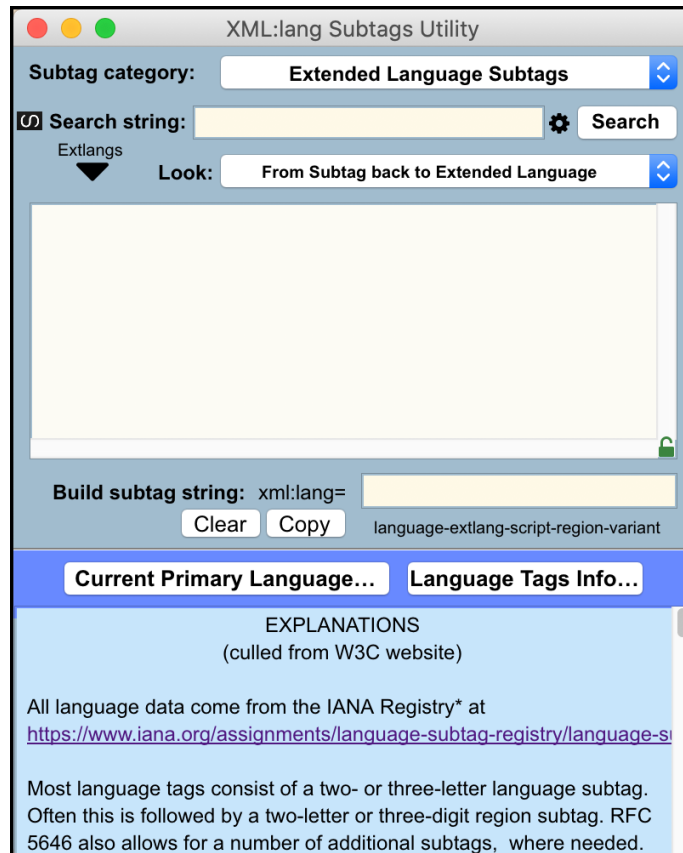


The XML:Lang Utility



Illustrated User Guide

THE XML:LANG UTILITY

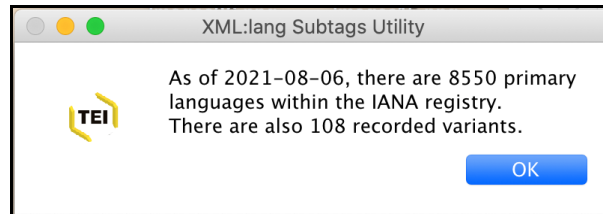
Table of Contents

1. Purpose	3
2. Exploring the Subtag Category Menu.....	4
2.a. Conducting a search.....	4
2.b. Displaying the registry’s subtag info	5
2.c. Updating the registry.....	6
3. The Subtag Identification Menu.....	6
4. Identifying Primary Language Subtags.....	7
4.a Looking up the name of a primary language from its subtag.....	7
4.b Learning more about a primary language through Glottolog.....	7
5. Identifying Extended Language Subtags.....	8
6. Identifying Language Script Subtags.....	10
7. Identifying Country or Region Subtags.....	10
8. Identifying Variant Idiom or Dialect Subtags.....	11
9. Exploring Registry Subtag Comments.....	12
10. Exploring Macrolanguages	14
11. Exploring Language Collections	15
12. Exploring the List of “Grandfathered” or Deprecated Tags.....	17
13. Exploring the List of “Redundant” Tags	18
14. Summary of rules for encoding @xml:lang values	20

The XML:LANG UTILITY

1. Purpose

Welcome to this special application. It may look small, but it packs a punch:



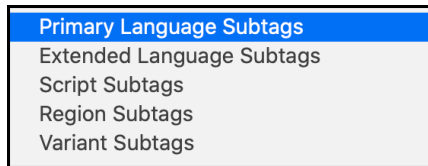
Indeed, choosing the command **Show Number of Primary Languages** in the middle of the app’s Subtag Category menu reveals the fact above: this utility allows users to navigate no fewer than 8,550 “primary languages,” not counting the “grandfathered” and “redundant” ones, as recorded within the IANA (Internet Assigned Numbers Authority)’s Language Subtags Registry. The W3C’s internationalization effort recommends the use of the IANA Registry for selecting codes for languages. That Registry—a work in progress for there are many more languages that linguists and ethnologists have yet to agree upon—depends on ISO 639-3 codes for languages which did not previously have codes in other parts (-1 and -2) of the ISO 639 standard.

The registry is downloaded within the app and turned into multiple searchable arrays. Those arrays allow users to figure out, for instance, what international standard-based language codification needs to be used when representing, via encoding, the language in which some text is written or some speech performed. Accuracy in this linguistic matter benefits electronic text processing, online searches, and scholarship at large.

In the XML universe, it is the attribute “xml:lang” that has been designed to host such language code within its values. The word “pain” for instance, if encoded `<w xml:lang="fr">pain</w>`, is a French word that means “bread” in English but, if encoded `<w xml:lang="en">pain</w>`, is an English word that means “douleur” in French. The IANA registry allows users to indicate the language of a text most precisely: an Ancient Greek word (up to 1453) will call for the “grc” code, while a modern Greek word will need the “el” code—not to mention Cappadocian Greek (“cpg”), Mycenaean Greek (“gmy”), and Romano-Greek (“rge”). A performance done in Greek sign language would be encoded “gss”, and even more precisely, though not indispensably, “sgn-gss”.

The app allows users to figure this out accurately and conveniently. It also allows users to explore the vast world of languages, dead or alive, discovering such interesting things as the existence of 209 kinds of scripts, of which our common Latin script (“Latn”) is but one, with two brothers: “Latf” for the Fraktur variant, and “Latg” for the Gaelic variant.

