Network Working Group
Request for Comments: 5646
BCP: 47
Obsoletes: 4646
Category: Best Current Practice

A. Phillips, Ed.
Lab126
M. Davis, Ed.
Google

September 2009

Tags for Identifying Languages

Abstract

This document describes the structure, content, construction, and semantics of language tags for use in cases where it is desirable to indicate the language used in an information object. It also describes how to register values for use in language tags and the creation of user-defined extensions for private interchange.

Status of This Memo

This document specifies an Internet Best Current Practices for the Internet Community, and requests discussion and suggestions for improvements. Distribution of this memo is unlimited.

Copyright Notice

Table of Contents

1. Introduction

Human beings on our planet have, past and present, used a number of languages. There are many reasons why one would want to identify the language used when presenting or requesting information.

The language of an information item or a user’s language preferences often need to be identified so that appropriate processing can be applied. For example, the user’s language preferences in a Web browser can be used to select Web pages appropriately. Language information can also be used to select among tools (such as dictionaries) to assist in the processing or understanding of content in different languages. Knowledge about the particular language used by some piece of information content might be useful or even required by some types of processing, for example, spell-checking, computer-synthesized speech, Braille transcription, or high-quality print renderings.

One means of indicating the language used is by labeling the information content with an identifier or "tag". These tags can also be used to specify the user’s preferences when selecting information content or to label additional attributes of content and associated resources.

Sometimes language tags are used to indicate additional language attributes of content. For example, indicating specific information about the dialect, writing system, or orthography used in a document or resource may enable the user to obtain information in a form that they can understand, or it can be important in processing or rendering the given content into an appropriate form or style.

This document specifies a particular identifier mechanism (the language tag) and a registration function for values to be used to

form tags. It also defines a mechanism for private use values and future extensions.

This document replaces [RFC4646] (which obsoleted [RFC3066] which, in turn, replaced [RFC1766]). This document, in combination with [RFC4647], comprises BCP 47. For a list of changes in this document, see Section 8.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

2. The Language Tag

Language tags are used to help identify languages, whether spoken, written, signed, or otherwise signaled, for the purpose of communication. This includes constructed and artificial languages but excludes languages not intended primarily for human communication, such as programming languages.

2.1. Syntax

A language tag is composed from a sequence of one or more "subtags", each of which refines or narrows the range of language identified by the overall tag. Subtags, in turn, are a sequence of alphanumeric characters (letters and digits), distinguished and separated from other subtags in a tag by a hyphen ("-", [Unicode] U+002D).

There are different types of subtag, each of which is distinguished by length, position in the tag, and content: each subtag’s type can be recognized solely by these features. This makes it possible to extract and assign some semantic information to the subtags, even if the specific subtag values are not recognized. Thus, a language tag processor need not have a list of valid tags or subtags (that is, a copy of some version of the IANA Language Subtag Registry) in order to perform common searching and matching operations. The only exceptions to this ability to infer meaning from subtag structure are the grandfathered tags listed in the productions ’regular’ and ’irregular’ below. These tags were registered under [RFC3066] and are a fixed list that can never change.

The syntax of the language tag in ABNF [RFC5234] is:

```
Language-Tag  = langtag             ; normal language tags
              / privateuse          ; private use tag
              / grandfathered       ; grandfathered tags
```

```
langtag       = language
                ["-" script]
                ["-" region]
                *("-" variant)
                *("-" extension)
                ["-" privateuse]

language      = 2*3ALPHA            ; shortest ISO 639 code
                ["-" extlang]       ; sometimes followed by
                                    ; extended language subtags
              / 4ALPHA              ; or reserved for future use
              / 5*8ALPHA            ; or registered language subtag

extlang       = 3ALPHA              ; selected ISO 639 codes
                *2("-" 3ALPHA)      ; permanently reserved

script        = 4ALPHA              ; ISO 15924 code

region        = 2ALPHA              ; ISO 3166-1 code
              / 3DIGIT              ; UN M.49 code

variant       = 5*8alphanum         ; registered variants
              / (DIGIT 3alphanum)

extension     = singleton 1*("-" (2*8alphanum))

                                    ; Single alphanumerics
                                    ; "x" reserved for private use
singleton     = DIGIT               ; 0 - 9
              / %x41-57             ; A - W
              / %x59-5A             ; Y - Z
              / %x61-77             ; a - w
              / %x79-7A             ; y - z

privateuse    = "x" 1*("-" (1*8alphanum))

grandfathered = irregular           ; non-redundant tags registered
              / regular             ; during the RFC 3066 era

irregular     = "en-GB-oed"         ; irregular tags do not match
              / "i-ami"             ; the 'langtag' production and
              / "i-bnn"             ; would not otherwise be
              / "i-default"         ; considered 'well-formed'
              / "i-enochian"        ; These tags are all valid,
              / "i-hak"             ; but most are deprecated
              / "i-klingon"         ; in favor of more modern
              / "i-lux"             ; subtags or subtag
              / "i-mingo"           ; combination
```

```
               / "i-navajo"
               / "i-pwn"
               / "i-tao"
               / "i-tay"
               / "i-tsu"
               / "sgn-BE-FR"
               / "sgn-BE-NL"
               / "sgn-CH-DE"

 regular       = "art-lojban"        ; these tags match the 'langtag'
               / "cel-gaulish"       ; production, but their subtags
               / "no-bok"            ; are not extended language
               / "no-nyn"            ; or variant subtags: their meaning
               / "zh-guoyu"          ; is defined by their registration
               / "zh-hakka"          ; and all of these are deprecated
               / "zh-min"            ; in favor of a more modern
               / "zh-min-nan"        ; subtag or sequence of subtags
               / "zh-xiang"

 alphanum      = (ALPHA / DIGIT)     ; letters and numbers
```

Figure 1: Language Tag ABNF

For examples of language tags, see Appendix A.

All subtags have a maximum length of eight characters. Whitespace is not permitted in a language tag. There is a subtlety in the ABNF production 'variant': a variant starting with a digit has a minimum length of four characters, while those starting with a letter have a minimum length of five characters.

Although [RFC5234] refers to octets, the language tags described in this document are sequences of characters from the US-ASCII [ISO646] repertoire. Language tags MAY be used in documents and applications that use other encodings, so long as these encompass the relevant part of the US-ASCII repertoire. An example of this would be an XML document that uses the UTF-16LE [RFC2781] encoding of [Unicode].

2.1.1. Formatting of Language Tags

At all times, language tags and their subtags, including private use and extensions, are to be treated as case insensitive: there exist conventions for the capitalization of some of the subtags, but these MUST NOT be taken to carry meaning.

Thus, the tag "mn-Cyrl-MN" is not distinct from "MN-cYRL-mn" or "mN-cYrL-Mn" (or any other combination), and each of these variations

conveys the same meaning: Mongolian written in the Cyrillic script as used in Mongolia.

The ABNF syntax also does not distinguish between upper- and lowercase: the uppercase US-ASCII letters in the range ’A’ through ’Z’ are always considered equivalent and mapped directly to their US-ASCII lowercase equivalents in the range ’a’ through ’z’. So the tag "I-AMI" is considered equivalent to that value "i-ami" in the ’irregular’ production.

Although case distinctions do not carry meaning in language tags, consistent formatting and presentation of language tags will aid users. The format of subtags in the registry is RECOMMENDED as the form to use in language tags. This format generally corresponds to the common conventions for the various ISO standards from which the subtags are derived.

These conventions include:

- [ISO639-1] recommends that language codes be written in lowercase (’mn’ Mongolian).

- [ISO15924] recommends that script codes use lowercase with the initial letter capitalized (’Cyrl’ Cyrillic).

- [ISO3166-1] recommends that country codes be capitalized (’MN’ Mongolia).

An implementation can reproduce this format without accessing the registry as follows. All subtags, including extension and private use subtags, use lowercase letters with two exceptions: two-letter and four-letter subtags that neither appear at the start of the tag nor occur after singletons. Such two-letter subtags are all uppercase (as in the tags "en-CA-x-ca" or "sgn-BE-FR") and four-letter subtags are titlecase (as in the tag "az-Latn-x-latn").

Note: Case folding of ASCII letters in certain locales, unless carefully handled, sometimes produces non-ASCII character values. The Unicode Character Database file "SpecialCasing.txt" [SpecialCasing] defines the specific cases that are known to cause problems with this. In particular, the letter ’i’ (U+0069) in Turkish and Azerbaijani is uppercased to U+0130 (LATIN CAPITAL LETTER I WITH DOT ABOVE). Implementers SHOULD specify a locale-neutral casing operation to ensure that case folding of subtags does not produce this value, which is illegal in language tags. For example, if one were to uppercase the region subtag ’in’ using Turkish locale rules, the sequence U+0130 U+004E would result, instead of the expected ’IN’.

2.2. Language Subtag Sources and Interpretation

The namespace of language tags and their subtags is administered by the Internet Assigned Numbers Authority (IANA) according to the rules in Section 5 of this document. The Language Subtag Registry maintained by IANA is the source for valid subtags: other standards referenced in this section provide the source material for that registry.

Terminology used in this document:

- "Tag" refers to a complete language tag, such as "sr-Latn-RS" or "az-Arab-IR". Examples of tags in this document are enclosed in double-quotes ("en-US").

- "Subtag" refers to a specific section of a tag, delimited by a hyphen, such as the subtags ’zh’, ’Hant’, and ’CN’ in the tag "zh-Hant-CN". Examples of subtags in this document are enclosed in single quotes (’Hant’).

- "Code" refers to values defined in external standards (and that are used as subtags in this document). For example, ’Hant’ is an [ISO15924] script code that was used to define the ’Hant’ script subtag for use in a language tag. Examples of codes in this document are enclosed in single quotes (’en’, ’Hant’).

Language tags are designed so that each subtag type has unique length and content restrictions. These make identification of the subtag’s type possible, even if the content of the subtag itself is unrecognized. This allows tags to be parsed and processed without reference to the latest version of the underlying standards or the IANA registry and makes the associated exception handling when parsing tags simpler.

Some of the subtags in the IANA registry do not come from an underlying standard. These can only appear in specific positions in a tag: they can only occur as primary language subtags or as variant subtags.

Sequences of private use and extension subtags MUST occur at the end of the sequence of subtags and MUST NOT be interspersed with subtags defined elsewhere in this document. These sequences are introduced by single-character subtags, which are reserved as follows:

- The single-letter subtag ’x’ introduces a sequence of private use subtags. The interpretation of any private use subtag is defined

solely by private agreement and is not defined by the rules in this section or in any standard or registry defined in this document.

- o The single-letter subtag ’i’ is used by some grandfathered tags, such as "i-default", where it always appears in the first position and cannot be confused with an extension.

- o All other single-letter and single-digit subtags are reserved to introduce standardized extension subtag sequences as described in Section 3.7.

2.2.1. Primary Language Subtag

The primary language subtag is the first subtag in a language tag and cannot be omitted, with two exceptions:

- o The single-character subtag ’x’ as the primary subtag indicates that the language tag consists solely of subtags whose meaning is defined by private agreement. For example, in the tag "x-fr-CH", the subtags ’fr’ and ’CH’ do not represent the French language or the country of Switzerland (or any other value in the IANA registry) unless there is a private agreement in place to do so. See Section 4.6.

- o The single-character subtag ’i’ is used by some grandfathered tags (see Section 2.2.8) such as "i-klingon" and "i-bnn". (Other grandfathered tags have a primary language subtag in their first position.)

The following rules apply to the primary language subtag:

1. Two-character primary language subtags were defined in the IANA registry according to the assignments found in the standard "ISO 639-1:2002, Codes for the representation of names of languages -- Part 1: Alpha-2 code" [ISO639-1], or using assignments subsequently made by the ISO 639-1 registration authority (RA) or governing standardization bodies.

2. Three-character primary language subtags in the IANA registry were defined according to the assignments found in one of these additional ISO 639 parts or assignments subsequently made by the relevant ISO 639 registration authorities or governing standardization bodies:

   A. "ISO 639-2:1998 - Codes for the representation of names of languages -- Part 2: Alpha-3 code - edition 1" [ISO639-2]

B. "ISO 639-3:2007 - Codes for the representation of names of languages -- Part 3: Alpha-3 code for comprehensive coverage of languages" [ISO639-3]

C. "ISO 639-5:2008 - Codes for the representation of names of languages -- Part 5: Alpha-3 code for language families and groups" [ISO639-5]

3. The subtags in the range ’qaa’ through ’qtz’ are reserved for private use in language tags. These subtags correspond to codes reserved by ISO 639-2 for private use. These codes MAY be used for non-registered primary language subtags (instead of using private use subtags following ’x-’). Please refer to Section 4.6 for more information on private use subtags.

4. Four-character language subtags are reserved for possible future standardization.

5. Any language subtags of five to eight characters in length in the IANA registry were defined via the registration process in Section 3.5 and MAY be used to form the primary language subtag. An example of what such a registration might include is the grandfathered IANA registration "i-enochian". The subtag ’enochian’ could be registered in the IANA registry as a primary language subtag (assuming that ISO 639 does not register this language first), making tags such as "enochian-AQ" and "enochian-Latn" valid.

   At the time this document was created, there were no examples of this kind of subtag. Future registrations of this type are discouraged: an attempt to register any new proposed primary language MUST be made to the ISO 639 registration authority. Proposals rejected by the ISO 639 registration authority are unlikely to meet the criteria for primary language subtags and are thus unlikely to be registered.

6. Other values MUST NOT be assigned to the primary subtag except by revision or update of this document.

When languages have both an ISO 639-1 two-character code and a three-character code (assigned by ISO 639-2, ISO 639-3, or ISO 639-5), only the ISO 639-1 two-character code is defined in the IANA registry.

When a language has no ISO 639-1 two-character code and the ISO 639-2/T (Terminology) code and the ISO 639-2/B (Bibliographic) code for that language differ, only the Terminology code is defined in the IANA registry. At the time this document was created, all languages that had both kinds of three-character codes were also assigned a

two-character code; it is expected that future assignments of this nature will not occur.

In order to avoid instability in the canonical form of tags, if a two-character code is added to ISO 639-1 for a language for which a three-character code was already included in either ISO 639-2 or ISO 639-3, the two-character code MUST NOT be registered. See Section 3.4.

For example, if some content were tagged with 'haw' (Hawaiian), which currently has no two-character code, the tag would not need to be changed if ISO 639-1 were to assign a two-character code to the Hawaiian language at a later date.

To avoid these problems with versioning and subtag choice (as experienced during the transition between RFC 1766 and RFC 3066), as well as to ensure the canonical nature of subtags defined by this document, the ISO 639 Registration Authority Joint Advisory Committee (ISO 639/RA-JAC) has included the following statement in [iso639.prin]:

> "A language code already in ISO 639-2 at the point of freezing ISO 639-1 shall not later be added to ISO 639-1. This is to ensure consistency in usage over time, since users are directed in Internet applications to employ the alpha-3 code when an alpha-2 code for that language is not available."

2.2.2. Extended Language Subtags

Extended language subtags are used to identify certain specially selected languages that, for various historical and compatibility reasons, are closely identified with or tagged using an existing primary language subtag. Extended language subtags are always used with their enclosing primary language subtag (indicated with a 'Prefix' field in the registry) when used to form the language tag. All languages that have an extended language subtag in the registry also have an identical primary language subtag record in the registry. This primary language subtag is RECOMMENDED for forming the language tag. The following rules apply to the extended language subtags:

1. Extended language subtags consist solely of three-letter subtags. All extended language subtag records defined in the registry were defined according to the assignments found in [ISO639-3]. Language collections and groupings, such as defined in [ISO639-5], are specifically excluded from being extended language subtags.

2. Extended language subtag records MUST include exactly one ’Prefix’ field indicating an appropriate subtag or sequence of subtags for that extended language subtag.

3. Extended language subtag records MUST include a ’Preferred-Value’. The ’Preferred-Value’ and ’Subtag’ fields MUST be identical.

4. Although the ABNF production ’extlang’ permits up to three extended language tags in the language tag, extended language subtags MUST NOT include another extended language subtag in their ’Prefix’. That is, the second and third extended language subtag positions in a language tag are permanently reserved and tags that include those subtags in that position are, and will always remain, invalid.

For example, the macrolanguage Chinese (’zh’) encompasses a number of languages. For compatibility reasons, each of these languages has both a primary and extended language subtag in the registry. A few selected examples of these include Gan Chinese (’gan’), Cantonese Chinese (’yue’), and Mandarin Chinese (’cmn’). Each is encompassed by the macrolanguage ’zh’ (Chinese). Therefore, they each have the prefix "zh" in their registry records. Thus, Gan Chinese is represented with tags beginning "zh-gan" or "gan", Cantonese with tags beginning either "yue" or "zh-yue", and Mandarin Chinese with "zh-cmn" or "cmn". The language subtag ’zh’ can still be used without an extended language subtag to label a resource as some unspecified variety of Chinese, while the primary language subtag (’gan’, ’yue’, ’cmn’) is preferred to using the extended language form ("zh-gan", "zh-yue", "zh-cmn").

2.2.3. Script Subtag

Script subtags are used to indicate the script or writing system variations that distinguish the written forms of a language or its dialects. The following rules apply to the script subtags:

1. Script subtags MUST follow any primary and extended language subtags and MUST precede any other type of subtag.

2. Script subtags consist of four letters and were defined according to the assignments found in [ISO15924] ("Information and documentation -- Codes for the representation of names of scripts"), or subsequently assigned by the ISO 15924 registration authority or governing standardization bodies. Only codes assigned by ISO 15924 will be considered for registration.

