbatim}

This will do \texttt{⟨true⟩} if one or more languages have been tracked otherwise it will do \texttt{⟨false⟩}. (Each detected dialect will automatically have the root language label added to the tracked language list, if it’s not already present.)
5. Detecting the User’s Requested Languages

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

\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

(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-Latn-GB been tracked?
\GetTrackedDialectFromLanguageTag{en-Latn-GB}\textbf{\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}
5. Detecting the User’s Requested Languages

Has de-DE-1996 been tracked?
\GetTrackedDialectFromLanguageTag{de-DE-1996}{\thisdialect}\
\ifx\thisdialect\empty
No!
\else
\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{⟨cs⟩}{⟨body⟩}

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

You can iterate through each tracked language using:

\ForEachTrackedLanguage{⟨cs⟩}{⟨body⟩}

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

The above for-loops use the same internal mechanism as \List’s \@for loop. Since this isn’t defined by \TeX, a similar command (\@tracklang@for) will be defined that works in the same way.

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.
5. Detecting the User’s Requested Languages

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

\IfTrackedLanguage{⟨language-label⟩}{⟨true⟩}{⟨false⟩}

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

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

\IfTrackedDialect{⟨dialect-label⟩}{⟨true⟩}{⟨false⟩}

where ⟨dialect-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.
5. Detecting the User’s Requested Languages

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

\( \text{\textbackslash TrackedLanguageFromDialect\{\textit{dialect}\}} \)

If \( \langle \text{dialect} \rangle \) 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:

\( \text{\textbackslash TrackedDialectsFromLanguage\{\textit{root language label}\}} \)

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:

\( \text{\textbackslash IfTrackedLanguageHasIsoCode\{\textit{code type}\}\{\textit{label}\}\{\text{true}\}\{\text{false}\}} \)

where \( \langle \text{code type} \rangle \) is the type of ISO code (for example, 639-1 for root languages or 3166-1 for regional dialects), and \( \langle \text{label} \rangle \) 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 \textbackslash TrackLanguage-Tag.

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

\( \text{\textbackslash IfTrackedIsoCode\{\textit{code type}\}\{\textit{code}\}\{\text{true}\}\{\text{false}\}} \)

where \( \langle \text{code type} \rangle \) is again the type of ISO code (for example, 639-1 or 3166-1), and \( \langle \text{code} \rangle \) 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:

\( \text{\textbackslash TrackedLanguageFromIsoCode\{\textit{code type}\}\{\textit{code}\}} \)

This does nothing if the given \( \langle \text{code} \rangle \) for the given ISO \( \langle \text{code type} \rangle \) 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:

\( \text{\textbackslash TrackedIsoCodeFromLanguage\{\textit{code type}\}\{\textit{label}\}} \)

where \( \langle \text{label} \rangle \) is the language or dialect label and \( \langle \text{code type} \rangle \) is the ISO code type (for example, 639-1 or 3166-1). Unlike \textbackslash TrackedLanguageFromIsoCode, this command only expands to
5. Detecting the User’s Requested Languages

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

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

\TwoLetterIsoLanguageCode

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

\ThreeLetterIsoLanguageCode

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

\ThreeLetterExtIsoLanguageCode

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{⟨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{⟨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 §6 for setting this value. Likewise for the following commands.)
You can test if a dialect has an associated modifier using:

\IfHasTrackedDialectModifier{\langle \text{dialect}\rangle}{\langle \text{true}\rangle}{\langle \text{false}\rangle}

If the dialect has an associated modifier this does \langle \text{true}\rangle otherwise it does \langle \text{false}\rangle.
For example:

\documentclass[british,francais,american,canadian,canadien,dutch]{article}
\usepackage{tracklang}
\begin{document}
Languages:
\ForEachTrackedLanguage{\langle \text{ThisLanguage}\rangle}{\langle \text{ISO ISO \TwoLetterIsoLanguageCode:} \langle \text{\TrackedIsoCodeFromLanguage{\TwoLetterIsoLanguageCode}{\ThisLanguage}}'\rangle\rangle. }\end{document}

Dialects:
\ForEachTrackedDialect{\langle \text{ThisDialect}\rangle}{\langle \text{\IfTrackedLanguageHasIsoCode{\TwoLetterIsoCountryCode}{\ThisDialect}}\rangle
\langle \text{\TrackedIsoCodeFromLanguage{\TwoLetterIsoCountryCode}{\ThisDialect}}'\rangle} {no specific region};
\text{root: } \TrackedLanguageFromDialect{\langle \ThisDialect\rangle}.

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

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

Country ISO \langle \TwoLetterIsoCountryCode\rangle code for `canadian':
\TrackedIsoCodeFromLanguage{\TwoLetterIsoCountryCode}{\langle \TwoLetterIsoCountryCode\rangle}{\langle \text{canadian}\rangle}.
\end{document}

This produces:
5. Detecting the User’s Requested Languages

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\{\langle dialect\rangle\}

where \langle dialect\rangle 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\{\langle dialect\rangle\}\{\langle true\rangle\}\{\langle false\rangle\}

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

\GetTrackedDialectScript\{\langle dialect\rangle\}

where \langle dialect\rangle is the label identifying the dialect.

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

\IfHasTrackedDialectScript\{\langle dialect\rangle\}\{\langle true\rangle\}\{\langle false\rangle\}

If the dialect has an associated script this does \langle true\rangle otherwise it does \langle false\rangle. This informa-
tion 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
Cyr1. 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 Cyr1 for Russian dialects. The
5. Detecting the User’s Requested Languages

default script for known languages can be obtained using:

```latex
\TrackLangGetDefaultScript{⟨language⟩}
```

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:

```latex
\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:

```latex
\input tracklang-scripts
```

There’s a convenient \LaTeX wrapper tracklang-scripts.sty:

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

See §4 for further details of that package.

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

```latex
\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{⟨dialect⟩}

where ⟨dialect⟩ is the label identifying the dialect.

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

\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{⟨dialect⟩}

where ⟨dialect⟩ is the label identifying the dialect.

You can test if a dialect has additional information using:

\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?
5. Detecting the User’s Requested Languages

To help with this, tracklang provides:

\IfTrackedLanguageFileExists{\langle dialect\rangle}{\langle prefix\rangle}{\langle suffix\rangle}{\langle true code\rangle}{\langle false code\rangle}

This attempts to find the file called \langle prefix\rangle\langle localeid\rangle\langle suffix\rangle where \langle localeid\rangle is determined from \langle dialect\rangle (see below). If the file is found then \CurrentTrackedTag is set to \langle localeid\rangle and \langle true code\rangle is done, otherwise \langle false code\rangle 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 \langle localeid\rangle.

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

\TrackLangRequireDialect{(load code)}{(pkgname)}{(dialect)}

which uses \IfTrackedLanguageFileExists{(load code)}{(pkgname)}{(dialect)} to input the resource file if found. The prefix is given by \langle pkgname\rangle- and the suffix is .ldf. A warning is issued if no resource file is found. Note that while it makes sense for \langle pkgname\rangle 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 to \langle pkgname\rangle, which allows the prefix to be picked up by resource file commands, such as \TrackLangProvidesResource and \TrackLangRequireResource. (See below.)

The optional argument \langle load code\rangle 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}
5. Detecting the User’s Requested Languages

which enables the following commands that may be used within `⟨true code⟩` or `⟨false code⟩`:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>\CurrentTrackedDialect</td>
<td>Expands to the dialect label.</td>
</tr>
<tr>
<td>\CurrentTrackedLanguage</td>
<td>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).</td>
</tr>
<tr>
<td>\CurrentTrackedRegion</td>
<td>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.</td>
</tr>
<tr>
<td>\CurrentTrackedIsoCode</td>
<td>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.</td>
</tr>
<tr>
<td>\CurrentTrackedDialectModifier</td>
<td>The dialect’s modifier or empty if not set. (This is set but not used in the set of possible <code>⟨localeid⟩</code> values.)</td>
</tr>
<tr>
<td>\CurrentTrackedDialectVariant</td>
<td>The dialect’s variant or empty if not set.</td>
</tr>
<tr>
<td>\CurrentTrackedDialectSubLang</td>
<td>The dialect’s sub-language code or empty if not set.</td>
</tr>
<tr>
<td>\CurrentTrackedDialectAdditional</td>
<td></td>
</tr>
</tbody>
</table>
5. Detecting the User’s Requested Languages

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 for the language is used instead.

\IfTrackedLanguageFileExists behaves as follows:

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

- If a dialect with the given label has been detected, then:
  - For each possible \langle localeid \rangle in an ordered set of tags determined by the dialect label (see below), the first file matching \langle prefix \rangle \langle localeid \rangle \langle suffix \rangle that’s found on \TeX’s path results in the condition evaluating to *true* and \CurrentTrackedTag is set to the current \langle localeid \rangle in the set. The rest of the set of possible values of \langle localeid \rangle is skipped.
  - If no file matching \langle prefix \rangle \langle localeid \rangle \langle suffix \rangle is found on \TeX’s path, then the condition evaluates to *false* and \CurrentTrackedTag is set to the final \langle localeid \rangle in the set (the language label).

The ordered set of possible values of \langle localeid \rangle is determined from the given dialect.

The ordering has changed in version 1.4, which now also includes the script and variant. This new ordering should typically make the more common combinations closer to the start of the search.

The possible values of \langle localeid \rangle are listed below in the order of priority used by \IfTrackedLanguageFileExists. Note that the set may contain repetitions (for example, if the dialect label is the same as the root language label). If an item contains an element that hasn’t been set (such as the ISO 639-3 code or a sub-language \langle sublang \rangle or variant) then that item is skipped.

1. \langle localeid \rangle is just the value of \CurrentTrackedLanguageTag.
2. \langle localeid \rangle is just the dialect label.
3. \langle localeid \rangle is \langle ISO 639-1\rangle–\langle sublang\rangle–\langle script\rangle–\langle region\rangle.
5. Detecting the User’s Requested Languages

4. \(\text{localeid}\) is \(\langle\text{ISO 639-1}\rangle-\langle\text{script}\rangle-\langle\text{region}\rangle\).

5. \(\text{localeid}\) is \(\langle\text{ISO 639-1}\rangle-\langle\text{sublang}\rangle-\langle\text{region}\rangle\) (if there’s no script or if the script is the default for the given language). \(\langle\text{ISO 639-1}\rangle-\langle\text{region}\rangle\) (if there’s no script or if the script is the default for the given language).

6. \(\text{localeid}\) is \(\langle\text{ISO 639-1}\rangle-\langle\text{sublang}\rangle-\langle\text{script}\rangle\).

7. \(\text{localeid}\) is \(\langle\text{ISO 639-1}\rangle-\langle\text{script}\rangle\).

8. \(\text{localeid}\) is \(\langle\text{ISO 639-1}\rangle-\langle\text{sublang}\rangle\).

9. \(\text{localeid}\) is just \(\langle\text{ISO 639-1}\rangle\).

10. \(\text{localeid}\) is \(\langle\text{ISO 639-2}\rangle-\langle\text{sublang}\rangle-\langle\text{script}\rangle-\langle\text{region}\rangle\).

11. \(\text{localeid}\) is \(\langle\text{ISO 639-2}\rangle-\langle\text{script}\rangle-\langle\text{region}\rangle\).

12. \(\text{localeid}\) is \(\langle\text{ISO 639-2}\rangle-\langle\text{sublang}\rangle-\langle\text{region}\rangle\) (if there’s no script or if the script is the default for the given language). \(\langle\text{ISO 639-2}\rangle-\langle\text{region}\rangle\) (if there’s no script or if the script is the default for the given language).

13. \(\text{localeid}\) is \(\langle\text{ISO 639-2}\rangle-\langle\text{script}\rangle\).

14. \(\text{localeid}\) is \(\langle\text{ISO 639-2}\rangle-\langle\text{script}\rangle\).

15. \(\text{localeid}\) is \(\langle\text{ISO 639-2}\rangle-\langle\text{sublang}\rangle\).

16. \(\text{localeid}\) is just \(\langle\text{ISO 639-2}\rangle\).

17. \(\text{localeid}\) is \(\langle\text{ISO 639-3}\rangle-\langle\text{sublang}\rangle-\langle\text{script}\rangle-\langle\text{region}\rangle\).

18. \(\text{localeid}\) is \(\langle\text{ISO 639-3}\rangle-\langle\text{script}\rangle-\langle\text{region}\rangle\).

19. \(\text{localeid}\) is \(\langle\text{ISO 639-3}\rangle-\langle\text{sublang}\rangle-\langle\text{region}\rangle\) (if there’s no script or if the script is the default for the given language). \(\langle\text{ISO 639-3}\rangle-\langle\text{region}\rangle\) (if there’s no script or if the script is the default for the given language).

20. \(\text{localeid}\) is \(\langle\text{ISO 639-3}\rangle-\langle\text{sublang}\rangle-\langle\text{script}\rangle\).

21. \(\text{localeid}\) is \(\langle\text{ISO 639-3}\rangle-\langle\text{script}\rangle\).

22. \(\text{localeid}\) is \(\langle\text{ISO 639-3}\rangle-\langle\text{sublang}\rangle\).

23. \(\text{localeid}\) is just \(\langle\text{ISO 639-3}\rangle\).

24. \(\text{localeid}\) is just \(\langle\text{region}\rangle\).

25. \(\text{localeid}\) is \(\langle\text{ISO 639-1}\rangle-\langle\text{sublang}\rangle-\langle\text{variant}\rangle\) or \(\langle\text{ISO 639-1}\rangle-\langle\text{variant}\rangle\) if \(\langle\text{sublang}\rangle\) is missing.
5. Detecting the User’s Requested Languages

26. \( \langle \text{localeid} \rangle \) is \( \langle \text{ISO 639-2} \rangle-\langle \text{sublang} \rangle-\langle \text{variant} \rangle \) or \( \langle \text{ISO 639-2} \rangle-\langle \text{variant} \rangle \) if \( \langle \text{sublang} \rangle \) is missing.

27. \( \langle \text{localeid} \rangle \) is \( \langle \text{ISO 639-3} \rangle-\langle \text{sublang} \rangle-\langle \text{variant} \rangle \) or \( \langle \text{ISO 639-3} \rangle-\langle \text{variant} \rangle \) if \( \langle \text{sublang} \rangle \) is missing.

28. \( \langle \text{localeid} \rangle \) is just the value of \( \text{\textbackslash CurrentTrackedLanguage} \) (the root language label).

For example (pre v1.3):

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

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

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

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

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

1. mypackage-en-GB.1df (language tag)
5. Detecting the User’s Requested Languages

2. mypackage-british.1df (dialect label)

3. mypackage-en-Latn-GB.1df (639-1 language code, script, region)

4. mypackage-en-GB.1df (639-1 language code, region)

5. mypackage-en-Latn.1df (639-1 language code, script)

6. mypackage-en.1df (639-1 language code)

7. mypackage-eng-Latn-GB.1df (639-2 language code, script, region)

8. mypackage-eng-GB.1df (639-2 language code, region)

9. mypackage-eng-Latn.1df (639-2 language code, script)

10. mypackage-eng.1df (639-2 language code)

11. mypackage-GB.1df (region)

12. mypackage-english.1df (language label)

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

1. mypackage-de-AT-1996.1df (language tag)

2. mypackage-naustrian.1df (dialect label)

3. mypackage-de-Latn-AT.1df (639-1 language code, script, region)

4. mypackage-de-AT.1df (639-1 language code, region)

5. mypackage-de-Latn.1df (639-1 language code, script)

6. mypackage-de.1df (639-1 language code)

7. mypackage-deu-Latn-AT.1df (639-2 language code, script, region)

8. mypackage-deu-AT.1df (639-2 language code, region)

9. mypackage-deu-Latn.1df (639-2 language code, script)

10. mypackage-deu.1df (639-2 language code)

11. mypackage-AT.1df (region)

12. mypackage-de-1996.1df (639-1 language code, variant)

13. mypackage-deu-1996.1df (639-2 language code, variant)

14. mypackage-german.1df (language label)
5. Detecting the User’s Requested Languages

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

1. `mypackage-fr.1df` (language tag)
2. `mypackage-francais.1df` (dialect label)
3. `mypackage-fr-Latn.1df` (639-1 language code, script)
4. `mypackage-fr.1df` (639-1 language code)
5. `mypackage-fra-Latn.1df` (639-2 language code, script)
6. `mypackage-fra.1df` (639-2 language code)
7. `mypackage-french.1df` (language)

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-fr.1df` (language tag)
2. `mypackage-french.1df` (dialect label)
3. `mypackage-fr-Latn.1df` (639-1 language code, script)
4. `mypackage-fr.1df` (639-1 language code)
5. `mypackage-fra-Latn.1df` (639-2 language code, script)
6. `mypackage-fra.1df` (639-2 language code)
7. `mypackage-french.1df` (language)

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

If the dialect label is identical to the root language label then it means that all associated information is the default for that language. For example, in the above case of `french`, the script is `Latn` and the region is unspecified. The root language label can therefore be used as the fallback in the event of no other match but for the specific case where the dialect is identical to the root language then all unnecessary file name checks can be skipped.

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

```verbatim
\AnyTrackedLanguages
{%
  \ForEachTrackedLanguage{\ThisLanguage}
  {%
    try to load the language file for this root language
    \IfTrackedLanguageFileExists{\ThisLanguage}
  %}
%}
```
5. Detecting the User’s Requested Languages

```latex
{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⟩-⟨localeid⟩.ldf, you can identify the file using (new
to version 1.3):

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

where ⟨tag⟩ is the locale identifier.

If \ProvidesFile is defined (through the \TeX 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⟩-⟨tag⟩.ldf, using (new to ver-
5. Detecting the User’s Requested Languages

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

```latex
\TrackLangRequireResource{⟨tag⟩}
```

% (In file foo-en-GB.ldf)
% Declare this regional file:
\TrackLangProvideResource{en-GB}
% load root language file foo-english.ldf:
\TrackLangRequireResource{english}

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

It may be that you want to load a file depending on the input encoding. The `inputenc` package defines `\inputencodingname`, but this is only used with `pdfLaTeX`. To avoid repeated tests to determine whether or not `\inputencodingname` has been defined, you can use:

```latex
\TrackLangEncodingName
```

This will expand to `utf8` if `\inputencodingname` hasn’t been defined, otherwise it will expand to `\inputencodingname`. For example:

```latex
\InputIfFileExists{foo-⟨\TrackLangEncodingName⟩.ldf}{% support available for the document encoding }{% no support for the document encoding }
```

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:

```latex
\TrackLangRequireResourceOrDo{⟨tag⟩}{⟨code1⟩}{⟨code2⟩}
```

If you want to load a resource file if it exists (without an error if it doesn’t exist), then you
5. Detecting the User’s Requested Languages

can use

\TrackLangRequestResource\{\langle\text{tag}\rangle\}\{\langle\text{not found code}\rangle\}

If the file doesn’t exist, \langle\text{not found code}\rangle 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.1df 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 language caption hooks in the form \captions\langle\text{language}\rangle (where \langle\text{language}\rangle is the root language label) whereas babel has dialect captions hooks in the form \captions\langle\text{dialect}\rangle (where \langle\text{dialect}\rangle is the dialect label). 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, tracklang provides a command to perform this set of conditionals using generic
5. Detecting the User’s Requested Languages

code:

\TrackLangAddToHook{(code)}{(type)}

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

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

Since \texttt{captions} is a commonly used hook type, there’s a shortcut command provided:

\TrackLangAddToCaptions{(code)}

This is equivalent to

\TrackLangAddToHook{(code)}{captions}

There may be some hooks, such as \texttt{\date{}\langle\texttt{dialect}\rangle}, that need redefining rather than appending to, so there’s an analogous command:

\TrackLangRedefHook{(code)}{(type)}

which will redefine the hook to do \texttt{(code)}.

Note that no expansion is performed on \texttt{(code)} when appending or redefining a hook.

5.1. Examples

The examples in this section illustrate the above commands.

5.1.1. animals.sty

This example is for a trivial package called animals.sty that defines three textual commands: \texttt{\catname}, \texttt{\dogname} and \texttt{\ladybirdname}. The default values are: “cat”, “dog” and “bishy-barney-bee”. The supported languages are defined in files animals-\langle\texttt{localeid}\rangle.ldf.

Here’s the code for animals.sty:

\texttt{\textasciitilde Thass\ Broad\ Norfolk,\ my\ bewties: -P}\footnote{Thass Broad Norfolk, my bewties :-P}
5. Detecting the User’s Requested Languages

Here’s a Plain TeX version that picks up the language from the locale environment variable:
5. Detecting the User’s Requested Languages

\input{tracklang}
\TrackLangFromEnv

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

\AnyTrackedLanguages
{%
 \ForEachTrackedDialect{\thisdialect}{%
   \TrackLangRequireDialect{animals}{\thisdialect}%
 }%
%
o% 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 L\LaTeX formats contains:

\TrackLangProvidesResource{english}
\def\englishanimals{%
 \def\catname{cat}%
 \def\dogname{dog}%
}
5. Detecting the User’s Requested Languages

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{Marienkäfer}%
}\TrackLangAddToCaptions{\germananimals}

This means that if babel or polyglossia are loaded, the redefinitions are automatically per-
5. Detecting the User’s Requested Languages

formed 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]{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]{article}
\usepackage{babel}
\usepackage{animals}
\begin{document}
\selectlanguage{american}
\catname.
\dogname.
\ladybirdname.
\selectlanguage{german}
\catname.
\dogname.
\ladybirdname.
5. Detecting the User’s Requested Languages

\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:
5. Detecting the User’s Requested Languages

\TrackLangProvidesResource{en-US}
\providecommand*{\enUSanimals}{}
englishanimals
\renewcommand*{\ladybirdname}{ladybug}{}

\TrackLangRequireResourceOrDo{english}
{
\TrackLangAddToCaptions{\renewcommand*{\ladybirdname}{ladybird}}{}
}
{
\TrackLangAddToCaptions{\enUSanimals}
}

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.

5.1.2. regions.sty

Earlier on page 49, I mentioned the search order for \IfTrackedLanguageFileExists where if, for example, the dialect is british, the file search (v1.4+) will be:

1. mypackage-en-GB.1df (language tag)
2. mypackage-british.1df (dialect label)
3. mypackage-en-Latn-GB.1df (639-1 language code, script, region)
4. mypackage-en-GB.1df (639-1 language code, region)
5. mypackage-en-Latn.1df (639-1 language code, script)
5. Detecting the User’s Requested Languages

6. mypackage-en.1df (639-1 language code)

7. mypackage-eng-Latn-GB.1df (639-2 language code, script, region)

8. mypackage-eng-GB.1df (639-2 language code, region)

9. mypackage-eng-Latn.1df (639-2 language code, script)

10. mypackage-eng.1df (639-2 language code)

11. mypackage-GB.1df (region)

12. mypackage-english.1df (language label)

You may have wondered why mypackage-GB.1df 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 ⟨localeid⟩ 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 1df 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⟨localeid⟩.1df 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:

% custom file loader for regions.sty
\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.

