Network Working Group
Request for Comments: 4646
BCP: 47
Obsoletes: 3066
Category: Best Current Practice

A. Phillips, Ed.
Yahoo! Inc.
M. Davis, Ed.
Google

September 2006

Tags for Identifying Languages

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

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. This document, in combination with RFC 4647, replaces RFC 3066, which replaced RFC 1766.

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.

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 preferences can also be used to select among tools (such as dictionaries) to assist in the processing or understanding of content in different languages.

In addition, 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 be used to specify user preferences when selecting information content, or for labeling additional attributes of content and associated resources.

Tags can also be 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 extension.

This document, in combination with [RFC4647], replaces [RFC3066], which replaced [RFC1766]. For a list of changes in this document, see Section 8.

The keywords "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

The language tag is composed of one or more parts, known as "subtags". Each subtag consists of a sequence of alphanumeric characters. Subtags are distinguished and separated from one another by a hyphen ("-", ABNF [RFC4234] %x2D). A language tag consists of a "primary language" subtag and a (possibly empty) series of subsequent subtags, each of which refines or narrows the range of languages identified by the overall tag.

Usually, each type of subtag is distinguished by length, position in the tag, and content: subtags can be recognized solely by these features. The only exception to this is a fixed list of grandfathered tags registered under RFC 3066 [RFC3066]. This makes it possible to construct a parser that can extract and assign some semantic information to the subtags, even if the specific subtag values are not recognized. Thus, a parser need not have an up-to-date copy (or any copy at all) of the subtag registry to perform most searching and matching operations.