3. The script subtags ’Qaaa’ through ’Qabx’ are reserved for private use in language tags. These subtags correspond to codes reserved by ISO 15924 for private use. These codes MAY be used for non-registered script values. Please refer to Section 4.6 for more information on private use subtags.

4. There MUST be at most one script subtag in a language tag, and the script subtag SHOULD be omitted when it adds no distinguishing value to the tag or when the primary or extended language subtag’s record in the subtag registry includes a ’Suppress-Script’ field listing the applicable script subtag.

For example: "sr-Latn" represents Serbian written using the Latin script.

2.2.4. Region Subtag

Region subtags are used to indicate linguistic variations associated with or appropriate to a specific country, territory, or region. Typically, a region subtag is used to indicate variations such as regional dialects or usage, or region-specific spelling conventions. It can also be used to indicate that content is expressed in a way that is appropriate for use throughout a region, for instance, Spanish content tailored to be useful throughout Latin America.

The following rules apply to the region subtags:

1. Region subtags MUST follow any primary language, extended language, or script subtags and MUST precede any other type of subtag.

2. Two-letter region subtags were defined according to the assignments found in [ISO3166-1] ("Codes for the representation of names of countries and their subdivisions -- Part 1: Country codes"), using the list of alpha-2 country codes or using assignments subsequently made by the ISO 3166-1 maintenance agency or governing standardization bodies. In addition, the codes that are "exceptionally reserved" (as opposed to "assigned") in ISO 3166-1 were also defined in the registry, with the exception of ’UK’, which is an exact synonym for the assigned code ’GB’.

3. The region subtags ’AA’, ’QM’-’QZ’, ’XA’-’XZ’, and ’ZZ’ are reserved for private use in language tags. These subtags correspond to codes reserved by ISO 3166 for private use. These codes MAY be used for private use region subtags (instead of using a private use subtag sequence). Please refer to Section 4.6 for more information on private use subtags.

4. Three-character region subtags consist solely of digit (number) characters and were defined according to the assignments found in the UN Standard Country or Area Codes for Statistical Use [UN_M.49] or assignments subsequently made by the governing standards body. Not all of the UN M.49 codes are defined in the IANA registry. The following rules define which codes are entered into the registry as valid subtags:

   A. UN numeric codes assigned to 'macro-geographical (continental)' or sub-regions MUST be registered in the registry. These codes are not associated with an assigned ISO 3166-1 alpha-2 code and represent supra-national areas, usually covering more than one nation, state, province, or territory.

   B. UN numeric codes for 'economic groupings' or 'other groupings' MUST NOT be registered in the IANA registry and MUST NOT be used to form language tags.

   C. When ISO 3166-1 reassigns a code formerly used for one country or area to another country or area and that code already is present in the registry, the UN numeric code for that country or area MUST be registered in the registry as described in Section 3.4 and MUST be used to form language tags that represent the country or region for which it is defined (rather than the recycled ISO 3166-1 code).

   D. UN numeric codes for countries or areas for which there is an associated ISO 3166-1 alpha-2 code in the registry MUST NOT be entered into the registry and MUST NOT be used to form language tags. Note that the ISO 3166-based subtag in the registry MUST actually be associated with the UN M.49 code in question.

   E. For historical reasons, the UN numeric code 830 (Channel Islands), which was not registered at the time this document was adopted and had, at that time, no corresponding ISO 3166-1 code, MAY be entered into the IANA registry via the process described in Section 3.5, provided no ISO 3166-1 code with that exact meaning has been previously registered.

   F. All other UN numeric codes for countries or areas that do not have an associated ISO 3166-1 alpha-2 code MUST NOT be entered into the registry and MUST NOT be used to form language tags. For more information about these codes, see Section 3.4.

5. The alphanumeric codes in Appendix X of the UN document MUST NOT be entered into the registry and MUST NOT be used to form language tags. (At the time this document was created, these values matched the ISO 3166-1 alpha-2 codes.)

6. There MUST be at most one region subtag in a language tag and the region subtag MAY be omitted, as when it adds no distinguishing value to the tag.

For example:

"de-AT" represents German (’de’) as used in Austria (’AT’).

"sr-Latn-RS" represents Serbian (’sr’) written using Latin script (’Latn’) as used in Serbia (’RS’).

"es-419" represents Spanish (’es’) appropriate to the UN-defined Latin America and Caribbean region (’419’).

2.2.5. Variant Subtags

Variant subtags are used to indicate additional, well-recognized variations that define a language or its dialects that are not covered by other available subtags. The following rules apply to the variant subtags:

1. Variant subtags MUST follow any primary language, extended language, script, or region subtags and MUST precede any extension or private use subtag sequences.

2. Variant subtags, as a collection, are not associated with any particular external standard. The meaning of variant subtags in the registry is defined in the course of the registration process defined in Section 3.5. Note that any particular variant subtag might be associated with some external standard. However, association with a standard is not required for registration.

3. More than one variant MAY be used to form the language tag.

4. Variant subtags MUST be registered with IANA according to the rules in Section 3.5 of this document before being used to form language tags. In order to distinguish variants from other types of subtags, registrations MUST meet the following length and content restrictions:

    1. Variant subtags that begin with a letter (a-z, A-Z) MUST be at least five characters long.

2. Variant subtags that begin with a digit (0-9) MUST be at least four characters long.

5. The same variant subtag MUST NOT be used more than once within a language tag.

   * For example, the tag "de-DE-1901-1901" is not valid.

Variant subtag records in the Language Subtag Registry MAY include one or more ’Prefix’ (Section 3.1.8) fields. Each ’Prefix’ indicates a suitable sequence of subtags for forming (with other subtags, as appropriate) a language tag when using the variant.

Most variants that share a prefix are mutually exclusive. For example, the German orthographic variations ’1996’ and ’1901’ SHOULD NOT be used in the same tag, as they represent the dates of different spelling reforms. A variant that can meaningfully be used in combination with another variant SHOULD include a ’Prefix’ field in its registry record that lists that other variant. For example, if another German variant ’example’ were created that made sense to use with ’1996’, then ’example’ should include two ’Prefix’ fields: "de" and "de-1996".

For example:

"sl-nedis" represents the Natisone or Nadiza dialect of Slovenian.

"de-CH-1996" represents German as used in Switzerland and as written using the spelling reform beginning in the year 1996 C.E.

2.2.6. Extension Subtags

Extensions provide a mechanism for extending language tags for use in various applications. They are intended to identify information that is commonly used in association with languages or language tags but that is not part of language identification. See Section 3.7. The following rules apply to extensions:

1. An extension MUST follow at least a primary language subtag. That is, a language tag cannot begin with an extension. Extensions extend language tags, they do not override or replace them. For example, "a-value" is not a well-formed language tag, while "de-a-value" is. Note that extensions cannot be used in tags that are entirely private use (that is, tags starting with "x-").

2. Extension subtags are separated from the other subtags defined in this document by a single-character subtag (called a "singleton"). The singleton MUST be one allocated to a registration authority via the mechanism described in Section 3.7 and MUST NOT be the letter ’x’, which is reserved for private use subtag sequences.

3. Each singleton subtag MUST appear at most one time in each tag (other than as a private use subtag). That is, singleton subtags MUST NOT be repeated. For example, the tag "en-a-bbb-a-ccc" is invalid because the subtag ’a’ appears twice. Note that the tag "en-a-bbb-x-a-ccc" is valid because the second appearance of the singleton ’a’ is in a private use sequence.

4. Extension subtags MUST meet whatever requirements are set by the document that defines their singleton prefix and whatever requirements are provided by the maintaining authority. Note that there might not be a registry of these subtags and validating processors are not required to validate extensions.

5. Each extension subtag MUST be from two to eight characters long and consist solely of letters or digits, with each subtag separated by a single ’-’. Case distinctions are ignored in extensions (as with any language subtag) and normalized subtags of this type are expected to be in lowercase.

6. Each singleton MUST be followed by at least one extension subtag. For example, the tag "tlh-a-b-foo" is invalid because the first singleton ’a’ is followed immediately by another singleton ’b’.

7. Extension subtags MUST follow all primary language, extended language, script, region, and variant subtags in a tag and MUST precede any private use subtag sequences.

8. All subtags following the singleton and before another singleton are part of the extension. Example: In the tag "fr-a-Latn", the subtag ’Latn’ does not represent the script subtag ’Latn’ defined in the IANA Language Subtag Registry. Its meaning is defined by the extension ’a’.

9. In the event that more than one extension appears in a single tag, the tag SHOULD be canonicalized as described in Section 4.5, by ordering the various extension sequences into case-insensitive ASCII order.

For example, if an extension were defined for the singleton ’r’ and it defined the subtags shown, then the following tag would be a valid example: "en-Latn-GB-boont-r-extended-sequence-x-private".

2.2.7. Private Use Subtags

Private use subtags are used to indicate distinctions in language that are important in a given context by private agreement. The following rules apply to private use subtags:

1. Private use subtags are separated from the other subtags defined in this document by the reserved single-character subtag ’x’.

2. Private use subtags MUST conform to the format and content constraints defined in the ABNF for all subtags; that is, they MUST consist solely of letters and digits and not exceed eight characters in length.

3. Private use subtags MUST follow all primary language, extended language, script, region, variant, and extension subtags in the tag. Another way of saying this is that all subtags following the singleton ’x’ MUST be considered private use. Example: The subtag ’US’ in the tag "en-x-US" is a private use subtag.

4. A tag MAY consist entirely of private use subtags.

5. No source is defined for private use subtags. Use of private use subtags is by private agreement only.

6. Private use subtags are NOT RECOMMENDED where alternatives exist or for general interchange. See Section 4.6 for more information on private use subtag choice.

For example, suppose a group of scholars is studying some texts in medieval Greek. They might agree to use some collection of private use subtags to identify different styles of writing in the texts. For example, they might use ’el-x-koine’ for documents in the "common" style while using ’el-x-attic’ for other documents that mimic the Attic style. These subtags would not be recognized by outside processes or systems, but might be useful in categorizing various texts for study by those in the group.

In the registry, there are also subtags derived from codes reserved by ISO 639, ISO 15924, or ISO 3166 for private use. Do not confuse these with private use subtag sequences following the subtag ’x’. See Section 4.6.

2.2.8. Grandfathered and Redundant Registrations

Prior to RFC 4646, whole language tags were registered according to the rules in RFC 1766 and/or RFC 3066. All of these registered tags remain valid as language tags.

Many of these registered tags were made redundant by the advent of either RFC 4646 or this document. A redundant tag is a grandfathered registration whose individual subtags appear with the same semantic meaning in the registry. For example, the tag "zh-Hant" (Traditional Chinese) can now be composed from the subtags 'zh' (Chinese) and 'Hant' (Han script traditional variant). These redundant tags are maintained in the registry as records of type 'redundant', mostly as a matter of historical curiosity.

The remainder of the previously registered tags are "grandfathered". These tags are classified into two groups: 'regular' and 'irregular'.

Grandfathered tags that (appear to) match the 'langtag' production in Figure 1 are considered 'regular' grandfathered tags. These tags contain one or more subtags that either do not individually appear in the registry or appear but with a different semantic meaning: each tag, in its entirety, represents a language or collection of languages.

Grandfathered tags that do not match the 'langtag' production in the ABNF and would otherwise be invalid are considered 'irregular' grandfathered tags. With the exception of "en-GB-oed", which is a variant of "en-GB", each of them, in its entirety, represents a language.

Many of the grandfathered tags have been superseded by the subsequent addition of new subtags: each superseded record contains a 'Preferred-Value' field that ought to be used to form language tags representing that value. For example, the tag "art-lojban" is superseded by the primary language subtag 'jbo'.

2.2.9. Classes of Conformance

Implementations sometimes need to describe their capabilities with regard to the rules and practices described in this document. Tags can be checked or verified in a number of ways, but two particular classes of tag conformance are formally defined here.

A tag is considered "well-formed" if it conforms to the ABNF (Section 2.1). Language tags may be well-formed in terms of syntax but not valid in terms of content. However, many operations involving language tags work well without knowing anything about the meaning or validity of the subtags.

A tag is considered "valid" if it satisfies these conditions:

- The tag is well-formed.

- o Either the tag is in the list of grandfathered tags or all of its primary language, extended language, script, region, and variant subtags appear in the IANA Language Subtag Registry as of the particular registry date.

- o There are no duplicate variant subtags.

- o There are no duplicate singleton (extension) subtags.

Note that a tag's validity depends on the date of the registry used to validate the tag. A more recent copy of the registry might contain a subtag that an older version does not.

A tag is considered valid for a given extension (Section 3.7) (as of a particular version, revision, and date) if it meets the criteria for "valid" above and also satisfies this condition:

> Each subtag used in the extension part of the tag is valid according to the extension.

Older specifications or language tag implementations sometimes reference [RFC3066]. A wider array of tags was considered well-formed under that document. Any tags that were valid for use under RFC 3066 are both well-formed and valid under this document's syntax; only invalid or illegal tags were well-formed under the earlier definition but no longer are. The language tag syntax under RFC 3066 was:

```
obs-language-tag = primary-subtag *( "-" subtag )
primary-subtag   = 1*8ALPHA
subtag           = 1*8(ALPHA / DIGIT)
```

Figure 2: RFC 3066 Language Tag Syntax

Subtags designated for private use as well as private use sequences introduced by the 'x' subtag are available for cases in which no assigned subtags are available and registration is not a suitable option. For example, one might use a tag such as "no-QQ", where 'QQ' is one of a range of private use ISO 3166-1 codes to indicate an otherwise undefined region. Users MUST NOT assign language tags that use subtags that do not appear in the registry other than in private use sequences (such as the subtag 'personal' in the tag "en-x-personal"). Besides not being valid, the user also risks collision with a future possible assignment or registrations.

Note well: although the 'Language-Tag' production appearing in this document is functionally equivalent to the one in [RFC4646], it has

been changed to prevent certain errors in well-formedness arising from the old ’grandfathered’ production.

3. Registry Format and Maintenance

The IANA Language Subtag Registry ("the registry") contains a comprehensive list of all of the subtags valid in language tags. This allows implementers a straightforward and reliable way to validate language tags. The registry will be maintained so that, except for extension subtags, it is possible to validate all of the subtags that appear in a language tag under the provisions of this document or its revisions or successors. In addition, the meaning of the various subtags will be unambiguous and stable over time. (The meaning of private use subtags, of course, is not defined by the registry.)

This section defines the registry along with the maintenance and update procedures associated with it, as well as a registry for extensions to language tags (Section 3.7).

3.1. Format of the IANA Language Subtag Registry

The IANA Language Subtag Registry is a machine-readable file in the format described in this section, plus copies of the registration forms approved in accordance with the process described in Section 3.5.

The existing registration forms for grandfathered and redundant tags taken from RFC 3066 have been maintained as part of the obsolete RFC 3066 registry. The subtags added to the registry by either [RFC4645] or [RFC5645] do not have separate registration forms (so no forms are archived for these additions).

3.1.1. File Format

The registry is a [Unicode] text file and consists of a series of records in a format based on "record-jar" (described in [record-jar]). Each record, in turn, consists of a series of fields that describe the various subtags and tags. The actual registry file is encoded using the UTF-8 [RFC3629] character encoding.

Each field can be considered a single, logical line of characters. Each field contains a "field-name" and a "field-body". These are separated by a "field-separator". The field-separator is a COLON character (U+003A) plus any surrounding whitespace. Each field is terminated by the newline sequence CRLF. The text in each field MUST be in Unicode Normalization Form C (NFC).

A collection of fields forms a "record". Records are separated by lines containing only the sequence "%%" (U+0025 U+0025).

Although fields are logically a single line of text, each line of text in the file format is limited to 72 bytes in length. To accommodate this, the field-body can be split into a multiple-line representation; this is called "folding". Folding is done according to customary conventions for line-wrapping. This is typically on whitespace boundaries, but can occur between other characters when the value does not include spaces, such as when a language does not use whitespace between words. In any event, there MUST NOT be breaks inside a multibyte UTF-8 sequence or in the middle of a combining character sequence. For more information, see [UAX14].

Although the file format uses the Unicode character set and the file itself is encoded using the UTF-8 encoding, fields are restricted to the printable characters from the US-ASCII [ISO646] repertoire unless otherwise indicated in the description of a specific field (Section 3.1.2).

The format of the registry is described by the following ABNF [RFC5234]. Character numbers (code points) are taken from Unicode, and terminals in the ABNF productions are in terms of characters rather than bytes.

```
registry   = record *("%%" CRLF record)
record     = 1*field
field      = ( field-name field-sep field-body CRLF )
field-name = (ALPHA / DIGIT) [*(ALPHA / DIGIT / "-") (ALPHA / DIGIT)]
field-sep  = *SP ":" *SP
field-body = *([[*SP CRLF] 1*SP] 1*CHARS)
CHARS      = (%x21-10FFFF)      ; Unicode code points
```

Figure 3: Registry Format ABNF

The sequence '..' (U+002E U+002E) in a field-body denotes a range of values. Such a range represents all subtags of the same length that are in alphabetic or numeric order within that range, including the values explicitly mentioned. For example, 'a..c' denotes the values 'a', 'b', and 'c', and '11..13' denotes the values '11', '12', and '13'.

All fields whose field-body contains a date value use the "full-date" format specified in [RFC3339]. For example, "2004-06-28" represents June 28, 2004, in the Gregorian calendar.

3.1.2. Record and Field Definitions

There are three types of records in the registry: "File-Date", "Subtag", and "Tag".