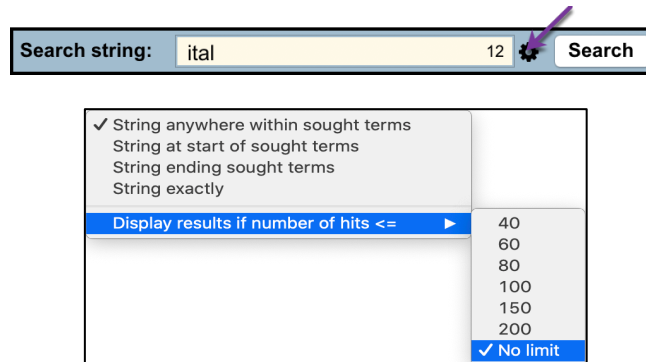
2. Exploring the Subtag Category Menu



The first menu at the top of the utility allows users to select any one of five ordered components, called “subtags,” when constructing an xml:lang entity, each corresponding to a distinct category: **language – extlang – script – region – variant**. The app knows the order in which each item needs to be sequenced. In practice, no one ever needs all five components to be declared: the fewer the better. Skipping intermediate components is legal. The goal is to supply the minimal information that is sufficient to avoid any ambiguity. It is for instance completely unnecessary to specify the script (-Latn) used in a French (fr) text because French is always written in the Latin script. Hence: “fr” suffices, and “fr-Latn” is proscribed (the database lists "Latn" under the label “Suppress-Script” in its characterization of the French language).

a. Conducting a search

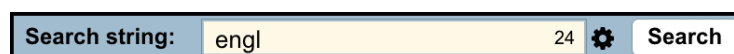
To conduct a search, the user first selects one of the five types of subtags in the pull-down menu: **primary language, extended language, script, region, and variant**. Note that such a search can be constrained using the popup menu that gets displayed when clicking the little gear icon. The choices are displayed below. Not every command in the menus is constrained by it, though.



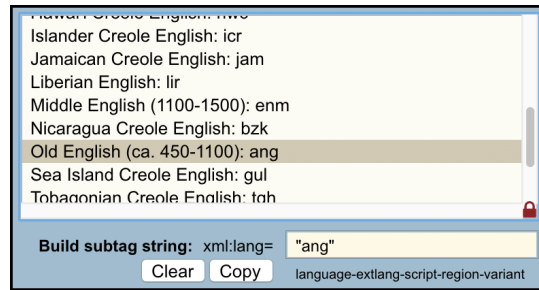
Search constraint menu (cog icon)

Sometimes if a search doesn’t yield much or anything at all, check whether it was being constrained too much.

Then the user types a telling string of characters in the Search field (“engl” for instance) and either tab, hit the return key, or click the Search button.



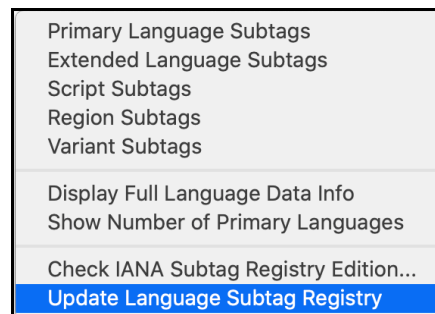
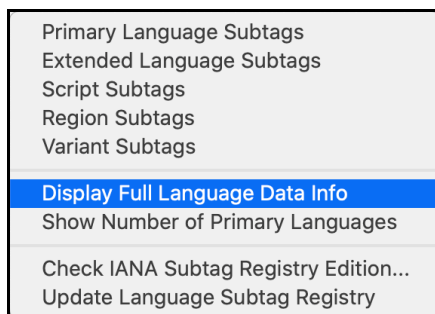
The results will be displayed in the field just below it (24 entries):



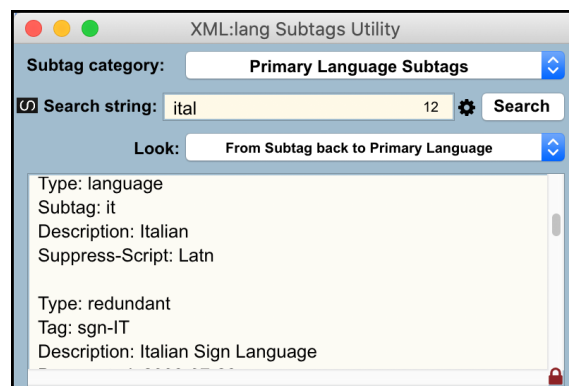
Clicking any line enters the language subtag ("ang" in this case) in the subtag string field.

b. Displaying the registry's subtag info

Other useful commands are also available in that top menu, as shown below.



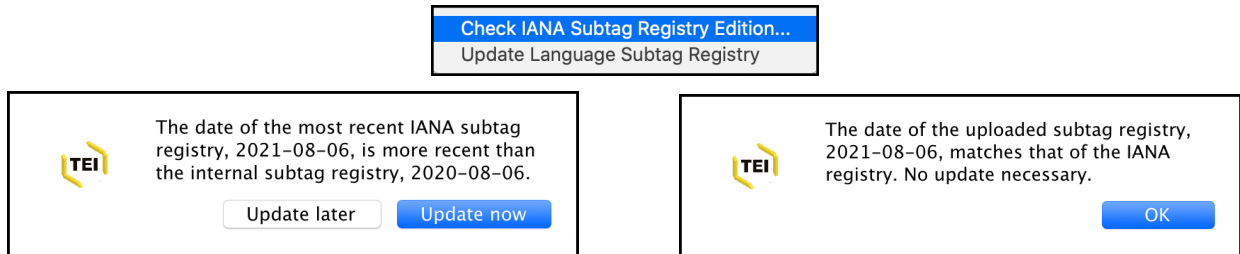
The command **Display Full Language Info** is pretty handy. Type any string in the search field, for instance "ital", and then choose that command. That command's search is not constrained, and the operation will yield 12 entries.