```
\% Example package regions.sty
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{regions}
\RequirePackage{tracklang}[2016/10/07] \% v1.3+
\DeclareOption*{\TrackLanguageTag{\CurrentOption}}
\ProcessOptions
\newcommand*{\weightunit}{kg}
\newcommand*{\lengthunit}{mm}
\newcommand*{\currencyunit}{EUR}
\newcommand*{\unitname}{units}
\newcommand*{\RequireRegionsDialect}[1]{%
   \TrackLangRequireDialect
   \TrackLangRequireResource{\CurrentTrackedTag}%
   \TrackLangRequireResource{\CurrentTrackedLanguage}%
}%
\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):
5. Detecting the User’s Requested Languages

- regions-CA.ldf (Canada):

```latex
\TrackLangProvidesResource{CA}
\providecommand*{\CAunits}{% 
  \renewcommand*{\weightunit}{kg}% 
  \renewcommand*{\lengthunit}{mm}% 
  \renewcommand*{\currencyunit}{CAD}% 
} \TrackLangAddToCaptions{\CAunits}
```

- regions-GB.ldf (Great Britain):

```latex
\TrackLangProvidesResource{GB}
\providecommand*{\GBunits}{% 
  \renewcommand*{\weightunit}{kg}% 
  \renewcommand*{\lengthunit}{mm}% 
  \renewcommand*{\currencyunit}{GBP}% 
} \TrackLangAddToCaptions{\GBunits}
```

- regions-US.ldf (USA):
5. Detecting the User’s Requested Languages

\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}
5. Detecting the User’s Requested Languages

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

- regions-german.ldf:

\providecommand*{\germannames}{%\renewcommand*{\unitname}{Ma\$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 \texttt{\langle localeid \rangle} 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 \texttt{†} in Table 1.2 on page 5, as then it will be picked up before the country code.

In the above example, \texttt{regions-CA.ldf} is matched rather than \texttt{regions-french.ldf}, so \texttt{regions-CA.ldf} is loaded by

\TrackLangRequireResource{\CurrentTrackedTag}

After this, the language file \texttt{regions-french.ldf} is then loaded:
5. Detecting the User's Requested Languages

\TrackLangRequireResource{\CurrentTrackedLanguage}

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

\% Modified example package regions.sty
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{regions}

\% Pass all options to tracklang.sty:
\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%

67
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}%
      \TrackLangRequireResource{\CurrentTrackedLanguage}%
    \fi
  \else
    \TrackLangRequireResource{\CurrentTrackedRegion}%
  \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{dialect}` hook for either the dialect label or the root language label, so mappings need to be defined from the tracklang
5. Detecting the User’s Requested Languages

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}
\selectlanguage{ngerman}
\end{document}
5. Detecting the User’s Requested Languages

\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{regions}
\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
5. Detecting the User’s Requested Languages

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 on page 7 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 on page 7.


6. 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. (See also: Integrating tracklang.tex into Language Packages.)

The tracklang package can be loaded using

\input tracklang

or (\TeX only)

\RequirePackage{tracklang}

When using \TeX, 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 your package is the language package, so you need to decide whether or not to allow the user to set up the localisation information through the document class options.

There’s a hook that, if defined, is performed by tracklang.sty after the package options have been loaded but before known language packages are checked:

\@tracklang@prelangpkgcheck@hook

If you prefer \RequirePackage over \input but you want to make tracklang.sty skip the check for known language packages then (as from v1.3.8) define the pre-language package check hook as follows:

\providecommand\@tracklang@prelangpkgcheck@hook{\endinput}
\RequirePackage{tracklang}[2019/10/06]% v1.3.8+

\footnote{dickimaw-books.com/latex/tracklang/langpkg.shtml}
6. Adding Support for Language Tracking

This will still pick up languages supplied 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 document has already input it.

To integrate tracklang into your language package, you need to consider the following steps:

1. Does tracklang define your supported ISO 15924 language scripts in the tracklang-scripts.tex file?
   
   If yes, then skip this step. Otherwise create a file with the relevant \TrackLang-ScriptMap command for each unknown script and identify this new file with \TrackLangAddExtraScriptFile (see §6.3). This usually won’t be necessary unless you have a custom script or a child script (a script that’s a sub-category of another script).

2. Does tracklang recognise the root language?
   
   If yes, then skip this step.
   
   If your package is setting up a language that tracklang doesn’t recognise then you will need to define the root language using \TrackLangNewLanguage (see §6.5).
   
   This usually won’t be the case as tracklang should support all languages that have an official ISO 639-1 alpha-2 code.
   
   If you simply have a different label from tracklang identifying the root language, then you can just set up your label as a dialect using \TrackLangProvidePredefinedDialect.

3. Does tracklang define the relevant ISO 3166-1 region codes in the tracklang-region-codes.tex file?
   
   If yes, then skip this step. Otherwise create a file with the relevant \TrackLangRegionMap command for each new region and identify this new file with \TrackLangAddExtraRegionFile (see §6.4). This usually won’t be necessary as tracklang should recognise all countries that have an alpha-2 region code, but you may require it if you need a broader region, such as EU.

4. Do you want to define some convenient dialect labels that can be used with \TrackPredefinedDialect?
   
   If no, then skip this step. Otherwise you can use \TrackLangProvidePredefinedLanguage for root languages and \TrackLangProvidePredefinedDialect for dialects with additional information, such as a region, sub-language or script (see §6.6).

5. In your language initialisation code, add the tracklang code to track the particular dialect (for example, use \TrackPredefinedDialect for recognised dialect labels or use the \AddTracked\langle Xxx\rangle set of commands). See §6.1.

6. In your language selection code (such as \selectlanguage), add \SetCurrentTrackedDialect\{(label)} to allow the document author to easily query the current localisation settings (such as the region). See §6.2.
6. Adding Support for Language Tracking

6.1. Initialising a New Language or Dialect

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

\TrackPredefinedDialect\{\langle dialect label \rangle\}

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

If there’s no matching dialect predefined by tracklang, you can just use \TrackLocale or \TrackLanguageTag (described in §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\{\langle dialect \rangle\}, then if \langle dialect \rangle doesn’t match the corresponding tracklang dialect label, you can provide a mapping using \SetTrackedDialectLabelMap, described below.

6.2. Switching Language or Dialect

When the document author switches to a different language or dialect, the current localisation information can be set with:

\SetCurrentTrackedDialect\{\langle dialect \rangle\}

where \langle dialect \rangle 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 \langle dialect \rangle).
- \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 §4 for further details.
6. Adding Support for Language Tracking

- \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 \language name needs to be converted to \dialect. See the accompanying sample file sample-setlang.tex for an example.)

6.3. Defining New Scripts

The tracklang-scripts.tex file isn’t automatically loaded, but if it is then, as from v1.4, it contains a hook at the end of the file that can be used to load additional files that define supplementary scripts. This entails creating a file called, say, mypackage-scripts.tex that contains:

\TrackLangScriptMap\{(alpha code)\}{(numeric code)\}{(name)\}{(direction)\}{(parent script)\}

The first argument \(alpha code\) is the four-letter ISO 15924 code (such as Latn), the second argument is the numeric code (such as 215), the third argument \(name\) is the name of the script (such as Latin), the fourth argument is the direction (such as LR for left-to-right) and the final argument is the parent script (leave blank if there’s no parent). Note that this command will override any previous mapping for those codes. No check is performed to determine if they have already been defined.

The supplementary file should be identified with:

\TrackLangAddExtraScriptFile\{(filename)\}

Additional information can be found in §4.

6.4. Defining New Regions

The tracklang-region-codes.tex file isn’t automatically loaded, but if it is then, as from v1.4, it contains a hook at the end of the file that can be used to load additional files that define supplementary regions. This entails creating a file called, say, mypackage-regions.tex that contains:
6. Adding Support for Language Tracking

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

where the first argument is the numeric region code (such as 826), the second argument is the alpha-2 region code (such as GB) and the third argument is the alpha-3 region code (such as GBR). Note that this command will override any previous mapping for those codes. No check is performed to determine if they have already been defined.

The supplementary file should be identified with:

\TrackLangAddExtraRegionFile{\langle filename \rangle}

Additional information can be found in §4.

6.5. Defining a New Language

(New to version 1.3.) If the root language isn’t recognised by tracklang (not listed in Table 1.2 on page 5), then it can be defined (but not tracked at this point) using:

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

where \langle language label \rangle is the root language label, \langle 639-1 code \rangle is the ISO 639-1 code for that language (may be empty if there isn’t one), \langle 639-2 (T) \rangle is the ISO 639-2 (T) code for that language (may be empty if there isn’t one), \langle 639-2 (B) \rangle is the ISO 639-2 (B) code for that language (may be empty if it’s the same as \langle 639-2 (T) \rangle), \langle 639-3 \rangle is the ISO 639-3 code for that language (empty if the same as the 639-2 code), \langle 3166-1 \rangle is the territory ISO 3166-1 code for languages that are only spoken in one territory (should be empty if the language is spoken in multiple territories), and \langle default script \rangle is the default script (empty if disputed or varies according to region).

You can then track this language using:

\AddTrackedDialect{\langle dialect label \rangle}\{\langle root language label \rangle\}

for dialects (where \langle dialect label \rangle is the dialect label and \langle root language label \rangle is the root language label) or, if no regional variant is needed, you can instead use:

\AddTrackedLanguage{\langle root language label \rangle}

This is equivalent to
6. Adding Support for Language Tracking

\AddTrackedDialect\{\langle root language label \rangle\}\{\langle root language label \rangle\}

Note that \AddTrackedDialect defines:

\TrackLangLastTrackedDialect

to the dialect label, which makes it easier to reference the last dialect to be tracked.

6.6. Defining New tracklang Labels

A dialect label may be predefined with associated information that allows that particular combination to be easily tracked with \TrackPredefinedDialect. In the case of a dialect label that only requires the information provided in \TrackLangNewLanguage you can use:

\TrackLangProvidePredefinedLanguage\{\langle language label \rangle\}

where \langle language label \rangle corresponds to the language label used in \TrackLangNewLanguage. This allows

\TrackPredefinedDialect\{\langle label \rangle\}

to not only track the root language but also the associated ISO codes.

If the dialect label doesn’t match the root language label then use:

\TrackLangProvidePredefinedDialect\{\langle dialect label \rangle\}\{\langle root language label \rangle\}\{\langle 3166-1 code \rangle\}\{\langle modifier \rangle\}\{\langle variant \rangle\}\{\langle map \rangle\}\{\langle script \rangle\}

where \langle dialect label \rangle is the new tracklang dialect label, \langle root language label \rangle is the tracklang root language label, \langle region \rangle is the ISO 3166-1 region code (may be empty), \langle modifier \rangle is the modifier (may be empty), \langle variant \rangle is the variant information (may be empty), \langle map \rangle is your package’s language label that corresponds to the tracklang dialect label supplied in the first argument (may be empty if identical), and \langle script \rangle is the ISO 15924 alpha-4 script code (may be empty if it’s the same as the default script for the root language).

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

\AddTrackedDialect\{\langle dialect \rangle\}\{\langle root language label \rangle\}
6. Adding Support for Language Tracking

where \(\langle\text{root language label}\rangle\) is the label for the dialect’s root language (Table 1.2 on page 5) and \(\langle\text{dialect}\rangle\) 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 \(\text{\textbackslash TrackLangLastTrackedDialect}\) to \(\langle\text{dialect}\rangle\) for convenient reference if required. Note that \(\text{\textbackslash AddTrackedDialect}\) is internally used by commands like \(\text{\textbackslash TrackPredefinedDialect}, \text{\textbackslash TrackLocale}\) and \(\text{\textbackslash TrackLanguage-Tag}\).

(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 \texttt{ngermanDE} but the closest match in \texttt{babel} is \texttt{ngerman}. This means that the caption hook \texttt{\captionsngerman} can’t be accessed through:

\[
\text{\textbackslash csname captions\CurrentTrackedDialect\endcsname}
\]

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

\[
\text{\textbackslash SetTrackedDialectLabelMap\{\langle\text{tracklang-label}\rangle\}\{\langle\text{hook-label}\rangle\}}
\]

where \(\langle\text{tracklang-label}\rangle\) is the tracklang label and \(\langle\text{hook-label}\rangle\) is the language hook label. For example:

\[
\text{\textbackslash TrackLanguageTag\{de-AR-1996\}}
\text{\textbackslash SetTrackedDialectLabelMap\{\text{\textbackslash TrackLangLastTrackedDialect}\}\{\text{ngerman}\}}
\]

Since \(\text{\textbackslash TrackLanguageTag}\) internally uses \(\text{\textbackslash AddTrackedDialect}\) the dialect label created by tracklang can be accessed using \(\text{\textbackslash TrackLangLastTrackedDialect}\). This means that \(\text{\textbackslash TrackLangAddToCaptions}\) can now find the \texttt{\captionsngerman} hook even though the tracklang dialect label isn’t \texttt{ngerman}.

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

\[
\text{\textbackslash AddTrackedLanguageIsoCodes\{\langle\text{root language label}\rangle\}}
\]

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

\[
\text{\textbackslash SetTrackedDialectModifier\{\langle\text{dialect}\rangle\}\{\langle\text{value}\rangle\}}
\]

where \(\langle\text{dialect}\rangle\) is the dialect label and \(\langle\text{value}\rangle\) is the modifier value. For example:

78
6. Adding Support for Language Tracking

Note that no sanitization is performed on \textit{(value)} when the modifier is set explicitly through \texttt{\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 \texttt{\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:

\[
\langle \text{language} \rangle \_\_\langle \text{territory} \rangle \_\_\langle \text{codeset} \rangle @\langle \text{modifier} \rangle
\]

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 \texttt{latin} whereas the script would be \texttt{Latn}:

\begin{itemize}
  \item \texttt{\AddTrackedDialect{serbianlatin}{serbian}}
  \item \texttt{\AddTrackedLanguageIsoCodes{serbian}}
  \item \texttt{\SetTrackedDialectModifier{serbianlatin}{latin}}
  \item \texttt{\SetTrackedDialectScript{serbianlatin}{Latn}}
\end{itemize}

As from v1.3, you can provide a script (for example, \texttt{Latn} or \texttt{Cyrl}) using:

\begin{itemize}
  \item \texttt{\SetTrackedDialectScript{\langle dialect \rangle}{\langle value \rangle}}
\end{itemize}

where \texttt{\langle dialect \rangle} is the dialect label and \texttt{\langle value \rangle} is the ISO 15924 alpha-4 script identifier. For example:

\begin{itemize}
  \item \texttt{\AddTrackedDialect{serbiancyrl}{serbian}}
  \item \texttt{\AddTrackedLanguageIsoCodes{serbian}}
  \item \texttt{\SetTrackedDialectScript{serbiancyrl}{Cyrl}}
\end{itemize}

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

\begin{itemize}
  \item \texttt{\SetTrackedDialectVariant{\langle dialect \rangle}{\langle value \rangle}}
\end{itemize}
6. Adding Support for Language Tracking

For example:

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

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

\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{(dialect)}{(value)}

where (dialect) is the dialect label and (value) is the additional information.

6.7. Example (alien.sty)

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}
\inputtracklang% v1.3
\DeclareOption{martian}{%
  \TrackPredefinedDialect{martian}
}\
\DeclareOption{lowermartian}{%
  \AddTrackedDialect{lowermartian}{martian}
6. Adding Support for Language Tracking

\AddTrackedLanguageIsoCodes{martian}
\AddTrackedIsoLanguage{3166-1}{YY}{lowermartian}
% other attributes such as
% \SetTrackedDialectVariant{lowermartian}{...}
}
\DeclareOption{uppermartian}{%
\AddTrackedDialect{uppermartian}{martian}
\AddTrackedLanguageIsoCodes{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 setup are in the files alien-⟨localeid⟩.ldf as in the examples from §5.1. 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.1df:

\TrackLangProvidesResource{xx-YY}
\TrackLangRequireResource{martian}% load common elements
\newcommand{\captionslowermartian}{%
6. Adding Support for Language Tracking

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.*
Part II.

Summaries
# A. Region and Script Mappings

Region mappings are listed in Table A.1, and script mappings are listed in Table A.2 on page 88.

### Table A.1: Region Mappings

<table>
<thead>
<tr>
<th>Alpha-2</th>
<th>Alpha-3</th>
<th>Numeric</th>
<th>Alpha-2</th>
<th>Alpha-3</th>
<th>Numeric</th>
</tr>
</thead>
<tbody>
<tr>
<td>AD</td>
<td>AND</td>
<td>020</td>
<td>AE</td>
<td>ARE</td>
<td>784</td>
</tr>
<tr>
<td>AF</td>
<td>AFG</td>
<td>004</td>
<td>AG</td>
<td>ATG</td>
<td>028</td>
</tr>
<tr>
<td>AI</td>
<td>AIA</td>
<td>660</td>
<td>AL</td>
<td>ALB</td>
<td>008</td>
</tr>
<tr>
<td>AM</td>
<td>ARM</td>
<td>051</td>
<td>AO</td>
<td>AGO</td>
<td>024</td>
</tr>
<tr>
<td>AQ</td>
<td>ATA</td>
<td>010</td>
<td>AR</td>
<td>ARG</td>
<td>032</td>
</tr>
<tr>
<td>AS</td>
<td>ASM</td>
<td>016</td>
<td>AT</td>
<td>AUT</td>
<td>040</td>
</tr>
<tr>
<td>AU</td>
<td>AUS</td>
<td>036</td>
<td>AW</td>
<td>ABW</td>
<td>533</td>
</tr>
<tr>
<td>AX</td>
<td>ALA</td>
<td>248</td>
<td>AZ</td>
<td>AZE</td>
<td>031</td>
</tr>
<tr>
<td>BA</td>
<td>BIH</td>
<td>070</td>
<td>BB</td>
<td>BRB</td>
<td>052</td>
</tr>
<tr>
<td>BD</td>
<td>BGD</td>
<td>050</td>
<td>BE</td>
<td>BEL</td>
<td>056</td>
</tr>
<tr>
<td>BF</td>
<td>BFA</td>
<td>854</td>
<td>BG</td>
<td>BGR</td>
<td>100</td>
</tr>
<tr>
<td>BH</td>
<td>BHR</td>
<td>048</td>
<td>BI</td>
<td>BDI</td>
<td>108</td>
</tr>
<tr>
<td>BJ</td>
<td>BEN</td>
<td>204</td>
<td>BL</td>
<td>BLM</td>
<td>652</td>
</tr>
<tr>
<td>BM</td>
<td>BMU</td>
<td>060</td>
<td>BN</td>
<td>BRN</td>
<td>096</td>
</tr>
<tr>
<td>BO</td>
<td>BOL</td>
<td>068</td>
<td>BQ</td>
<td>BES</td>
<td>535</td>
</tr>
<tr>
<td>BR</td>
<td>BRA</td>
<td>076</td>
<td>BS</td>
<td>BHS</td>
<td>044</td>
</tr>
<tr>
<td>BT</td>
<td>BTN</td>
<td>064</td>
<td>BV</td>
<td>BVT</td>
<td>074</td>
</tr>
<tr>
<td>BW</td>
<td>BWA</td>
<td>072</td>
<td>BY</td>
<td>BLR</td>
<td>112</td>
</tr>
<tr>
<td>BZ</td>
<td>BLZ</td>
<td>084</td>
<td>CA</td>
<td>CAN</td>
<td>124</td>
</tr>
<tr>
<td>CC</td>
<td>CCK</td>
<td>166</td>
<td>CD</td>
<td>COD</td>
<td>180</td>
</tr>
<tr>
<td>CF</td>
<td>CAF</td>
<td>140</td>
<td>CG</td>
<td>COG</td>
<td>178</td>
</tr>
<tr>
<td>CH</td>
<td>CHE</td>
<td>756</td>
<td>CI</td>
<td>CIV</td>
<td>384</td>
</tr>
<tr>
<td>CK</td>
<td>COK</td>
<td>184</td>
<td>CL</td>
<td>CHL</td>
<td>152</td>
</tr>
<tr>
<td>CM</td>
<td>CMR</td>
<td>120</td>
<td>CN</td>
<td>CHN</td>
<td>156</td>
</tr>
<tr>
<td>CO</td>
<td>COL</td>
<td>170</td>
<td>CR</td>
<td>CRI</td>
<td>188</td>
</tr>
<tr>
<td>CU</td>
<td>CUB</td>
<td>192</td>
<td>CV</td>
<td>CPV</td>
<td>132</td>
</tr>
<tr>
<td>CW</td>
<td>CUW</td>
<td>531</td>
<td>CX</td>
<td>CXR</td>
<td>162</td>
</tr>
<tr>
<td>CY</td>
<td>CYP</td>
<td>196</td>
<td>CZ</td>
<td>CZE</td>
<td>203</td>
</tr>
<tr>
<td>DE</td>
<td>DEU</td>
<td>276</td>
<td>DJ</td>
<td>DJI</td>
<td>262</td>
</tr>
<tr>
<td>DK</td>
<td>DNK</td>
<td>208</td>
<td>DM</td>
<td>DMA</td>
<td>212</td>
</tr>
</tbody>
</table>
### Table A.1.: Region Mappings (Continued)