The first record in the registry is always the "File-Date" record. This record occurs only once in the file and contains a single field whose field-name is "File-Date". The field-body of this record contains a date (see Section 5.1), making it possible to easily recognize different versions of the registry.

```
File-Date: 2004-06-28
%%
```

Figure 4: Example of the File-Date Record

Subsequent records contain multiple fields and represent information about either subtags or tags. Both types of records have an identical structure, except that "Subtag" records contain a field with a field-name of "Subtag", while, unsurprisingly, "Tag" records contain a field with a field-name of "Tag". Field-names MUST NOT occur more than once per record, with the exception of the 'Description', 'Comments', and 'Prefix' fields.

Each record MUST contain at least one of each of the following fields:

- o 'Type'
  - * Type's field-body MUST consist of one of the following strings: "language", "extlang", "script", "region", "variant", "grandfathered", and "redundant"; it denotes the type of tag or subtag.
- o Either 'Subtag' or 'Tag'
  - * Subtag's field-body contains the subtag being defined. This field MUST appear in all records whose 'Type' has one of these values: "language", "extlang", "script", "region", or "variant".
  - * Tag's field-body contains a complete language tag. This field MUST appear in all records whose 'Type' has one of these values: "grandfathered" or "redundant". If the 'Type' is "grandfathered", then the 'Tag' field-body will be one of the tags listed in either the 'regular' or 'irregular' production found in Section 2.1.

- o ’Description’
  - * Description’s field-body contains a non-normative description of the subtag or tag.
- o ’Added’
  - * Added’s field-body contains the date the record was registered or, in the case of grandfathered or redundant tags, the date the corresponding tag was registered under the rules of [RFC1766] or [RFC3066].

Each record MAY also contain the following fields:

- o ’Deprecated’
  - * Deprecated’s field-body contains the date the record was deprecated. In some cases, this value is earlier than that of the ’Added’ field in the same record. That is, the date of deprecation preceded the addition of the record to the registry.
- o ’Preferred-Value’
  - * Preferred-Value’s field-body contains a canonical mapping from this record’s value to a modern equivalent that is preferred in its place. Depending on the value of the ’Type’ field, this value can take different forms:
    - + For fields of type ’language’, ’Preferred-Value’ contains the primary language subtag that is preferred when forming the language tag.
    - + For fields of type ’script’, ’region’, or ’variant’, ’Preferred-Value’ contains the subtag of the same type that is preferred for forming the language tag.
    - + For fields of type ’extlang’, ’grandfathered’, or ’redundant’, ’Preferred-Value’ contains an "extended language range" [RFC4647] that is preferred for forming the language tag. That is, the preferred language tag will contain, in order, each of the subtags that appears in the ’Preferred-Value’; additional fields can be included in a language tag, as described elsewhere in this document. For example, the replacement for the grandfathered tag "zh-min-nan" (Min Nan Chinese) is "nan", which can be used as the

basis for tags such as "nan-Hant" or "nan-TW" (note that the extended language subtag form such as "zh-nan-Hant" or "zh-nan-TW" can also be used).

- o ’Prefix’
  - * Prefix’s field-body contains a valid language tag that is RECOMMENDED as one possible prefix to this record’s subtag. This field MAY appear in records whose ’Type’ field-body is either ’extlang’ or ’variant’ (it MUST NOT appear in any other record type).
- o ’Suppress-Script’
  - * Suppress-Script’s field-body contains a script subtag that SHOULD NOT be used to form language tags with the associated primary or extended language subtag. This field MUST appear only in records whose ’Type’ field-body is ’language’ or ’extlang’. See Section 4.1.
- o ’Macrolanguage’
  - * Macrolanguage’s field-body contains a primary language subtag defined by ISO 639 as the "macrolanguage" that encompasses this language subtag. This field MUST appear only in records whose ’Type’ field-body is either ’language’ or ’extlang’.
- o ’Scope’
  - * Scope’s field-body contains information about a primary or extended language subtag indicating the type of language code according to ISO 639. The values permitted in this field are "macrolanguage", "collection", "special", and "private-use". This field only appears in records whose ’Type’ field-body is either ’language’ or ’extlang’. When this field is omitted, the language is an individual language.
- o ’Comments’
  - * Comments’s field-body contains additional information about the subtag, as deemed appropriate for understanding the registry and implementing language tags using the subtag or tag.

Future versions of this document might add additional fields to the registry; implementations SHOULD ignore fields found in the registry that are not defined in this document.

3.1.3. Type Field

The field ’Type’ contains the string identifying the record type in which it appears. Values for the ’Type’ field-body are: "language" (Section 2.2.1); "extlang" (Section 2.2.2); "script" (Section 2.2.3); "region" (Section 2.2.4); "variant" (Section 2.2.5); "grandfathered" or "redundant" (Section 2.2.8).

3.1.4. Subtag and Tag Fields

The field ’Subtag’ contains the subtag defined in the record. The field ’Tag’ appears in records whose ’Type’ is either ’grandfathered’ or ’redundant’ and contains a tag registered under [RFC3066].

The ’Subtag’ field-body MUST follow the casing conventions described in Section 2.1.1. All subtags use lowercase letters in the field-body, with two exceptions:

> Subtags whose ’Type’ field is ’script’ (in other words, subtags defined by ISO 15924) MUST use titlecase.
>
> Subtags whose ’Type’ field is ’region’ (in other words, the non-numeric region subtags defined by ISO 3166-1) MUST use all uppercase.

The ’Tag’ field-body MUST be formatted according to the rules described in Section 2.1.1.

3.1.5. Description Field

The field ’Description’ contains a description of the tag or subtag in the record. The ’Description’ field MAY appear more than once per record. The ’Description’ field MAY include the full range of Unicode characters. At least one of the ’Description’ fields MUST be written or transcribed into the Latin script; additional ’Description’ fields MAY be in any script or language.

The ’Description’ field is used for identification purposes. Descriptions SHOULD contain all and only that information necessary to distinguish one subtag from others with which it might be confused. They are not intended to provide general background information or to provide all possible alternate names or designations. ’Description’ fields don’t necessarily represent the actual native name of the item in the record, nor are any of the descriptions guaranteed to be in any particular language (such as English or French, for example).

Descriptions in the registry that correspond to ISO 639, ISO 15924, ISO 3166-1, or UN M.49 codes are intended only to indicate the meaning of that identifier as defined in the source standard at the time it was added to the registry or as subsequently modified, within the bounds of the stability rules (Section 3.4), via subsequent registration. The 'Description' does not replace the content of the source standard itself. 'Description' fields are not intended to be the localized English names for the subtags. Localization or translation of language tag and subtag descriptions is out of scope of this document.

For subtags taken from a source standard (such as ISO 639 or ISO 15924), the 'Description' fields in the record are also initially taken from that source standard. Multiple descriptions in the source standard are split into separate 'Description' fields. The source standard's descriptions MAY be edited or modified, either prior to insertion or via the registration process, and additional or extraneous descriptions omitted or removed. Each 'Description' field MUST be unique within the record in which it appears, and formatting variations of the same description SHOULD NOT occur in that specific record. For example, while the ISO 639-1 code 'fy' has both the description "Western Frisian" and the description "Frisian, Western" in that standard, only one of these descriptions appears in the registry.

To help ensure that users do not become confused about which subtag to use, 'Description' fields assigned to a record of any specific type ('language', 'extlang', 'script', and so on) MUST be unique within that given record type with the following exception: if a particular 'Description' field occurs in multiple records of a given type, then at most one of the records can omit the 'Deprecated' field. All deprecated records that share a 'Description' MUST have the same 'Preferred-Value', and all non-deprecated records MUST be that 'Preferred-Value'. This means that two records of the same type that share a 'Description' are also semantically equivalent and no more than one record with a given 'Description' is preferred for that meaning.

For example, consider the 'language' subtags 'zza' (Zaza) and 'diq' (Dimli). It so happens that 'zza' is a macrolanguage enclosing 'diq' and thus also has a description in ISO 639-3 of "Dimli". This description was edited to read "Dimli (macrolanguage)" in the registry record for 'zza' to prevent a collision.

By contrast, the subtags 'he' and 'iw' share a 'Description' value of "Hebrew"; this is permitted because 'iw' is deprecated and its 'Preferred-Value' is 'he'.

For fields of type 'language', the first 'Description' field appearing in the registry corresponds whenever possible to the Reference Name assigned by ISO 639-3. This helps facilitate cross-referencing between ISO 639 and the registry.

When creating or updating a record due to the action of one of the source standards, the Language Subtag Reviewer MAY edit descriptions to correct irregularities in formatting (such as misspellings, inappropriate apostrophes or other punctuation, or excessive or missing spaces) prior to submitting the proposed record to the ietf-languages@iana.org list for consideration.

3.1.6. Deprecated Field

The field 'Deprecated' contains the date the record was deprecated and MAY be added, changed, or removed from any record via the maintenance process described in Section 3.3 or via the registration process described in Section 3.5. Usually, the addition of a 'Deprecated' field is due to the action of one of the standards bodies, such as ISO 3166, withdrawing a code. Although valid in language tags, subtags and tags with a 'Deprecated' field are deprecated, and validating processors SHOULD NOT generate these subtags. Note that a record that contains a 'Deprecated' field and no corresponding 'Preferred-Value' field has no replacement mapping.

In some historical cases, it might not have been possible to reconstruct the original deprecation date. For these cases, an approximate date appears in the registry. Some subtags and some grandfathered or redundant tags were deprecated before the initial creation of the registry. The exact rules for this appear in Section 2 of [RFC4645]. Note that these records have a 'Deprecated' field with an earlier date then the corresponding 'Added' field!

3.1.7. Preferred-Value Field

The field 'Preferred-Value' contains a mapping between the record in which it appears and another tag or subtag (depending on the record's 'Type'). The value in this field is used for canonicalization (see Section 4.5). In cases where the subtag or tag also has a 'Deprecated' field, then the 'Preferred-Value' is RECOMMENDED as the best choice to represent the value of this record when selecting a language tag.

Records containing a 'Preferred-Value' fall into one of these four groups:

1. ISO 639 language codes that were later withdrawn in favor of other codes. These values are mostly a historical curiosity. The ’he’/’iw’ pairing above is an example of this.

2. Subtags (with types other than language or extlang) taken from codes or values that have been withdrawn in favor of a new code. In particular, this applies to region subtags taken from ISO 3166-1, because sometimes a country will change its name or administration in such a way that warrants a new region code. In some cases, countries have reverted to an older name, which might already be encoded. For example, the subtag ’ZR’ (Zaire) was replaced by the subtag ’CD’ (Democratic Republic of the Congo) when that country’s name was changed.

3. Tags or subtags that have become obsolete because the values they represent were later encoded. Many of the grandfathered or redundant tags were later encoded by ISO 639, for example, and fall into this grouping. For example, "i-klingon" was deprecated when the subtag ’tlh’ was added. The record for "i-klingon" has a ’Preferred-Value’ of ’tlh’.

4. Extended language subtags always have a mapping to their identical primary language subtag. For example, the extended language subtag ’yue’ (Cantonese) can be used to form the tag "zh-yue". It has a ’Preferred-Value’ mapping to the primary language subtag ’yue’, meaning that a tag such as "zh-yue-Hant-HK" can be canonicalized to "yue-Hant-HK".

Records other than those of type ’extlang’ that contain a ’Preferred-Value’ field MUST also have a ’Deprecated’ field. This field contains the date on which the tag or subtag was deprecated in favor of the preferred value.

For records of type ’extlang’, the ’Preferred-Value’ field appears without a corresponding ’Deprecated’ field. An implementation MAY ignore these preferred value mappings, although if it ignores the mapping, it SHOULD do so consistently. It SHOULD also treat the ’Preferred-Value’ as equivalent to the mapped item. For example, the tags "zh-yue-Hant-HK" and "yue-Hant-HK" are semantically equivalent and ought to be treated as if they were the same tag.

Occasionally, the deprecated code is preferred in certain contexts. For example, both "iw" and "he" can be used in the Java programming language, but "he" is converted on input to "iw", which is thus the canonical form in Java.

’Preferred-Value’ mappings in records of type ’region’ sometimes do not represent exactly the same meaning as the original value. There are many reasons for a country code to be changed, and the effect this has on the formation of language tags will depend on the nature of the change in question. For example, the region subtag ’YD’ (Democratic Yemen) was deprecated in favor of the subtag ’YE’ (Yemen) when those two countries unified in 1990.

A ’Preferred-Value’ MAY be added to, changed, or removed from records according to the rules in Section 3.3. Addition, modification, or removal of a ’Preferred-Value’ field in a record does not imply that content using the affected subtag needs to be retagged.

The ’Preferred-Value’ fields in records of type "grandfathered" and "redundant" each contain an "extended language range" [RFC4647] that is strongly RECOMMENDED for use in place of the record’s value. In many cases, these mappings were created via deprecation of the tags during the period before [RFC4646] was adopted. For example, the tag "no-nyn" was deprecated in favor of the ISO 639-1-defined language code ’nn’.

The ’Preferred-Value’ field in subtag records of type "extlang" also contains an "extended language range". This allows the subtag to be deprecated in favor of either a single primary language subtag or a new language-extlang sequence.

Usually, the addition, removal, or change of a ’Preferred-Value’ field for a subtag is done to reflect changes in one of the source standards. For example, if an ISO 3166-1 region code is deprecated in favor of another code, that SHOULD result in the addition of a ’Preferred-Value’ field.

Changes to one subtag can affect other subtags as well: when proposing changes to the registry, the Language Subtag Reviewer MUST review the registry for such effects and propose the necessary changes using the process in Section 3.5, although anyone MAY request such changes. For example:

> Suppose that subtag ’XX’ has a ’Preferred-Value’ of ’YY’. If ’YY’ later changes to have a ’Preferred-Value’ of ’ZZ’, then the ’Preferred-Value’ for ’XX’ MUST also change to be ’ZZ’.
>
> Suppose that a registered language subtag ’dialect’ represents a language not yet available in any part of ISO 639. The later addition of a corresponding language code in ISO 639 SHOULD result in the addition of a ’Preferred-Value’ for ’dialect’.

3.1.8. Prefix Field

The field ’Prefix’ contains a valid language tag that is RECOMMENDED as one possible prefix to this record’s subtag, perhaps with other subtags. That is, when including an extended language or a variant subtag that has at least one ’Prefix’ in a language tag, the resulting tag SHOULD match at least one of the subtag’s ’Prefix’ fields using the "Extended Filtering" algorithm (see [RFC4647]), and each of the subtags in that ’Prefix’ SHOULD appear before the subtag itself.

The ’Prefix’ field MUST appear exactly once in a record of type ’extlang’. The ’Prefix’ field MAY appear multiple times (or not at all) in records of type ’variant’. Additional fields of this type MAY be added to a ’variant’ record via the registration process, provided the ’variant’ record already has at least one ’Prefix’ field.

Each ’Prefix’ field indicates a particular sequence of subtags that form a meaningful tag with this subtag. For example, the extended language subtag ’cmn’ (Mandarin Chinese) only makes sense with its prefix ’zh’ (Chinese). Similarly, ’rozaj’ (Resian, a dialect of Slovenian) would be appropriate when used with its prefix ’sl’ (Slovenian), while tags such as "is-1994" are not appropriate (and probably not meaningful). Although the ’Prefix’ for ’rozaj’ is "sl", other subtags might appear between them. For example, the tag "sl-IT-rozaj" (Slovenian, Italy, Resian) matches the ’Prefix’ "sl".

The ’Prefix’ also indicates when variant subtags make sense when used together (many that otherwise share a ’Prefix’ are mutually exclusive) and what the relative ordering of variants is supposed to be. For example, the variant ’1994’ (Standardized Resian orthography) has several ’Prefix’ fields in the registry ("sl-rozaj", "sl-rozaj-biske", "sl-rozaj-njiva", "sl-rozaj-osojs", and "sl-rozaj-solba"). This indicates not only that ’1994’ is appropriate to use with each of these five Resian variant subtags (’rozaj’, ’biske’, ’njiva’, ’osojs’, and ’solba’), but also that it SHOULD appear following any of these variants in a tag. Thus, the language tag ought to take the form "sl-rozaj-biske-1994", rather than "sl-1994-rozaj-biske" or "sl-rozaj-1994-biske".

If a record includes no ’Prefix’ field, a ’Prefix’ field MUST NOT be added to the record at a later date. Otherwise, changes (additions, deletions, or modifications) to the set of ’Prefix’ fields MAY be registered, as long as they strictly widen the range of language tags that are recommended. For example, a ’Prefix’ with the value "be-Latn" (Belarusian, Latin script) could be replaced by the value "be" (Belarusian) but not by the value "ru-Latn" (Russian, Latin script)

or the value "be-Latn-BY" (Belarusian, Latin script, Belarus), since these latter either change or narrow the range of suggested tags.

The field-body of the ’Prefix’ field MUST NOT conflict with any ’Prefix’ already registered for a given record. Such a conflict would occur when no valid tag could be constructed that would contain the prefix, such as when two subtags each have a ’Prefix’ that contains the other subtag. For example, suppose that the subtag ’avariant’ has the prefix "es-bvariant". Then the subtag ’bvariant’ cannot be assigned the prefix ’avariant’, for that would require a tag of the form "es-avariant-bvariant-avariant", which would not be valid.

3.1.9. Suppress-Script Field

The field ’Suppress-Script’ contains a script subtag (whose record appears in the registry). The field ’Suppress-Script’ MUST appear only in records whose ’Type’ field-body is either ’language’ or ’extlang’. This field MUST NOT appear more than one time in a record.

This field indicates a script used to write the overwhelming majority of documents for the given language. The subtag for such a script therefore adds no distinguishing information to a language tag and thus SHOULD NOT be used for most documents in that language. Omitting the script subtag indicated by this field helps ensure greater compatibility between the language tags generated according to the rules in this document and language tags and tag processors or consumers based on RFC 3066. For example, virtually all Icelandic documents are written in the Latin script, making the subtag ’Latn’ redundant in the tag "is-Latn".