Each entry displayed by that command shows the IANA representation of a language, using such labels as Type, Tag, Subtag, Description, Suppress-Script, Scope, Prefix, Deprecated, Preferred-Value, and Comments.

c. Updating the registry

The other two commands, **Check IANA Subtag Registry Edition...** and **Update Language Subtag Registry**, acknowledge the fact that the IANA updates its language subtag registry from time to time. The utility allows you to check whether there is a newer registry and if so to update the app's internal registry in just a few seconds.



3. The Subtag Identification Menu

The second menu allows users to identify what a subtag they may have encountered in some encoded document stands for. It helps to know that primary language subtags are either two-letter or three-letter long; that extlang subtags are always three-letter long; that script subtags are always four-letter long; that region subtags consist either of two-letter codes or three-digit numeric codes (for larger regions); and that variant subtags (which indicate dialects or script variations only specialists are concerned with) are generally 5 to 8-character long.

Here is the full **Subtag identification menu**. It is full of commands, the totality of which enables users to explore the IANA registry and take advantage of its classifications to the full.

From Subtag back to Primary Language
From Subtag back to Extended Language
From Subtag back to Language Script
From Subtag back to Country or Region
From Subtag back to Variant Idiom or Dialect
Display Subtag Comment If Any
List of Extlang Subtags
List of Script Subtags
List of Country/Region Subtags
From Macrolanguage Subtag to Language Name
List of Macrolanguage Subtags
From Collection Subtag to Language Name
List of Language Collection Subtags
From Subtag Prefix to Variants needing it
List of Prefixed Subtags
List of Grandfathered Subtags
List of Redundant Subtags

4. Identifying Primary Language Subtags

It is one thing to use the Primary Language Subtags command in the Subtag Category Menu to search the registry, it is another to properly identify and recognize a subtag.


a. Looking up the name of a primary language from its subtag


To use that menu, users type a subtag, say "seo" in the search box and then choose the first command, **From Subtag back to Primary Language**.

The result indicates that "seo" is the code for the Suarmin language. Typing any subtag correctly will work in the same way: type the string, and then select the command.

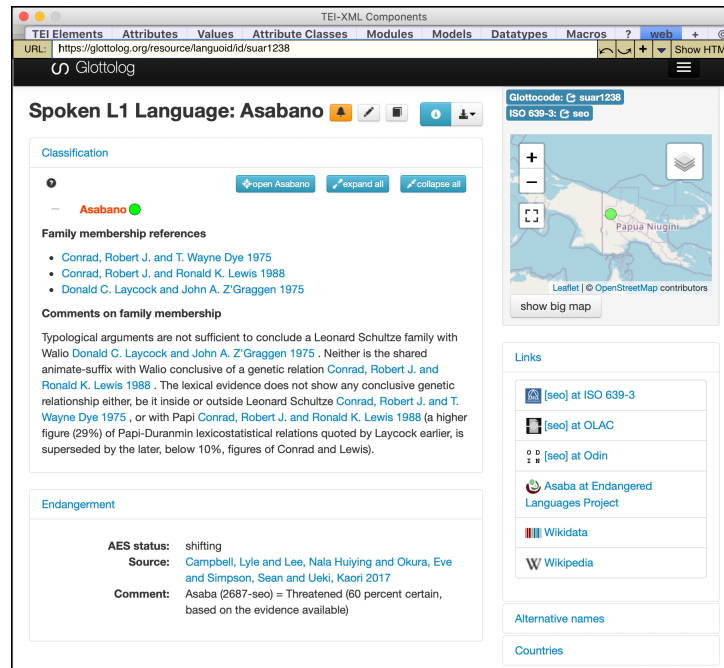
b. Learning more about a primary language through Glottolog

What is “Suarmin” though? What kind of language is it? Where is it spoken in the world? In what location that language is spoken cannot be derived directly from the registry. But no matter! There are two ways of conducting that search from this app.

(1) While "seo" is in the search box, click the button  on the left side (it's the Glottolog logo). The app sends a request to Glottolog.org regarding that symbol (an ISO 639-3 code). It then sends you to TEI-XML Components' internal web browser to reveal the following page:

(2) Or the user types the language name “Suarmin” in the search box and click . This time the app figures out what is Glottolog's code (suar128) for that language via a hidden Wikipedia search, and then sends a distinct query to Glottolog.org. This time a more

informative page is shown in the internal browser (the same one revealed if the user had clicked the language name “Asabano” directly in the webpage shown above):

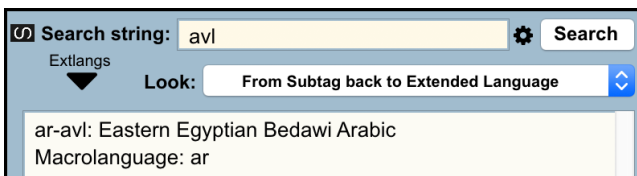


Further inquiry in Glottolog (<https://glottolog.org/glottolog/language>) explains that Suarmin is a threatened language—spoken by only about 145 native speakers—and is better known under the name Asaba, as well as Duranmin, Akiapmin, and Wani. The language is spoken in the Telefomin District of Sandaun Province in Papua New Guinea.

This app allows therefore users to expand their knowledge about the languages of the world much beyond their XML encodings!

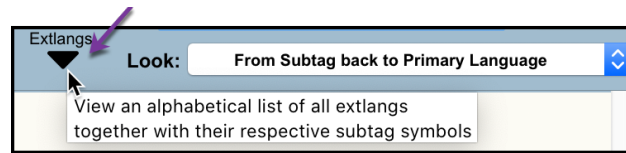
5. Identifying Extended Language Subtags

Typing the extlang subtag "avl" and then selecting the command **From Subtag back to Extended Language** yields the following result: it tells us that when the extlang "avl" is prefixed with the macrolanguage symbol "ar" we are dealing with a special form of Arabic: Eastern Egyptian Bedawi.