<table>
<thead>
<tr>
<th>Alpha-2</th>
<th>Alpha-3</th>
<th>Numeric</th>
<th>Alpha-2</th>
<th>Alpha-3</th>
<th>Numeric</th>
</tr>
</thead>
<tbody>
<tr>
<td>DO</td>
<td>DOM</td>
<td>214</td>
<td>DZ</td>
<td>DZA</td>
<td>012</td>
</tr>
<tr>
<td>EC</td>
<td>ECU</td>
<td>218</td>
<td>EE</td>
<td>EST</td>
<td>233</td>
</tr>
<tr>
<td>EG</td>
<td>EGY</td>
<td>818</td>
<td>EH</td>
<td>ESH</td>
<td>732</td>
</tr>
<tr>
<td>ER</td>
<td>ERI</td>
<td>232</td>
<td>ES</td>
<td>ESP</td>
<td>724</td>
</tr>
<tr>
<td>ET</td>
<td>ETH</td>
<td>231</td>
<td>FI</td>
<td>FIN</td>
<td>246</td>
</tr>
<tr>
<td>FJ</td>
<td>FJI</td>
<td>242</td>
<td>FK</td>
<td>FLK</td>
<td>238</td>
</tr>
<tr>
<td>FM</td>
<td>FSM</td>
<td>583</td>
<td>FO</td>
<td>FRO</td>
<td>234</td>
</tr>
<tr>
<td>FR</td>
<td>FRA</td>
<td>250</td>
<td>GA</td>
<td>GAB</td>
<td>266</td>
</tr>
<tr>
<td>GB</td>
<td>GBR</td>
<td>826</td>
<td>GD</td>
<td>GRD</td>
<td>308</td>
</tr>
<tr>
<td>GE</td>
<td>GEO</td>
<td>268</td>
<td>GF</td>
<td>GUF</td>
<td>254</td>
</tr>
<tr>
<td>GG</td>
<td>GGY</td>
<td>831</td>
<td>GH</td>
<td>GHA</td>
<td>288</td>
</tr>
<tr>
<td>GI</td>
<td>GIB</td>
<td>292</td>
<td>GL</td>
<td>GRL</td>
<td>304</td>
</tr>
<tr>
<td>GM</td>
<td>GMB</td>
<td>270</td>
<td>GN</td>
<td>GIN</td>
<td>324</td>
</tr>
<tr>
<td>GP</td>
<td>GLP</td>
<td>312</td>
<td>GQ</td>
<td>GNQ</td>
<td>226</td>
</tr>
<tr>
<td>GR</td>
<td>GRC</td>
<td>300</td>
<td>GS</td>
<td>SGS</td>
<td>239</td>
</tr>
<tr>
<td>GT</td>
<td>GTM</td>
<td>320</td>
<td>GU</td>
<td>GUM</td>
<td>316</td>
</tr>
<tr>
<td>GW</td>
<td>GNB</td>
<td>624</td>
<td>GY</td>
<td>GUY</td>
<td>328</td>
</tr>
<tr>
<td>HK</td>
<td>HKG</td>
<td>344</td>
<td>HM</td>
<td>HMD</td>
<td>334</td>
</tr>
<tr>
<td>HN</td>
<td>HND</td>
<td>340</td>
<td>HR</td>
<td>HRV</td>
<td>191</td>
</tr>
<tr>
<td>HT</td>
<td>HTI</td>
<td>332</td>
<td>HU</td>
<td>HUN</td>
<td>348</td>
</tr>
<tr>
<td>ID</td>
<td>IDN</td>
<td>360</td>
<td>IE</td>
<td>IRL</td>
<td>372</td>
</tr>
<tr>
<td>IL</td>
<td>ISR</td>
<td>376</td>
<td>IM</td>
<td>IMN</td>
<td>833</td>
</tr>
<tr>
<td>IN</td>
<td>IND</td>
<td>356</td>
<td>IO</td>
<td>IOT</td>
<td>086</td>
</tr>
<tr>
<td>IQ</td>
<td>IRQ</td>
<td>368</td>
<td>IR</td>
<td>IRN</td>
<td>364</td>
</tr>
<tr>
<td>IS</td>
<td>ISL</td>
<td>352</td>
<td>IT</td>
<td>ITA</td>
<td>380</td>
</tr>
<tr>
<td>JE</td>
<td>JEF</td>
<td>832</td>
<td>JM</td>
<td>JAM</td>
<td>388</td>
</tr>
<tr>
<td>JO</td>
<td>JOR</td>
<td>400</td>
<td>JP</td>
<td>JPN</td>
<td>392</td>
</tr>
<tr>
<td>KE</td>
<td>KEN</td>
<td>404</td>
<td>KG</td>
<td>KGZ</td>
<td>417</td>
</tr>
<tr>
<td>KH</td>
<td>KHM</td>
<td>116</td>
<td>KI</td>
<td>KIR</td>
<td>296</td>
</tr>
<tr>
<td>KM</td>
<td>COM</td>
<td>174</td>
<td>KN</td>
<td>KNA</td>
<td>659</td>
</tr>
<tr>
<td>KP</td>
<td>PRK</td>
<td>408</td>
<td>KR</td>
<td>KOR</td>
<td>410</td>
</tr>
<tr>
<td>KW</td>
<td>KWT</td>
<td>414</td>
<td>KY</td>
<td>CYM</td>
<td>136</td>
</tr>
<tr>
<td>KZ</td>
<td>KAZ</td>
<td>398</td>
<td>LA</td>
<td>LAO</td>
<td>418</td>
</tr>
<tr>
<td>LB</td>
<td>LBN</td>
<td>422</td>
<td>LC</td>
<td>LCA</td>
<td>662</td>
</tr>
<tr>
<td>LI</td>
<td>LIE</td>
<td>438</td>
<td>LK</td>
<td>LKA</td>
<td>144</td>
</tr>
<tr>
<td>LR</td>
<td>LBR</td>
<td>430</td>
<td>LS</td>
<td>LSO</td>
<td>426</td>
</tr>
<tr>
<td>LT</td>
<td>LTU</td>
<td>440</td>
<td>LU</td>
<td>LUX</td>
<td>442</td>
</tr>
<tr>
<td>LV</td>
<td>LVA</td>
<td>428</td>
<td>LY</td>
<td>LBY</td>
<td>434</td>
</tr>
<tr>
<td>MA</td>
<td>MAR</td>
<td>504</td>
<td>MC</td>
<td>MCO</td>
<td>492</td>
</tr>
<tr>
<td>MD</td>
<td>MDA</td>
<td>498</td>
<td>ME</td>
<td>MNE</td>
<td>499</td>
</tr>
</tbody>
</table>
### A. Region and Script Mappings

Table A.1.: Region Mappings (Continued)

<table>
<thead>
<tr>
<th>Alpha-2</th>
<th>Alpha-3</th>
<th>Numeric</th>
<th>Alpha-2</th>
<th>Alpha-3</th>
<th>Numeric</th>
</tr>
</thead>
<tbody>
<tr>
<td>MF</td>
<td>MAF</td>
<td>663</td>
<td>MG</td>
<td>MDG</td>
<td>450</td>
</tr>
<tr>
<td>MH</td>
<td>MHL</td>
<td>584</td>
<td>MK</td>
<td>MKD</td>
<td>807</td>
</tr>
<tr>
<td>ML</td>
<td>MLI</td>
<td>466</td>
<td>MM</td>
<td>MMR</td>
<td>104</td>
</tr>
<tr>
<td>MN</td>
<td>MNG</td>
<td>496</td>
<td>MO</td>
<td>MAC</td>
<td>446</td>
</tr>
<tr>
<td>MP</td>
<td>MNP</td>
<td>580</td>
<td>MQ</td>
<td>MTQ</td>
<td>474</td>
</tr>
<tr>
<td>MR</td>
<td>MRT</td>
<td>478</td>
<td>MS</td>
<td>MSR</td>
<td>500</td>
</tr>
<tr>
<td>MT</td>
<td>MLT</td>
<td>470</td>
<td>MU</td>
<td>MUS</td>
<td>480</td>
</tr>
<tr>
<td>MV</td>
<td>MDV</td>
<td>462</td>
<td>MW</td>
<td>MWI</td>
<td>454</td>
</tr>
<tr>
<td>MX</td>
<td>MEX</td>
<td>484</td>
<td>MY</td>
<td>MYS</td>
<td>458</td>
</tr>
<tr>
<td>MZ</td>
<td>MOZ</td>
<td>508</td>
<td>NA</td>
<td>NAM</td>
<td>516</td>
</tr>
<tr>
<td>NC</td>
<td>NCL</td>
<td>540</td>
<td>NE</td>
<td>NER</td>
<td>562</td>
</tr>
<tr>
<td>NF</td>
<td>NFK</td>
<td>574</td>
<td>NG</td>
<td>NGA</td>
<td>566</td>
</tr>
<tr>
<td>NI</td>
<td>NIC</td>
<td>558</td>
<td>NL</td>
<td>NLD</td>
<td>528</td>
</tr>
<tr>
<td>NO</td>
<td>NOR</td>
<td>578</td>
<td>NP</td>
<td>NPL</td>
<td>524</td>
</tr>
<tr>
<td>NR</td>
<td>NRU</td>
<td>520</td>
<td>NU</td>
<td>NIU</td>
<td>570</td>
</tr>
<tr>
<td>NZ</td>
<td>NZL</td>
<td>554</td>
<td>OM</td>
<td>OMN</td>
<td>512</td>
</tr>
<tr>
<td>PA</td>
<td>PAN</td>
<td>591</td>
<td>PE</td>
<td>PER</td>
<td>604</td>
</tr>
<tr>
<td>PF</td>
<td>PYF</td>
<td>258</td>
<td>PG</td>
<td>PNG</td>
<td>598</td>
</tr>
<tr>
<td>PH</td>
<td>PHL</td>
<td>608</td>
<td>PK</td>
<td>PAK</td>
<td>586</td>
</tr>
<tr>
<td>PL</td>
<td>POL</td>
<td>616</td>
<td>PM</td>
<td>SPM</td>
<td>666</td>
</tr>
<tr>
<td>PN</td>
<td>PCN</td>
<td>612</td>
<td>PR</td>
<td>PRI</td>
<td>630</td>
</tr>
<tr>
<td>PS</td>
<td>PSE</td>
<td>275</td>
<td>PT</td>
<td>PRT</td>
<td>620</td>
</tr>
<tr>
<td>PW</td>
<td>PLW</td>
<td>585</td>
<td>PY</td>
<td>PRY</td>
<td>600</td>
</tr>
<tr>
<td>QA</td>
<td>QAT</td>
<td>634</td>
<td>RE</td>
<td>REU</td>
<td>638</td>
</tr>
<tr>
<td>RO</td>
<td>ROU</td>
<td>642</td>
<td>RS</td>
<td>SRB</td>
<td>688</td>
</tr>
<tr>
<td>RU</td>
<td>RUS</td>
<td>643</td>
<td>RW</td>
<td>RWA</td>
<td>646</td>
</tr>
<tr>
<td>SA</td>
<td>SAU</td>
<td>682</td>
<td>SB</td>
<td>SLB</td>
<td>090</td>
</tr>
<tr>
<td>SC</td>
<td>SYC</td>
<td>690</td>
<td>SD</td>
<td>SDN</td>
<td>729</td>
</tr>
<tr>
<td>SE</td>
<td>SWE</td>
<td>752</td>
<td>SG</td>
<td>SGP</td>
<td>702</td>
</tr>
<tr>
<td>SH</td>
<td>SHN</td>
<td>654</td>
<td>SI</td>
<td>SVN</td>
<td>705</td>
</tr>
<tr>
<td>SJ</td>
<td>SJM</td>
<td>744</td>
<td>SK</td>
<td>SVK</td>
<td>703</td>
</tr>
<tr>
<td>SL</td>
<td>SLE</td>
<td>694</td>
<td>SM</td>
<td>SMR</td>
<td>674</td>
</tr>
<tr>
<td>SN</td>
<td>SEN</td>
<td>686</td>
<td>SO</td>
<td>SOM</td>
<td>706</td>
</tr>
<tr>
<td>SR</td>
<td>SUR</td>
<td>740</td>
<td>SS</td>
<td>SSD</td>
<td>728</td>
</tr>
<tr>
<td>ST</td>
<td>STP</td>
<td>678</td>
<td>SV</td>
<td>SLV</td>
<td>222</td>
</tr>
<tr>
<td>SX</td>
<td>SXM</td>
<td>534</td>
<td>SY</td>
<td>SYR</td>
<td>760</td>
</tr>
<tr>
<td>SZ</td>
<td>SWZ</td>
<td>748</td>
<td>TC</td>
<td>TCA</td>
<td>796</td>
</tr>
<tr>
<td>TD</td>
<td>TCD</td>
<td>148</td>
<td>TF</td>
<td>ATF</td>
<td>260</td>
</tr>
<tr>
<td>TG</td>
<td>TGO</td>
<td>768</td>
<td>TH</td>
<td>THA</td>
<td>764</td>
</tr>
<tr>
<td>TJ</td>
<td>TJK</td>
<td>762</td>
<td>TK</td>
<td>TKL</td>
<td>772</td>
</tr>
</tbody>
</table>
## A. Region and Script Mappings

Table A.1.: Region Mappings (Continued)

<table>
<thead>
<tr>
<th>Alpha-2</th>
<th>Alpha-3</th>
<th>Numeric</th>
<th>Alpha-2</th>
<th>Alpha-3</th>
<th>Numeric</th>
</tr>
</thead>
<tbody>
<tr>
<td>TL</td>
<td>TLS</td>
<td>626</td>
<td>TM</td>
<td>TKM</td>
<td>795</td>
</tr>
<tr>
<td>TN</td>
<td>TUN</td>
<td>788</td>
<td>TO</td>
<td>TON</td>
<td>776</td>
</tr>
<tr>
<td>TR</td>
<td>TUR</td>
<td>792</td>
<td>TT</td>
<td>TTO</td>
<td>780</td>
</tr>
<tr>
<td>TV</td>
<td>TUV</td>
<td>798</td>
<td>TW</td>
<td>TWN</td>
<td>158</td>
</tr>
<tr>
<td>TZ</td>
<td>TZA</td>
<td>834</td>
<td>UA</td>
<td>UKR</td>
<td>804</td>
</tr>
<tr>
<td>UD</td>
<td>UGA</td>
<td>800</td>
<td>UM</td>
<td>UMI</td>
<td>581</td>
</tr>
<tr>
<td>US</td>
<td>USA</td>
<td>840</td>
<td>UY</td>
<td>URY</td>
<td>858</td>
</tr>
<tr>
<td>UZ</td>
<td>UZB</td>
<td>860</td>
<td>VA</td>
<td>VAT</td>
<td>336</td>
</tr>
<tr>
<td>VC</td>
<td>VCT</td>
<td>670</td>
<td>VE</td>
<td>VEN</td>
<td>862</td>
</tr>
<tr>
<td>VG</td>
<td>VGB</td>
<td>092</td>
<td>VI</td>
<td>VIR</td>
<td>850</td>
</tr>
<tr>
<td>VN</td>
<td>VNM</td>
<td>704</td>
<td>VU</td>
<td>VUT</td>
<td>548</td>
</tr>
<tr>
<td>WF</td>
<td>WLF</td>
<td>876</td>
<td>WS</td>
<td>WSM</td>
<td>882</td>
</tr>
<tr>
<td>YE</td>
<td>YEM</td>
<td>887</td>
<td>YT</td>
<td>MYT</td>
<td>175</td>
</tr>
<tr>
<td>ZA</td>
<td>ZAF</td>
<td>710</td>
<td>ZM</td>
<td>ZMB</td>
<td>894</td>
</tr>
<tr>
<td>ZW</td>
<td>ZWE</td>
<td>716</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### A. Region and Script Mappings

Table A.2.: Script Mappings

<table>
<thead>
<tr>
<th>Alpha-2</th>
<th>Numeric</th>
<th>Direction</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adlm</td>
<td>166</td>
<td>RL</td>
<td>Adlam.</td>
</tr>
<tr>
<td>Afak</td>
<td>439</td>
<td>varies</td>
<td>Afaka.</td>
</tr>
<tr>
<td>Aghb</td>
<td>239</td>
<td>LR</td>
<td>Caucasian Albanian.</td>
</tr>
<tr>
<td>Ahom</td>
<td>338</td>
<td>LR</td>
<td>Ahom, Tai Ahom.</td>
</tr>
<tr>
<td>Arab</td>
<td>160</td>
<td>RL</td>
<td>Arabic.</td>
</tr>
<tr>
<td>Aram</td>
<td>161</td>
<td>RL</td>
<td>Arabic (Nastaliq variant).</td>
</tr>
<tr>
<td>Armi</td>
<td>124</td>
<td>RL</td>
<td>Imperial Aramaic.</td>
</tr>
<tr>
<td>Armn</td>
<td>230</td>
<td>LR</td>
<td>Armenian.</td>
</tr>
<tr>
<td>Avst</td>
<td>134</td>
<td>RL</td>
<td>Avestan.</td>
</tr>
<tr>
<td>Bali</td>
<td>360</td>
<td>LR</td>
<td>Balinese.</td>
</tr>
<tr>
<td>Bamu</td>
<td>435</td>
<td>LR</td>
<td>Bamum.</td>
</tr>
<tr>
<td>Bass</td>
<td>259</td>
<td>LR</td>
<td>Bassa Vah.</td>
</tr>
<tr>
<td>Batk</td>
<td>365</td>
<td>LR</td>
<td>Batak.</td>
</tr>
<tr>
<td>Beng</td>
<td>334</td>
<td>LR</td>
<td>Bhaiksuki.</td>
</tr>
<tr>
<td>Blis</td>
<td>550</td>
<td>varies</td>
<td>Blissymbols.</td>
</tr>
<tr>
<td>Bopo</td>
<td>285</td>
<td>LR</td>
<td>Bopomofo.</td>
</tr>
<tr>
<td>Brah</td>
<td>300</td>
<td>LR</td>
<td>Brahmi.</td>
</tr>
<tr>
<td>Brai</td>
<td>570</td>
<td>LR</td>
<td>Braille.</td>
</tr>
<tr>
<td>Bugi</td>
<td>367</td>
<td>LR</td>
<td>Buginese.</td>
</tr>
<tr>
<td>Buhd</td>
<td>372</td>
<td>LR</td>
<td>Buhid.</td>
</tr>
<tr>
<td>Cakm</td>
<td>349</td>
<td>LR</td>
<td>Chakma.</td>
</tr>
<tr>
<td>Cans</td>
<td>440</td>
<td>LR</td>
<td>Unified Canadian Aboriginal Syllabics.</td>
</tr>
<tr>
<td>Cari</td>
<td>201</td>
<td>LR</td>
<td>Carian.</td>
</tr>
<tr>
<td>Cham</td>
<td>358</td>
<td>LR</td>
<td>Cham.</td>
</tr>
<tr>
<td>Cher</td>
<td>445</td>
<td>LR</td>
<td>Cherokee.</td>
</tr>
<tr>
<td>Cirt</td>
<td>291</td>
<td>varies</td>
<td>Cirth.</td>
</tr>
<tr>
<td>Copt</td>
<td>204</td>
<td>LR</td>
<td>Coptic.</td>
</tr>
<tr>
<td>Cprt</td>
<td>403</td>
<td>RL</td>
<td>Cypriot.</td>
</tr>
<tr>
<td>Cyril</td>
<td>220</td>
<td>LR</td>
<td>Cyrillic.</td>
</tr>
<tr>
<td>Cyrs</td>
<td>221</td>
<td>varies</td>
<td>Cyrillic (Old Church Slavonic variant).</td>
</tr>
<tr>
<td>Deva</td>
<td>315</td>
<td>LR</td>
<td>Devanagari (Nagari).</td>
</tr>
<tr>
<td>Dsrt</td>
<td>250</td>
<td>LR</td>
<td>Deseret (Mormon).</td>
</tr>
<tr>
<td>Dupl</td>
<td>755</td>
<td>LR</td>
<td>Duployan shorthand, Duployan stenography.</td>
</tr>
<tr>
<td>Egyd</td>
<td>070</td>
<td>RL</td>
<td>Egyptian demotic.</td>
</tr>
<tr>
<td>Eghy</td>
<td>060</td>
<td>RL</td>
<td>Egyptian hieratic.</td>
</tr>
<tr>
<td>Egyp</td>
<td>050</td>
<td>LR</td>
<td>Egyptian hieroglyphs.</td>
</tr>
<tr>
<td>Elba</td>
<td>226</td>
<td>LR</td>
<td>Elbasan.</td>
</tr>
<tr>
<td>Ethi</td>
<td>430</td>
<td>LR</td>
<td>Ethiopic (Ge’ez).</td>
</tr>
<tr>
<td>Geok</td>
<td>241</td>
<td>LR</td>
<td>Khutsuri (Asomtavruli and Nuskhuri).</td>
</tr>
<tr>
<td>Geor</td>
<td>240</td>
<td>LR</td>
<td>Georgian (Mkhedruli).</td>
</tr>
</tbody>
</table>
## A. Region and Script Mappings