Many language subtag records do not have a ’Suppress-Script’ field. The lack of a ’Suppress-Script’ might indicate that the language is customarily written in more than one script or that the language is not customarily written at all. It might also mean that sufficient information was not available when the record was created and thus remains a candidate for future registration.

3.1.10. Macrolanguage Field

The field ’Macrolanguage’ contains a primary language subtag (whose record appears in the registry). This field indicates a language that encompasses this subtag’s language according to assignments made by ISO 639-3.

ISO 639-3 labels some languages in the registry as "macrolanguages". ISO 639-3 defines the term "macrolanguage" to mean "clusters of

closely-related language varieties that [...] can be considered distinct individual languages, yet in certain usage contexts a single language identity for all is needed". These correspond to codes registered in ISO 639-2 as individual languages that were found to correspond to more than one language in ISO 639-3.

A language contained within a macrolanguage is called an "encompassed language". The record for each encompassed language contains a 'Macrolanguage' field in the registry; the macrolanguages themselves are not specially marked. Note that some encompassed languages have ISO 639-1 or ISO 639-2 codes.

The 'Macrolanguage' field can only occur in records of type 'language' or 'extlang'. Only values assigned by ISO 639-3 will be considered for inclusion. 'Macrolanguage' fields MAY be added or removed via the normal registration process whenever ISO 639-3 defines new values or withdraws old values. Macrolanguages are informational, and MAY be removed or changed if ISO 639-3 changes the values. For more information on the use of this field and choosing between macrolanguage and encompassed language subtags, see Section 4.1.1.

For example, the language subtags 'nb' (Norwegian Bokmal) and 'nn' (Norwegian Nynorsk) each have a 'Macrolanguage' field with a value of 'no' (Norwegian). For more information, see Section 4.1.

3.1.11. Scope Field

The field 'Scope' contains classification information about a primary or extended language subtag derived from ISO 639. Most languages have a scope of 'individual', which means that the language is not a macrolanguage, collection, special code, or private use. That is, it is what one would normally consider to be 'a language'. Any primary or extended language subtag that has no 'Scope' field is an individual language.

'Scope' information can sometimes be helpful in selecting language tags, since it indicates the purpose or "scope" of the code assignment within ISO 639. The available values are:

- 'macrolanguage' - Indicates a macrolanguage as defined by ISO 639-3 (see Section 3.1.10). A macrolanguage is a cluster of closely related languages that are sometimes considered to be a single language.

- 'collection' - Indicates a subtag that represents a collection of languages, typically related by some type of historical, geographical, or linguistic association. Unlike a macrolanguage,

a collection can contain languages that are only loosely related and a collection cannot be used interchangeably with languages that belong to it.

- o ’special’ - Indicates a special language code. These are subtags used for identifying linguistic attributes not particularly associated with a concrete language. These include codes for when the language is undetermined or for non-linguistic content.

- o ’private-use’ - Indicates a code reserved for private use in the underlying standard. Subtags with this scope can be used to indicate a primary language for which no ISO 639 or registered assignment exists.

The ’Scope’ field MAY appear in records of type ’language’ or ’extlang’. Note that many of the prefixes for extended language subtags will have a ’Scope’ of ’macrolanguage’ (although some will not) and that many languages that have a ’Scope’ of ’macrolanguage’ will have extended language subtags associated with them.

The ’Scope’ field MAY be added, modified, or removed via the registration process, provided the change mirrors changes made by ISO 639 to the assignment’s classification. Such a change is expected to be rare.

For example, the primary language subtag ’zh’ (Chinese) has a ’Scope’ of ’macrolanguage’, while its enclosed language ’nan’ (Min Nan Chinese) has a ’Scope’ of ’individual’. The special value ’und’ (Undetermined) has a ’Scope’ of ’special’. The ISO 639-5 collection ’gem’ (Germanic languages) has a ’Scope’ of ’collection’.

3.1.12. Comments Field

The field ’Comments’ contains additional information about the record and MAY appear more than once per record. The field-body MAY include the full range of Unicode characters and is not restricted to any particular script. This field MAY be inserted or changed via the registration process, and no guarantee of stability is provided.

The content of this field is not restricted, except by the need to register the information, the suitability of the request, and by reasonable practical size limitations. The primary reason for the ’Comments’ field is subtag identification -- to help distinguish the subtag from others with which it might be confused as an aid to usage. Large amounts of information about the use, history, or general background of a subtag are frowned upon, as these generally belong in a registration request rather than in the registry.

3.2. Language Subtag Reviewer

The Language Subtag Reviewer moderates the ietf-languages@iana.org mailing list, responds to requests for registration, and performs the other registry maintenance duties described in Section 3.3. Only the Language Subtag Reviewer is permitted to request IANA to change, update, or add records to the Language Subtag Registry. The Language Subtag Reviewer MAY delegate list moderation and other clerical duties as needed.

The Language Subtag Reviewer is appointed by the IESG for an indefinite term, subject to removal or replacement at the IESG's discretion. The IESG will solicit nominees for the position (upon adoption of this document or upon a vacancy) and then solicit feedback on the nominees' qualifications. Qualified candidates should be familiar with BCP 47 and its requirements; be willing to fairly, responsively, and judiciously administer the registration process; and be suitably informed about the issues of language identification so that the reviewer can assess the claims and draw upon the contributions of language experts and subtag requesters.

The subsequent performance or decisions of the Language Subtag Reviewer MAY be appealed to the IESG under the same rules as other IETF decisions (see [RFC2026]). The IESG can reverse or overturn the decisions of the Language Subtag Reviewer, provide guidance, or take other appropriate actions.

3.3. Maintenance of the Registry

Maintenance of the registry requires that, as codes are assigned or withdrawn by ISO 639, ISO 15924, ISO 3166, and UN M.49, the Language Subtag Reviewer MUST evaluate each change and determine the appropriate course of action according to the rules in this document. Such updates follow the registration process described in Section 3.5. Usually, the Language Subtag Reviewer will start the process for the new or updated record by filling in the registration form and submitting it. If a change to one of these standards takes place and the Language Subtag Reviewer does not do this in a timely manner, then any interested party MAY submit the form. Thereafter, the registration process continues normally.

Note that some registrations affect other subtags--perhaps more than one--as when a region subtag is being deprecated in favor of a new value. The Language Subtag Reviewer is responsible for ensuring that any such changes are properly registered, with each change requiring its own registration form.

The Language Subtag Reviewer MUST ensure that new subtags meet the requirements elsewhere in this document (and most especially in Section 3.4) or submit an appropriate registration form for an alternate subtag as described in that section. Each individual subtag affected by a change MUST be sent to the ietf-languages@iana.org list with its own registration form and in a separate message.

3.4. Stability of IANA Registry Entries

The stability of entries and their meaning in the registry is critical to the long-term stability of language tags. The rules in this section guarantee that a specific language tag's meaning is stable over time and will not change.

These rules specifically deal with how changes to codes (including withdrawal and deprecation of codes) maintained by ISO 639, ISO 15924, ISO 3166, and UN M.49 are reflected in the IANA Language Subtag Registry. Assignments to the IANA Language Subtag Registry MUST follow the following stability rules:

1. Values in the fields 'Type', 'Subtag', 'Tag', and 'Added' MUST NOT be changed and are guaranteed to be stable over time.

2. Values in the fields 'Preferred-Value' and 'Deprecated' MAY be added, altered, or removed via the registration process. These changes SHOULD be limited to changes necessary to mirror changes in one of the underlying standards (ISO 639, ISO 15924, ISO 3166-1, or UN M.49) and typically alteration or removal of a 'Preferred-Value' is limited specifically to region codes.

3. Values in the 'Description' field MUST NOT be changed in a way that would invalidate any existing tags. The description MAY be broadened somewhat in scope, changed to add information, or adapted to the most common modern usage. For example, countries occasionally change their names; a historical example of this is "Upper Volta" changing to "Burkina Faso".

4. Values in the field 'Prefix' MAY be added to existing records of type 'variant' via the registration process, provided the 'variant' already has at least one 'Prefix'. A 'Prefix' field SHALL NOT be registered for any 'variant' that has no existing 'Prefix' field. If a prefix is added to a variant record, 'Comment' fields MAY be used to explain different usages with the various prefixes.

5. Values in the field ’Prefix’ in records of type ’variant’ MAY also be modified, so long as the modifications broaden the set of prefixes. That is, a prefix MAY be replaced by one of its own prefixes. For example, the prefix "en-US" could be replaced by "en", but not by the prefixes "en-Latn", "fr", or "en-US-boont". If one of those prefix values were needed, it would have to be separately registered.

6. Values in the field ’Prefix’ in records of type ’extlang’ MUST NOT be added, modified, or removed.

7. The field ’Prefix’ MUST NOT be removed from any record in which it appears. This field SHOULD be included in the initial registration of any records of type ’variant’ and MUST be included in any records of type ’extlang’.

8. The field ’Comments’ MAY be added, changed, modified, or removed via the registration process or any of the processes or considerations described in this section.

9. The field ’Suppress-Script’ MAY be added or removed via the registration process.

10. The field ’Macrolanguage’ MAY be added or removed via the registration process, but only in response to changes made by ISO 639. The ’Macrolanguage’ field appears whenever a language has a corresponding macrolanguage in ISO 639. That is, the ’Macrolanguage’ fields in the registry exactly match those of ISO 639. No other macrolanguage mappings will be considered for registration.

11. The field ’Scope’ MAY be added or removed from a primary or extended language subtag after initial registration, and it MAY be modified in order to match any changes made by ISO 639. Changes to the ’Scope’ field MUST mirror changes made by ISO 639. Note that primary or extended language subtags whose records do not contain a ’Scope’ field (that is, most of them) are individual languages as described in Section 3.1.11.

12. Primary and extended language subtags (other than independently registered values created using the registration process) are created according to the assignments of the various parts of ISO 639, as follows:

    A. Codes assigned by ISO 639-1 that do not conflict with existing two-letter primary language subtags and that have no corresponding three-letter primary defined in the registry are entered into the IANA registry as new records

of type ’language’. Note that languages given an ISO 639-1 code cannot be given extended language subtags, even if encompassed by a macrolanguage.

B. Codes assigned by ISO 639-3 or ISO 639-5 that do not conflict with existing three-letter primary language subtags and that do not have ISO 639-1 codes assigned (or expected to be assigned) are entered into the IANA registry as new records of type ’language’. Note that these two standards now comprise a superset of ISO 639-2 codes. Codes that have a defined ’macrolanguage’ mapping at the time of their registration MUST contain a ’Macrolanguage’ field.

C. Codes assigned by ISO 639-3 MAY also be considered for an extended language subtag registration. Note that they MUST be assigned a primary language subtag record of type ’language’ even when an ’extlang’ record is proposed. When considering extended language subtag assignment, these criteria apply:

1. If a language has a macrolanguage mapping, and that macrolanguage has other encompassed languages that are assigned extended language subtags, then the new language SHOULD have an ’extlang’ record assigned to it as well. For example, any language with a macrolanguage of ’zh’ or ’ar’ would be assigned an ’extlang’ record.

2. ’Extlang’ records SHOULD NOT be created for languages if other languages encompassed by the macrolanguage do not also include ’extlang’ records. For example, if a new Serbo-Croatian (’sh’) language were registered, it would not get an extlang record because other languages encompassed, such as Serbian (’sr’), do not include one in the registry.

3. Sign languages SHOULD have an ’extlang’ record with a ’Prefix’ of ’sgn’.

4. ’Extlang’ records MUST NOT be created for items already in the registry. Extended language subtags will only be considered at the time of initial registration.

5. Extended language subtag records MUST include the fields ’Prefix’ and ’Preferred-Value’ with field values assigned as described in Section 2.2.2.

D. Any other codes assigned by ISO 639-2 that do not conflict with existing three-letter primary or extended language

subtags and that do not have ISO 639-1 two-letter codes assigned are entered into the IANA registry as new records of type 'language'. This type of registration is not supposed to occur in the future.

13. Codes assigned by ISO 15924 and ISO 3166-1 that do not conflict with existing subtags of the associated type and whose meaning is not the same as an existing subtag of the same type are entered into the IANA registry as new records.

14. Codes assigned by ISO 639, ISO 15924, or ISO 3166-1 that are withdrawn by their respective maintenance or registration authority remain valid in language tags. A 'Deprecated' field containing the date of withdrawal MUST be added to the record. If a new record of the same type is added that represents a replacement value, then a 'Preferred-Value' field MAY also be added. The registration process MAY be used to add comments about the withdrawal of the code by the respective standard.

    For example: the region code 'TL' was assigned to the country 'Timor-Leste', replacing the code 'TP' (which was assigned to 'East Timor' when it was under administration by Portugal). The subtag 'TP' remains valid in language tags, but its record contains the 'Preferred-Value' of 'TL' and its field 'Deprecated' contains the date the new code was assigned ('2004-07-06').

15. Codes assigned by ISO 639, ISO 15924, or ISO 3166-1 that conflict with existing subtags of the associated type, including subtags that are deprecated, MUST NOT be entered into the registry. The following additional considerations apply to subtag values that are reassigned:

    A. For ISO 639 codes, if the newly assigned code's meaning is not represented by a subtag in the IANA registry, the Language Subtag Reviewer, as described in Section 3.5, SHALL prepare a proposal for entering in the IANA registry, as soon as practical, a registered language subtag as an alternate value for the new code. The form of the registered language subtag will be at the discretion of the Language Subtag Reviewer and MUST conform to other restrictions on language subtags in this document.

    B. For all subtags whose meaning is derived from an external standard (that is, by ISO 639, ISO 15924, ISO 3166-1, or UN M.49), if a new meaning is assigned to an existing code and the new meaning broadens the meaning of that code, then the meaning for the associated subtag MAY be changed to match.

The meaning of a subtag MUST NOT be narrowed, however, as this can result in an unknown proportion of the existing uses of a subtag becoming invalid. Note: the ISO 639 registration authority (RA) has adopted a similar stability policy.

C. For ISO 15924 codes, if the newly assigned code’s meaning is not represented by a subtag in the IANA registry, the Language Subtag Reviewer, as described in Section 3.5, SHALL prepare a proposal for entering in the IANA registry, as soon as practical, a registered variant subtag as an alternate value for the new code. The form of the registered variant subtag will be at the discretion of the Language Subtag Reviewer and MUST conform to other restrictions on variant subtags in this document.

D. For ISO 3166-1 codes, if the newly assigned code’s meaning is associated with the same UN M.49 code as another ’region’ subtag, then the existing region subtag remains as the preferred value for that region and no new entry is created. A comment MAY be added to the existing region subtag indicating the relationship to the new ISO 3166-1 code.

E. For ISO 3166-1 codes, if the newly assigned code’s meaning is associated with a UN M.49 code that is not represented by an existing region subtag, then the Language Subtag Reviewer, as described in Section 3.5, SHALL prepare a proposal for entering the appropriate UN M.49 country code as an entry in the IANA registry.

F. For ISO 3166-1 codes, if there is no associated UN numeric code, then the Language Subtag Reviewer SHALL petition the UN to create one. If there is no response from the UN within 90 days of the request being sent, the Language Subtag Reviewer SHALL prepare a proposal for entering in the IANA registry, as soon as practical, a registered variant subtag as an alternate value for the new code. The form of the registered variant subtag will be at the discretion of the Language Subtag Reviewer and MUST conform to other restrictions on variant subtags in this document. This situation is very unlikely to ever occur.

16. UN M.49 has codes for both "countries and areas" (such as ’276’ for Germany) and "geographical regions and sub-regions" (such as ’150’ for Europe). UN M.49 country or area codes for which there is no corresponding ISO 3166-1 code MUST NOT be registered, except as a surrogate for an ISO 3166-1 code that is blocked from registration by an existing subtag.

If such a code becomes necessary, then the maintenance agency for ISO 3166-1 SHALL first be petitioned to assign a code to the region. If the petition for a code assignment by ISO 3166-1 is refused or not acted on in a timely manner, the registration process described in Section 3.5 can then be used to register the corresponding UN M.49 code. This way, UN M.49 codes remain available as the value of last resort in cases where ISO 3166-1 reassigns a deprecated value in the registry.

17. The redundant and grandfathered entries together form the complete list of tags registered under [RFC3066]. The redundant tags are those previously registered tags that can now be formed using the subtags defined in the registry. The grandfathered entries include those that can never be legal because they are 'irregular' (that is, they do not match the 'langtag' production in Figure 1), are limited by rule (subtags such as 'nyn' and 'min' look like the extlang production, but cannot be registered as extended language subtags), or their subtags are inappropriate for registration. All of the grandfathered tags are listed in either the 'regular' or the 'irregular' productions in the ABNF. Under [RFC4646] it was possible for grandfathered tags to become redundant. However, all of the tags for which this was possible became redundant before this document was produced. So the set of redundant and grandfathered tags is now permanent and immutable: new entries of either type MUST NOT be added and existing entries MUST NOT be removed. The decision-making process about which tags were initially grandfathered and which were made redundant is described in [RFC4645].

    Many of the grandfathered tags are deprecated -- indeed, they were deprecated even before [RFC4646]. For example, the tag "art-lojban" was deprecated in favor of the primary language subtag 'jbo'. These tags could have been made 'redundant' by registering some of their subtags as 'variants'. The 'variant-like' subtags in the grandfathered registrations SHALL NOT be registered in the future, even with a similar or identical meaning.