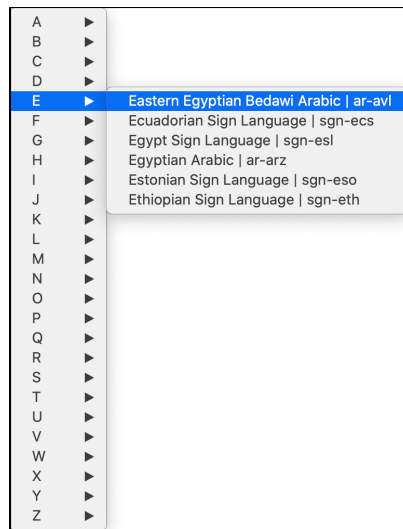


← Clicking  will confirm this fact online.

NOTE that when either that command or the command **List of Extlang Subtags** is selected in this menu, or the command **Extended Language Subtags** is selected in the top menu, something else of interest occurs at the same time:

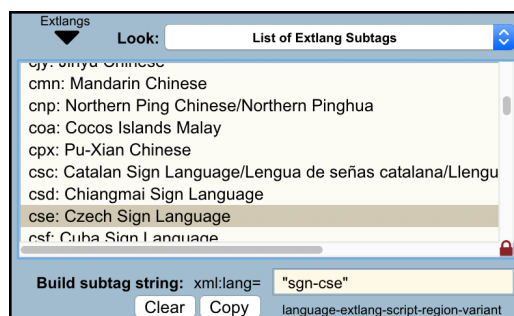


This down-arrow indicates the presence of a pop-up menu full of informative data. The content of that menu varies according to the selected category of subtags: extlangs, scripts, regions, and variants. It may also load data concerning macrolanguages, grandfathered subtags, and redundant subtags when their related commands are selected. The tooltip content will vary accordingly. In the present case we are considering the extlang-related content.



Clicking the down arrow pops up the menu above, exploration of which shows a multitude of languages whose codification requires a prefix followed by an extlang subtag. Most prominent in that category are the many sign languages: all of them are signaled by the language collection symbol "sgn" followed by a hyphen and a 3-character extlang denoting the language.

As to the command **List of Extlang Subtags**, selecting it displays the list of 245 three-character codes, each followed by their respective language names. Clicking any line will enter the related code inside the subtag string box.



6. Identifying Language Script Subtags

Typing the script subtag "Pcun" and then selecting the command **From Subtag back to Language Script** yields the following clarification: that such a subtag stands for the Proto-Cuneiform script. The Glottolog symbol is hidden from view since that resource is not concerned with scripts.

The screenshot shows the 'Script Subtags' section of the utility. The 'Subtag category' is set to 'Script Subtags'. The 'Search string' is 'Pcun'. The 'Look:' dropdown is set to 'From Subtag back to Language Script'. The search results display 'Pcun: Proto-Cuneiform'.

NOTE that when either that command or the command **List of Script Subtags** is selected in this menu, or the command **Script Subtags** is selected in the top menu, the popup menu triggered by the down arrow button loads itself with script-related data: a list of all script names followed by their 4-letter code.

The screenshot shows the 'Look:' dropdown menu. The selected option is 'From Subtag back to Primary Language'. A tooltip is visible, stating: 'View an alphabetical list of all scripts together with their respective subtag symbols'.

A	▶	
B	▶	
C	▶	
D	▶	
E	▶	Egyptian demotic Egyd
F	▶	Egyptian hieratic Egyh
G	▶	Egyptian hieroglyphs Egyg
H	▶	Elbasan Elba
I	▶	Elymaic Elym
J	▶	Ethiopic Ethi
K	▶	
L	▶	
M	▶	
N	▶	
O	▶	

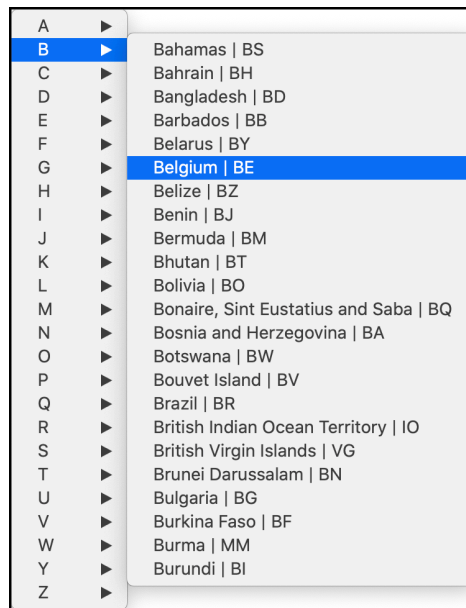
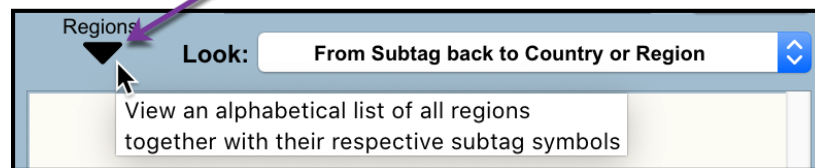
As to the command **List of Script Subtags** in that same menu, selecting it reversely displays that same list of 209 three-character codes, each followed by their respective language names. Clicking any line will enter the related code inside the subtag string box.

7. Identifying Country or Region Subtags

Typing the script subtag "BE" and then selecting the command **From Subtag back to Country or Region** yields the following clarification: that such a subtag stands for Belgium. The Glottolog symbol is hidden from view since that resource is not concerned with scripts.

The screenshot shows the 'Region Subtags' section of the utility. The 'Subtag category' is set to 'Region Subtags'. The 'Search string' is 'BE'. The 'Look:' dropdown is set to 'From Subtag back to Country or Region'. The search results display 'BE: Belgium'.