Table A.2.: Script Mappings (Continued)

<table>
<thead>
<tr>
<th>Alpha-2</th>
<th>Numeric</th>
<th>Direction</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Glag</td>
<td>225</td>
<td>LR</td>
<td>Glagolitic.</td>
</tr>
<tr>
<td>Goth</td>
<td>206</td>
<td>LR</td>
<td>Gothic.</td>
</tr>
<tr>
<td>Gran</td>
<td>343</td>
<td>LR</td>
<td>Grantha.</td>
</tr>
<tr>
<td>Grek</td>
<td>200</td>
<td>LR</td>
<td>Greek.</td>
</tr>
<tr>
<td>Gujr</td>
<td>320</td>
<td>LR</td>
<td>Gujarati.</td>
</tr>
<tr>
<td>Guru</td>
<td>310</td>
<td>LR</td>
<td>Gurmukhi.</td>
</tr>
<tr>
<td>Hanb</td>
<td>503</td>
<td>LR</td>
<td>Han with Bopomofo (alias for Han + Bopomofo).</td>
</tr>
<tr>
<td>Hang</td>
<td>286</td>
<td>LR</td>
<td>Hangul.</td>
</tr>
<tr>
<td>Han1</td>
<td>500</td>
<td>LR</td>
<td>Han (Hanzi, Kanji, Hanja).</td>
</tr>
<tr>
<td>Hano</td>
<td>371</td>
<td>LR</td>
<td>Hanunoo.</td>
</tr>
<tr>
<td>Hans</td>
<td>501</td>
<td>varies</td>
<td>Han (Simplified variant).</td>
</tr>
<tr>
<td>Hant</td>
<td>502</td>
<td>varies</td>
<td>Han (Traditional variant).</td>
</tr>
<tr>
<td>Hatr</td>
<td>127</td>
<td>RL</td>
<td>Hatran.</td>
</tr>
<tr>
<td>Hebr</td>
<td>125</td>
<td>RL</td>
<td>Hebrew.</td>
</tr>
<tr>
<td>Hira</td>
<td>410</td>
<td>LR</td>
<td>Hiragana.</td>
</tr>
<tr>
<td>Hluw</td>
<td>080</td>
<td>LR</td>
<td>Anatolian Hieroglyphs (Luwian Hieroglyphs, Hittite Hieroglyphs).</td>
</tr>
<tr>
<td>Hmng</td>
<td>450</td>
<td>LR</td>
<td>Pahawh Hmong.</td>
</tr>
<tr>
<td>Hrkt</td>
<td>412</td>
<td>varies</td>
<td>Japanese syllabaries (alias for Hiragana + Katakana).</td>
</tr>
<tr>
<td>Hung</td>
<td>176</td>
<td>RL</td>
<td>Old Hungarian (Hungarian Runic).</td>
</tr>
<tr>
<td>Inds</td>
<td>610</td>
<td>RL</td>
<td>Indus (Harappan).</td>
</tr>
<tr>
<td>Ital</td>
<td>210</td>
<td>LR</td>
<td>Old Italic (Etruscan, Oscan, etc.)</td>
</tr>
<tr>
<td>Jamo</td>
<td>284</td>
<td>LR</td>
<td>Jamo (alias for Jamo subset of Hangul).</td>
</tr>
<tr>
<td>Java</td>
<td>361</td>
<td>LR</td>
<td>Javanese.</td>
</tr>
<tr>
<td>Jpan</td>
<td>413</td>
<td>varies</td>
<td>Japanese (alias for Han + Hiragana + Katakana).</td>
</tr>
<tr>
<td>Jurc</td>
<td>510</td>
<td>LR</td>
<td>Jurchen.</td>
</tr>
<tr>
<td>Kali</td>
<td>357</td>
<td>LR</td>
<td>Kayah Li.</td>
</tr>
<tr>
<td>Kana</td>
<td>411</td>
<td>LR</td>
<td>Katakana.</td>
</tr>
<tr>
<td>Khar</td>
<td>305</td>
<td>RL</td>
<td>Kharoshthi.</td>
</tr>
<tr>
<td>Khmr</td>
<td>355</td>
<td>LR</td>
<td>Khmer.</td>
</tr>
<tr>
<td>Khoj</td>
<td>322</td>
<td>LR</td>
<td>Khojki.</td>
</tr>
<tr>
<td>Kttl</td>
<td>505</td>
<td>LR</td>
<td>Khitan large script.</td>
</tr>
<tr>
<td>Kts</td>
<td>288</td>
<td>TB</td>
<td>Khitan small script.</td>
</tr>
<tr>
<td>Knda</td>
<td>345</td>
<td>LR</td>
<td>Kannada.</td>
</tr>
<tr>
<td>Kore</td>
<td>287</td>
<td>LR</td>
<td>Korean (alias for Hangul + Han).</td>
</tr>
<tr>
<td>Kpel</td>
<td>436</td>
<td>LR</td>
<td>Kpelle.</td>
</tr>
<tr>
<td>Kthi</td>
<td>317</td>
<td>LR</td>
<td>Kaithi.</td>
</tr>
<tr>
<td>Lana</td>
<td>351</td>
<td>LR</td>
<td>Tai Tham (Lanna).</td>
</tr>
<tr>
<td>Lao</td>
<td>356</td>
<td>LR</td>
<td>Lao.</td>
</tr>
<tr>
<td>Latf</td>
<td>217</td>
<td>varies</td>
<td>Latin (Fraktur variant).</td>
</tr>
</tbody>
</table>
### A. Region and Script Mappings

#### Table A.2.: Script Mappings (Continued)

<table>
<thead>
<tr>
<th>Alpha-2</th>
<th>Numeric</th>
<th>Direction</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Latg</td>
<td>216</td>
<td>LR</td>
<td>Latin (Gaelic variant).</td>
</tr>
<tr>
<td>Latn</td>
<td>215</td>
<td>LR</td>
<td>Latin.</td>
</tr>
<tr>
<td>Leke</td>
<td>364</td>
<td>LR</td>
<td>Leke.</td>
</tr>
<tr>
<td>Lepc</td>
<td>335</td>
<td>LR</td>
<td>Lepcha.</td>
</tr>
<tr>
<td>Limb</td>
<td>336</td>
<td>LR</td>
<td>Limbu.</td>
</tr>
<tr>
<td>Lina</td>
<td>400</td>
<td>LR</td>
<td>Linear A.</td>
</tr>
<tr>
<td>Linb</td>
<td>401</td>
<td>LR</td>
<td>Linear B.</td>
</tr>
<tr>
<td>Lisu</td>
<td>399</td>
<td>LR</td>
<td>Lisu (Fraser).</td>
</tr>
<tr>
<td>Loma</td>
<td>437</td>
<td>LR</td>
<td>Loma.</td>
</tr>
<tr>
<td>Lyci</td>
<td>202</td>
<td>LR</td>
<td>Lycian.</td>
</tr>
<tr>
<td>Lydi</td>
<td>116</td>
<td>RL</td>
<td>Lydian.</td>
</tr>
<tr>
<td>Mahj</td>
<td>314</td>
<td>LR</td>
<td>Mahajani.</td>
</tr>
<tr>
<td>Mand</td>
<td>140</td>
<td>RL</td>
<td>Mandaic, Mandaean.</td>
</tr>
<tr>
<td>Mani</td>
<td>139</td>
<td>RL</td>
<td>Manichaean.</td>
</tr>
<tr>
<td>Marc</td>
<td>332</td>
<td>LR</td>
<td>Marchen.</td>
</tr>
<tr>
<td>Maya</td>
<td>090</td>
<td>varies</td>
<td>Mayan hieroglyphs.</td>
</tr>
<tr>
<td>Mend</td>
<td>438</td>
<td>RL</td>
<td>Mende Kikakui.</td>
</tr>
<tr>
<td>Merc</td>
<td>101</td>
<td>RL</td>
<td>Meroitic Cursive.</td>
</tr>
<tr>
<td>Mero</td>
<td>100</td>
<td>RL</td>
<td>Meroitic Hieroglyphs.</td>
</tr>
<tr>
<td>Mlym</td>
<td>347</td>
<td>LR</td>
<td>Malayalam.</td>
</tr>
<tr>
<td>Modi</td>
<td>324</td>
<td>LR</td>
<td>Modi.</td>
</tr>
<tr>
<td>Mong</td>
<td>145</td>
<td>TB</td>
<td>Mongolian.</td>
</tr>
<tr>
<td>Moon</td>
<td>218</td>
<td>varies</td>
<td>Moon (Moon code, Moon script, Moon type).</td>
</tr>
<tr>
<td>Mroo</td>
<td>199</td>
<td>LR</td>
<td>Mro, Mru.</td>
</tr>
<tr>
<td>Mtei</td>
<td>337</td>
<td>LR</td>
<td>Meitei Mayek (Meithei, Meetei).</td>
</tr>
<tr>
<td>Mult</td>
<td>323</td>
<td>LR</td>
<td>Multani.</td>
</tr>
<tr>
<td>Mymr</td>
<td>350</td>
<td>LR</td>
<td>Myanmar (Burmese).</td>
</tr>
<tr>
<td>Nbat</td>
<td>159</td>
<td>RL</td>
<td>Nabataean.</td>
</tr>
<tr>
<td>Newa</td>
<td>333</td>
<td>LR</td>
<td>Newa, Newar, Newari.</td>
</tr>
<tr>
<td>Nkgb</td>
<td>420</td>
<td>LR</td>
<td>Nakhi Geba.</td>
</tr>
<tr>
<td>Nkoo</td>
<td>165</td>
<td>RL</td>
<td>N’Ko.</td>
</tr>
<tr>
<td>Nshu</td>
<td>499</td>
<td>LR</td>
<td>Nushu.</td>
</tr>
<tr>
<td>Ogam</td>
<td>212</td>
<td>varies</td>
<td>Ogham.</td>
</tr>
<tr>
<td>Olck</td>
<td>261</td>
<td>LR</td>
<td>Ol Chiki.</td>
</tr>
<tr>
<td>Orkh</td>
<td>175</td>
<td>RL</td>
<td>Old Turkic, Orkhon Runic.</td>
</tr>
<tr>
<td>Orya</td>
<td>327</td>
<td>LR</td>
<td>Oriya.</td>
</tr>
<tr>
<td>Osge</td>
<td>219</td>
<td>LR</td>
<td>Osage.</td>
</tr>
<tr>
<td>Osma</td>
<td>260</td>
<td>LR</td>
<td>Osmanyia.</td>
</tr>
<tr>
<td>Palm</td>
<td>126</td>
<td>RL</td>
<td>Palmyrene.</td>
</tr>
</tbody>
</table>
### A. Region and Script Mappings

**Table A.2.: Script Mappings (Continued)**

<table>
<thead>
<tr>
<th>Alpha-2</th>
<th>Numeric</th>
<th>Direction</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pauc</td>
<td>263</td>
<td>LR</td>
<td>Pau Cin Hau.</td>
</tr>
<tr>
<td>Perm</td>
<td>227</td>
<td>LR</td>
<td>Old Permic.</td>
</tr>
<tr>
<td>Phag</td>
<td>331</td>
<td>TB</td>
<td>Phags-pa.</td>
</tr>
<tr>
<td>Phli</td>
<td>131</td>
<td>RL</td>
<td>Inscriptional Pahlavi.</td>
</tr>
<tr>
<td>Phlp</td>
<td>132</td>
<td>RL</td>
<td>Psalter Pahlavi.</td>
</tr>
<tr>
<td>Phlv</td>
<td>133</td>
<td>RL</td>
<td>Book Pahlavi.</td>
</tr>
<tr>
<td>Phnx</td>
<td>115</td>
<td>RL</td>
<td>Phoenician.</td>
</tr>
<tr>
<td>Piqd</td>
<td>293</td>
<td>LR</td>
<td>Klingon (KLI plqaD).</td>
</tr>
<tr>
<td>Plrd</td>
<td>282</td>
<td>LR</td>
<td>Miao (Pollard).</td>
</tr>
<tr>
<td>Prti</td>
<td>130</td>
<td>RL</td>
<td>Inscriptional Parthian.</td>
</tr>
<tr>
<td>Qaaa</td>
<td>900</td>
<td>varies</td>
<td>Reserved for private use (start).</td>
</tr>
<tr>
<td>Qaai</td>
<td>908</td>
<td>varies</td>
<td>Private use.</td>
</tr>
<tr>
<td>Qabx</td>
<td>949</td>
<td>varies</td>
<td>Reserved for private use (end).</td>
</tr>
<tr>
<td>Rjng</td>
<td>363</td>
<td>LR</td>
<td>Rejang (Redjang, Kaganga).</td>
</tr>
<tr>
<td>Roro</td>
<td>620</td>
<td>varies</td>
<td>Rongorongo.</td>
</tr>
<tr>
<td>Runr</td>
<td>211</td>
<td>LR</td>
<td>Runic.</td>
</tr>
<tr>
<td>Samr</td>
<td>123</td>
<td>RL</td>
<td>Samaritan.</td>
</tr>
<tr>
<td>Sara</td>
<td>292</td>
<td>varies</td>
<td>Sarati.</td>
</tr>
<tr>
<td>Sarb</td>
<td>105</td>
<td>RL</td>
<td>Old South Arabian.</td>
</tr>
<tr>
<td>Saur</td>
<td>344</td>
<td>LR</td>
<td>Saurashtra.</td>
</tr>
<tr>
<td>Sgnw</td>
<td>095</td>
<td>TB</td>
<td>SignWriting.</td>
</tr>
<tr>
<td>Shaw</td>
<td>281</td>
<td>LR</td>
<td>Shavian (Shaw).</td>
</tr>
<tr>
<td>Shrd</td>
<td>319</td>
<td>LR</td>
<td>Sharada.</td>
</tr>
<tr>
<td>Sidd</td>
<td>302</td>
<td>LR</td>
<td>Siddham.</td>
</tr>
<tr>
<td>Sind</td>
<td>318</td>
<td>LR</td>
<td>Khudawadi, Sindhi.</td>
</tr>
<tr>
<td>Sinh</td>
<td>348</td>
<td>LR</td>
<td>Sinhala.</td>
</tr>
<tr>
<td>Sora</td>
<td>398</td>
<td>LR</td>
<td>Sora Sompeng.</td>
</tr>
<tr>
<td>Sund</td>
<td>362</td>
<td>LR</td>
<td>Sundanese.</td>
</tr>
<tr>
<td>Sylo</td>
<td>316</td>
<td>LR</td>
<td>Syloti Nagri.</td>
</tr>
<tr>
<td>Syrc</td>
<td>135</td>
<td>RL</td>
<td>Syriac.</td>
</tr>
<tr>
<td>Syre</td>
<td>138</td>
<td>RL</td>
<td>Syriac (Estrangelo variant).</td>
</tr>
<tr>
<td>Syrj</td>
<td>137</td>
<td>RL</td>
<td>Syriac (Western variant).</td>
</tr>
<tr>
<td>Syrn</td>
<td>136</td>
<td>RL</td>
<td>Syriac (Eastern variant).</td>
</tr>
<tr>
<td>Tagb</td>
<td>373</td>
<td>LR</td>
<td>Tagbanwa.</td>
</tr>
<tr>
<td>Takr</td>
<td>321</td>
<td>LR</td>
<td>Takri.</td>
</tr>
<tr>
<td>Tale</td>
<td>353</td>
<td>LR</td>
<td>Tai Le.</td>
</tr>
<tr>
<td>Talu</td>
<td>354</td>
<td>LR</td>
<td>New Tai Lue.</td>
</tr>
<tr>
<td>Tamil</td>
<td>346</td>
<td>LR</td>
<td>Tamil.</td>
</tr>
<tr>
<td>Tang</td>
<td>520</td>
<td>LR</td>
<td>Tangut.</td>
</tr>
<tr>
<td>Tavt</td>
<td>359</td>
<td>LR</td>
<td>Tai Viet.</td>
</tr>
</tbody>
</table>
A. Region and Script Mappings

Table A.2.: Script Mappings (Continued)

<table>
<thead>
<tr>
<th>Alpha-2</th>
<th>Numeric</th>
<th>Direction</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Telu</td>
<td>340</td>
<td>LR</td>
<td>Telugu.</td>
</tr>
<tr>
<td>Teng</td>
<td>290</td>
<td>LR</td>
<td>Tengwar.</td>
</tr>
<tr>
<td>Tfng</td>
<td>120</td>
<td>LR</td>
<td>Tifinagh (Berber).</td>
</tr>
<tr>
<td>Tglg</td>
<td>370</td>
<td>LR</td>
<td>Tagalog (Baybayin, Alibata).</td>
</tr>
<tr>
<td>Thaa</td>
<td>170</td>
<td>RL</td>
<td>Thaana.</td>
</tr>
<tr>
<td>Thai</td>
<td>352</td>
<td>LR</td>
<td>Thai.</td>
</tr>
<tr>
<td>Tibt</td>
<td>330</td>
<td>LR</td>
<td>Tibetan.</td>
</tr>
<tr>
<td>Tirh</td>
<td>326</td>
<td>LR</td>
<td>Tirhuta.</td>
</tr>
<tr>
<td>Ugar</td>
<td>040</td>
<td>LR</td>
<td>Ugaritic.</td>
</tr>
<tr>
<td>Vaii</td>
<td>470</td>
<td>LR</td>
<td>Vai.</td>
</tr>
<tr>
<td>Visp</td>
<td>280</td>
<td>LR</td>
<td>Visible Speech.</td>
</tr>
<tr>
<td>Wara</td>
<td>262</td>
<td>LR</td>
<td>Warang Citi (Varang Kshiti).</td>
</tr>
<tr>
<td>Wole</td>
<td>480</td>
<td>RL</td>
<td>Woleai.</td>
</tr>
<tr>
<td>Xpeo</td>
<td>030</td>
<td>LR</td>
<td>Old Persian.</td>
</tr>
<tr>
<td>Xsux</td>
<td>020</td>
<td>LR</td>
<td>Cuneiform, Sumero-Akkadian.</td>
</tr>
<tr>
<td>Yiii</td>
<td>460</td>
<td>LR</td>
<td>Yi.</td>
</tr>
<tr>
<td>Zinh</td>
<td>994</td>
<td>inherited</td>
<td>Inherited script.</td>
</tr>
<tr>
<td>Zmth</td>
<td>995</td>
<td>LR</td>
<td>Mathematical notation.</td>
</tr>
<tr>
<td>Zsys</td>
<td>993</td>
<td>varies</td>
<td>Symbols (emoji variant).</td>
</tr>
<tr>
<td>Zsym</td>
<td>996</td>
<td>varies</td>
<td>Symbols.</td>
</tr>
<tr>
<td>Zxxx</td>
<td>997</td>
<td>varies</td>
<td>Unwritten documents.</td>
</tr>
<tr>
<td>Zzyy</td>
<td>998</td>
<td>varies</td>
<td>Undetermined script.</td>
</tr>
<tr>
<td>Zzzz</td>
<td>999</td>
<td>varies</td>
<td>Uncoded script.</td>
</tr>
</tbody>
</table>
# Symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Tape" /></td>
<td>The syntax and usage of a command, environment or option etc.</td>
</tr>
<tr>
<td><img src="image" alt="Information" /></td>
<td>An important message.</td>
</tr>
<tr>
<td><img src="image" alt="Information" /></td>
<td>Prominent information.</td>
</tr>
<tr>
<td><img src="image" alt="LaTeX code" /></td>
<td>\LaTeX{} code to insert into your document.</td>
</tr>
<tr>
<td><img src="image" alt="PDF code" /></td>
<td>How the example code should appear in the PDF.</td>
</tr>
<tr>
<td><img src="image" alt="Command" /></td>
<td>A command-line application invocation that needs to be entered into a terminal or command prompt.</td>
</tr>
</tbody>
</table>
Glossary

Command-line interface (CLI)
An application that doesn’t have a graphical user interface. That is, an application that doesn’t have any windows, buttons or menus and can be run in a command prompt or terminal.¹

Shell escape
\LaTeX{} Has the ability to run CLI applications while it’s typesetting a document. Whilst this is a convenient way of using tools to help build the document, it’s a security risk. To help protect users from arbitrary — and potentially dangerous — code from being executed, \LaTeX{} has a restricted mode, where only trusted applications are allowed to run. This is usually the default mode, but your \LaTeX{} installation may be set up so that the shell escape is disabled by default. The unrestricted mode allows you to run any application from the shell escape. Take care about enabling this option. If you receive a document or package from an untrusted source, first run \LaTeX{} with the shell escape disabled or in restricted mode and search the log file for “runsystem” before using the unrestricted mode. Note that Lua\LaTeX{} additionally requires the shellesc package.