3.5. Registration Procedure for Subtags

The procedure given here MUST be used by anyone who wants to use a subtag not currently in the IANA Language Subtag Registry or who wishes to add, modify, update, or remove information in existing records as permitted by this document.

Only subtags of type 'language' and 'variant' will be considered for independent registration of new subtags. Subtags needed for

stability and subtags necessary to keep the registry synchronized with ISO 639, ISO 15924, ISO 3166, and UN M.49 within the limits defined by this document also use this process, as described in Section 3.3 and subject to stability provisions as described in Section 3.4.

Registration requests are accepted relating to information in the 'Comments', 'Deprecated', 'Description', 'Prefix', 'Preferred-Value', 'Macrolanguage', or 'Suppress-Script' fields in a subtag's record as described in Section 3.4. Changes to all other fields in the IANA registry are NOT permitted.

Registering a new subtag or requesting modifications to an existing tag or subtag starts with the requester filling out the registration form reproduced below. Note that each response is not limited in size so that the request can adequately describe the registration. The fields in the "Record Requested" section need to follow the requirements in Section 3.1 before the record will be approved.

```
LANGUAGE SUBTAG REGISTRATION FORM
1. Name of requester:
2. E-mail address of requester:
3. Record Requested:

   Type:
   Subtag:
   Description:
   Prefix:
   Preferred-Value:
   Deprecated:
   Suppress-Script:
   Macrolanguage:
   Comments:

4. Intended meaning of the subtag:
5. Reference to published description
   of the language (book or article):
6. Any other relevant information:
```

Figure 5: The Language Subtag Registration Form

Examples of completed registration forms can be found in Appendix B. A complete list of approved registration forms is online through http://www.iana.org; readers should note that the Language Tag Registry is now obsolete and should instead look for the Language Subtag Registry.

The subtag registration form MUST be sent to <ietf-languages@iana.org>. Registration requests receive a two-week review period before being approved and submitted to IANA for inclusion in the registry. If modifications are made to the request during the course of the registration process (such as corrections to meet the requirements in Section 3.1 or to make the 'Description' fields unique for the given record type), the modified form MUST also be sent to <ietf-languages@iana.org> at least one week prior to submission to IANA.

The ietf-languages list is an open list and can be joined by sending a request to <ietf-languages-request@iana.org>. The list can be hosted by IANA or any third party at the request of IESG.

Before forwarding any registration to IANA, the Language Subtag Reviewer MUST ensure that all requirements in this document are met. This includes ensuring that values in the 'Subtag' field match case according to the description in Section 3.1.4 and that 'Description' fields are unique for the given record type as described in Section 3.1.5. The Reviewer MUST also ensure that an appropriate File-Date record is included in the request, to assist IANA when updating the registry (see Section 5.1).

Some fields in both the registration form as well as the registry record itself permit the use of non-ASCII characters. Registration requests SHOULD use the UTF-8 encoding for consistency and clarity. However, since some mail clients do not support this encoding, other encodings MAY be used for the registration request. The Language Subtag Reviewer is responsible for ensuring that the proper Unicode characters appear in both the archived request form and the registry record. In the case of a transcription or encoding error by IANA, the Language Subtag Reviewer will request that the registry be repaired, providing any necessary information to assist IANA.

Extended language subtags (type 'extlang'), by definition, are always encompassed by another language. All records of type 'extlang' MUST, therefore, contain a 'Prefix' field at the time of registration. This 'Prefix' field can never be altered or removed, and requests to do so MUST be rejected.

Variant subtags are usually registered for use with a particular range of language tags, and variant subtags based on the terminology of the language to which they are apply are encouraged. For example, the subtag 'rozaj' (Resian) is intended for use with language tags that start with the primary language subtag "sl" (Slovenian), since Resian is a dialect of Slovenian. Thus, the subtag 'rozaj' would be appropriate in tags such as "sl-Latn-rozaj" or "sl-IT-rozaj". This information is stored in the 'Prefix' field in the registry. Variant

registration requests SHOULD include at least one ’Prefix’ field in the registration form.

Requests to assign an additional record of a given type with an existing subtag value MUST be rejected. For example, the variant subtag ’rozaj’ already exists in the registry, so adding a second record of type ’variant’ with the subtag ’rozaj’ is prohibited.

The ’Prefix’ field for a given registered variant subtag exists in the IANA registry as a guide to usage. Additional ’Prefix’ fields MAY be added by filing an additional registration form. In that form, the "Any other relevant information:" field MUST indicate that it is the addition of a prefix.

Requests to add a ’Prefix’ field to a variant subtag that imply a different semantic meaning SHOULD be rejected. For example, a request to add the prefix "de" to the subtag ’1994’ so that the tag "de-1994" represented some German dialect or orthographic form would be rejected. The ’1994’ subtag represents a particular Slovenian orthography, and the additional registration would change or blur the semantic meaning assigned to the subtag. A separate subtag SHOULD be proposed instead.

Requests to add a ’Prefix’ to a variant subtag that has no current ’Prefix’ field MUST be rejected. Variants are registered with no prefix because they are potentially useful with many or even all languages. Adding one or more ’Prefix’ fields would be potentially harmful to the use of the variant, since it dramatically reduces the scope of the subtag (which is not allowed under the stability rules (Section 3.4) as opposed to broadening the scope of the subtag, which is what the addition of a ’Prefix’ normally does. An example of such a "no-prefix" variant is the subtag ’fonipa’, which represents the International Phonetic Alphabet, a scheme that can be used to transcribe many languages.

The ’Description’ fields provided in the request MUST contain at least one description written or transcribed into the Latin script; the request MAY also include additional ’Description’ fields in any script or language. The ’Description’ field is used for identification purposes and doesn’t necessarily represent the actual native name of the language or variation. It also doesn’t have to be in any particular language, but SHOULD be both suitable and sufficient to identify the item in the record. The Language Subtag Reviewer will check and edit any proposed ’Description’ fields so as to ensure uniqueness and prevent collisions with ’Description’ fields in other records of the same type. If this occurs in an independent registration request, the Language Subtag Reviewer MUST resubmit the record to <ietf-languages@iana.org>, treating it as a modification of

a request due to discussion, as described in Section 3.5, unless the request's sole purpose is to introduce a duplicate 'Description' field, in which case the request SHALL be rejected.

The 'Description' field is not guaranteed to be stable. Corrections or clarifications of intent are examples of possible changes. Attempts to provide translations or transcriptions of entries in the registry (which, by definition, provide no new information) are unlikely to be approved.

Soon after the two-week review period has passed, the Language Subtag Reviewer MUST take one of the following actions:

- o Explicitly accept the request and forward the form containing the record to be inserted or modified to <iana@iana.org> according to the procedure described in Section 3.3.

- o Explicitly reject the request because of significant objections raised on the list or due to problems with constraints in this document (which MUST be explicitly cited).

- o Extend the review period by granting an additional two-week increment to permit further discussion. After each two-week increment, the Language Subtag Reviewer MUST indicate on the list whether the registration has been accepted, rejected, or extended.

Note that the Language Subtag Reviewer MAY raise objections on the list if he or she so desires. The important thing is that the objection MUST be made publicly.

Sometimes the request needs to be modified as a result of discussion during the review period or due to requirements in this document. The applicant, Language Subtag Reviewer, or others MAY submit a modified version of the completed registration form, which will be considered in lieu of the original request with the explicit approval of the applicant. Such changes do not restart the two-week discussion period, although an application containing the final record submitted to IANA MUST appear on the list at least one week prior to the Language Subtag Reviewer forwarding the record to IANA. The applicant MAY modify a rejected application with more appropriate or additional information and submit it again; this starts a new two-week comment period.

Registrations initiated due to the provisions of Section 3.3 or Section 3.4 SHALL NOT be rejected altogether (since they have to ultimately appear in the registry) and SHOULD be completed as quickly as possible. The review process allows list members to comment on the specific information in the form and the record it contains and

thus help ensure that it is correct and consistent. The Language Subtag Reviewer MAY reject a specific version of the form, but MUST propose a suitable replacement, extending the review period as described above, until the form is in a format worthy of the reviewer's approval and meets with rough consensus of the list.

Decisions made by the Language Subtag Reviewer MAY be appealed to the IESG [RFC2028] under the same rules as other IETF decisions [RFC2026]. This includes a decision to extend the review period or the failure to announce a decision in a clear and timely manner.

The approved records appear in the Language Subtag Registry. The approved registration forms are available online from http://www.iana.org.

Updates or changes to existing records follow the same procedure as new registrations. The Language Subtag Reviewer decides whether there is consensus to update the registration following the two-week review period; normally, objections by the original registrant will carry extra weight in forming such a consensus.

Registrations are permanent and stable. Once registered, subtags will not be removed from the registry and will remain a valid way in which to specify a specific language or variant.

Note: The purpose of the "Reference to published description" section in the registration form is to aid in verifying whether a language is registered or to which language or language variation a particular subtag refers. In most cases, reference to an authoritative grammar or dictionary of that language will be useful; in cases where no such work exists, other well-known works describing that language or in that language MAY be appropriate. The Language Subtag Reviewer decides what constitutes "good enough" reference material. This requirement is not intended to exclude particular languages or dialects due to the size of the speaker population or lack of a standardized orthography. Minority languages will be considered equally on their own merits.

3.6. Possibilities for Registration

Possibilities for registration of subtags or information about subtags include:

- Primary language subtags for languages not listed in ISO 639 that are not variants of any listed or registered language MAY be registered. At the time this document was created, there were no examples of this form of subtag. Before attempting to register a language subtag, there MUST be an attempt to register the language

with ISO 639. Subtags MUST NOT be registered for languages defined by codes that exist in ISO 639-1, ISO 639-2, or ISO 639-3; that are under consideration by the ISO 639 registration authorities; or that have never been attempted for registration with those authorities. If ISO 639 has previously rejected a language for registration, it is reasonable to assume that there must be additional, very compelling evidence of need before it will be registered as a primary language subtag in the IANA registry (to the extent that it is very unlikely that any subtags will be registered of this type).

- o Dialect or other divisions or variations within a language, its orthography, writing system, regional or historical usage, transliteration or other transformation, or distinguishing variation MAY be registered as variant subtags. An example is the 'rozaj' subtag (the Resian dialect of Slovenian).

- o The addition or maintenance of fields (generally of an informational nature) in tag or subtag records as described in Section 3.1 is allowed. Such changes are subject to the stability provisions in Section 3.4. This includes 'Description', 'Comments', 'Deprecated', and 'Preferred-Value' fields for obsolete or withdrawn codes, or the addition of 'Suppress-Script' or 'Macrolanguage' fields to primary language subtags, as well as other changes permitted by this document, such as the addition of an appropriate 'Prefix' field to a variant subtag.

- o The addition of records and related field value changes necessary to reflect assignments made by ISO 639, ISO 15924, ISO 3166-1, and UN M.49 as described in Section 3.4 is allowed.

Subtags proposed for registration that would cause all or part of a grandfathered tag to become redundant but whose meaning conflicts with or alters the meaning of the grandfathered tag MUST be rejected.

This document leaves the decision on what subtags or changes to subtags are appropriate (or not) to the registration process described in Section 3.5.

Note: Four-character primary language subtags are reserved to allow for the possibility of alpha4 codes in some future addition to the ISO 639 family of standards.

ISO 639 defines a registration authority for additions to and changes in the list of languages in ISO 639. This agency is:

International Information Centre for Terminology (Infoterm)
Aichholzgasse 6/12, AT-1120
Wien, Austria
Phone: +43 1 26 75 35 Ext. 312 Fax: +43 1 216 32 72

ISO 639-2 defines a registration authority for additions to and changes in the list of languages in ISO 639-2. This agency is:

Library of Congress
Network Development and MARC Standards Office
Washington, DC 20540, USA
Phone: +1 202 707 6237 Fax: +1 202 707 0115
URL: http://www.loc.gov/standards/iso639-2

ISO 639-3 defines a registration authority for additions to and changes in the list of languages in ISO 639-3. This agency is:

SIL International
ISO 639-3 Registrar
7500 W. Camp Wisdom Rd.
Dallas, TX 75236, USA
Phone: +1 972 708 7400, ext. 2293
Fax: +1 972 708 7546
Email: iso639-3@sil.org
URL: http://www.sil.org/iso639-3

ISO 639-5 defines a registration authority for additions to and changes in the list of languages in ISO 639-5. This agency is the same as for ISO 639-2 and is:

Library of Congress
Network Development and MARC Standards Office
Washington, DC 20540, USA
Phone: +1 202 707 6237
Fax: +1 202 707 0115
URL: http://www.loc.gov/standards/iso639-5

The maintenance agency for ISO 3166-1 (country codes) is:

ISO 3166 Maintenance Agency
c/o International Organization for Standardization
Case postale 56
CH-1211 Geneva 20, Switzerland
Phone: +41 22 749 72 33 Fax: +41 22 749 73 49
URL: http://www.iso.org/iso/en/prods-services/iso3166ma/index.html

The registration authority for ISO 15924 (script codes) is:

Unicode Consortium
Box 391476
Mountain View, CA 94039-1476, USA
URL: http://www.unicode.org/iso15924

The Statistics Division of the United Nations Secretariat maintains the Standard Country or Area Codes for Statistical Use and can be reached at:

Statistical Services Branch
Statistics Division
United Nations, Room DC2-1620
New York, NY 10017, USA
Fax: +1-212-963-0623
Email: statistics@un.org
URL: http://unstats.un.org/unsd/methods/m49/m49alpha.htm

3.7. Extensions and the Extensions Registry

Extension subtags are those introduced by single-character subtags ("singletons") other than ’x’. They are reserved for the generation of identifiers that contain a language component and are compatible with applications that understand language tags.

The structure and form of extensions are defined by this document so that implementations can be created that are forward compatible with applications that might be created using singletons in the future. In addition, defining a mechanism for maintaining singletons will lend stability to this document by reducing the likely need for future revisions or updates.

Single-character subtags are assigned by IANA using the "IETF Review" policy defined by [RFC5226]. This policy requires the development of an RFC, which SHALL define the name, purpose, processes, and procedures for maintaining the subtags. The maintaining or registering authority, including name, contact email, discussion list email, and URL location of the registry, MUST be indicated clearly in the RFC. The RFC MUST specify or include each of the following:

- The specification MUST reference the specific version or revision of this document that governs its creation and MUST reference this section of this document.

- The specification and all subtags defined by the specification MUST follow the ABNF and other rules for the formation of tags and subtags as defined in this document. In particular, it MUST

specify that case is not significant and that subtags MUST NOT exceed eight characters in length.

- The specification MUST specify a canonical representation.

- The specification of valid subtags MUST be available over the Internet and at no cost.

- The specification MUST be in the public domain or available via a royalty-free license acceptable to the IETF and specified in the RFC.

- The specification MUST be versioned, and each version of the specification MUST be numbered, dated, and stable.

- The specification MUST be stable. That is, extension subtags, once defined by a specification, MUST NOT be retracted or change in meaning in any substantial way.

- The specification MUST include, in a separate section, the registration form reproduced in this section (below) to be used in registering the extension upon publication as an RFC.

- IANA MUST be informed of changes to the contact information and URL for the specification.

IANA will maintain a registry of allocated single-character (singleton) subtags. This registry MUST use the record-jar format described by the ABNF in Section 3.1.1. Upon publication of an extension as an RFC, the maintaining authority defined in the RFC MUST forward this registration form to <iesg@ietf.org>, who MUST forward the request to <iana@iana.org>. The maintaining authority of the extension MUST maintain the accuracy of the record by sending an updated full copy of the record to <iana@iana.org> with the subject line "LANGUAGE TAG EXTENSION UPDATE" whenever content changes. Only the ’Comments’, ’Contact_Email’, ’Mailing_List’, and ’URL’ fields MAY be modified in these updates.

Failure to maintain this record, maintain the corresponding registry, or meet other conditions imposed by this section of this document MAY be appealed to the IESG [RFC2028] under the same rules as other IETF decisions (see [RFC2026]) and MAY result in the authority to maintain the extension being withdrawn or reassigned by the IESG.

```
%%
Identifier:
Description:
Comments:
Added:
RFC:
Authority:
Contact_Email:
Mailing_List:
URL:
%%
```

Figure 6: Format of Records in the Language Tag Extensions Registry

'Identifier' contains the single-character subtag (singleton) assigned to the extension. The Internet-Draft submitted to define the extension SHOULD specify which letter or digit to use, although the IESG MAY change the assignment when approving the RFC.

'Description' contains the name and description of the extension.

'Comments' is an OPTIONAL field and MAY contain a broader description of the extension.

'Added' contains the date the extension's RFC was published in the "full-date" format specified in [RFC3339]. For example: 2004-06-28 represents June 28, 2004, in the Gregorian calendar.

'RFC' contains the RFC number assigned to the extension.

'Authority' contains the name of the maintaining authority for the extension.

'Contact_Email' contains the email address used to contact the maintaining authority.

'Mailing_List' contains the URL or subscription email address of the mailing list used by the maintaining authority.

'URL' contains the URL of the registry for this extension.

The determination of whether an Internet-Draft meets the above conditions and the decision to grant or withhold such authority rests solely with the IESG and is subject to the normal review and appeals process associated with the RFC process.

Extension authors are strongly cautioned that many (including most well-formed) processors will be unaware of any special relationships

or meaning inherent in the order of extension subtags. Extension authors SHOULD avoid subtag relationships or canonicalization mechanisms that interfere with matching or with length restrictions that sometimes exist in common protocols where the extension is used. In particular, applications MAY truncate the subtags in doing matching or in fitting into limited lengths, so it is RECOMMENDED that the most significant information be in the most significant (left-most) subtags and that the specification gracefully handle truncated subtags.