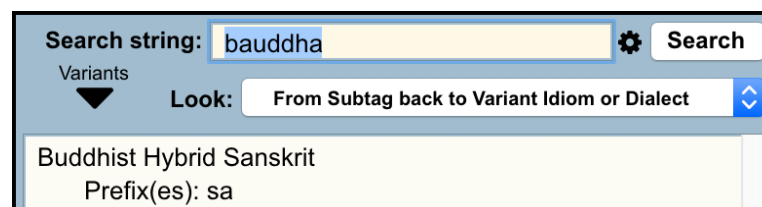
Note that when either that command or the command **List of Country/ Region Subtags** is selected in this menu, or the command **Region Subtags** is selected in the top menu, the popup menu triggered by the down arrow button loads itself with region-related data: a list of all countries followed by their 2-letter code or regions followed by their 4-digit code (“World” being of course 0001).



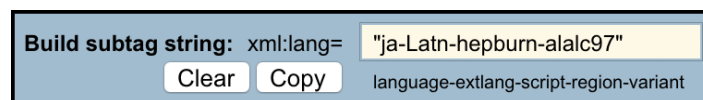
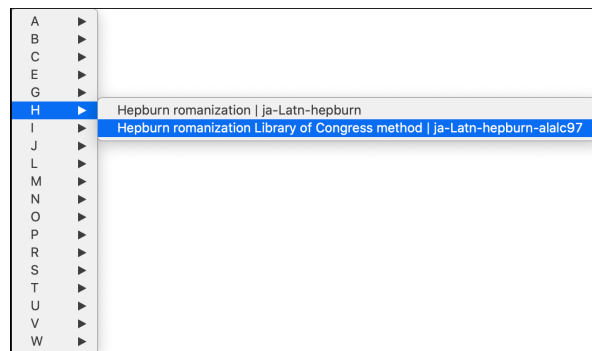
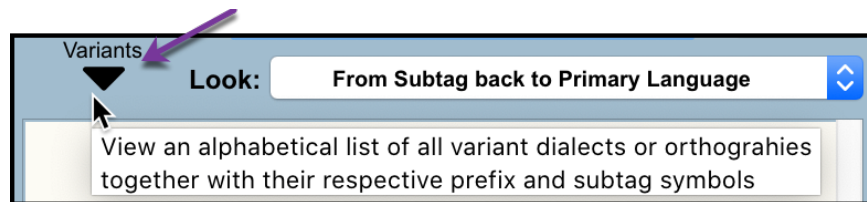
As to the command **List of Country/ Region Subtag** in that same menu, selecting it reversely displays that same list of 298 two-character or four-digit codes, each followed by their respective country or region names. Clicking any line will enter the related code inside the subtag string box.

8. Identifying Variant Idiom or Dialect Subtags

Typing the variant subtag "bauddha" and then selecting the command **From Subtag back to Country or Region** yields the following clarification: that such a subtag stands for the variety of Sanskrit known as Buddhist Hybrid Sanskrit. The Glottolog symbol is hidden from view since that resource is not concerned with variants.



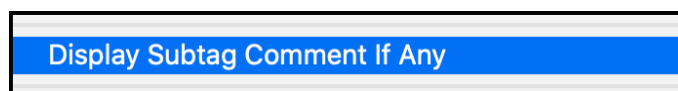
Note that when either that command, or the command **From Subtag Prefix to Variants needing it**, or the command **List of Prefixed Subtags** are selected in this menu, or the command **Variant Subtags** is selected in the top menu, the popup menu triggered by the down arrow button loads itself with variant-related data: a list of all variant dialects, idioms, or orthographies followed by their full code, including as the case may be a language prefix, an extlang, a script, and even (in only one case) a region. Choosing any choice fills the subtag string with the full code. Users need only click the adjacent **Copy** button to get hold of it.



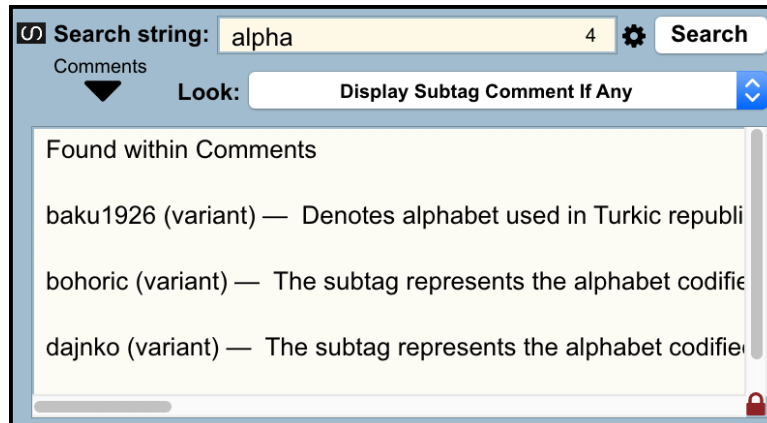
As to the command **List of Prefixed Subtags** in that same menu, selecting it reversely displays that same list of 195 variant codes, each followed by their respective country or region names. Clicking any line will enter the related code inside the subtag string box.

9. Exploring Registry Subtag Comments

The menu command **Display Subtag Comment If Any** is atypical. It comes from the fact that the IANA registry includes, for a number of language, script, region, and variant subtags (mostly), explanatory or advisory cross-referential comments that may be useful. This app allows users to search for such comments in two distinct ways.



The first method consists in typing some telling string in the search box, say “alpha”. Then select the command Display Subtag Comment If Any. If Nothing is found, a message will announce it. Otherwise the following illustration shows one example of search results: the string “alpha” is found within the word “alphabet” in three comments linked each to a variant, and the latter are displayed accordingly.