¹dickimaw-books.com/latex/novices/html/terminal.html
Command Summary

0

\@tracklang@declareoption{\textit{\texttt{dialect}}} \texttt{tracklang v1.1+} \textsection{3; 19}

Provided by \texttt{tracklang.sty} to declare \textit{\texttt{dialect}} as a package option that tracks \textit{\texttt{dialect}}. Provided by \texttt{tracklang.tex}, if not already defined, to ignore its argument.

\@tracklang@for\textit{\texttt{cs}}:=(\textit{\texttt{list}})\texttt{do\textit{\texttt{body}}} \texttt{tracklang.tex v1.0+} \textsection{5; 37}

As \LaTeX{}'s \texttt{\texttt{\at \for}}.

\@tracklang@prelangpkgcheck@hook \textsection{6; 72}

If defined before \texttt{tracklang.sty v1.3.8+} is loaded, this command will be done after package options have been processed but before the check for language packages, such as babel and polyglossia.

A

\AddTrackedCountryIsoCode{\textit{\texttt{language}}} \texttt{tracklang.tex v1.3+} \textsection{6.5; 76}

Adds the ISO 3166-1 code.

\AddTrackedDialect{\textit{\texttt{dialect label}}}{\textit{\texttt{root language label}}} \texttt{tracklang.tex v1.0+}

Tracks a dialect. This command defines \texttt{\texttt{\at \TrackLangLastTrackedDialect}} to provide a convenient way to reference the last dialect to be tracked.
Command Summary

\texttt{\textbackslash AddTrackedIsoLanguage\{\langle code type\rangle\}\{\langle code\rangle\}\{\langle language\rangle\}} \quad \text{tracklang.tex v1.0+}

Adds a mapping between the given ISO code and language name.

\texttt{\textbackslash AddTrackedLanguage\{\langle root language label\rangle\}} \quad \text{tracklang.tex v1.0+} \hspace{1cm} \S 6.5; 76

Shortcut for \texttt{\textbackslash AddTrackedDialect\{\langle root language label\rangle\}\{\langle root language label\rangle\}}.

\texttt{\textbackslash AddTrackedLanguageIsoCodes\{\langle root language label\rangle\}} \quad \text{tracklang.tex v1.3+} \hspace{1cm} \S 6.6; 78

Adds the ISO 639-1, 639-2 and 639-3 codes, which must have previously been declared using \texttt{\textbackslash TrackLangNewLanguage}.

\texttt{\textbackslash AnyTrackedLanguages\{\langle true\rangle\}\{\langle false\rangle\}} \quad \text{tracklang.tex v1.0+} \hspace{1cm} \S 5; 35

Expands to \texttt{\langle true\rangle} if there are any tracked languages, otherwise expands to \texttt{\langle false\rangle}.

\textbf{C}

\texttt{\textbackslash CurrentTrackedDialect} \quad \text{tracklang.tex v1.3+} \hspace{1cm} \S 5; 46

Defined by \texttt{\textbackslash SetCurrentTrackedDialect} to the dialect label, which may be the supplied \texttt{\langle dialect\rangle} label or the mapped label or, if \texttt{\langle dialect\rangle} is a root language label, the last tracked dialect for the given root language.

\texttt{\textbackslash CurrentTrackedDialectAdditional} \quad \text{tracklang.tex v1.3+} \hspace{1cm} \S 5; 46

Defined by \texttt{\textbackslash SetCurrentTrackedDialect} to the additional part associated with the dialect (may be empty).

\texttt{\textbackslash CurrentTrackedDialectModifier} \quad \text{tracklang.tex v1.3+} \hspace{1cm} \S 5; 46
Defined by \SetCurrentTrackedDialect to the associated modifier (may be empty).

\CurrentTrackedDialectScript \quad \text{tracklang.tex v1.3+} \quad §5; 47

Defined by \SetCurrentTrackedDialect to the script associated with the dialect, or to the default script for the language.

\CurrentTrackedDialectSubLang \quad \text{tracklang.tex v1.3+} \quad §5; 46

Defined by \SetCurrentTrackedDialect to the sub language associated with the dialect (may be empty).

\CurrentTrackedDialectVariant \quad \text{tracklang.tex v1.3+} \quad §5; 46

Defined by \SetCurrentTrackedDialect to the associated variant (may be empty).

\CurrentTrackedIsoCode \quad \text{tracklang.tex v1.3+} \quad §5; 46

Defined by \SetCurrentTrackedDialect to the ISO 639-1 or 639-2 or 639-3 language code (may be empty).

\CurrentTrackedLanguage \quad \text{tracklang.tex v1.3+} \quad §5; 46

Defined by \SetCurrentTrackedDialect to the associated root language label.

\CurrentTrackedLanguageTag \quad \text{tracklang.tex v1.3+} \quad §5; 47

Defined by \SetCurrentTrackedDialect to the language tag that identifies the dialect or und if no match.

\CurrentTrackedRegion \quad \text{tracklang.tex v1.3+} \quad §5; 46
Command Summary

Defined by \SetCurrentTrackedDialect to the ISO 3166-1 region code associated with the dialect (may be empty).

\CurrentTrackedTag \hspace{1cm} \texttt{tracklang.tex v1.0+} \hspace{1cm} \S 5; 45

Expands to the current tracked tag.

\ForEachTrackedDialect{\langle cs \rangle}{\langle body \rangle} \hspace{1cm} \texttt{tracklang.tex v1.0+} \hspace{1cm} \S 5; 37

Iterates through the list of tracked dialects. On each iteration \langle cs \rangle is set to the dialect tag and \langle body \rangle is performed.

\ForEachTrackedLanguage{\langle cs \rangle}{\langle body \rangle} \hspace{1cm} \texttt{tracklang.tex v1.0+} \hspace{1cm} \S 5; 37

Iterates through the list of tracked languages. On each iteration \langle cs \rangle is set to the language tag and \langle body \rangle is performed.

\GetTrackedDialectAdditional{\langle dialect \rangle} \hspace{1cm} \texttt{tracklang.tex v1.3+} \hspace{1cm} \S 5; 44

Expands to the extra information for \langle dialect \rangle.

\GetTrackedDialectFromLanguageTag{\langle tag \rangle}{\langle cs \rangle} \hspace{1cm} \texttt{tracklang.tex v1.3+} \hspace{1cm} \S 5; 36

Finds the tracked dialect that matches the given language tag and stores the dialect label in \langle cs \rangle. If no match found, \langle cs \rangle will be empty.

\GetTrackedDialectModifier{\langle dialect \rangle} \hspace{1cm} \texttt{tracklang.tex v1.3+} \hspace{1cm} \S 5; 40

98
Expands to the modifier for the given dialect.

\texttt{\textbackslash GetTrackedDialectScript\{}(\texttt{dialect})\}\texttt{\}} \quad \textit{tracklang.tex v1.3+} \quad §5; 42

Expands to the script for \texttt{\}dialect\}.\texttt{\}

\texttt{\textbackslash GetTrackedDialectSubLang\{}(\texttt{dialect})\}\texttt{\}} \quad \textit{tracklang.tex v1.3+} \quad §5; 44

Expands to the sub-language for \texttt{\}dialect\}.\texttt{\}

\texttt{\textbackslash GetTrackedDialectVariant\{}(\texttt{dialect})\}\texttt{\}} \quad \textit{tracklang.tex v1.3+} \quad §5; 42

Expands to the modifier for \texttt{\}dialect\}.\texttt{\}

\texttt{\textbackslash GetTrackedLanguageTag\{}(\texttt{dialect})\}\texttt{\}} \quad \textit{tracklang.tex v1.3+} \quad §5; 40

Gets the language tag for \texttt{\}dialect\}.\texttt{\}

\texttt{\textbackslash IfHasTrackedDialectAdditional\{}(\texttt{dialect})\}\{\texttt{\{}true\}\}\{\texttt{\}}false\} \quad \textit{tracklang.tex v1.3+} \quad §5; 44

Expands to \texttt{\true\} if there’s extra information for \texttt{\}dialect\}, otherwise expands to \texttt{\false\}.

\texttt{\textbackslash IfHasTrackedDialectModifier\{}(\texttt{dialect})\}\{\texttt{\{}true\}\}\{\texttt{\}}false\} \quad \textit{tracklang.tex v1.3+} \quad §5; 41

Expands to \texttt{\true\} if there’s a modifier for the given dialect, otherwise expands to \texttt{\false\}.
Command Summary

Expands to \langle true \rangle if there’s a script for \langle dialect \rangle, otherwise expands to \langle false \rangle.

\texttt{\IfHasTrackedDialectSubLang\{\langle dialect \rangle\}\{\langle true \rangle\}\{\langle false \rangle\}} \quad \text{tracklang.tex v1.3+} \quad \S 5; 44

Expands to \langle true \rangle if there’s a sub-language for \langle dialect \rangle, otherwise expands to \langle false \rangle.

\texttt{\IfHasTrackedDialectVariant\{\langle dialect \rangle\}\{\langle true \rangle\}\{\langle false \rangle\}} \quad \text{tracklang.tex v1.3+} \quad \S 5; 42

Expands to \langle true \rangle if there’s a modifier for \langle dialect \rangle, otherwise expands to \langle false \rangle.

\texttt{\IfTrackedDialect\{\langle dialect-label \rangle\}\{\langle true \rangle\}\{\langle false \rangle\}} \quad \text{tracklang.tex v1.0+} \quad \S 5; 38

Does \langle true \rangle if the dialect identified by \langle dialect-label \rangle has been tracked, otherwise does \langle false \rangle.

\texttt{\IfTrackedDialectIsScriptCs\{\langle dialect \rangle\}\{\langle cs \rangle\}\{\langle true \rangle\}\{\langle false \rangle\}} \quad \text{tracklang.tex v1.3+} \quad \S 5; 43

If the given tracked dialect has an associated script and that script code matches the replacement text for the control sequence \langle cs \rangle then do \langle true \rangle otherwise to \langle false \rangle. If the tracked dialect doesn’t have an associated script then the default script for the root language is tested.

\texttt{\IfTrackedIsoCode\{\langle code type \rangle\}\{\langle code \rangle\}\{\langle true \rangle\}\{\langle false \rangle\}} \quad \text{tracklang.tex v1.0+} \quad \S 5; 39

Does \langle true \rangle if the given ISO code has been defined otherwise does \langle false \rangle.

\texttt{\IfTrackedLanguage\{\langle language-label \rangle\}\{\langle true \rangle\}\{\langle false \rangle\}} \quad \text{tracklang.tex v1.0+} \quad \S 5; 38

Does \langle true \rangle if the language identified by \langle language-label \rangle has been tracked, otherwise does \langle false \rangle.
Command Summary

\IfTrackedLanguageFileExists\{}\{}\{\texttt{true}\}\{\texttt{false}\}\}\\texttt{code}\}{\texttt{prefix}}\{\texttt{suffix}\}\texttt{tracklang.tex v1.0+}

Does \texttt{\SetCurrentTrackedDialect\{}\}\ and if the dialect is recognised, then determines if the file \texttt{\{}\texttt{prefix}\}\texttt{tag}\texttt{suffix}\ exists. If it does, \texttt{\CurrentTrackedTag} is set to \texttt{tag} and \texttt{true} is done, otherwise \texttt{false} is done.

\IfTrackedLanguageHasIsoCode\{}\\texttt{code type}\}{\texttt{label}}\{\texttt{true}\}{\texttt{false}}\texttt{tracklang.tex v1.0+}

Does \texttt{true} if the given language or dialect has a corresponding ISO code of the given type, otherwise does \texttt{false}.

\ifTrackLangShowInfo \texttt{true}\texttt{false}\texttt{fi} \texttt{initial: iftrue tracklang.tex v1.3+}

Conditional that indicates whether or not to show information messages.

\ifTrackLangShowVerbose \texttt{true}\texttt{false}\texttt{fi} \texttt{initial: iffalse tracklang.tex v1.4+}

Conditional that indicates whether or not to show verbose messages.

\ifTrackLangShowWarnings \texttt{true}\texttt{false}\texttt{fi} \texttt{initial: iftrue tracklang.tex v1.3+}

Conditional that indicates whether or not to show warnings.

S

\SetCurrentTrackedDialect\{}\texttt{dialect}\texttt{tracklang.tex v1.3+}

Sets the current tracked dialect.
### Command Summary

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>\SetTrackedDialectAdditional{&lt;dialect&gt;}{&lt;value&gt;}</code></td>
<td>Sets the extra information for <code>&lt;dialect&gt;</code> to <code>&lt;value&gt;</code>.</td>
</tr>
<tr>
<td><code>\SetTrackedDialectLabelMap{&lt;tracklang-label&gt;}{&lt;hook-label&gt;}</code></td>
<td>Defines a mapping between a tracklang dialect label and the corresponding dialect label used by a language hook, such as <code>\captions{&lt;dialect&gt;}</code>.</td>
</tr>
<tr>
<td><code>\SetTrackedDialectModifier{&lt;dialect&gt;}{&lt;value&gt;}</code></td>
<td>Sets the modifier for the given <code>&lt;dialect&gt;</code> to <code>&lt;value&gt;</code>.</td>
</tr>
<tr>
<td><code>\SetTrackedDialectScript{&lt;dialect&gt;}{&lt;value&gt;}</code></td>
<td>Sets the script for <code>&lt;dialect&gt;</code> to <code>&lt;value&gt;</code>.</td>
</tr>
<tr>
<td><code>\SetTrackedDialectSubLang{&lt;dialect&gt;}{&lt;value&gt;}</code></td>
<td>Sets the sub-language for <code>&lt;dialect&gt;</code> to <code>&lt;value&gt;</code>.</td>
</tr>
<tr>
<td><code>\SetTrackedDialectVariant{&lt;dialect&gt;}{&lt;value&gt;}</code></td>
<td>Sets the modifier for <code>&lt;dialect&gt;</code> to <code>&lt;value&gt;</code>.</td>
</tr>
<tr>
<td><code>\ThreeLetterExtIsoLanguageCode</code></td>
<td>Expands to 639-3 (should not be redefined).</td>
</tr>
</tbody>
</table>

Expands to 639-3 (should not be redefined).
## Command Summary

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
<th>Version</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>\ThreeLetterIsoLanguageCode</td>
<td>Expands to 639-2 (should not be redefined).</td>
<td>tracklang.tex v1.0+</td>
<td>§5; 40</td>
</tr>
<tr>
<td>\TrackedDialectClosestSubMatch</td>
<td>Defined by \GetTrackedDialectFromLanguageTag to the closest match.</td>
<td>tracklang.tex v1.3.6+</td>
<td>§5; 36</td>
</tr>
<tr>
<td>\TrackedDialectsFromLanguage{⟨root language label⟩}</td>
<td>Expands to a comma-separated list of the tracked dialects with the given language.</td>
<td>tracklang.tex v1.0+</td>
<td>§5; 39</td>
</tr>
<tr>
<td>\TrackedIsoCodeFromLanguage{⟨code type⟩}{⟨label⟩}</td>
<td>Expands to the code associated with the given language or dialect identified by ⟨label⟩.</td>
<td>tracklang.tex v1.0+</td>
<td>§5; 39</td>
</tr>
<tr>
<td>\TrackedLanguageFromDialect{⟨dialect⟩}</td>
<td>Expands to the language from the given dialect.</td>
<td>tracklang.tex v1.0+</td>
<td>§5; 39</td>
</tr>
<tr>
<td>\TrackedLanguageFromIsoCode{⟨code type⟩}{⟨code⟩}</td>
<td>Expands to a comma-separated list of language or dialect labels associated with the given code.</td>
<td>tracklang.tex v1.0+</td>
<td>§5; 39</td>
</tr>
<tr>
<td>\TrackIfKnownLanguage{⟨tag⟩}{⟨success code⟩}{⟨fail code⟩}</td>
<td>As \TrackLanguageTag but does ⟨fail code⟩ if the tag doesn’t contain a valid language code. If successful, does ⟨success code⟩ after tracking the language.</td>
<td>tracklang.tex v1.0+</td>
<td>§3; 21</td>
</tr>
<tr>
<td>\TrackLangAddExtraRegionFile{(file)}</td>
<td></td>
<td>tracklang.tex v1.4+</td>
<td>§4; 31</td>
</tr>
</tbody>
</table>

103
Command Summary

Adds \texttt{⟨file⟩} to the list of extra region code files that should be input by \texttt{tracklang-region-codes.tex}.

\texttt{\TrackLangAddExtraScriptFile{⟨file⟩}} \hspace{1cm} \texttt{tracklang.tex v1.4+} \hspace{1cm} §4; 33

Adds \texttt{⟨file⟩} to the list of files that should be input by \texttt{tracklang-scripts.tex}.

\texttt{\TrackLangAddToCaptions{⟨code⟩}} \hspace{1cm} \texttt{tracklang.tex v1.3+} \hspace{1cm} §5; 55

A shortcut that just does \texttt{\TrackLangAddToHook{⟨code⟩}{captions}}.

\texttt{\TrackLangAddToHook{⟨code⟩}{⟨type⟩}} \hspace{1cm} \texttt{tracklang.tex v1.3+} \hspace{1cm} §5; 55

For use within resource files, this can be used to add \texttt{⟨code⟩} to the appropriate hook.

\texttt{\TrackLangAlphaIIIToNumericRegion{⟨alpha-3 code⟩}} \hspace{1cm} \texttt{tracklang-region-codes.tex v1.3+} \hspace{1cm} §4; 30

Expands to the numeric code corresponding to the given alpha-3 code.

\texttt{\TrackLangAlphaIIIToNumericRegion{⟨alpha-2 code⟩}} \hspace{1cm} \texttt{tracklang-region-codes.tex v1.3+} \hspace{1cm} §4; 30

Expands to the numeric code corresponding to the given alpha-2 code.

\texttt{\TrackLangDeclareDialectOption{(dialect){⟨root language⟩}{⟨3166-1 code⟩}{⟨modifier⟩}{⟨variant⟩}{⟨map⟩}{⟨script⟩}}} \hspace{1cm} \texttt{tracklang.tex v1.3+}

Defines a predefined dialect label that can be used by \texttt{\TrackPredefinedDialect}. 

104
## Command Summary

\TrackLangDeclareLanguageOption{⟨language name⟩}{⟨639-1 code⟩}{⟨639-2 (T)⟩}{⟨639-2 (B)⟩}{⟨639-3⟩}{⟨3166-1⟩}{⟨default script⟩} \textit{tracklang.tex v1.3+}

Defines a new root language that's declared as an option.

\TrackLangEncodingName \textit{tracklang.tex v1.6.1+}  
§5; 53

Expands to \texttt{\inputencodingname} if it has been defined or utf8 otherwise.

\TrackLangEnv (user defined) \textit{§3; 25}

May be defined using the same format as \texttt{LC\_ALL} before using \TrackLangParseFromEnv to skip the environment variable query.

\TrackLangEnvCodeSet \textit{tracklang.tex v1.3+}  
§3; 26

Set by \TrackLangParseFromEnv to the code-set.

\TrackLangEnvLang \textit{tracklang.tex v1.3+}  
§3; 26

Set by \TrackLangParseFromEnv to the language code.

\TrackLangEnvModifier \textit{tracklang.tex v1.3+}  
§3; 26

Set by \TrackLangParseFromEnv to the modifier.

\TrackLangEnvTerritory \textit{tracklang.tex v1.3+}  
§3; 26

Set by \TrackLangParseFromEnv to the territory.

\TrackLangFromEnv \textit{tracklang.tex v1.3+}  
§3; 23

105
\textbf{Command Summary}

Queries environment variable if \texttt{\textbackslash TrackLangEnv} not already set, parses \texttt{\textbackslash TrackLangEnv} if it has been set, and adds the dialect if it’s recognised.

\begin{verbatim}
\%5\;43
\texttt{\textbackslash TrackLangGetDefaultScript\{\langle language\rangle\}} \hspace{1cm} \texttt{tracklang.tex v1.3+}
\end{verbatim}

Expands to the default script for the given language.

\begin{verbatim}
\%5\;43
\texttt{\textbackslash TrackLangGetKnownCountry\{\langle language\rangle\}} \hspace{1cm} \texttt{tracklang.tex v1.3+}
\end{verbatim}

Expands to the ISO 3166-1 country code for the given language.

\begin{verbatim}
\%5\;43
\texttt{\textbackslash TrackLangGetKnownIsoThreeLetterLang\{\langle language\rangle\}} \hspace{1cm} \texttt{tracklang.tex v1.3+}
\end{verbatim}

Expands to the ISO 639-2 language code associated with \texttt{(language)}.

\begin{verbatim}
\%5\;43
\texttt{\textbackslash TrackLangGetKnownIsoThreeLetterLangB\{\langle language\rangle\}} \hspace{1cm} \texttt{tracklang.tex v1.3+}
\end{verbatim}

Expands to the ISO 639-2 (B) language code associated with \texttt{(language)}.

\begin{verbatim}
\%5\;43
\texttt{\textbackslash TrackLangGetKnownIsoTwoLetterLang\{\langle language\rangle\}} \hspace{1cm} \texttt{tracklang.tex v1.3+}
\end{verbatim}

Expands to the ISO 639-1 language code associated with \texttt{(language)}.

\begin{verbatim}
\%5\;43
\texttt{\textbackslash TrackLangGetKnownLangFromIso\{\langle ISO code\rangle\}} \hspace{1cm} \texttt{tracklang.tex v1.3+}
\end{verbatim}

Expands to the root language label from the given ISO code (639-1 or 639-2 or 639-3).

\begin{verbatim}
\%5\;43
\texttt{\textbackslash TrackLangIfAlphaNumericChar\{\langle tag\rangle\}\{\langle true\rangle\}\{\langle false\rangle\}} \hspace{1cm} \texttt{tracklang.tex v1.3+}
\end{verbatim}

Does \texttt{(true)} if the argument is a single alphanumeric character otherwise does \texttt{(false)}.
\TrackLangIfHasDefaultScript{(language)}{(true)}{(false)}

Does \textit{true} if the given language has a default script (but is not necessarily tracked), otherwise does \textit{false}.