When a language tag is to be used in a specific, known protocol, it is RECOMMENDED that the language tag not contain extensions not supported by that protocol. In addition, note that some protocols MAY impose upper limits on the length of the strings used to store or transport the language tag.

3.8. Update of the Language Subtag Registry

After the adoption of this document, the IANA Language Subtag Registry needed an update so that it would contain the complete set of subtags valid in a language tag. [RFC5645] describes the process used to create this update.

Registrations that are in process under the rules defined in [RFC4646] when this document is adopted MUST be completed under the rules contained in this document.

3.9. Applicability of the Subtag Registry

The Language Subtag Registry is the source of data elements used to construct language tags, following the rules described in this document. Language tags are designed for indicating linguistic attributes of various content, including not only text but also most media formats, such as video or audio. They also form the basis for language and locale negotiation in various protocols and APIs.

The registry is therefore applicable to many applications that need some form of language identification, with these limitations:

- It is not designed to be the sole data source in the creation of a language-selection user interface. For example, the registry does not contain translations for subtag descriptions or for tags composed from the subtags. Sources for localized data based on the registry are generally available, notably [CLDR]. Nor does the registry indicate which subtag combinations are particularly useful or relevant.

- It does not provide information indicating relationships between different languages, such as might be used in a user interface to select language tags hierarchically, regionally, or on some other organizational model.

- It does not supply information about potential overlap between different language tags, as the notion of what constitutes a language is not precise: several different language tags might be reasonable choices for the same given piece of content.

- It does not contain information about appropriate fallback choices when performing language negotiation. A good fallback language might be linguistically unrelated to the specified language. The fact that one language is often used as a fallback language for another is usually a result of outside factors, such as geography, history, or culture -- factors that might not apply in all cases. For example, most people who use Breton (a Celtic language used in the Northwest of France) would probably prefer to be served French (a Romance language) if Breton isn’t available.

4. Formation and Processing of Language Tags

This section addresses how to use the information in the registry with the tag syntax to choose, form, and process language tags.

4.1. Choice of Language Tag

The guiding principle in forming language tags is to "tag content wisely." Sometimes there is a choice between several possible tags for the same content. The choice of which tag to use depends on the content and application in question, and some amount of judgment might be necessary when selecting a tag.

Interoperability is best served when the same language tag is used consistently to represent the same language. If an application has requirements that make the rules here inapplicable, then that application risks damaging interoperability. It is strongly RECOMMENDED that users not define their own rules for language tag choice.

Standards, protocols, and applications that reference this document normatively but apply different rules to the ones given in this section MUST specify how language tag selection varies from the guidelines given here.

To ensure consistent backward compatibility, this document contains several provisions to account for potential instability in the standards used to define the subtags that make up language tags.

These provisions mean that no valid language tag can become invalid, nor will a language tag have a narrower scope in the future (it may have a broader scope). The most appropriate language tag for a given application or content item might evolve over time, but once applied, the tag itself cannot become invalid or have its meaning wholly change.

A subtag SHOULD only be used when it adds useful distinguishing information to the tag. Extraneous subtags interfere with the meaning, understanding, and processing of language tags. In particular, users and implementations SHOULD follow the 'Prefix' and 'Suppress-Script' fields in the registry (defined in Section 3.1): these fields provide guidance on when specific additional subtags SHOULD be used or avoided in a language tag.

The choice of subtags used to form a language tag SHOULD follow these guidelines:

1. Use as precise a tag as possible, but no more specific than is justified. Avoid using subtags that are not important for distinguishing content in an application.

   * For example, 'de' might suffice for tagging an email written in German, while "de-CH-1996" is probably unnecessarily precise for such a task.

   * Note that some subtag sequences might not represent the language a casual user might expect. For example, the Swiss German (Schweizerdeutsch) language is represented by "gsw-CH" and not by "de-CH". This latter tag represents German ('de') as used in Switzerland ('CH'), also known as Swiss High German (Schweizer Hochdeutsch). Both are real languages, and distinguishing between them could be important to an application.

2. The script subtag SHOULD NOT be used to form language tags unless the script adds some distinguishing information to the tag. Script subtags were first formally defined in [RFC4646]. Their use can affect matching and subtag identification for implementations of [RFC1766] or [RFC3066] (which are obsoleted by this document), as these subtags appear between the primary language and region subtags. Some applications can benefit from the use of script subtags in language tags, as long as the use is consistent for a given context. Script subtags are never appropriate for unwritten content (such as audio recordings). The field 'Suppress-Script' in the primary or extended language record in the registry indicates script subtags that do not add distinguishing information for most applications; this field

defines when users SHOULD NOT include a script subtag with a particular primary language subtag.

For example, if an implementation selects content using Basic Filtering [RFC4647] (originally described in Section 14.4 of [RFC2616]) and the user requested the language range "en-US", content labeled "en-Latn-US" will not match the request and thus not be selected. Therefore, it is important to know when script subtags will customarily be used and when they ought not be used.

For example:

* The subtag ’Latn’ should not be used with the primary language ’en’ because nearly all English documents are written in the Latin script and it adds no distinguishing information. However, if a document were written in English mixing Latin script with another script such as Braille (’Brai’), then it might be appropriate to choose to indicate both scripts to aid in content selection, such as the application of a style sheet.

* When labeling content that is unwritten (such as a recording of human speech), the script subtag should not be used, even if the language is customarily written in several scripts. Thus, the subtitles to a movie might use the tag "uz-Arab" (Uzbek, Arabic script), but the audio track for the same language would be tagged simply "uz". (The tag "uz-Zxxx" could also be used where content is not written, as the subtag ’Zxxx’ represents the "Code for unwritten documents".)

3. If a tag or subtag has a ’Preferred-Value’ field in its registry entry, then the value of that field SHOULD be used to form the language tag in preference to the tag or subtag in which the preferred value appears.

   * For example, use ’jbo’ for Lojban in preference to the grandfathered tag "art-lojban".

4. Use subtags or sequences of subtags for individual languages in preference to subtags for language collections. A "language collection" is a group of languages that are descended from a common ancestor, are spoken in the same geographical area, or are otherwise related. Certain language collections are assigned codes by [ISO639-5] (and some of these [ISO639-5] codes are also defined as collections in [ISO639-2]). These codes are included as primary language subtags in the registry. Subtags for a language collection in the registry have a ’Scope’ field with a value of ’collection’. A subtag for a language collection is

always preferred to less specific alternatives such as ’mul’ and ’und’ (see below), and a subtag representing a language collection MAY be used when more specific language information is not available. However, most users and implementations do not know there is a relationship between the collection and its individual languages. In addition, the relationship between the individual languages in the collection is not well defined; in particular, the languages are usually not mutually intelligible. Since the subtags are different, a request for the collection will typically only produce items tagged with the collection’s subtag, not items tagged with subtags for the individual languages contained in the collection.

* For example, collections are interpreted inclusively, so the subtag ’gem’ (Germanic languages) could, but SHOULD NOT, be used with content that would be better tagged with "en" (English), "de" (German), or "gsw" (Swiss German, Alemannic). While ’gem’ collects all of these (and other) languages, most implementations will not match ’gem’ to the individual languages; thus, using the subtag will not produce the desired result.

5. [ISO639-2] has defined several codes included in the subtag registry that require additional care when choosing language tags. In most of these cases, where omitting the language tag is permitted, such omission is preferable to using these codes. Language tags SHOULD NOT incorporate these subtags as a prefix, unless the additional information conveys some value to the application.

   * The ’mul’ (Multiple) primary language subtag identifies content in multiple languages. This subtag SHOULD NOT be used when a list of languages or individual tags for each content element can be used instead. For example, the ’Content-Language’ header [RFC3282] allows a list of languages to be used, not just a single language tag.

   * The ’und’ (Undetermined) primary language subtag identifies linguistic content whose language is not determined. This subtag SHOULD NOT be used unless a language tag is required and language information is not available or cannot be determined. Omitting the language tag (where permitted) is preferred. The ’und’ subtag might be useful for protocols that require a language tag to be provided or where a primary language subtag is required (such as in "und-Latn"). The ’und’ subtag MAY also be useful when matching language tags in certain situations.

* The ’zxx’ (Non-Linguistic, Not Applicable) primary language subtag identifies content for which a language classification is inappropriate or does not apply. Some examples might include instrumental or electronic music; sound recordings consisting of nonverbal sounds; audiovisual materials with no narration, dialog, printed titles, or subtitles; machine-readable data files consisting of machine languages or character codes; or programming source code.

* The ’mis’ (Uncoded) primary language subtag identifies content whose language is known but that does not currently have a corresponding subtag. This subtag SHOULD NOT be used. Because the addition of other codes in the future can render its application invalid, it is inherently unstable and hence incompatible with the stability goals of BCP 47. It is always preferable to use other subtags: either ’und’ or (with prior agreement) private use subtags.

6. Use variant subtags sparingly and in the correct order. Most variant subtags have one or more ’Prefix’ fields in the registry that express the list of subtags with which they are appropriate. Variants SHOULD only be used with subtags that appear in one of these ’Prefix’ fields. If a variant lists a second variant in one of its ’Prefix’ fields, the first variant SHOULD appear directly after the second variant in any language tag where both occur. General purpose variants (those with no ’Prefix’ fields at all) SHOULD appear after any other variant subtags. Order any remaining variants by placing the most significant subtag first. If none of the subtags is more significant or no relationship can be determined, alphabetize the subtags. Because variants are very specialized, using many of them together generally makes the tag so narrow as to override the additional precision gained. Putting the subtags into another order interferes with interoperability, as well as the overall interpretation of the tag.

   For example:

   * The tag "en-scotland-fonipa" (English, Scottish dialect, IPA phonetic transcription) is correctly ordered because ’scotland’ has a ’Prefix’ of "en", while ’fonipa’ has no ’Prefix’ field.

   * The tag "sl-IT-rozaj-biske-1994" is correctly ordered: ’rozaj’ lists "sl" as its sole ’Prefix’; ’biske’ lists "sl-rozaj" as its sole ’Prefix’. The subtag ’1994’ has several prefixes,

including "sl-rozaj". However, it follows both ’rozaj’ and ’biske’ because one of its ’Prefix’ fields is "sl-rozaj-biske".

7. The grandfathered tag "i-default" (Default Language) was originally registered according to [RFC1766] to meet the needs of [RFC2277]. It is not used to indicate a specific language, but rather to identify the condition or content used where the language preferences of the user cannot be established. It SHOULD NOT be used except as a means of labeling the default content for applications or protocols that require default language content to be labeled with that specific tag. It MAY also be used by an application or protocol to identify when the default language content is being returned.

4.1.1. Tagging Encompassed Languages

Some primary language records in the registry have a ’Macrolanguage’ field (Section 3.1.10) that contains a mapping from each "encompassed language" to its macrolanguage. The ’Macrolanguage’ mapping doesn’t define what the relationship between the encompassed language and its macrolanguage is, nor does it define how languages encompassed by the same macrolanguage are related to each other. Two different languages encompassed by the same macrolanguage may differ from one another more than, say, French and Spanish do.

A few specific macrolanguages, such as Chinese (’zh’) and Arabic (’ar’), are handled differently. See Section 4.1.2.

The more specific encompassed language subtag SHOULD be used to form the language tag, although either the macrolanguage’s primary language subtag or the encompassed language’s subtag MAY be used. This means, for example, tagging Plains Cree with ’crk’ rather than ’cr’ (Cree), and so forth.

Each macrolanguage subtag’s scope, by definition, includes all of its encompassed languages. Since the relationship between encompassed languages varies, users cannot assume that the macrolanguage subtag means any particular encompassed language, nor that any given pair of encompassed languages are mutually intelligible or otherwise interchangeable.

Applications MAY use macrolanguage information to improve matching or language negotiation. For example, the information that ’sr’ (Serbian) and ’hr’ (Croatian) share a macrolanguage expresses a closer relation between those languages than between, say, ’sr’ (Serbian) and ’ma’ (Macedonian). However, this relationship is not guaranteed nor is it exclusive. For example, Romanian (’ro’) and

Moldavian (’mo’) do not share a macrolanguage, but are far more closely related to each other than Cantonese (’yue’) and Wu (’wuu’), which do share a macrolanguage.

4.1.2. Using Extended Language Subtags

To accommodate language tag forms used prior to the adoption of this document, language tags provide a special compatibility mechanism: the extended language subtag. Selected languages have been provided with both primary and extended language subtags. These include macrolanguages, such as Malay (’ms’) and Uzbek (’uz’), that have a specific dominant variety that is generally synonymous with the macrolanguage. Other languages, such as the Chinese (’zh’) and Arabic (’ar’) macrolanguages and the various sign languages (’sgn’), have traditionally used their primary language subtag, possibly coupled with various region subtags or as part of a registered grandfathered tag, to indicate the language.

With the adoption of this document, specific ISO 639-3 subtags became available to identify the languages contained within these diverse language families or groupings. This presents a choice of language tags where previously none existed:

- Each encompassed language’s subtag SHOULD be used as the primary language subtag. For example, a document in Mandarin Chinese would be tagged "cmn" (the subtag for Mandarin Chinese) in preference to "zh" (Chinese).

- If compatibility is desired or needed, the encompassed subtag MAY be used as an extended language subtag. For example, a document in Mandarin Chinese could be tagged "zh-cmn" instead of either "cmn" or "zh".

- The macrolanguage or prefixing subtag MAY still be used to form the tag instead of the more specific encompassed language subtag. That is, tags such as "zh-HK" or "sgn-RU" are still valid.

Chinese (’zh’) provides a useful illustration of this. In the past, various content has used tags beginning with the ’zh’ subtag, with application-specific meaning being associated with region codes, private use sequences, or grandfathered registered values. This is because historically only the macrolanguage subtag ’zh’ was available for forming language tags. However, the languages encompassed by the Chinese subtag ’zh’ are, in the main, not mutually intelligible when spoken, and the written forms of these languages also show wide variation in form and usage.

To provide compatibility, Chinese languages encompassed by the 'zh' subtag are in the registry both as primary language subtags and as extended language subtags. For example, the ISO 639-3 code for Cantonese is 'yue'. Content in Cantonese might historically have used a tag such as "zh-HK" (since Cantonese is commonly spoken in Hong Kong), although that tag actually means any type of Chinese as used in Hong Kong. With the availability of ISO 639-3 codes in the registry, content in Cantonese can be directly tagged using the 'yue' subtag. The content can use it as a primary language subtag, as in the tag "yue-HK" (Cantonese, Hong Kong). Or it can use an extended language subtag with 'zh', as in the tag "zh-yue-Hant" (Chinese, Cantonese, Traditional script).

As noted above, applications can choose to use the macrolanguage subtag to form the tag instead of using the more specific encompassed language subtag. For example, an application with large quantities of data already using tags with the 'zh' (Chinese) subtag might continue to use this more general subtag even for new data, even though the content could be more precisely tagged with 'cmn' (Mandarin), 'yue' (Cantonese), 'wuu' (Wu), and so on. Similarly, an application already using tags that start with the 'ar' (Arabic) subtag might continue to use this more general subtag even for new data, which could be more precisely tagged with 'arb' (Standard Arabic).

In some cases, the encompassed languages had tags registered for them during the RFC 3066 era. Those grandfathered tags not already deprecated or rendered redundant were deprecated in the registry upon adoption of this document. As grandfathered values, they remain valid for use, and some content or applications might use them. As with other grandfathered tags, since implementations might not be able to associate the grandfathered tags with the encompassed language subtag equivalents that are recommended by this document, implementations are encouraged to canonicalize tags for comparison purposes. Some examples of this include the tags "zh-hakka" (Hakka) and "zh-guoyu" (Mandarin or Standard Chinese).

Sign languages share a mode of communication rather than a linguistic heritage. There are many sign languages that have developed independently, and the subtag 'sgn' indicates only the presence of a sign language. A number of sign languages also had grandfathered tags registered for them during the RFC 3066 era. For example, the grandfathered tag "sgn-US" was registered to represent 'American Sign Language' specifically, without reference to the United States. This is still valid, but deprecated: a document in American Sign Language can be labeled either "ase" or "sgn-ase" (the 'ase' subtag is for the language called 'American Sign Language').

4.2. Meaning of the Language Tag

The meaning of a language tag is related to the meaning of the subtags that it contains. Each subtag, in turn, implies a certain range of expectations one might have for related content, although it is not a guarantee. For example, the use of a script subtag such as 'Arab' (Arabic script) does not mean that the content contains only Arabic characters. It does mean that the language involved is predominantly in the Arabic script. Thus, a language tag and its subtags can encompass a very wide range of variation and yet remain appropriate in each particular instance.

Validity of a tag is not the only factor determining its usefulness. While every valid tag has a meaning, it might not represent any real-world language usage. This is unavoidable in a system in which subtags can be combined freely. For example, tags such as "ar-Cyrl-CO" (Arabic, Cyrillic script, as used in Colombia) or "tlh-Kore-AQ-fonipa" (Klingon, Korean script, as used in Antarctica, IPA phonetic transcription) are both valid and unlikely to represent a useful combination of language attributes.

The meaning of a given tag doesn't depend on the context in which it appears. The relationship between a tag's meaning and the information objects to which that tag is applied, however, can vary.

- For a single information object, the associated language tags might be interpreted as the set of languages that is necessary for a complete comprehension of the complete object. Example: Plain text documents.

- For an aggregation of information objects, the associated language tags could be taken as the set of languages used inside components of that aggregation. Examples: Document stores and libraries.

- For information objects whose purpose is to provide alternatives, the associated language tags could be regarded as a hint that the content is provided in several languages and that one has to inspect each of the alternatives in order to find its language or languages. In this case, the presence of multiple tags might not mean that one needs to be multilingual to get complete understanding of the document. Example: MIME multipart/alternative [RFC2046].