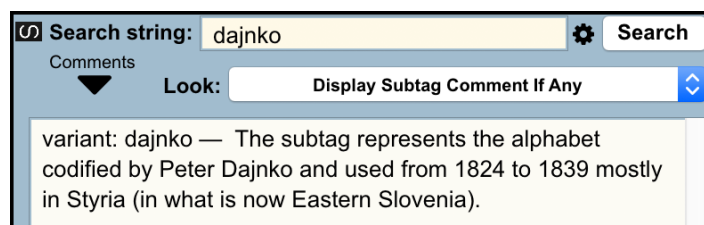


Note that clicking the lock icon in the bottom corner of the field unlocks and wraps around the field content:

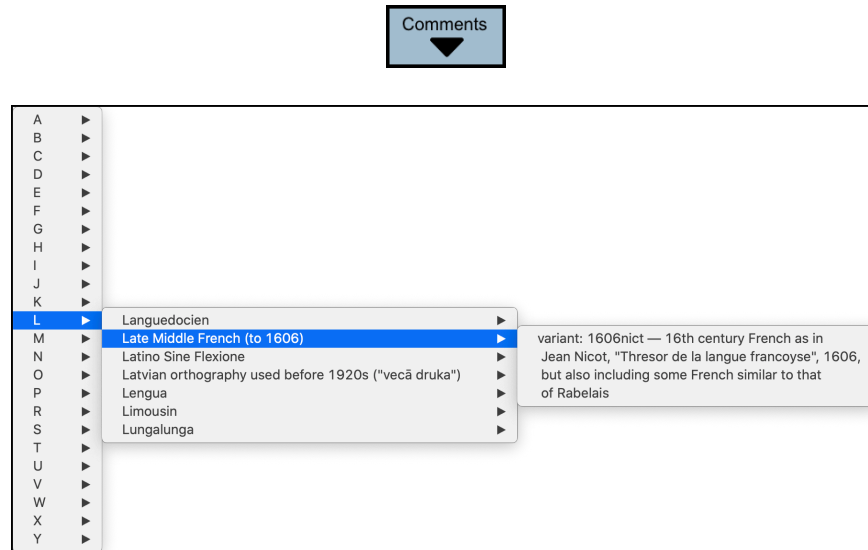


Reclick that button to lock the field and thus allow yourself to click individual lines to grab subtag codes and bring them into the subtag string box. Not every view allows this, including this particular one, for one examines comments not to select a given subtag but to learn about it.

Instead of a string, you may also type the full code of a variant, say “dajnko”, to display the comment related to it only.



The **second method** consists in using the power of the down arrow button, once more:

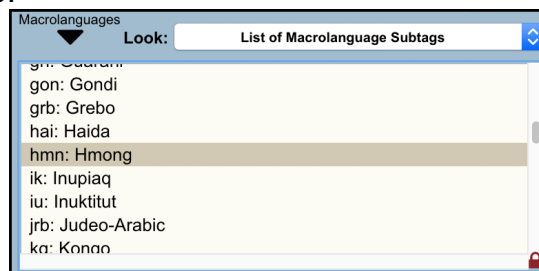


The menu displays in alphabetical order all the language names that have explanatory or advisory comments attached to them. Exploring that menu is a fascinating exercise, most profitable for anyone interested in matters of tongues.

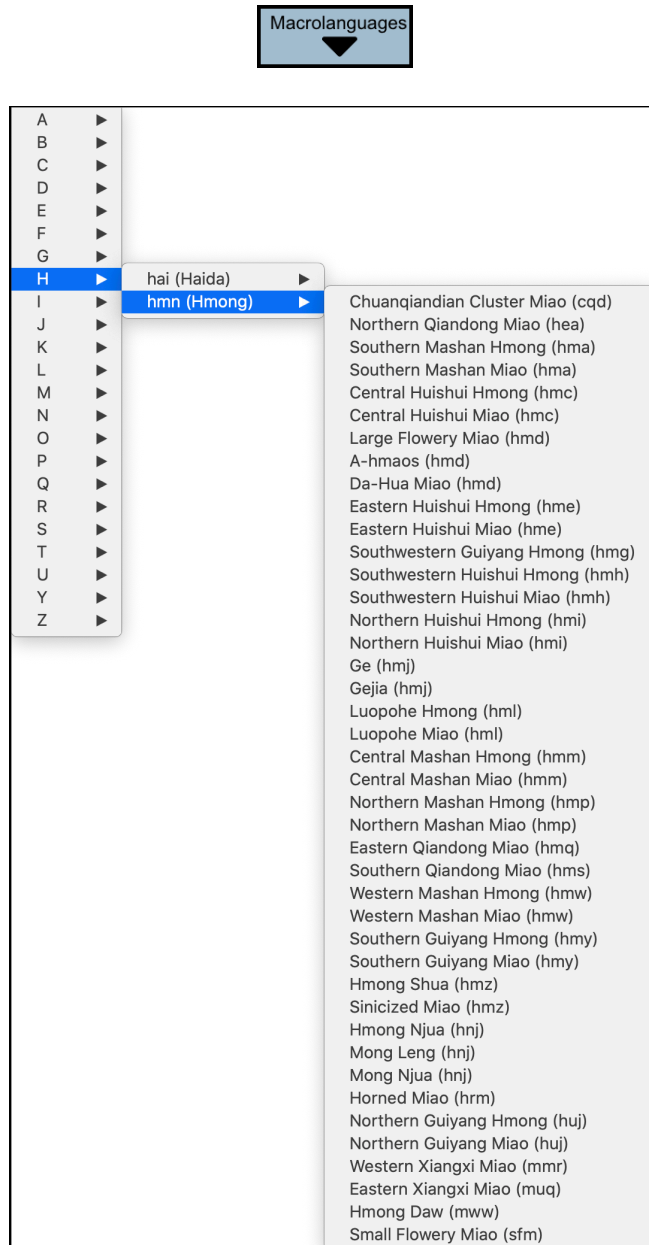
10. Exploring Macrolanguages

A “macrolanguage” yields a primary language subtag that encompasses several more specific primary languages in the IANA registry. For instance, Hmong is a macrolanguage whose subtag is "hmn". Looking up “hmn” no fewer than 42 primary languages! The latter are scattered throughout the registry BUT this app displays all of them within the arrow down menu. How so?