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

Does \textit{true} if the given language has an ISO 3166-1 country code (but is not necessarily tracked), otherwise does \textit{false}.

\TrackLangIfKnownAlphaIIIRegion{(alpha-3 code)}{(true)}{(false)}

Expands to \textit{true} if there’s a known mapping for the given \textit{alpha-3 code}, otherwise expands to \textit{false}.

\TrackLangIfKnownAlphaIIRegion{(alpha-2 code)}{(true)}{(false)}

Expands to \textit{true} if there’s a known mapping for the given alpha-2 region code, otherwise expands to \textit{false}.

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

Does \textit{true} if \textit{language} has an ISO 639-2 code (but is not necessarily tracked), otherwise does \textit{false}.

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

Does \(\langle true\rangle\) if \(\langle language\rangle\) has an ISO 639-2 (B) code (but is not necessarily tracked), otherwise does \(\langle false\rangle\).

\[
\textbackslash \text{TrackLangIfKnownIsoTwoLetterLang}\{\langle language\rangle\}\{\langle true\rangle\}\{\langle false\rangle\}
\]

tracklang.tex v1.3+

Does \(\langle true\rangle\) if \(\langle language\rangle\) has an ISO 639-1 code (but is not necessarily tracked), otherwise does \(\langle false\rangle\).

\[
\textbackslash \text{TrackLangIfKnownLang}\{\langle language\rangle\}\{\langle true\rangle\}\{\langle false\rangle\}
\]

tracklang.tex v1.3+

Does \(\langle true\rangle\) if \(\langle language\rangle\) is known (but not necessarily tracked), otherwise does \(\langle false\rangle\).

\[
\textbackslash \text{TrackLangIfKnownLangFromIso}\{\langle ISO \ code\rangle\}\{\langle true\rangle\}\{\langle false\rangle\}
\]

tracklang.tex v1.3+

Does \(\langle true\rangle\) if the given language code (639-1 or 639-2 or 639-3) is recognised (but not necessarily tracked), otherwise does \(\langle false\rangle\).

\[
\textbackslash \text{TrackLangIfKnownNumericRegion}\{\langle numeric \ code\rangle\}\{\langle true\rangle\}\{\langle false\rangle\}
\]

tracklang-region-codes.tex v1.3+

Expands to \(\langle true\rangle\) if there’s a known mapping for the given numeric region code, otherwise expands to \(\langle false\rangle\).

\[
\textbackslash \text{TrackLangIfLanguageTag}\{\langle tag\rangle\}\{\langle true\rangle\}\{\langle false\rangle\}
\]

tracklang.tex v1.3+

Does \(\langle true\rangle\) if the argument is a language tag otherwise does \(\langle false\rangle\).

\[
\textbackslash \text{TrackLangIfRegionTag}\{\langle tag\rangle\}\{\langle true\rangle\}\{\langle false\rangle\}
\]

tracklang.tex v1.3+

Does \(\langle true\rangle\) if the argument is a region tag otherwise does \(\langle false\rangle\).
Command Summary

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

tracklang.tex v1.3+

Does \langle true \rangle if the argument is a script tag otherwise does \langle false \rangle.

\TrackLangIfVariantTag\{⟨tag⟩\}\{⟨true⟩\}⟨false⟩\}

tracklang.tex v1.3+

Does \langle true \rangle if the argument is a variant tag otherwise does \langle false \rangle.

\TrackLangLastTrackedDialect

tracklang.tex v1.3+

§6.5; 77

Expands to the label of the last tracked dialect.

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

tracklang.tex v1.3+

§6.5; 76

Identifies a new language that may be tracked. Apart from \langle language label \rangle, the other arguments may be empty if the information is unavailable.

\TrackLangNumericToAlphaIIIRegion\{⟨numeric code⟩\}

tracklang-region-codes.tex v1.3+

§4; 31

Expands to the alpha-3 code corresponding to the given numeric code.

\TrackLangNumericToAlphaIIRegion\{⟨numeric code⟩\}

tracklang-region-codes.tex v1.3+

§4; 30

Expands to the alpha-2 code corresponding to the given numeric code.

\TrackLangParseFromEnv

tracklang.tex v1.3+

§3; 27

Attempts to obtain locale information from the expansion of \TrackLangEnv.
\TrackLangProvidePredefinedDialect{\langle dialect label \rangle}{\langle root language label \rangle}{\langle 3166-1 code \rangle}{\langle modifier \rangle}{\langle variant \rangle}{\langle map \rangle}{\langle script \rangle}

tracklang.tex v1.4+

§6.6; 77

Defines a predefined dialect label that can be used by \TrackPredefinedDialect.

\TrackLangProvidePredefinedLanguage{\langle language label \rangle}

tracklang.tex v1.4+

§6.6; 77

Sets up a language label for use with \TrackPredefinedDialect.

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

tracklang.tex v1.3+

§5; 52

Analogous to \ProvidesFile.

\TrackLangQueryEnv

tracklang.tex v1.3+

§3; 26

Attempts to obtain locale information from the LC_ALL environment variable via the shell escape or, with LuaTeX, \directlua.

\TrackLangQueryOtherEnv{\langle env-name \rangle}

tracklang.tex v1.3+

§3; 26

Attempts to obtain locale information from the LC_ALL environment variable and then by the \langle env-name \rangle environment variable via the shell escape or, with LuaTeX, \directlua.

\TrackLangRedefHook{\langle code \rangle}{\langle type \rangle}

tracklang.tex v1.4+

§5; 55

Similar to \TrackLangAddToHook but redefines the hook rather than appending to it.

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

tracklang-region-codes.tex v1.3+

§4; 31

Establishes a mapping between a numeric region code and alpha-2 and alpha-3 codes.
\textit{Command Summary}

\texttt{\TrackLangRequestResource\{\langle\text{tag}\rangle\}\{\langle\text{not found code}\rangle\}} \quad \text{tracklang.tex v1.3+} \quad §5; 54

As \texttt{\TrackLangRequireResource} but does \texttt{\langle\text{not found code}\rangle} if the file doesn’t exist.

\texttt{\TrackLangRequireDialect\{\langle\text{load code}\rangle\}\{\langle\text{pkgname}\rangle\}\{\langle\text{dialect}\rangle\}} \quad \text{tracklang.tex v1.3+} \quad §5; 45

Loads the dialect for the given package.

\texttt{\TrackLangRequireDialectPrefix} \quad \text{tracklang.tex v1.3+} \quad §5; 45

Defined by \texttt{\TrackLangRequireDialect}.

\texttt{\TrackLangRequireResource\{\langle\text{tag}\rangle\}} \quad \text{tracklang.tex v1.3+} \quad §5; 53

Loads the appropriate 1df file if it hasn’t already been loaded.

\texttt{\TrackLangRequireResourceOrDo\{\langle\text{tag}\rangle\}\{\langle\text{code1}\rangle\}\{\langle\text{code2}\rangle\}} \quad \text{tracklang.tex v1.3+} \quad §5; 53

As \texttt{\TrackLangRequireResource} but does \texttt{\langle\text{code1}\rangle} if the file is now loaded or \texttt{\langle\text{code2}\rangle} if the file has already been loaded.

\texttt{\TrackLangScriptAlphaToDir\{\langle\text{alpha code}\rangle\}} \quad \text{tracklang-scripts.tex v1.3+} \quad §4; 33

Expands to the direction associated with the given alpha script code.

\texttt{\TrackLangScriptAlphaToName\{\langle\text{alpha code}\rangle\}} \quad \text{tracklang-scripts.tex v1.3+} \quad §4; 32

Expands to the name associated with the given alpha script code.

\texttt{\TrackLangScriptAlphaToNumeric\{\langle\text{alpha code}\rangle\}} \quad \text{tracklang-scripts.tex v1.3+} \quad §4; 32
Command Summary

Expands to the numeric script code corresponding to the given alpha code.

\TrackLangScript{Code} \quad \text{tracklang-scripts.tex v1.3+} \quad §4; 32

Set by \TrackLangScriptMap to the associated alpha code \textit{(Code)}.

\TrackLangScriptGetParent\{\langle alpha code\rangle\} \quad \text{tracklang-scripts.tex v1.3+} \quad §4; 33

Expands to the parent of the given alpha script code.

\TrackLangScriptIfHasParent\{\langle alpha code\rangle\}\{\langle true\rangle\}\{\langle false\rangle\} \quad \text{tracklang-scripts.tex v1.3+} \quad §4; 33

Expands to \textit{(true)} if the given alpha script code has a parent otherwise expands to \textit{(false)}.

\TrackLangScriptIfKnownAlpha\{\langle alpha code\rangle\}\{\langle true\rangle\}\{\langle false\rangle\} \quad \text{tracklang-scripts.tex v1.3+} \quad §4; 32

Expands to \textit{(true)} if there’s a known mapping for the given alpha script code otherwise expands to \textit{(false)}.

\TrackLangScriptIfKnownNumeric\{\langle numeric code\rangle\}\{\langle true\rangle\}\{\langle false\rangle\} \quad \text{tracklang-scripts.tex v1.3+} \quad §4; 32

Expands to \textit{(true)} if there’s a known mapping for the given numeric script code otherwise expands to \textit{(false)}.

\TrackLangScriptMap\{\langle letter code\rangle\}\{\langle numeric code\rangle\}\{\langle script name\rangle\}\{\langle direction\rangle\}\{\langle parent script\rangle\} \quad \text{tracklang-scripts.tex v1.3+} \quad §4; 32

Defines a mapping between an alpha code and a numeric code.
\TrackLangScriptNumericToAlpha{⟨numeric code⟩} \textit{tracklang-scripts.tex} v1.3+

Expands to the alpha script code corresponding to the given numeric code.

\TrackLangScriptSetParent{⟨alpha code⟩}{⟨parent alpha code⟩} \textit{tracklang-scripts.tex} v1.3+

Sets the parent for the given alpha script code.

\TrackLangShowWarningsfalse \textit{tracklang.tex} v1.3+

Sets \texttt{\ifTrackLangShowWarnings} to false.

\TrackLangShowWarningstrue \textit{tracklang.tex} v1.3+

Sets \texttt{\ifTrackLangShowWarnings} to true.

\TrackLanguageTag{⟨tag⟩} \textit{tracklang.tex} v1.3+

Parse ⟨tag⟩, which should be a regular, well-formed RFC 5646 language tag (not an irregular grandfather tag) and track the dialect.

\TrackLocale{⟨locale⟩} \textit{tracklang.tex} v1.3+

Tracks the dialect identified by the given ⟨locale⟩, which may either be a predefined language/dialect or in the same format as \TrackLangEnv.

\TrackPredefinedDialect{⟨dialect label⟩} \textit{tracklang.tex} v1.0+

Tracks a predefined language or dialect.
\TwoLetterIsoCountryCode  tracklang.tex v1.0+

Expands to 3166-1 (should not be redefined).

\TwoLetterIsoLanguageCode  tracklang.tex v1.0+

Expands to 639-1 (should not be redefined).
Index

Symbols

_ (separator) 20, 24, 25, 79, 81
. (code-set) 20, 24, 79
\ (escaped backslash) 25
$ (environment variable) 25
- (separator) 20, 25
--shell-escape §23, 24

@ (catcode 11) 19
@ (modifier) 20, 79
\@for §37, 95
\@nil §37
\@tracklang@declareoption §3; 19, 95
\@tracklang@for §5; 37, 95
\@tracklang@prelangpkgcheck@hook §6; 72, 95

A

aa (ISO code) see afar
ab (ISO code) see abkhaz
abkhaz (root language) §1; Table 1.2
acadian (option) §1; Table 1.3
AD (region) §A; Table A.1
\AddTrackedCountryIsoCode §95
\AddTrackedDialect §6.5; 76–80, 95, 96
\AddTrackedIsoLanguage §80, 96
\AddTrackedLanguage §6.5; 76, 96
\AddTrackedLanguageIsoCodes §6.6; 78–80, 96
Adlm (script) §A; Table A.2
AE (region) §A; Table A.1
ae (ISO code) see avestan
AF (region) §A; Table A.1
af (ISO code) see afrikaans
Afak (script) §A; Table A.2
afar (root language) §1; Table 1.2
afrikaans (root language) §1; Table 1.2
AG (region) §A; Table A.1
Aghb (script) §A; Table A.2
Ahom (script) §A; Table A.2
AI (region) §A; Table A.1
ak (ISO code) see akan
akan (root language) §1; Table 1.2
AL (region) §A; Table A.1
albanian (root language) §1; Table 1.2
alien.sty 80–82
alien package see alien.sty
AM (region) §A; Table A.1
am (ISO code) see amharic
american (option) §1; Table 1.3; 60, 61, 71
amharic (root language) §1; Table 1.2
an (ISO code) see aragonese
ang (ISO code) see anglosaxon
anglosaxon (root language) §1; Table 1.2
animals.sty 55
\AnyTrackedLanguages §5; 14, 15, 35, 96
AO (region) §A; Table A.1
apa (ISO code) see apache
apache (root language) §1; Table 1.2
AQ (region) §A; Table A.1
AR (region) §A; Table A.1
ar (ISO code) see arabic
Arab (script) §A; Table A.2
arabic (root language) §1; Table 1.2
aragonese (root language) §1; Table 1.2
Armi (script) §A; Table A.2

Index

<table>
<thead>
<tr>
<th>Armn (script)</th>
<th>§A; Table A.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>AS (region)</td>
<td>§A; Table A.1</td>
</tr>
<tr>
<td>as (ISO code)</td>
<td>see assamese</td>
</tr>
<tr>
<td>assamese (root language)</td>
<td>§1; Table 1.2</td>
</tr>
<tr>
<td>ast (ISO code)</td>
<td>see asturian</td>
</tr>
<tr>
<td>asturian (root language)</td>
<td>§1; Table 1.2</td>
</tr>
<tr>
<td>AT (region)</td>
<td>§A; Table A.1</td>
</tr>
<tr>
<td>\AtBeginDocument</td>
<td>37</td>
</tr>
<tr>
<td>AU (region)</td>
<td>§A; Table A.1</td>
</tr>
<tr>
<td>australian (option)</td>
<td>§1; Table 1.3</td>
</tr>
<tr>
<td>austrain (option)</td>
<td>§1; Table 1.3</td>
</tr>
<tr>
<td>av (ISO code)</td>
<td>see avaric</td>
</tr>
<tr>
<td>avaric (root language)</td>
<td>§1; Table 1.2</td>
</tr>
<tr>
<td>avestan (root language)</td>
<td>§1; Table 1.2</td>
</tr>
<tr>
<td>Avst (script)</td>
<td>§A; Table A.2</td>
</tr>
<tr>
<td>AW (region)</td>
<td>§A; Table A.1</td>
</tr>
<tr>
<td>AX (region)</td>
<td>§A; Table A.1</td>
</tr>
<tr>
<td>ay (ISO code)</td>
<td>see aymara</td>
</tr>
<tr>
<td>aymara (root language)</td>
<td>§1; Table 1.2</td>
</tr>
<tr>
<td>AZ (region)</td>
<td>§A; Table A.1</td>
</tr>
<tr>
<td>az (ISO code)</td>
<td>see azerbaijani</td>
</tr>
<tr>
<td>azerbaijani (root language)</td>
<td>§1; Table 1.2</td>
</tr>
</tbody>
</table>

B

<table>
<thead>
<tr>
<th>BA (region)</th>
<th>§A; Table A.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>ba (ISO code)</td>
<td>see bashkir</td>
</tr>
<tr>
<td>babel package</td>
<td>a, 2, 3, 9, 12, 13, 35, 54, 58–60, 68–71, 78, 95</td>
</tr>
<tr>
<td>babelprovide</td>
<td>2, 13</td>
</tr>
<tr>
<td>babel</td>
<td>§1; Table 1.3</td>
</tr>
<tr>
<td>bahasa (option)</td>
<td>§1; Table 1.3</td>
</tr>
<tr>
<td>bahasa (root language)</td>
<td>§1; Table 1.2</td>
</tr>
<tr>
<td>bali (script)</td>
<td>§A; Table A.2</td>
</tr>
<tr>
<td>bambara (root language)</td>
<td>§1; Table 1.2</td>
</tr>
<tr>
<td>bamu (script)</td>
<td>§A; Table A.2</td>
</tr>
<tr>
<td>bashkir (root language)</td>
<td>§1; Table 1.2</td>
</tr>
<tr>
<td>basque (root language)</td>
<td>§1; Table 1.2</td>
</tr>
<tr>
<td>Bass (script)</td>
<td>§A; Table A.2</td>
</tr>
<tr>
<td>Batak (script)</td>
<td>§A; Table A.2</td>
</tr>
<tr>
<td>BB (region)</td>
<td>§A; Table A.1</td>
</tr>
<tr>
<td>\bb@loaded</td>
<td>2, 35</td>
</tr>
<tr>
<td>BCP 47</td>
<td>35</td>
</tr>
<tr>
<td>BD (region)</td>
<td>§A; Table A.1</td>
</tr>
<tr>
<td>BE (region)</td>
<td>§A; Table A.1</td>
</tr>
<tr>
<td>be (ISO code)</td>
<td>see belarusian</td>
</tr>
<tr>
<td>belarusian (root language)</td>
<td>§1; Table 1.2</td>
</tr>
<tr>
<td>belgiangerman (option)</td>
<td>§1; Table 1.3</td>
</tr>
<tr>
<td>belgique (option)</td>
<td>§1; Table 1.3</td>
</tr>
<tr>
<td>Beng (script)</td>
<td>§A; Table A.2</td>
</tr>
<tr>
<td>bengali (root language)</td>
<td>§1; Table 1.2</td>
</tr>
<tr>
<td>ber (ISO code)</td>
<td>see berber</td>
</tr>
<tr>
<td>berber (root language)</td>
<td>§1; Table 1.2</td>
</tr>
<tr>
<td>bg (ISO code)</td>
<td>see bulgarian</td>
</tr>
<tr>
<td>BH (region)</td>
<td>§A; Table A.1</td>
</tr>
<tr>
<td>bh (ISO code)</td>
<td>see bihari</td>
</tr>
<tr>
<td>BI (region)</td>
<td>§A; Table A.1</td>
</tr>
<tr>
<td>bi (ISO code)</td>
<td>see bislama</td>
</tr>
<tr>
<td>bihari (root language)</td>
<td>§1; Table 1.2</td>
</tr>
<tr>
<td>bislama (root language)</td>
<td>§1; Table 1.2</td>
</tr>
<tr>
<td>BJ (region)</td>
<td>§A; Table A.1</td>
</tr>
<tr>
<td>BL (region)</td>
<td>§A; Table A.1</td>
</tr>
<tr>
<td>Blis (script)</td>
<td>§A; Table A.2</td>
</tr>
<tr>
<td>BM (region)</td>
<td>§A; Table A.1</td>
</tr>
<tr>
<td>bm (ISO code)</td>
<td>see bambara</td>
</tr>
<tr>
<td>BN (region)</td>
<td>§A; Table A.1</td>
</tr>
<tr>
<td>bn (ISO code)</td>
<td>see bosnian</td>
</tr>
<tr>
<td>BO (region)</td>
<td>§A; Table A.1</td>
</tr>
<tr>
<td>bo (ISO code)</td>
<td>see tibetan</td>
</tr>
<tr>
<td>bokmal (root language)</td>
<td>§1; Table 1.2</td>
</tr>
<tr>
<td>bolivianspanish (option)</td>
<td>§1; Table 1.3</td>
</tr>
<tr>
<td>Bopo (script)</td>
<td>§A; Table A.2</td>
</tr>
<tr>
<td>bosnian (root language)</td>
<td>§1; Table 1.2</td>
</tr>
<tr>
<td>BQ (region)</td>
<td>§A; Table A.1</td>
</tr>
<tr>
<td>br (ISO code)</td>
<td>see breton</td>
</tr>
<tr>
<td>BR (region)</td>
<td>§A; Table A.1</td>
</tr>
<tr>
<td>Brai (script)</td>
<td>§A; Table A.2</td>
</tr>
<tr>
<td>brazил (option)</td>
<td>§1; Table 1.3</td>
</tr>
<tr>
<td>brazilian (option)</td>
<td>§1; Table 1.3; 21</td>
</tr>
<tr>
<td>breton (root language)</td>
<td>§1; Table 1.2</td>
</tr>
<tr>
<td>british (option)</td>
<td>§1; Table 1.3; 2, 3, 12, 13, 15, 17, 34, 36, 49, 54, 60, 61, 63, 71</td>
</tr>
</tbody>
</table>