- For markup languages, such as HTML and XML, language information can be added to each part of the document identified by the markup structure (including the whole document itself). For example, one could write C'est la vie. inside a German document; the German-speaking user could then access a French-

German dictionary to find out what the marked section meant. If the user were listening to that document through a speech synthesis interface, this formation could be used to signal the synthesizer to appropriately apply French text-to-speech pronunciation rules to that span of text, instead of applying the inappropriate German rules.

- For markup languages and document formats that allow the audience to be identified, a language tag could indicate the audience(s) appropriate for that document. For example, the same HTML document described in the preceding bullet might have an HTTP header "Content-Language: de" to indicate that the intended audience for the file is German (even though three words appear and are identified as being in French within it).

- For systems and APIs, language tags form the basis for most implementations of locale identifiers. For example, see Unicode's CLDR (Common Locale Data Repository) (see UTS #35 [UTS35]) project.

Language tags are related when they contain a similar sequence of subtags. For example, if a language tag B contains language tag A as a prefix, then B is typically "narrower" or "more specific" than A. Thus, "zh-Hant-TW" is more specific than "zh-Hant".

This relationship is not guaranteed in all cases: specifically, languages that begin with the same sequence of subtags are NOT guaranteed to be mutually intelligible, although they might be. For example, the tag "az" shares a prefix with both "az-Latn" (Azerbaijani written using the Latin script) and "az-Cyrl" (Azerbaijani written using the Cyrillic script). A person fluent in one script might not be able to read the other, even though the linguistic content (e.g., what would be heard if both texts were read aloud) might be identical. Content tagged as "az" most probably is written in just one script and thus might not be intelligible to a reader familiar with the other script.

Similarly, not all subtags specify an actual distinction in language. For example, the tags "en-US" and "en-CA" mean, roughly, English with features generally thought to be characteristic of the United States and Canada, respectively. They do not imply that a significant dialectical boundary exists between any arbitrarily selected point in the United States and any arbitrarily selected point in Canada. Neither does a particular region subtag imply that linguistic distinctions do not exist within that region.

4.3. Lists of Languages

In some applications, a single content item might best be associated with more than one language tag. Examples of such a usage include:

- o Content items that contain multiple, distinct varieties. Often this is used to indicate an appropriate audience for a given content item when multiple choices might be appropriate. Examples of this could include:

  - * Metadata about the appropriate audience for a movie title. For example, a DVD might label its individual audio tracks 'de' (German), 'fr' (French), and 'es' (Spanish), but the overall title would list "de, fr, es" as its overall audience.

  - * A French/English, English/French dictionary tagged as both "en" and "fr" to specify that it applies equally to French and English.

  - * A side-by-side or interlinear translation of a document, as is commonly done with classical works in Latin or Greek.

- o Content items that contain a single language but that require multiple levels of specificity. For example, a library might wish to classify a particular work as both Norwegian ('no') and as Nynorsk ('nn') for audiences capable of appreciating the distinction or needing to select content more narrowly.

4.4. Length Considerations

There is no defined upper limit on the size of language tags. While historically most language tags have consisted of language and region subtags with a combined total length of up to six characters, larger tags have always been both possible and have actually appeared in use.

Neither the language tag syntax nor other requirements in this document impose a fixed upper limit on the number of subtags in a language tag (and thus an upper bound on the size of a tag). The language tag syntax suggests that, depending on the specific language, more subtags (and thus a longer tag) are sometimes necessary to completely identify the language for certain applications; thus, it is possible to envision long or complex subtag sequences.

4.4.1. Working with Limited Buffer Sizes

Some applications and protocols are forced to allocate fixed buffer sizes or otherwise limit the length of a language tag. A conformant implementation or specification MAY refuse to support the storage of language tags that exceed a specified length. Any such limitation SHOULD be clearly documented, and such documentation SHOULD include what happens to longer tags (for example, whether an error value is generated or the language tag is truncated). A protocol that allows tags to be truncated at an arbitrary limit, without giving any indication of what that limit is, has the potential to cause harm by changing the meaning of tags in substantial ways.

In practice, most language tags do not require more than a few subtags and will not approach reasonably sized buffer limitations; see Section 4.1.

Some specifications or protocols have limits on tag length but do not have a fixed length limitation. For example, [RFC2231] has no explicit length limitation: the length available for the language tag is constrained by the length of other header components (such as the charset’s name) coupled with the 76-character limit in [RFC2047]. Thus, the "limit" might be 50 or more characters, but it could potentially be quite small.

The considerations for assigning a buffer limit are:

Implementations SHOULD NOT truncate language tags unless the meaning of the tag is purposefully being changed, or unless the tag does not fit into a limited buffer size specified by a protocol for storage or transmission.

Implementations SHOULD warn the user when a tag is truncated since truncation changes the semantic meaning of the tag.

Implementations of protocols or specifications that are space constrained but do not have a fixed limit SHOULD use the longest possible tag in preference to truncation.

Protocols or specifications that specify limited buffer sizes for language tags MUST allow for language tags of at least 35 characters. Note that [RFC4646] recommended a minimum field size of 42 characters because it included all three elements of the ’extlang’ production. Two of these are now permanently reserved, so a registered primary language subtag of the maximum length of 8 characters is now longer than the longest language-extlang combination. Protocols or specifications that commonly use

extensions or private use subtags might wish to reserve or recommend a longer "minimum buffer" size.

The following illustration shows how the 35-character recommendation was derived:

```
language      =  8 ; longest allowed registered value
                   ;   longer than primary+extlang
                   ;   which requires 7 characters
script        =  5 ; if not suppressed: see Section 4.1
region        =  4 ; UN M.49 numeric region code
                   ;   ISO 3166-1 codes require 3
variant1      =  9 ; needs 'language' as a prefix
variant2      =  9 ; very rare, as it needs
                   ;   'language-variant1' as a prefix

total         = 35 characters
```

Figure 7: Derivation of the Limit on Tag Length

4.4.2. Truncation of Language Tags

Truncation of a language tag alters the meaning of the tag, and thus SHOULD be avoided. However, truncation of language tags is sometimes necessary due to limited buffer sizes. Such truncation MUST NOT permit a subtag to be chopped off in the middle or the formation of invalid tags (for example, one ending with the "-" character).

This means that applications or protocols that truncate tags MUST do so by progressively removing subtags along with their preceding "-" from the right side of the language tag until the tag is short enough for the given buffer. If the resulting tag ends with a single-character subtag, that subtag and its preceding "-" MUST also be removed. For example:

Tag to truncate: zh-Latn-CN-variant1-a-extend1-x-wadegile-private1

1. zh-Latn-CN-variant1-a-extend1-x-wadegile
2. zh-Latn-CN-variant1-a-extend1
3. zh-Latn-CN-variant1
4. zh-Latn-CN
5. zh-Latn
6. zh

Figure 8: Example of Tag Truncation

4.5. Canonicalization of Language Tags

Since a particular language tag can be used by many processes, language tags SHOULD always be created or generated in canonical form.

A language tag is in ’canonical form’ when the tag is well-formed according to the rules in Sections 2.1 and 2.2 and it has been canonicalized by applying each of the following steps in order, using data from the IANA registry (see Section 3.1):

1. Extension sequences are ordered into case-insensitive ASCII order by singleton subtag.

   * For example, the subtag sequence ’-a-babble’ comes before ’-b-warble’.

2. Redundant or grandfathered tags are replaced by their ’Preferred-Value’, if there is one.

   * The field-body of the ’Preferred-Value’ for grandfathered and redundant tags is an "extended language range" [RFC4647] and might consist of more than one subtag.

   * ’Preferred-Value’ fields in the registry provide mappings from deprecated tags to modern equivalents. Many of these were created before the adoption of this document (such as the mapping of "no-nyn" to "nn" or "i-klingon" to "tlh"). Others are the result of later registrations or additions to the registry as permitted or required by this document (for example, "zh-hakka" was deprecated in favor of the ISO 639-3 code ’hak’ when this document was adopted).

3. Subtags are replaced by their ’Preferred-Value’, if there is one. For extlangs, the original primary language subtag is also replaced if there is a primary language subtag in the ’Preferred-Value’.

   * The field-body of the ’Preferred-Value’ for extlangs is an "extended language range" and typically maps to a primary language subtag. For example, the subtag sequence "zh-hak" (Chinese, Hakka) is replaced with the subtag ’hak’ (Hakka).

   * Most of the non-extlang subtags are either Region subtags where the country name or designation has changed or clerical corrections to ISO 639-1.

The canonical form contains no ’extlang’ subtags. There is an alternate ’extlang form’ that maintains or reinstates extlang subtags. This form can be useful in environments where the presence of the ’Prefix’ subtag is considered beneficial in matching or selection (see Section 4.1.2).

A language tag is in ’extlang form’ when the tag is well-formed according to the rules in Sections 2.1 and 2.2 and it has been processed by applying each of the following two steps in order, using data from the IANA registry:

1. The language tag is first transformed into canonical form, as described above.

2. If the language tag starts with a primary language subtag that is also an extlang subtag, then the language tag is prepended with the extlang’s ’Prefix’.

   * For example, "hak-CN" (Hakka, China) has the primary language subtag ’hak’, which in turn has an ’extlang’ record with a ’Prefix’ ’zh’ (Chinese). The extlang form is "zh-hak-CN" (Chinese, Hakka, China).

   * Note that Step 2 (prepending a prefix) can restore a subtag that was removed by Step 1 (canonicalizing).

Example: The language tag "en-a-aaa-b-ccc-bbb-x-xyz" is in canonical form, while "en-b-ccc-bbb-a-aaa-X-xyz" is well-formed and potentially valid (extensions ’a’ and ’b’ are not defined as of the publication of this document) but not in canonical form (the extensions are not in alphabetical order).

Example: Although the tag "en-BU" (English as used in Burma) maintains its validity, the language tag "en-BU" is not in canonical form because the ’BU’ subtag has a canonical mapping to ’MM’ (Myanmar).

Canonicalization of language tags does not imply anything about the use of upper- or lowercase letters when processing or comparing subtags (and as described in Section 2.1). All comparisons MUST be performed in a case-insensitive manner.

When performing canonicalization of language tags, processors MAY regularize the case of the subtags (that is, this process is OPTIONAL), following the case used in the registry (see Section 2.1.1).

If more than one variant appears within a tag, processors MAY reorder the variants to obtain better matching behavior or more consistent presentation. Reordering of the variants SHOULD follow the recommendations for variant ordering in Section 4.1.

If the field 'Deprecated' appears in a registry record without an accompanying 'Preferred-Value' field, then that tag or subtag is deprecated without a replacement. These values are canonical when they appear in a language tag. However, tags that include these values SHOULD NOT be selected by users or generated by implementations.

An extension MUST define any relationships that exist between the various subtags in the extension and thus MAY define an alternate canonicalization scheme for the extension's subtags. Extensions MAY define how the order of the extension's subtags is interpreted. For example, an extension could define that its subtags are in canonical order when the subtags are placed into ASCII order: that is, "en-a-aaa-bbb-ccc" instead of "en-a-ccc-bbb-aaa". Another extension might define that the order of the subtags influences their semantic meaning (so that "en-b-ccc-bbb-aaa" has a different value from "en-b-aaa-bbb-ccc"). However, extension specifications SHOULD be designed so that they are tolerant of the typical processes described in Section 3.7.

4.6. Considerations for Private Use Subtags

Private use subtags, like all other subtags, MUST conform to the format and content constraints in the ABNF. Private use subtags have no meaning outside the private agreement between the parties that intend to use or exchange language tags that employ them. The same subtags MAY be used with a different meaning under a separate private agreement. They SHOULD NOT be used where alternatives exist and SHOULD NOT be used in content or protocols intended for general use.

Private use subtags are simply useless for information exchange without prior arrangement. The value and semantic meaning of private use tags and of the subtags used within such a language tag are not defined by this document.

Private use sequences introduced by the 'x' singleton are completely opaque to users or implementations outside of the private use agreement. So, in addition to private use subtag sequences introduced by the singleton subtag 'x', the Language Subtag Registry provides private use language, script, and region subtags derived from the private use codes assigned by the underlying standards. These subtags are valid for use in forming language tags; they are RECOMMENDED over the 'x' singleton private use subtag sequences

because they convey more information via their linkage to the language tag's inherent structure.

For example, the region subtags 'AA', 'ZZ', and those in the ranges 'QM'-'QZ' and 'XA'-'XZ' (derived from the ISO 3166-1 private use codes) can be used to form a language tag. A tag such as "zh-Hans-XQ" conveys a great deal of public, interchangeable information about the language material (that it is Chinese in the simplified Chinese script and is suitable for some geographic region 'XQ'). While the precise geographic region is not known outside of private agreement, the tag conveys far more information than an opaque tag such as "x-somelang" or even "zh-Hans-x-xq" (where the 'xq' subtag's meaning is entirely opaque).

However, in some cases content tagged with private use subtags can interact with other systems in a different and possibly unsuitable manner compared to tags that use opaque, privately defined subtags, so the choice of the best approach sometimes depends on the particular domain in question.

5. IANA Considerations

This section deals with the processes and requirements necessary for IANA to maintain the subtag and extension registries as defined by this document and in accordance with the requirements of [RFC5226].

The impact on the IANA maintainers of the two registries defined by this document will be a small increase in the frequency of new entries or updates. IANA also is required to create a new mailing list (described below in Section 5.1) to announce registry changes and updates.

5.1. Language Subtag Registry

IANA updated the registry using instructions and content provided in a companion document [RFC5645]. The criteria and process for selecting the updated set of records are described in that document. The updated set of records represents no impact on IANA, since the work to create it will be performed externally.

Future work on the Language Subtag Registry includes the following activities:

- Inserting or replacing whole records. These records are preformatted for IANA by the Language Subtag Reviewer, as described in Section 3.3.

- Archiving and making publicly available the registration forms.

- o Announcing each updated version of the registry on the "ietf-languages-announcements@iana.org" mailing list.

Each registration form sent to IANA contains a single record for incorporation into the registry. The form will be sent to <iana@iana.org> by the Language Subtag Reviewer. It will have a subject line indicating whether the enclosed form represents an insertion of a new record (indicated by the word "INSERT" in the subject line) or a replacement of an existing record (indicated by the word "MODIFY" in the subject line). At no time can a record be deleted from the registry.

IANA will extract the record from the form and place the inserted or modified record into the appropriate section of the Language Subtag Registry, grouping the records by their 'Type' field. Inserted records can be placed anywhere within the appropriate section; there is no guarantee that the registry's records will be placed in any particular order except that they will always be grouped by 'Type'. Modified records overwrite the record they replace.

Whenever an entry is created or modified in the registry, the 'File-Date' record at the start of the registry is updated to reflect the most recent modification date. The date format SHALL be the "full-date" format of [RFC3339]. The date SHALL be the date on which that version of the registry was first published by IANA. There SHALL be at most one version of the registry published in a day. A 'File-Date' record is also included in each request to IANA to insert or modify records, indicating the acceptance date of the records in the request.

The updated registry file MUST use the UTF-8 character encoding, and IANA MUST check the registry file for proper encoding. Non-ASCII characters can be sent to IANA by attaching the registration form to the email message or by using various encodings in the mail message body (UTF-8 is recommended). IANA will verify any unclear or corrupted characters with the Language Subtag Reviewer prior to posting the updated registry.

IANA will also archive and make publicly available from http://www.iana.org each registration form. Note that multiple registrations can pertain to the same record in the registry.

Developers who are dependent upon the Language Subtag Registry sometimes would like to be informed of changes in the registry so that they can update their implementations. When any change is made to the Language Subtag Registry, IANA will send an announcement message to <ietf-languages-announcements@iana.org> (a self-subscribing list to which only IANA can post).

5.2. Extensions Registry

The Language Tag Extensions Registry can contain at most 35 records, and thus changes to this registry are expected to be very infrequent.

Future work by IANA on the Language Tag Extensions Registry is limited to two cases. First, the IESG MAY request that new records be inserted into this registry from time to time. These requests MUST include the record to insert in the exact format described in Section 3.7. In addition, there MAY be occasional requests from the maintaining authority for a specific extension to update the contact information or URLs in the record. These requests MUST include the complete, updated record. IANA is not responsible for validating the information provided, only that it is properly formatted. IANA SHOULD take reasonable steps to ascertain that the request comes from the maintaining authority named in the record present in the registry.

6. Security Considerations

Language tags used in content negotiation, like any other information exchanged on the Internet, might be a source of concern because they might be used to infer the nationality of the sender, and thus identify potential targets for surveillance.

This is a special case of the general problem that anything sent is visible to the receiving party and possibly to third parties as well. It is useful to be aware that such concerns can exist in some cases.

The evaluation of the exact magnitude of the threat, and any possible countermeasures, is left to each application protocol (see BCP 72 [RFC3552] for best current practice guidance on security threats and defenses).

The language tag associated with a particular information item is of no consequence whatsoever in determining whether that content might contain possible homographs. The fact that a text is tagged as being in one language or using a particular script subtag provides no assurance whatsoever that it does not contain characters from scripts other than the one(s) associated with or specified by that language tag.

Since there is no limit to the number of variant, private use, and extension subtags, and consequently no limit on the possible length of a tag, implementations need to guard against buffer overflow attacks. See Section 4.4 for details on language tag truncation, which can occur as a consequence of defenses against buffer overflow.

To prevent denial-of-service attacks, applications SHOULD NOT depend on either the Language Subtag Registry or the Language Tag Extensions Registry being always accessible. Additionally, although the specification of valid subtags for an extension (see Section 3.7) MUST be available over the Internet, implementations SHOULD NOT mechanically depend on those sources being always accessible.