The first step is to choose the **List of Macrolanguage Subtags** command in the menu. This does two things: (1) All 62 macrolanguages get listed in the field, each preceded by their macrolanguage code.



And (2) the Macrolanguages menu gets loaded with information concerning each macrolanguage: it provides an alphabetic list of them all (by subtag and macrolanguage name) followed each by the list of the more specific primary languages they encompass.



Choosing any specialized primary language will display its subtag code in the subtag string box, though not preceded by the macrolanguage subtag itself as that is unnecessary. The macrolanguage subtag is useful only in situations where backward compatibility in computer processing is paramount.

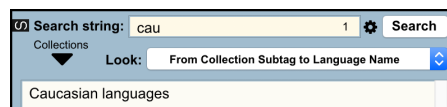
11. Exploring Language Collections

A “collection” yields a subtag that represents a group of languages that are descended from a common ancestor, are spoken in the same geographical area, or are otherwise related. As in the case of macrolanguages, users should preferably use a more specific subtag for the language they are targeting. Collection subtags, however, may be useful if

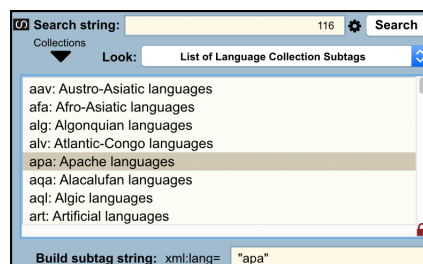
there is no more specific subtag available. It is always preferable to use one of these rather than the subtags "MUL" (multiple languages) or "UND" (undefined).

The hard fact of the matter, however, is that there is no way to use the IANA registry to figure out what are the languages that fall within any particular collection. Hence this app's limitations in that regard, but it comes with ways for compensating for it.

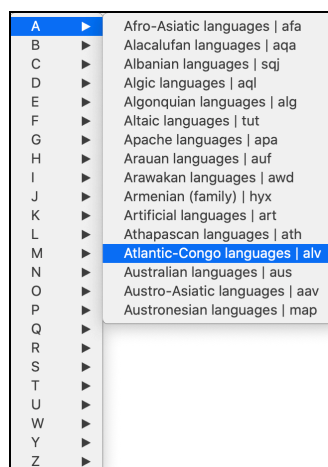
The command **From Collection Subtag to Language Name** merely checks whether a particular 3-character code matches that of a language collection and if so displays which one.





The command **List of Language Collection Subtags**, when selected, displays all 116 collections in the alphabetical order of their respective three-character subtag codes.




That command also loads the Collections menu tag with an alphabetized list of collection names followed by their respective subtag code.



But now, choosing for instance “Atlantic-Congo languages” not only inscribes its subtag code "alv" within the subtag string box, but it also inscribes “Atlantic-Congo languages” in the app's search box.


Search string:


Do click the glottolog.org icon  on the left side. In rare cases will that website provide a successful hit, given that language collections are not well recognized by linguists. But the app will. then check whether Wikipedia might not be helpful, which it often is. When that is the case, users will be directed to the corresponding Wikipedia page in TEI-XML Components' internal web browser.



The screenshot shows a browser window with the URL `https://en.wikipedia.org/wiki/Atlantic-Congo_languages`. The page title is "Atlantic-Congo languages" and it is noted as being redirected from "Atlantic-Congo languages". The main text states that the Atlantic-Congo languages are the largest demonstrated family of languages in Africa, forming the core of the Niger-Congo family hypothesis. It lists various branches like Mande, Dogon, Ijoid, Siamou, Kru, Katla, and Rashad. A sidebar on the right provides a summary table:

Atlantic-Congo	
Geographic distribution	Africa
Linguistic classification	Niger-Congo? <ul style="list-style-type: none"> Atlantic-Congo
Subdivisions	<ul style="list-style-type: none"> Talodi-Heiban? (Kordofanian) Senegambian Nalu ? Rio Nunez Mel Sua Gola Volta-Congo
ISO 639-5	aLv
Glottolog	atla1278 

Below the table is a map of Africa showing the distribution of Niger-Congo languages, with a legend for non-Atlantic-Congo, Atlantic-Congo, and other groups. The map shows a high density of languages in West and Central Africa.

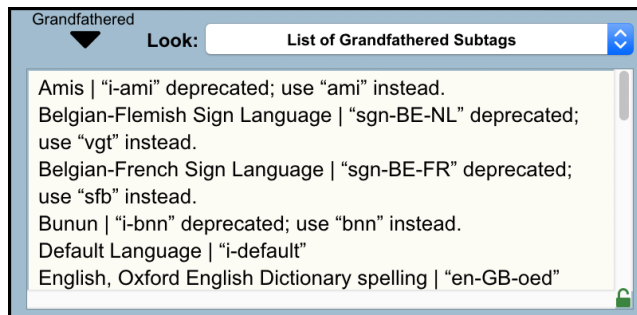
12. Exploring the List of “Grandfathered” or Deprecated Tags


Grandfathered tags are special cases provided only for backwards compatibility. Registered before RFC 4646, they are either tags that cannot be completely composed from the subtags in the current IANA registry, or tags that do not fit the syntax currently defined for language tags.

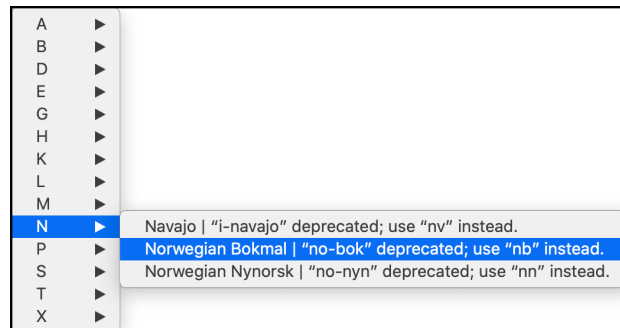
Nearly all grandfathered **tags** have been superseded by **subtags** or combinations of **subtags** in the registry. Such grandfathered **tags** are now deprecated, and their registry entry usually contains a Preferred-Value field that indicates how to represent that language instead. For instance, the registry entry for the grandfathered tag "sgn-BE-FR" indicates that the "sfb" language subtag should be used instead when referring to the Belgian-French sign language.