116
Table 1.2

<table>
<thead>
<tr>
<th>Language/Code</th>
<th>Section</th>
<th>Page(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bugi (script)</td>
<td>§A; Table A.2</td>
<td>55, 60, 61, 68, 70, 74, 78, 81, 82, 102</td>
</tr>
<tr>
<td>Burmese (root language)</td>
<td>§1; Table 1.2</td>
<td>55, 60, 61, 68, 70, 74, 78, 81, 82, 102</td>
</tr>
<tr>
<td>Copt (script)</td>
<td>§A; Table A.2</td>
<td>55, 60, 61, 68, 70, 74, 78, 81, 82, 102</td>
</tr>
<tr>
<td>Copt (root language)</td>
<td>§A; Table A.2</td>
<td>55, 60, 61, 68, 70, 74, 78, 81, 82, 102</td>
</tr>
<tr>
<td>Corsican (root language)</td>
<td>§A; Table A.2</td>
<td>55, 60, 61, 68, 70, 74, 78, 81, 82, 102</td>
</tr>
<tr>
<td>Czecho-Slovak (root language)</td>
<td>§A; Table A.2</td>
<td>55, 60, 61, 68, 70, 74, 78, 81, 82, 102</td>
</tr>
<tr>
<td>Czecho-Slovakian (root language)</td>
<td>§A; Table A.2</td>
<td>55, 60, 61, 68, 70, 74, 78, 81, 82, 102</td>
</tr>
<tr>
<td>Cubanspanish (script)</td>
<td>11, 46, 74, 78, 81, 82, 102</td>
<td>55, 60, 61, 68, 70, 74, 78, 81, 82, 102</td>
</tr>
<tr>
<td>Cubanspanish (root language)</td>
<td>11, 46, 74, 78, 81, 82, 102</td>
<td>55, 60, 61, 68, 70, 74, 78, 81, 82, 102</td>
</tr>
<tr>
<td>Cubanspanishian (root language)</td>
<td>11, 46, 74, 78, 81, 82, 102</td>
<td>55, 60, 61, 68, 70, 74, 78, 81, 82, 102</td>
</tr>
<tr>
<td>Cubanspanishianian (root language)</td>
<td>11, 46, 74, 78, 81, 82, 102</td>
<td>55, 60, 61, 68, 70, 74, 78, 81, 82, 102</td>
</tr>
</tbody>
</table>

C
Index

CX (region) .................................... §A; Table A.1
cy-GB (option) ................................ §1; Table 1.1
CY (region) ..................................... §A; Table A.1
cy (ISO code) .................................. see welsh
cymraeg (option) ............................... §1; Table 1.3
Cyr1 (script) ..................................... §A; Table A.2; 16, 32, 42, 79
Cyr2 (script) ..................................... §A; Table A.2
CZ (region) ...................................... §A; Table A.1
czech (root language) ......................... §1; Table 1.2

D

da (ISO code) ................................... see danish
danish (root language) ......................... §1; Table 1.2
\date{dialect} .................................. Table 1.3; 15, 55
datetime2 package ............................. 9, 21
de-AT (option) .................................. §1; Table 1.1
de-AT-1996 (option) ......................... §1; Table 1.1
de-BE (option) .................................. §1; Table 1.1
de-CH (option) .................................. §1; Table 1.1
de-CH-1996 (option) ......................... §1; Table 1.1
de-DE (option) .................................. §1; Table 1.1
de-DE-1996 (option) ......................... §1; Table 1.1
DE (region) ..................................... §A; Table A.1
de (ISO code) .................................. see german
\DeclareOption .................................. 13, 19
deutsch (option) ................................ §1; Table 1.3
Deva (script) .................................... §A; Table A.2
\directlua ..................................... a, 23, 110
divehi (root language) ....................... §1; Table 1.2
DJ (region) ..................................... §A; Table A.1
DK (region) ..................................... §A; Table A.1
DM (region) ..................................... §A; Table A.1
DØ (region) ..................................... §A; Table A.1
dominicanspanish (option) ............... §1; Table 1.3
dsrb (ISO code) ................................ see isorbian
Dsrt (script) .................................... §A; Table A.2
Dupl (script) .................................... §A; Table A.2
dutch (root language) ....................... §1; Table 1.2
dv (ISO code) .................................. see divehi
DZ (region) ..................................... §A; Table A.1
dz (ISO code) .................................. see dzongkha
dzongkha (root language) .................. §1; Table 1.2
easternpunjabi (root language) ........... §1; Table 1.2
EC (region) ..................................... §A; Table A.1
ecuadorianspanish (option) ............... §1; Table 1.3
EE (region) ..................................... §A; Table A.1
ee (ISO code) .................................. see ewe
EG (region) ..................................... §A; Table A.1
Egyd (script) .................................... §A; Table A.2
Egyh (script) .................................... §A; Table A.2
Egyp (script) .................................... §A; Table A.2
EH (region) ..................................... §A; Table A.1
e1 (ISO code) .................................. see greek
Elba (script) ..................................... §A; Table A.2
elsalvadorspanish (option) ............... §1; Table 1.3
en-AU (option) .................................. §1; Table 1.1
en-CA (option) .................................. §1; Table 1.1
en-GB (option) .................................. §1; Table 1.1; 12, 15, 24
en-GG (option) .................................. §1; Table 1.1
en-IE (option) .................................. §1; Table 1.1; 12
en-IM (option) .................................. §1; Table 1.1
en-JE (option) .................................. §1; Table 1.1
en-MT (option) .................................. §1; Table 1.1; 12, 13, 17, 70
en-NZ (option) .................................. §1; Table 1.1
en-US (option) .................................. §1; Table 1.1
en (ISO code) .................................. see english
english (root language) ..................... §1; Table 1.2; 2, 3, 15, 34, 36, 39
environment variables

LANG ............................................. a, 23–26, 81
LC_ALL ........................................... a, 23, 26, 105, 110
LC_MONETARY .................................. 26
eo (ISO code) .................................. see esperanto
ER (region) ..................................... §A; Table A.1
es-AR (option) .................................. §1; Table 1.1
es-B0 (option) .................................. §1; Table 1.1
es-CL (option) .................................. §1; Table 1.1
es-C0 (option) .................................. §1; Table 1.1
es-CR (option) .................................. §1; Table 1.1
es-CU (option) .................................. §1; Table 1.1
es-D0 (option) .................................. §1; Table 1.1
es-EC (option) .................................. §1; Table 1.1
es-ES (option) .................................. §1; Table 1.1

118
es–GT (option) ........................................§1; Table 1.1  \ForEachTrackedDialect §5; 2, 15, 37, 98
es–HN (option) ........................................ §1; Table 1.1  \ForEachTrackedLanguage §5; 37, 98
es–MX (option) ........................................ §1; Table 1.1  fr–BE (option) §1; Table 1.1
es–NL (option) ........................................ §1; Table 1.1  fr–CA (option) §1; Table 1.1
es–PA (option) ........................................ §1; Table 1.1  fr–CH (option) §1; Table 1.1
es–PE (option) ........................................ §1; Table 1.1  fr–FR (option) §1; Table 1.1
es–PR (option) ........................................ §1; Table 1.1  fr–GG (option) §1; Table 1.1
es–PY (option) ........................................ §1; Table 1.1  fr–JE (option) §1; Table 1.1
es–SV (option) ........................................ §1; Table 1.1  fr (region) §A; Table A.1
es–UY (option) ........................................ §1; Table 1.1  fr (ISO code) see french
es–VE (option) ........................................ §1; Table 1.1  franais (option) §1; Table 1.3; 51
ES (region) ........................................... §A; Table A.1  france (option) §1; Table 1.3
es (ISO code) .......................................... see spanish  french (root language) §1; Table 1.2; 51
esperanto (root language) ......................... §1; Table 1.2  frenchb (option) §1; Table 1.3
estonian (root language) ......................... §1; Table 1.2  friulian (root language) §1; Table 1.2
ET (region) ........................................... §A; Table A.1  friulano (option) §1; Table 1.3
et (ISO code) .......................................... see estonian  friulian (option) §1; Table 1.3
etex (application) .................................. 23  fula (root language) §1; Table 1.2
Ethi (script) .......................................... §A; Table A.2  fur (ISO code) see friulian
toolbox package ..................................... 54  furlan (option) §1; Table 1.3
eu (ISO code) .......................................... see basque  fy (ISO code) see westernfrisian
ewe (root language) ................................... §1; Table 1.2

F

fa (ISO code) .......................................... see farsi  ga–GB (option) §1; Table 1.1
faroese (root language) ............................ §1; Table 1.2  ga–IE (option) §1; Table 1.1; 12
farsi (root language) ................................ §1; Table 1.2  ga (region) §A; Table A.1
gb (ISO code) ........................................ see irish
ff (ISO code) .......................................... see fula  gaelige (option) §1; Table 1.3
FI (region) ........................................... §A; Table A.1  gaelic (option) §1; Table 1.3
fi (ISO code) .......................................... see finnish  galician (root language) §1; Table 1.2
fijian (root language) ............................... §1; Table 1.2  galicien (option) §1; Table 1.3
file formats ...........................................
  ldf ............................................... 13, 15, 45, 62  ganda (root language) §1; Table 1.2
  log .............................................. 94  GB (region) ........................................... §A; Table A.1; 36, 39
  tex .............................................. 9  GBirish (option) §1; Table 1.3
  gb (ISO code) ..................................... see irish  GBscottish (option) §1; Table 1.3
  gd (region) ...................................... §A; Table A.1  GBwelsh (option) §1; Table 1.3
  gd–GB (option) ................................... §1; Table 1.1
  fijian (ISO code) .................................. see fijian
gd (ISO code) ...................................... see scottish
gd (region) ......................................... §A; Table A.1
ge (region) .......................................... §A; Table A.1  Geok (script) §A; Table A.2
geor (script) ........................................ §A; Table A.2
Geor (script) ........................................ §A; Table A.2

G

119
Index

german (root language) §1; Table 1.2; 35, 79
german package 2, 3, 35
germanb (option) §1; Table 1.3
germanDE (option) §1; Table 1.3
\GetTrackedDialectAdditional §5; 44, 75, 98
\GetTrackedDialectFromLanguageTag §5; 36, 98, 103
\GetTrackedDialectModifier §5; 40, 98
\GetTrackedDialectScript §5; 42, 75, 99
\GetTrackedDialectSubLang §5; 44, 99
\GetTrackedDialectVariant §5; 42, 99
\GetTrackedLanguageTag §5; Tables 1.2, 1.3; 40, 99
GF (region) §A; Table A.1
GG (region) §A; Table A.1
GH (region) §A; Table A.1
GI (region) §A; Table A.1
GL (region) §A; Table A.1
g1 (ISO code) see galician
G1ag (script) §A; Table A.2
glossaries package 2, 9
GM (region) §A; Table A.1
GN (region) §A; Table A.1
gn (ISO code) see guarani
Goth (script) §A; Table A.2
GP (region) §A; Table A.1
GQ (region) §A; Table A.1
GR (region) §A; Table A.1
Gran (script) §A; Table A.2
greek (root language) §1; Table 1.2
Grek (script) §A; Table A.2
GS (region) §A; Table A.1
GT (region) §A; Table A.1
GU (region) §A; Table A.1
gu (ISO code) see gujarati
guarani (root language) §1; Table 1.2
guatemalanspanish (option) §1; Table 1.3
guernseyenglish (option) §1; Table 1.3
guernseyfrench (option) §1; Table 1.3
gujarati (root language) §1; Table 1.2
Guru (script) §A; Table A.2
Gv (ISO code) see manx
Gw (region) §A; Table A.1
GY (region) §A; Table A.1

ha (ISO code) see hausa
haitian (root language) §1; Table 1.2
Hanb (script) §A; Table A.2
Hang (script) §A; Table A.2
Hani (script) §A; Table A.2
Hano (script) §A; Table A.2
Hans (script) §A; Table A.2
Hant (script) §A; Table A.2
Hatr (script) §A; Table A.2
hausa (root language) §1; Table 1.2
he (ISO code) see hebrew
Hebr (script) §A; Table A.2
hebrew (root language) §1; Table 1.2
herero (root language) §1; Table 1.2
hi (ISO code) see hindi
hindi (root language) §1; Table 1.2
Hira (script) §A; Table A.2
hirimotu (root language) §1; Table 1.2
honduranspanish (option) §1; Table 1.3
hr–HR (option) §1; Table 1.1
HR (region) §A; Table A.1
hr (ISO code) see croatian
Hrkt (script) §A; Table A.2
hsb (ISO code) see usorbian
HO (region) §A; Table A.1
ht (ISO code) see haitian
hu–HU (option) §1; Table 1.1
HU (region) §A; Table A.1
hu (ISO code) see magyar
hz (ISO code) see herero

H

120
Index

ia (ISO code) ........................................ 39, 40, 46, 50, 51, 61, 62, 73, 75, 76, 82
icelandic (root language) .................................................. §1; Table 1.2
id-IN (option) ........................................ 39, 40, 46, 50, 51, 61, 62, 73, 75, 76, 82
ID (region) .................................................. §A; Table A.1
ido (root language) ........................................ see bahasa
IE (region) .................................................. §A; Table A.1
ie (ISO code) ........................................ see interlingue
IEEnglish (option) ........................................ §1; Table 1.3
IEirish (option) ........................................ §1; Table 1.3
\IfHasTrackedDialectAdditional ........................................ §5; 44, 99
\IfHasTrackedDialectModifier ........................................ §5; 41, 99
\IfHasTrackedDialectScript ........................................ §5; 42, 99
\IfHasTrackedDialectSubLang ........................................ §5; 44, 100
\IfHasTrackedDialectVariant ........................................ §5; 42, 100
\IfTrackedDialect ........................................ §5; 38, 100
\IfTrackedDialectIsScriptCs ........................................ §5; 32, 43, 100
\IfTrackedIsoCode ........................................ §5; 39, 100
\IfTrackedLanguage ........................................ §5; 38, 100
\IfTrackedLanguageFileExists ........................................ §5; 15, 45, 47, 54, 61, 101
\IfTrackedLanguageHasIsoCode ........................................ §5; 39, 101
\ifTrackLangShowInfo ........................................ 101
\ifTrackLangShowVerbose ........................................ 101
\ifTrackLangShowWarnings ........................................ 101, 113
ig (ISO code) ........................................ see igbo
igbo (root language) ........................................ §1; Table 1.2
ii (ISO code) ........................................ see nuosu
ik (ISO code) ........................................ see inupiaq
IL (region) .................................................. §A; Table A.1
IM (region) .................................................. §A; Table A.1
IN (region) .................................................. §A; Table A.1
indon (option) ........................................ §1; Table 1.3
indonesian (option) ........................................ §1; Table 1.3
Inds (script) .................................................. §A; Table A.2
\input ........................................ 9–11, 13, 17, 19, 25, 72, 73, 82
\inputencodingname ........................................ 53
\inputencodingname ........................................ 53
interlingua (root language) ........................................ §1; Table 1.2
interlingue (root language) ........................................ §1; Table 1.2
inuktitut (root language) ........................................ §1; Table 1.2
inupiaq (root language) ........................................ §1; Table 1.2
IO (region) .................................................. §A; Table A.1
io (ISO code) ........................................ see ido
IQ (region) .................................................. §A; Table A.1
IR (region) .................................................. §A; Table A.1
irish (root language) ........................................ §1; Table 1.2
IS (region) .................................................. §A; Table A.1
is (ISO code) ........................................ see icelandic
isleofmanenglish (option) ........................................ §1; Table 1.3
ISO ........................................ Table 1.1; b, 15, 39
ISO 15924 ........................................ 31, 42, 43, 73, 77, 79
ISO 3166-1 ........................................ 20, 22, 35, 39, 40, 46, 73, 75–77, 95, 98, 106, 107, 114
ISO 639-1 ........................................ 20, 22, 34, 39, 40, 46, 50, 51, 61, 62, 73, 75, 76, 82, 96, 97, 106, 108, 114
ISO 639-2 ........................................ 20, 34, 40, 46, 50, 51, 62, 75, 76, 82, 96, 97, 103, 106–108
ISO 639-2 (B) ........................................ 21, 35, 76, 106, 107
ISO 639-2 (T) ........................................ 21, 35, 76
ISO 639-3 ........................................ 22, 39, 40, 46, 47, 75, 76, 96, 97, 102, 106, 108
istriacountycroatian (option) ........................................ §1; Table 1.3
istriacountyitalian (option) ........................................ §1; Table 1.3
it-CH (option) ........................................ §1; Table 1.1
it-HR (option) ........................................ §1; Table 1.1
it-TT (option) ........................................ §1; Table 1.1
it-SI (option) ........................................ §1; Table 1.1
it-SM (option) ........................................ §1; Table 1.1
it-VA (option) ........................................ §1; Table 1.1
it-VA (option) ........................................ §1; Table 1.1
italian (root language) ........................................ §1; Table 1.2
italy (option) ........................................ §1; Table 1.3
iu (ISO code) ........................................ see inuktitut

121
Index

J

ja (ISO code) .......... see japanese
Jamo (script) .......... §A; Table A.2
japanese (root language) §1; Table 1.2
Java (script) .......... §A; Table A.2
javanese (root language) §1; Table 1.2
JE (region) .......... §A; Table A.1
jerseyenglish (option) §1; Table 1.3
jerseyfrench (option) §1; Table 1.3
J.M (region) .......... §A; Table A.1
J0 (region) .......... §A; Table A.1
JP (region) .......... §A; Table A.1
Jpan (script) .......... §A; Table A.2
Jurc (script) .......... §A; Table A.2
jv (ISO code) .......... see javanese

K

ka (ISO code) .......... see georgian
kalaallisut (root language) §1; Table 1.2
Kali (script) .......... §A; Table A.2
Kana (script) .......... §A; Table A.2
kannada (root language) §1; Table 1.2
kanuri (root language) §1; Table 1.2
kashmiri (root language) §1; Table 1.2
kazakh (root language) §1; Table 1.2
KE (region) .......... §A; Table A.1
KG (region) .......... §A; Table A.1
kg (ISO code) .......... see kongo
KH (region) .......... §A; Table A.1
Khar (script) .......... §A; Table A.2
khmer (root language) §1; Table 1.2
Khmr (script) .......... §A; Table A.2
Khoj (script) .......... §A; Table A.2
KI (region) .......... §A; Table A.1
ki (ISO code) .......... see kikuyu
kikuyu (root language) §1; Table 1.2
kinyarwanda (root language) §1; Table 1.2
kirundi (root language) §1; Table 1.2
Kitl (script) .......... §A; Table A.2
Kits (script) .......... §A; Table A.2
kj (ISO code) .......... see kwanyama
kk (ISO code) .......... see kazakh
kl (ISO code) .......... see kalaallisut

KM (region) .......... §A; Table A.1
km (ISO code) .......... see khmer
KN (region) .......... §A; Table A.1
kn (ISO code) .......... see kannada
Knda (script) .......... §A; Table A.2
ko (ISO code) .......... see korean
komi (root language) §1; Table 1.2
kongo (root language) §1; Table 1.2
Kore (script) .......... §A; Table A.2
korean (root language) §1; Table 1.2
KP (region) .......... §A; Table A.1
Kpel (script) .......... §A; Table A.2
KR (region) .......... §A; Table A.1
kr (ISO code) .......... see kanuri
ks (ISO code) .......... see kashmiri
Kthi (script) .......... §A; Table A.2
ku (ISO code) .......... see kurdish
kurdish (root language) §1; Table 1.2
kurmanji (option) §1; Table 1.3
kv (ISO code) .......... see komi
KW (region) .......... §A; Table A.1
kw (ISO code) .......... see cornish
kwanyama (root language) §1; Table 1.2
KY (region) .......... §A; Table A.1
ky (ISO code) .......... see kyrgyz
kyrgyzz (root language) §1; Table 1.2
KZ (region) .......... §A; Table A.1

L

LA (region) .......... §A; Table A.1
la (ISO code) .......... see latin
Lana (script) .......... §A; Table A.2
\languagename 10, 75
\lao (root language) §1; Table 1.2
Laox (script) .......... §A; Table A.2
\lausein (option) §1; Table 1.3
Latf (script) .......... §A; Table A.2
Latg (script) .......... §A; Table A.2
\latin (root language) §1; Table 1.2
Latn (script) §A; Table A.2; 10, 11, 16, 32, 36, 42, 51, 75, 79, 82
LB (region) .......... §A; Table A.1