The registries specified in this document are not suitable for frequent or real-time access to, or retrieval of, the full registry contents. Most applications do not need registry data at all. For others, being able to validate or canonicalize language tags as of a particular registry date will be sufficient, as the registry contents change only occasionally. Changes are announced to <ietf-languages-announcements@iana.org>. This mailing list is intended for interested organizations and individuals, not for bulk subscription to trigger automatic software updates. The size of the registry makes it unsuitable for automatic software updates. Implementers considering integrating the Language Subtag Registry in an automatic updating scheme are strongly advised to distribute only suitably encoded differences, and only via their own infrastructure -- not directly from IANA.

Changes, or the absence thereof, can also easily be detected by looking at the 'File-Date' record at the start of the registry, or by using features of the protocol used for downloading, without having to download the full registry. At the time of publication of this document, IANA is making the Language Tag Registry available over HTTP 1.1. The proper way to update a local copy of the Language Subtag Registry using HTTP 1.1 is to use a conditional GET [RFC2616].

7. Character Set Considerations

The syntax in this document requires that language tags use only the characters A-Z, a-z, 0-9, and HYPHEN-MINUS, which are present in most character sets, so the composition of language tags shouldn't have any character set issues.

The rendering of text based on the language tag is not addressed here. Historically, some processes have relied on the use of character set/encoding information (or other external information) in order to infer how a specific string of characters should be rendered. Notably, this applies to language- and culture-specific variations of Han ideographs as used in Japanese, Chinese, and Korean, where use of, for example, a Japanese character encoding such as EUC-JP implies that the text itself is in Japanese. When language tags are applied to spans of text, rendering engines might be able to use that information to better select fonts or make other rendering

choices, particularly where languages with distinct writing traditions use the same characters.

8. Changes from RFC 4646

The main goal for this revision of RFC 4646 was to incorporate two new parts of ISO 639 (ISO 639-3 and ISO 639-5) and their attendant sets of language codes into the IANA Language Subtag Registry. This permits the identification of many more languages and language collections than previously supported.

The specific changes in this document to meet these goals are:

- Defined the incorporation of ISO 639-3 and ISO 639-5 codes for use as primary and extended language subtags. It also permanently reserves and disallows the use of additional ’extlang’ subtags. The changes necessary to achieve this were:

  * Modified the ABNF comments.

  * Updated various registration and stability requirements sections to reference ISO 639-3 and ISO 639-5 in addition to ISO 639-1 and ISO 639-2.

  * Edited the text to eliminate references to extended language subtags where they are no longer used.

  * Explained the change in the section on extended language subtags.

- Changed the ABNF related to grandfathered tags. The irregular tags are now listed. Well-formed grandfathered tags are now described by the ’langtag’ production, and the ’grandfathered’ production was removed as a result. Also: added description of both types of grandfathered tags to Section 2.2.8.

- Added the paragraph on "collections" to Section 4.1.

- Changed the capitalization rules for ’Tag’ fields in Section 3.1.

- Split Section 3.1 up into subsections.

- Modified Section 3.5 to allow ’Suppress-Script’ fields to be added, modified, or removed via the registration process. This was an erratum from RFC 4646.

- Modified examples that used region code ’CS’ (formerly Serbia and Montenegro) to use ’RS’ (Serbia) instead.

- o Modified the rules for creating and maintaining record ’Description’ fields to prevent duplicates, including inverted duplicates.

- o Removed the lengthy description of why RFC 4646 was created from this section, which also caused the removal of the reference to XML Schema.

- o Modified the text in Section 2.1 to place more emphasis on the fact that language tags are not case sensitive.

- o Replaced the example "fr-Latn-CA" in Section 2.1 with "sr-Latn-RS" and "az-Arab-IR" because "fr-Latn-CA" doesn’t respect the ’Suppress-Script’ on ’Latn’ with ’fr’.

- o Changed the requirements for well-formedness to make singleton repetition checking optional (it is required for validity checking) in Section 2.2.9.

- o Changed the text in Section 2.2.9 referring to grandfathered checking to note that the list is now included in the ABNF.

- o Modified and added text to Section 3.2. The job description was placed first. A note was added making clear that the Language Subtag Reviewer may delegate various non-critical duties, including list moderation. Finally, additional text was added to make the appointment process clear and to clarify that decisions and performance of the reviewer are appealable.

- o Added text to Section 3.5 clarifying that the ietf-languages@iana.org list is operated by whomever the IESG appoints.

- o Added text to Section 3.1.5 clarifying that the first Description in a ’language’ record matches the corresponding Reference Name for the language in ISO 639-3.

- o Modified Section 2.2.9 to define classes of conformance related to specific tags (formerly ’well-formed’ and ’valid’ referred to implementations). Notes were added about the removal of ’extlang’ from the ABNF provided in RFC 4646, allowing for well-formedness using this older definition. Reference to RFC 3066 well-formedness was also added.

- o Added text to the end of Section 3.1.2 noting that future versions of this document might add new field types to the registry format and recommending that implementations ignore any unrecognized fields.

- Added text about what the lack of a ’Suppress-Script’ field means in a record to Section 3.1.9.

- Added text allowing the correction of misspellings and typographic errors to Section 3.1.5.

- Added text to Section 3.1.8 disallowing ’Prefix’ field conflicts (such as circular prefix references).

- Modified text in Section 3.5 to require the subtag reviewer to announce his/her decision (or extension) following the two-week period. Also clarified that any decision or failure to decide can be appealed.

- Modified text in Section 4.1 to include the (heretofore anecdotal) guiding principle of tag choice, and clarifying the non-use of script subtags in non-written applications.

- Prohibited multiple use of the same variant in a tag (i.e., "de-1901-1901"). Previously, this was only a recommendation ("SHOULD").

- Removed inappropriate [RFC2119] language from the illustration in Section 4.4.1.

- Replaced the example of deprecating "zh-guoyu" with "zh-hakka"->"hak" in Section 4.5, noting that it was this document that caused the change.

- Replaced the section in Section 4.1 dealing with "mul"/"und" to include the subtags ’zxx’ and ’mis’, as well as the tag "i-default". A normative reference to RFC 2277 was added.

- Added text to Section 3.5 clarifying that any modifications of a registration request must be sent to the <ietf-languages@iana.org> list before submission to IANA.

- Changed the ABNF for the record-jar format from using the LWSP production to use a folding whitespace production similar to obs-FWS in [RFC5234]. This effectively prevents unintentional blank lines inside a field.

- Clarified and revised text in Sections 3.3, 3.5, and 5.1 to clarify that the Language Subtag Reviewer sends the complete registration forms to IANA, that IANA extracts the record from the form, and that the forms must also be archived separately from the registry.

- o Added text to Section 5 requiring IANA to send an announcement to an ietf-languages-announcements list whenever the registry is updated.

- o Modification of the registry to use UTF-8 as its character encoding. This also entails additional instructions to IANA and the Language Subtag Reviewer in the registration process.

- o Modified the rules in Section 2.2.4 so that "exceptionally reserved" ISO 3166-1 codes other than ’UK’ were included into the registry. In particular, this allows the code ’EU’ (European Union) to be used to form language tags or (more commonly) for applications that use the registry for region codes to reference this subtag.

- o Modified the IANA considerations section (Section 5) to remove unnecessary normative [RFC2119] language.

9. References

9.1. Normative References

[ISO15924] International Organization for Standardization, "ISO 15924:2004. Information and documentation -- Codes for the representation of names of scripts", January 2004.

[ISO3166-1] International Organization for Standardization, "ISO 3166-1:2006. Codes for the representation of names of countries and their subdivisions -- Part 1: Country codes", November 2006.

[ISO639-1] International Organization for Standardization, "ISO 639-1:2002. Codes for the representation of names of languages -- Part 1: Alpha-2 code", July 2002.

[ISO639-2] International Organization for Standardization, "ISO 639-2:1998. Codes for the representation of names of languages -- Part 2: Alpha-3 code", October 1998.

[ISO639-3] International Organization for Standardization, "ISO 639-3:2007. Codes for the representation of names of languages - Part 3: Alpha-3 code for comprehensive coverage of languages", February 2007.

[ISO639-5] International Organization for Standardization, "ISO 639-5:2008. Codes for the representation of names of languages -- Part 5: Alpha-3 code for language families and groups", May 2008.

[ISO646] International Organization for Standardization, "ISO/IEC 646:1991, Information technology -- ISO 7-bit coded character set for information interchange.", 1991.

[RFC2026] Bradner, S., "The Internet Standards Process -- Revision 3", BCP 9, RFC 2026, October 1996.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2277] Alvestrand, H., "IETF Policy on Character Sets and Languages", BCP 18, RFC 2277, January 1998.

[RFC3339] Klyne, G., Ed. and C. Newman, "Date and Time on the Internet: Timestamps", RFC 3339, July 2002.

[RFC4647] Phillips, A. and M. Davis, "Matching of Language Tags", BCP 47, RFC 4647, September 2006.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

[RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.

[SpecialCasing] The Unicode Consoritum, "Unicode Character Database, Special Casing Properties", March 2008, <http://unicode.org/Public/UNIDATA/SpecialCasing.txt>.

[UAX14] Freitag, A., "Unicode Standard Annex #14: Line Breaking Properties", August 2006, <http://www.unicode.org/reports/tr14/>.

[UN_M.49] Statistics Division, United Nations, "Standard Country or Area Codes for Statistical Use", Revision 4 (United Nations publication, Sales No. 98.XVII.9, June 1999.

[Unicode] Unicode Consortium, "The Unicode Consortium. The Unicode Standard, Version 5.0, (Boston, MA, Addison-Wesley, 2003. ISBN 0-321-49081-0)", January 2007.

9.2. Informative References

[CLDR] "The Common Locale Data Repository Project", <http://cldr.unicode.org>.

[RFC1766] Alvestrand, H., "Tags for the Identification of Languages", RFC 1766, March 1995.

[RFC2028] Hovey, R. and S. Bradner, "The Organizations Involved in the IETF Standards Process", BCP 11, RFC 2028, October 1996.

[RFC2046] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", RFC 2046, November 1996.

[RFC2047] Moore, K., "MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text", RFC 2047, November 1996.

[RFC2231] Freed, N. and K. Moore, "MIME Parameter Value and Encoded Word Extensions: Character Sets, Languages, and Continuations", RFC 2231, November 1997.

[RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

[RFC2781] Hoffman, P. and F. Yergeau, "UTF-16, an encoding of ISO 10646", RFC 2781, February 2000.

[RFC3066] Alvestrand, H., "Tags for the Identification of Languages", RFC 3066, January 2001.

[RFC3282] Alvestrand, H., "Content Language Headers", RFC 3282, May 2002.

[RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, July 2003.

[RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, November 2003.

[RFC4645] Ewell, D., "Initial Language Subtag Registry", RFC 4645, September 2006.

[RFC4646] Phillips, A. and M. Davis, "Tags for Identifying Languages", BCP 47, RFC 4646, September 2006.

[RFC5645] Ewell, D., Ed., "Update to the Language Subtag Registry", September 2009.

[UTS35] Davis, M., "Unicode Technical Standard #35: Locale Data Markup Language (LDML)", December 2007, <http://www.unicode.org/reports/tr35/>.

[iso639.prin] ISO 639 Joint Advisory Committee, "ISO 639 Joint Advisory Committee: Working principles for ISO 639 maintenance", March 2000, <http://www.loc.gov/standards/iso639-2/iso639jac_n3r.html>.

[record-jar] Raymond, E., "The Art of Unix Programming", 2003, <urn:isbn:0-13-142901-9>.

Appendix A. Examples of Language Tags (Informative)

Simple language subtag:

de (German)

fr (French)

ja (Japanese)

i-enochian (example of a grandfathered tag)

Language subtag plus Script subtag:

zh-Hant (Chinese written using the Traditional Chinese script)

zh-Hans (Chinese written using the Simplified Chinese script)

sr-Cyrl (Serbian written using the Cyrillic script)

sr-Latn (Serbian written using the Latin script)

Extended language subtags and their primary language subtag counterparts:

zh-cmn-Hans-CN (Chinese, Mandarin, Simplified script, as used in China)

cmn-Hans-CN (Mandarin Chinese, Simplified script, as used in China)

zh-yue-HK (Chinese, Cantonese, as used in Hong Kong SAR)

yue-HK (Cantonese Chinese, as used in Hong Kong SAR)

Language-Script-Region:

zh-Hans-CN (Chinese written using the Simplified script as used in mainland China)

sr-Latn-RS (Serbian written using the Latin script as used in Serbia)

Language-Variant:

sl-rozaj (Resian dialect of Slovenian)

sl-rozaj-biske (San Giorgio dialect of Resian dialect of Slovenian)

sl-nedis (Nadiza dialect of Slovenian)

Language-Region-Variant:

de-CH-1901 (German as used in Switzerland using the 1901 variant [orthography])

sl-IT-nedis (Slovenian as used in Italy, Nadiza dialect)

Language-Script-Region-Variant:

hy-Latn-IT-arevela (Eastern Armenian written in Latin script, as used in Italy)

Language-Region:

de-DE (German for Germany)

en-US (English as used in the United States)

es-419 (Spanish appropriate for the Latin America and Caribbean region using the UN region code)

Private use subtags:

de-CH-x-phonebk

az-Arab-x-AZE-derbend

Private use registry values:

x-whatever (private use using the singleton 'x')

qaa-Qaaa-QM-x-southern (all private tags)

de-Qaaa (German, with a private script)

sr-Latn-QM (Serbian, Latin script, private region)

sr-Qaaa-RS (Serbian, private script, for Serbia)

Tags that use extensions (examples ONLY -- extensions MUST be defined by revision or update to this document, or by RFC):

en-US-u-islamcal

zh-CN-a-myext-x-private

en-a-myext-b-another

Some Invalid Tags:

de-419-DE (two region tags)

a-DE (use of a single-character subtag in primary position; note that there are a few grandfathered tags that start with "i-" that are valid)

ar-a-aaa-b-bbb-a-ccc (two extensions with same single-letter prefix)

Appendix B. Examples of Registration Forms

LANGUAGE SUBTAG REGISTRATION FORM

1. Name of requester: Han Steenwijk
2. E-mail address of requester: han.steenwijk @ unipd.it
3. Record Requested:

```
Type:        variant
Subtag:      biske
Description: The San Giorgio dialect of Resian
Description: The Bila dialect of Resian
Prefix:      sl-rozaj
Comments:    The dialect of San Giorgio/Bila is one of the
   four major local dialects of Resian
```

4. Intended meaning of the subtag:

The local variety of Resian as spoken in San Giorgio/Bila

5. Reference to published description of the language (book or article):

-- Jan I.N. Baudouin de Courtenay - Opyt fonetiki rez’janskich govorov, Varsava - Peterburg: Vende - Kozancikov, 1875.

LANGUAGE SUBTAG REGISTRATION FORM

1. Name of requester: Jaska Zedlik
2. E-mail address of requester: jz53 @ zedlik.com
3. Record Requested:

Type: variant
Subtag: tarask
Description: Belarusian in Taraskievica orthography
Prefix: be
Comments: The subtag represents Branislau Taraskievic's Belarusian orthography as published in "Bielaruski klasycny pravapis" by Juras Buslakou, Vincuk Viacorka, Zmicier Sanko, and Zmicier Sauka (Vilnia-Miensk 2005).

4. Intended meaning of the subtag:

The subtag is intended to represent the Belarusian orthography as published in "Bielaruski klasycny pravapis" by Juras Buslakou, Vincuk Viacorka, Zmicier Sanko, and Zmicier Sauka (Vilnia-Miensk 2005).

5. Reference to published description of the language (book or article):

Taraskievic, Branislau. Bielaruskaja gramatyka dla skol. Vilnia: Vyd. "Bielaruskaha kamitetu", 1929, 5th edition.

Buslakou, Juras; Viacorka, Vincuk; Sanko, Zmicier; Sauka, Zmicier. Bielaruski klasycny pravapis. Vilnia-Miensk, 2005.

6. Any other relevant information:

Belarusian in Taraskievica orthography became widely used, especially in Belarusian-speaking Internet segment, but besides this some books and newspapers are also printed using this orthography of Belarusian.

Appendix C. Acknowledgements

Any list of contributors is bound to be incomplete; please regard the following as only a selection from the group of people who have contributed to make this document what it is today.

The contributors to RFC 4646, RFC 4647, RFC 3066, and RFC 1766, the precursors of this document, made enormous contributions directly or indirectly to this document and are generally responsible for the success of language tags.

The following people contributed to this document:

Stephane Bortzmeyer, Karen Broome, Peter Constable, John Cowan, Martin Duerst, Frank Ellerman, Doug Ewell, Deborah Garside, Marion Gunn, Alfred Hoenes, Kent Karlsson, Chris Newman, Randy Presuhn, Stephen Silver, Shawn Steele, and many, many others.

Very special thanks must go to Harald Tveit Alvestrand, who originated RFCs 1766 and 3066, and without whom this document would not have been possible.

Special thanks go to Michael Everson, who served as the Language Tag Reviewer for almost the entire RFC 1766/RFC 3066 period, as well as the Language Subtag Reviewer since the adoption of RFC 4646.

Special thanks also go to Doug Ewell, for his production of the first complete subtag registry, his work to support and maintain new registrations, and his careful editorship of both RFC 4645 and [RFC5645].

Authors' Addresses

Addison Phillips (editor)
Lab126

EMail: addison@inter-locale.com
URI: http://www.inter-locale.com

Mark Davis (editor)
Google

EMail: markdavis@google.com