Important to know is that no grandfathered tag can be augmented with additional subtags: that is a completely logical corollary.

This app provides two ways of displaying such obsolete tags. One is through the command **List of Grandfathered Tags**, which, when selected, displays the list as shown in the illustration below.



The other method is of course through the “Grandfathered” pull-down menu,  .



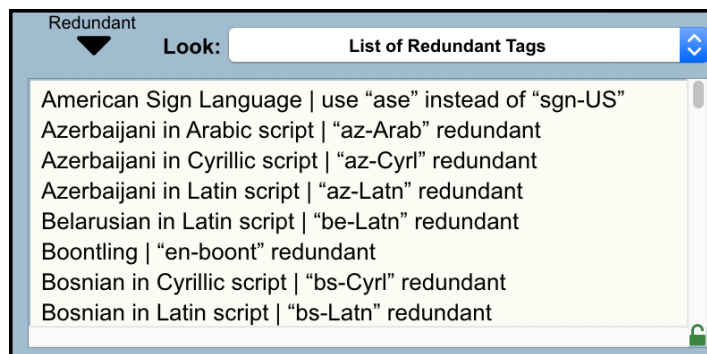
Selecting any line within this menu will lead to no action in the subtag string box since deprecated grandfathered tags need to be avoided

13. Exploring the List of “Redundant” Tags

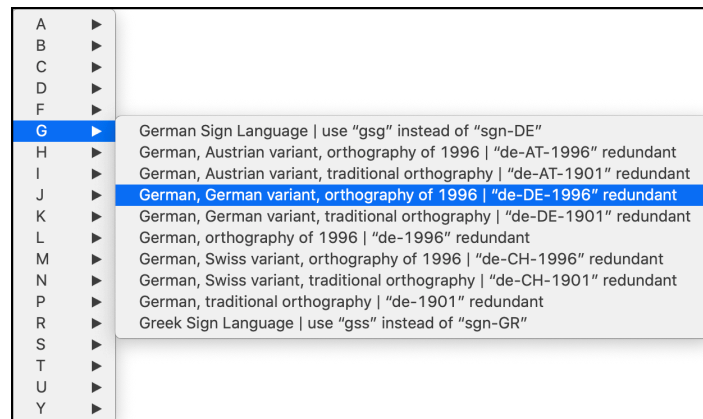
Registered before RFC 4646, so-called Redundant Tags are language **tags** composed of a sequence of subtags that can now be formed by **combining separate subtags** from the current registry. They remain in the registry mostly to satisfy historical curiosity.

A good example is the representation of Germany orthography following the revision of 1996. The way to code it is to use the **regional subtag** prefix "de" followed by the **variant subtag** "1996"—this, "de-1996". Pre RFC 4646, the “tag” was either "de-1996" or "de-DE-1996" as such. It looks nearly identical save for the regional and indeed redundant DE, but it was a “tag” and not a combination of “subtags” from distinct categories. Hence the formal, even double, redundancy.

This app provides two ways of displaying those curiously redundant tags. One is through the command **List of Redundant Tags**, which, when selected, displays the list as shown in the illustration below.



The other method is of course through the “Redundant” pull-down menu,



Selecting any line within this menu will lead to no action in the subtag string box since redundant tags need to be avoided.

14. Summary of rules for encoding @xml:lang values

(culled from W3C website)

All language data come from the IANA Registry* at

<http://www.iana.org/assignments/language-subtag-registry/language-subtag-registry>.

Most language tags consist of a two- or three-letter language subtag. Often this is followed by a two-letter or three-digit region subtag. RFC 5646 also allows for a number of additional subtags, where needed.

The golden rule when creating language tags is to keep the tag as short as possible.

Avoid region, script or other subtags except where they add useful distinguishing information.

Subtags have fixed positions and lengths.

The order of subtags is: primary language, extended language ("extlang"), script, region, and variant (separated by hyphens).

Or again: language-extlang-script-region-variant (where any item can be skipped as long as the order is respected).

More useful explanation regarding language tag choice is provided at

<https://www.w3.org/International/questions/qa-choosing-language-tags>.

EXAMPLES

Language:

French: fr

English: en

Spanish: es

Old Spanish: osp

Japanese: ja

Old Japanese: ojp

Language+region:

Dutch spoken in Sint Maarten: nl-SX

French as spoken in Canada: fr-CA

Spanish as used in Latin America: es-419

Language+script:

Chinese written with Simplified script: zh-Hans
Arabic in Old South Arabian; ar-Sarb

Language + extlang:

Gulf Arabic: ar-afb
Cantonese Chinese: zh-yue

Extlang subtags (always three letters long) must be preceded by a specific primary language subtag. There can only be one in a language tag, and it comes before any other subtags.

Script subtags (always four letters long) must immediately follow the language or any extlang subtag. There can only be one such subtag in any language tag. Add one only if it is absolutely necessary to make a distinction, for script tags are generally discouraged.

Region subtags are two-letter alpha codes (countries) or 3-digit numeric code (larger regions). They must appear (no more than one per tag) after the language subtag and any extlang and script tags. Use them only if absolutely needed.

Variant subtags indicate dialects or script variations not already covered by combinations of language, script and region subtags. They must appear after any language, script or region subtags, but script and region subtags do not need to precede them. Use variant subtags only when working in a specialized field.

* Internet Assigned Numbers Authority