122
Index

1b (ISO code) . . . . . . see luxembourgish  maltaenglish (option) §1; Table 1.3; 13, 17
LC (region) . . . . . . §A; Table A.1  maltaenglish (option) §1; Table 1.3
Leke (script) . . . . . . §A; Table A.2  maltese (root language) §1; Table 1.2
Lepc (script) . . . . . . §A; Table A.2  Mand (script) . . . . . . §A; Table A.2
1g (ISO code) . . . . . see ganda  Mani (script) . . . . . . §A; Table A.2
LI (region) . . . . . . §A; Table A.1  manx (root language) §1; Table 1.2; 68
li (ISO code) . . . . see limburgish  maori (root language) §1; Table 1.2
Limb (script) . . . . . . §A; Table A.2  marathi (root language) §1; Table 1.2
limburgish (root language) §1; Table 1.2  Marc (script) . . . . . . §A; Table A.2
Lina (script) . . . . . . §A; Table A.2  marshalleneese (root language) §1; Table 1.2
Linb (script) . . . . . . §A; Table A.2  Maya (script) . . . . . . §A; Table A.2
lingala (root language) §1; Table 1.2  MC (region) . . . . . . §A; Table A.1
Lisu (script) . . . . . . §A; Table A.2  MD (region) . . . . . . §A; Table A.1
lithuanian (root language) §1; Table 1.2  ME (region) . . . . . . §A; Table A.1
LK (region) . . . . . . §A; Table A.1  Mend (script) . . . . . . §A; Table A.2
ln (ISO code) . . . . see lingala  Merc (script) . . . . . . §A; Table A.2
lo (ISO code) . . . . see lao  Mero (script) . . . . . . §A; Table A.2
donovan (script) . . . . . . §A; Table A.2  mexicanspanish (option) §1; Table 1.3; 3
Loma (script) . . . . . . §A; Table A.2  meyalu (option) §1; Table 1.3
lowsorbian (option) §1; Table 1.3  MF (region) . . . . . . §A; Table A.1
LR (region) . . . . . . §A; Table A.1  MG (region) . . . . . . §A; Table A.1
LS (region) . . . . . . §A; Table A.1  mg (ISO code) . . . . see malagasy
lsorbian (root language) §1; Table 1.2  MH (region) . . . . . . §A; Table A.1
LT (region) . . . . . . §A; Table A.1  mh (ISO code) . . . . see marshalleneese
lt (ISO code) . . . . see lithuanian  mi (ISO code) . . . . . . see maori
LU (region) . . . . . . §A; Table A.1  MK (region) . . . . . . §A; Table A.1
lu (ISO code) . . . . see lubakatanga  MK (ISO code) . . . . see macedonian
luxembourgish (root language) §1; Table 1.2  ML (region) . . . . . . §A; Table A.1
Table 1.2  ml (ISO code) . . . . see malayalam
LV (region) . . . . . . §A; Table A.1  Mlym (script) . . . . . . §A; Table A.2
lv (ISO code) . . . . see latvian  MM (region) . . . . . . §A; Table A.1
LY (region) . . . . . . §A; Table A.1  MN (region) . . . . . . §A; Table A.1
Lyci (script) . . . . . . §A; Table A.2  mn (ISO code) . . . . see mongolian
Lydi (script) . . . . . . §A; Table A.2  MO (region) . . . . . . §A; Table A.1

M

MA (region) . . . . . . §A; Table A.1  modifier . . . . . . 20, 26, 34, 36, 41, 77, 79, 105
macedonian (root language) §1; Table 1.2  Mong (script) . . . . . . §A; Table A.2
magyar (root language) §1; Table 1.2  Moon (script) . . . . . . §A; Table A.2
Mahj (script) . . . . . . §A; Table A.2  MP (region) . . . . . . §A; Table A.1
malagasy (root language) §1; Table 1.2  MQ (region) . . . . . . §A; Table A.1
malay (option) §1; Table 1.3  MR (region) . . . . . . §A; Table A.1
malayalam (root language) §1; Table 1.2  MRoo (script) . . . . . . §A; Table A.2
Index

ms–MY (option) §1; Table 1.1
MS (region) §A; Table A.1
ms (ISO code) see bahasam
mt–MT (option) §1; Table 1.1
MT (region) §A; Table A.1
mt (ISO code) see maltese
Mtei (script) §A; Table A.2
MU (region) §A; Table A.1
Multi (script) §A; Table A.2
MV (region) §A; Table A.1
MW (region) §A; Table A.1
MX (region) §A; Table A.1
MY (region) §A; Table A.1
my (ISO code) see burmese
Mymr (script) §A; Table A.2
MZ (region) §A; Table A.1

N

NA (region) §A; Table A.1
na (ISO code) see nauruan
Narb (script) §A; Table A.2
nauruan (root language) §1; Table 1.3
naussian (option) §1; Table 1.3; 50
navajo (root language) §1; Table 1.2
nb (ISO code) see bokmal
Nbat (script) §A; Table A.2
nbelgiangerman (option) §1; Table A.3
NC (region) §A; Table A.1
nd (ISO code) see northernndebele
ndonga (root language) §1; Table 1.2
NE (region) §A; Table A.1
ne (ISO code) see nepali
nepali (root language) §1; Table 1.2
netherlands (option) §1; Table 1.3
Newa (script) §A; Table A.2
newzealand (option) §1; Table 1.3
NF (region) §A; Table A.1
NG (region) §A; Table A.1
ng (ISO code) see ndonga
ngerman (option) §1; Table 1.3; 35, 71, 78
ngerman package 2, 3, 35, 55
ngermanb (option) §1; Table 1.3
ngermanDE (option) §1; Table 1.3; 71, 78

N (root language) §A; Table A.1

NI (region) §A; Table A.1
nicaraguanspanish (option) §1; Table 1.3
nil (option) §1; Table 1.3
Nkgb (script) §A; Table A.2
nko (root language) §1; Table 1.2
Nkoo (script) §A; Table A.2
n1–BE (option) §1; Table 1.1
n1–NL (option) §1; Table 1.1
NL (region) §A; Table A.1
n1 (ISO code) see dutch
nn (ISO code) see nynorsk
northernndebele (root language) §1; Table 1.2
northernsotho (root language) §1; Table 1.2
norskk (root language) §1; Table 1.2
nsi (ISO code) see northernndebele
nsi (ISO code). see southernndebele
nsi (ISO code). see southernndebele
nsi (ISO code). see southernndebele
nsi (ISO code). see southernndebele
nsi (ISO code). see nswissgerman (option) §1; Table 1.3
nu (ISO code) see nswissgerman (option) §1; Table 1.3
nu (ISO code) see nswissgerman (option) §1; Table 1.3
nu (ISO code) see nswissgerman (option) §1; Table 1.3
nu (ISO code) see nswissgerman (option) §1; Table 1.3
nu (ISO code) see nswissgerman (option) §1; Table 1.3
nu (ISO code) see nswissgerman (option) §1; Table 1.3

N (option) §1; Table 1.3

oc (ISO code) see occitan
occitan (root language) §1; Table 1.2
ogam (script) §A; Table A.2
oj (ISO code) see ojambwe
ojambwe (root language) §1; Table 1.2
0lk (script) §A; Table A.2
om (ISO code) see oromo
or (ISO code) see or iya

124
Index

oriya (root language) ........... §1; Table 1.2
Orkh (script) ................... §A; Table A.2
oromo (root language) .......... §1; Table 1.2
Orya (script) ................... §A; Table A.2
os (ISO code) ................... see ossetian
Osge (script) ................... §A; Table A.2
Osma (script) ................... §A; Table A.2
ossetian (root language) ..... §1; Table 1.2

P

PA (region) ..................... §A; Table A.1
pa (ISO code) ........ see easternpunjabi
Pali (root language) .......... §1; Table 1.2
Palm (script) ................... §A; Table A.2
Panamaspanish (option) ...... §1; Table 1.3
paraguayspanish (option) ... §1; Table 1.3
pashto (root language) ...... §1; Table 1.2
Pauc (script) ................... §A; Table A.2
pdflatex (application) ...... 24
PE (region) ..................... §A; Table A.1
Perm (script) ................... §A; Table A.2
persian (option) .............. §1; Table 1.3
peruvianspanish (option) ... §1; Table 1.3
PF (region) ..................... §A; Table A.1
PG (region) ..................... §A; Table A.1
PH (region) ..................... §A; Table A.1
Phag (script) ................... §A; Table A.2
Phli (script) ................... §A; Table A.2
Phlp (script) ................... §A; Table A.2
Phlv (script) ................... §A; Table A.2
Phnx (script) ................... §A; Table A.2
pi (ISO code) ................... see pali
piedmontese (root language) §1; Table 1.2
piemontes (option) .......... §1; Table 1.3
Piqd (script) ................... §A; Table A.2
PK (region) ..................... §A; Table A.1
PL (region) ..................... §A; Table A.1
Pi (ISO code) ................... see polish
Pldr (script) ................... §A; Table A.2
PM (region) ..................... §A; Table A.1
PMS (ISO code) ................. see piedmontese
PN (region) ..................... §A; Table A.1
polish (root language) ...... §1; Table 1.2

polutonik (option) .......... §1; Table A.3
polutonikogreek (option) ... §1; Table A.3
polyglossia package a, 2, 3, 9, 11–13, 35, 44, 54, 58, 61, 95
portugal (option) .......... §1; Table A.3
portuges (root language) ... §1; Table A.2
portuguese (option) ......... §1; Table A.3
POSIX ......................... 23, 40, 79
PR (region) ..................... §A; Table A.1

-prefix \-localeid .ldf 15, 16, 49–55, 57, 58, 60–66, 68, 81

ProvidesFile .................. 52, 110
Pṛi (script) ................... §A; Table A.2
PS (region) ..................... §A; Table A.1
Ps (ISO code) ........ see pashto
pt-BR (option) .............. §1; Table 1.1
pt-PT (option) .............. §1; Table 1.1
PT (region) ..................... §A; Table A.1
pt (ISO code) ........ see portuges
puertoricospanish (option) ... §1; Table A.3
PW (region) ..................... §A; Table A.1
PY (region) ..................... §A; Table A.1

Q

QA (region) ..................... §A; Table A.1
Qaaa (script) ................... §A; Table A.2
Qaaı̈ (script) ................... §A; Table A.2
Qabx (script) ................... §A; Table A.2
qu (ISO code) ........ see quechua
quechua (root language) ..... §1; Table 1.2

R

RE (region) ..................... §A; Table A.1
region ................ see territory
regions.sty ................... 61, 63
\RequirePackage ............. 13, 17, 72
Rjng (script) ................... §A; Table A.2
rm–CH (option) ............. §1; Table 1.1
rm (ISO code) ........ see romansh
rn (ISO code) ........ see kirundi
RO (region) ..................... §A; Table A.1
ro (ISO code) ........ see romanian
romanche (option) .......... §1; Table 1.3

125
Index

romanian (root language) §1; Table 1.2
romansch (option) §1; Table 1.3
romansh (root language) §1; Table 1.2
Roro (script) §A; Table A.2
RS (region) §A; Table A.1
RU (region) §A; Table A.1
ru (ISO code) see russian
rumantsch (option) §1; Table 1.3
Runr (script) §A; Table A.2
russian (root language) §1; Table 1.2
russianb (option) §1; Table A.1
rW (region) §A; Table A.1
rw (ISO code) see kinyarwanda

S

SA (region) §A; Table A.1
sa (ISO code) see sanskrit
samin (root language) §1; Table 1.2
samoan (root language) §1; Table 1.2
Samr (script) §A; Table A.2
sango (root language) §1; Table 1.2
sanmarino (option) §1; Table 1.3
sanskrit (root language) §1; Table 1.2
Sara (script) §A; Table A.2
Sarb (script) §A; Table A.2
sardinian (root language) §1; Table 1.2
Saur (script) §A; Table A.2
SB (region) §A; Table A.1
SC (region) §A; Table A.1
sc (ISO code) see sardinian
scottish (root language) §1; Table 1.2
script 10, 11, 13, 15, 16, 21, 36, 43, 47, 50, 51, 61, 62
SD (region) §A; Table A.1
sd (ISO code) see sindhi
SE (region) §A; Table A.1
se (ISO code) see samin
\selectlanguage 17, 73
serbian (root language) §1; Table 1.2; 79
serbianc (option) §1; Table 1.3
serbianl (option) §1; Table 1.3
\setCurrentTrackedDialect §6.2; 10, 18, 45, 73–75, 96–98, 101
\SetTrackedDialectAdditional §6.6; 80, 102
\SetTrackedDialectLabelMap §6.6; 12, 18, 74, 78, 82, 102
\SetTrackedDialectModifier §6.6; 34, 78, 79, 102
\SetTrackedDialectScript §6.6; 79, 102
\SetTrackedDialectSubLang §6.6; 80, 102
\SetTrackedDialectVariant §6.6; 79, 80, 102

SG (region) §A; Table A.1
sg (ISO code) see sango
Sgnw (script) §A; Table A.2
SH (region) §A; Table A.1
Shaw (script) §A; Table A.2
shell escape a, 10, 22–25, 27, 94, 110
shellesc package 94
shona (root language) §1; Table 1.2
Shrd (script) §A; Table A.2
SI (region) §A; Table A.1
si (ISO code) see sinhalese
Sidd (script) §A; Table A.2
Sind (script) §A; Table A.2
sindhi (root language) §1; Table 1.2
Sinh (script) §A; Table A.2
sinhalese (root language) §1; Table 1.2
SJ (region) §A; Table A.1
SK (region) §A; Table A.1
sk (ISO code) see slovak
s1-SI (option) §1; Table 1.1
SL (region) §A; Table A.1
s1 (ISO code) see slovene
slovak (root language) §1; Table 1.2
slovene (root language) §1; Table 1.2
sloveneistriaitalian (option) §1;
Table 1.3
tsloveneistriaslovak (option) §1;
Table 1.3
tslovenia (option) §1; Table 1.3	slovenian (option) §1; Table 1.3
SM (region) §A; Table A.1
sm (ISO code) see samoan
SN (region) §A; Table A.1
sn (ISO code) see shona
Index

S0 (region) ........................................... §A; Table A.1
s0 (ISO code) ........................................ see somali
somali (root language) §1; Table 1.2
tahitian (root language) §1; Table 1.2
Sora (script) §A; Table A.2
tai (root language) §1; Table 1.2
southernndebele (root language) §1; Table 1.2
tajik (root language) §1; Table 1.2
southernsotho (root language) §1;
Table 1.2
takr (script) §A; Table A.2
tale (script) §A; Table A.2
talu (script) §A; Table A.2
tamil (root language) §1; Table 1.2
tam (script) §A; Table A.2
tang (script) §A; Table A.2
tatar (root language) §1; Table 1.2
tavt (script) §A; Table A.2
telugu (root language) §1; Table 1.2
tensor 20, 26, 30, 31, 35, 36, 50, 61, 62, 105,
107, 108, 110
tex (application) ...................................... 25
texosquery.cfg ........................................ a
texosquery.tex ........................................ 9, 10
texosquery (application) ................................ a, 10
texosquery package .................................... 10, 22, 23, 26
\TeXOSQueryLangTag ................................ 10, 14, 22
\TeXOSQueryLocale ................................... 23, 26
TF (region) .......................................... §A; Table A.1
tfng (script) §A; Table A.2
TG (region) .......................................... §A; Table A.1
tg (ISO code) ........................................ see tajik
tglg (script) §A; Table A.2
TH (region) .......................................... §A; Table A.1
th (ISO code) ........................................ see thai
tha (script) §A; Table A.2
Thai (script) §A; Table A.2
thai (root language) §1; Table 1.2
\ThreeLetterExtIsoLanguageCode §5;
40, 102
\ThreeLetterIsoLanguageCode §5;
40, 103
ti (ISO code) ........................................ see tigrinya
tibetan (root language) §1; Table 1.2

T

ta (ISO code) ........................................ see tamil
Index

Tibt (script) ...................................... §A; Table A.2
 tigrinya (root language) ........................... §1; Table 1.2
Tirh (script) ........................................ §A; Table A.2
 TJ (region) ........................................... §A; Table A.1
 TK (region) ........................................... §A; Table A.1
 tk (ISO code) ...................................... see turkmen
 TL (region) ........................................... §A; Table A.1
tl (ISO code) ...................................... see tagalog
 TM (region) ........................................... §A; Table A.1
 TN (region) ........................................... §A; Table A.1
 tn (ISO code) ...................................... see tswana
 to (ISO code) ...................................... see tonga
 TR (region) ........................................... §A; Table A.1
 tracklang.sty ...................................... see tracklang
 tracklang.tex ...................................... §4; 11, 31–33, 43, 73, 75, 104, 111–113
 tracklang-scripts.sty .............................. see tracklang-scripts
 tracklang-scripts.tex .............................. §4; 11, 31–33, 43, 73, 75, 104, 111–113
 tracklang package ................................ a, 9, 19, 30, 35, 72, 95
 tracklang-region-codes.tex ................. §4; 30, 31, 73, 75, 104, 107–110
 tracklang-scripts package ...................... a, 9, 19, 30, 35, 72, 95
 TrackLangAddExtraRegionFile .................. §4; 31, 73, 76, 103
 TrackLangAddExtraScriptFile ................... §4; 33, 73, 75, 104
 TrackLangAddToCaptions ......................... §5; 15, 16, 55, 70, 78, 104
 TrackLangAlphaIIIToNumericRegion .......... §4; 30, 104
 TrackLangAlphaIIIToNumericRegion .......... §4; 30, 104
 TrackLangDeclareDialectOption .............. 104
 TrackLangDeclareLanguageOption .......... §5; 25–27, 105, 109, 113
 TrackLangEnv ...................................... §3; 26, 105
 TrackLangEnvCodeSet ............................ §3; 26, 105
 TrackLangEnvLang ................................ §3; 26, 105
 TrackLangEnvModifier ......................... §3; 26, 105
 TrackLangEnvTerritory ......................... §3; 26, 105
 TrackLangFromEnv ................................ §3; 9, 10, 14, 23, 25–28, 34, 40, 81, 82, 105
 TrackLangGetDefaultScript ................... §5; 43, 106
 TrackLangGetKnownCountry ................. §5; 106
 TrackLangGetKnownIsoThreeLetterLang ......... 106
 TrackLangGetKnownIsoThreeLetterLangB ........ 106
 TrackLangGetKnownIsoTwoLetterLang .......... 106
 TrackLangGetKnownLangFromIso ............ §5; 82, 108
 TrackLangGetKnownAlphaIIRegion ........... §4; 31, 107
 TrackLangIfAlphaNumericChar ............... §6; 106
 TrackLangIfHasDefaultScript ............... §6; 106
 TrackLangIfKnownCountry ................... §6; 107
 TrackLangIfKnownIsoThreeLetterLang ......... 107
 TrackLangIfKnownIsoThreeLetterLangB ........ 107
 TrackLangIfKnownIsoTwoLetterLang ........... 107
 TrackLangIfLanguageTag ..................... 108
 TrackLangIfRegionTag ......................... 108

128
ukraine (option) §1; Table 1.3
ukraineanb (option) §1; Table 1.3
ukrainian (root language) §1; Table 1.2
UM (region) §A; Table A.1
und (ISO code) see undetermined undetermined (root language) §1; Table 1.2, 40, 97
uppersorbian (option) §1; Table 1.3
ur (ISO code) see urdu
urdu (root language) §1; Table 1.2
uruguayspanish (option) §1; Table 1.3
US (region) §A; Table A.1
USenglish (option) §1; Table 1.3; 71
usorbian (root language) §1; Table 1.2
UY (region) §A; Table A.1
uyghur (root language) §1; Table 1.2
UZ (region) §A; Table A.1
uz (ISO code) see uzbek
uzbek (root language) §1; Table 1.2

V

VA (region) §A; Table A.1
Vaii (script) §A; Table A.2
valencian (option) §1; Table 1.3
valencien (option) §1; Table 1.3
variant 10, 15, 20, 36, 47, 50, 77, 79
vatican §1; Table 1.3
VC (region) §A; Table A.1
VE (region) §A; Table A.1
ve (ISO code) see venda
venda (root language) §1; Table 1.2
venezuelanspanish (option) §1; Table 1.3
VG (region) §A; Table A.1
VI (region) §A; Table A.1
vi (ISO code) see vietnamese
vietnamese (root language) §1; Table 1.2
Visp (script) §A; Table A.2
VN (region) §A; Table A.1
vo (ISO code) see volapuk
volapuk (root language) §1; Table 1.2
VU (region) §A; Table A.1

W

wa (ISO code) see walloon
walloon (root language) §1; Table 1.2
Wara (script) §A; Table A.2
welsh (root language) §1; Table 1.2
westernfrisian (root language) §1; Table 1.2

X

Xf (region) §A; Table A.1
xh (ISO code) see xhosa
xhosa (root language) §1; Table 1.2
Xpeo (script) §A; Table A.2
\xpg@bcp@loaded 2, 11, 35
\xpg@loaded 35
Xsux (script) §A; Table A.2

Y

YE (region) §A; Table A.1
yi (ISO code) see yiddish
yiddish (root language) §1; Table 1.2
Yiii (script) §A; Table A.2
yo (ISO code) see yoruba
yoruba (root language) §1; Table 1.2
YT (region) §A; Table A.1

Z

ZA (region) §A; Table A.1
za (ISO code) see zhuang
zh (ISO code) see chinese
zhuang (root language) §1; Table 1.2
Zinh (script) §A; Table A.2
Zsy (script) §A; Table A.2
Zsym (script) §A; Table A.2
zu (ISO code) see zulu
zulu (root language) §1; Table 1.2
ZW (region) §A; Table A.1
Zxxx (script) §A; Table A.2
Zyy (script) §A; Table A.2
Zzzz (script) §A; Table A.2