The syntax of the language tag in ABNF [RFC4234] is:

```
Language-Tag  = langtag
              / privateuse            ; private use tag
              / grandfathered         ; grandfathered registrations

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

language      = (2*3ALPHA [ extlang ]) ; shortest ISO 639 code
              / 4ALPHA                 ; reserved for future use
              / 5*8ALPHA               ; registered language subtag

extlang       = *3("-" 3ALPHA)         ; reserved for future use

script        = 4ALPHA                 ; ISO 15924 code

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

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

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

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

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

grandfathered = 1*3ALPHA 1*2("-" (2*8alphanum))
                ; grandfathered registration
                ; Note: i is the only singleton
                ; that starts a grandfathered tag

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

Figure 1: Language Tag ABNF

Note: There is a subtlety in the ABNF for ’variant’: variants starting with a digit MAY be four characters long, while those starting with a letter MUST be at least five characters long.

All subtags have a maximum length of eight characters and whitespace is not permitted in a language tag. For examples of language tags, see Appendix B.

Note that although [RFC4234] 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 US-ASCII repertoire. An example of this would be an XML document that uses the UTF-16LE [RFC2781] encoding of [Unicode].

The 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.

For example:

- [ISO639-1] recommends that language codes be written in lowercase ('mn' Mongolian).
- [ISO3166-1] recommends that country codes be capitalized ('MN' Mongolia).
- [ISO15924] recommends that script codes use lowercase with the initial letter capitalized ('Cyrl' Cyrillic).

However, in the tags defined by this document, the uppercase US-ASCII letters in the range 'A' through 'Z' are considered equivalent and mapped directly to their US-ASCII lowercase equivalents in the range 'a' through 'z'. 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.

Although case distinctions do not carry meaning in language tags, consistent formatting and presentation of the tags will aid users. The format of the tags and subtags in the registry is RECOMMENDED. In this format, all non-initial two-letter subtags are uppercase, all non-initial four-letter subtags are titlecase, and all other subtags are lowercase.

2.2. Language Subtag Sources and Interpretation

The namespace of language tags and their subtags is administered by the Internet Assigned Numbers Authority (IANA) [RFC2860] 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 in this section:

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

- Subtag refers to a specific section of a tag, delimited by hyphen, such as the subtag 'Latn' in "fr-Latn-CA". Examples of subtags in this document are enclosed in single quotes ('Latn').

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

The definitions in this section apply to the various subtags within the language tags defined by this document, excepting those "grandfathered" tags defined in Section 2.2.8.

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.

Subtags in the IANA registry that do not come from an underlying standard can only appear in specific positions in a tag. Specifically, they can only occur as primary language subtags or as variant subtags.

Note that 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.

Single-letter and single-digit subtags are reserved for current or future use. These include the following current uses:

- The single-letter subtag ’x’ is reserved to introduce a sequence of private use subtags. The interpretation of any private use subtags 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.

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

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

2.2.1. Primary Language Subtag

The primary language subtag is the first subtag in a language tag (with the exception of private use and certain grandfathered tags) and cannot be omitted. The following rules apply to the primary language subtag:

1. All two-character language subtags were defined in the IANA registry according to the assignments found in the standard ISO 639 Part 1, "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 Part 1 maintenance agency or governing standardization bodies.

2. All three-character language subtags were defined in the IANA registry according to the assignments found in ISO 639 Part 2, "ISO 639-2:1998 - Codes for the representation of names of languages -- Part 2: Alpha-3 code - edition 1" [ISO639-2], or assignments subsequently made by the ISO 639 Part 2 maintenance agency or governing standardization bodies.

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 4.5 for more information on private use subtags.

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

5. All language subtags of 5 to 8 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. At the time

this document was created, there were no examples of this kind of subtag and future registrations of this type will be discouraged: primary languages are strongly RECOMMENDED for registration with ISO 639, and proposals rejected by ISO 639/RA will be closely scrutinized before they are registered with IANA.

6. 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' SHOULD NOT be taken to 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 4.5.

7. 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.)

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

Note: For languages that have both an ISO 639-1 two-character code and an ISO 639-2 three-character code, only the ISO 639-1 two-character code is defined in the IANA registry.

Note: For languages that have no ISO 639-1 two-character code and for which the ISO 639-2/T (Terminology) code and the ISO 639-2/B (Bibliographic) codes 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 code were also assigned a two-character code; it is not expected that future assignments of this nature will occur.

Note: To avoid problems with versioning and subtag choice as experienced during the transition between RFC 1766 and RFC 3066, as well as 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."

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 ISO 639-2, 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 be invalidated if ISO 639-1 were to assign a two-character code to the Hawaiian language at a later date.

For example, one of the grandfathered IANA registrations is "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.

2.2.2. Extended Language Subtags

The following rules apply to the extended language subtags:

1. Three-letter subtags immediately following the primary subtag are reserved for future standardization, anticipating work that is currently under way on ISO 639.

2. Extended language subtags MUST follow the primary subtag and precede any other subtags.

3. There MAY be up to three extended language subtags.

4. Extended language subtags MUST NOT be registered or used to form language tags. Their syntax is described here so that implementations can be compatible with any future revision of this document that does provide for their registration.

Extended language subtag records, once they appear in the registry, MUST include exactly one 'Prefix' field indicating an appropriate language subtag or sequence of subtags that MUST always appear as a prefix to the extended language subtag.

Example: In a future revision or update of this document, the tag "zh-gan" (registered under RFC 3066) might become a valid non-grandfathered (that is, redundant) tag in which the subtag 'gan' might represent the Chinese dialect 'Gan'.

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. All four-character subtags were defined according to [ISO15924]--"Codes for the representation of names of scripts": alpha-4 script codes, or subsequently assigned by the ISO 15924 maintenance agency or governing standardization bodies, denoting the script or writing system used in conjunction with this language.

2. Script subtags MUST immediately follow the primary language subtag and all extended language subtags and MUST occur before any other type of subtag described below.

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 4.5 for more information on private use subtags.

4. Script subtags MUST NOT be registered using the process in Section 3.5 of this document. Variant subtags MAY be considered for registration for that purpose.

5. 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 language subtag’s record includes a Suppress-Script field listing the applicable script subtag.

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 regional dialects or usage, or region-specific spelling conventions. A region subtag 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 language, extended language, or script subtags and MUST precede all other subtags.

2. All two-character subtags following the primary subtag were defined in the IANA registry 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 maintenance agency or governing standardization bodies.

3. All three-character subtags consisting of digit (numeric) characters following the primary subtag were defined in the IANA registry according to the assignments found in UN Standard Country or Area Codes for Statistical Use [UN_M.49] or assignments subsequently made by the governing standards body. Note that 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 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. UN numeric codes for countries or areas with ambiguous ISO 3166 alpha-2 codes, when entered into the registry, MUST be defined according to the rules in Section 3.4 and MUST be used to form language tags that represent the country or region for which they are defined.

   D. UN numeric codes for countries or areas for which there is an associated ISO 3166 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. UN numeric codes and ISO 3166 alpha-2 codes for countries or areas listed as eligible for registration in [RFC4645] but not presently registered MAY be entered into the IANA registry via the process described in Section 3.5. Once registered, these codes MAY be used to form language tags.

F. All other UN numeric codes for countries or areas that do not have an associated ISO 3166 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.

4. Note: 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 alpha-2 codes.)

5. 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.

6. 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 4.5 for more information on private use subtags.

"de-CH" represents German ('de') as used in Switzerland ('CH').

"sr-Latn-CS" represents Serbian ('sr') written using Latin script ('Latn') as used in Serbia and Montenegro ('CS').

"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 are not associated with any external standard. Variant subtags and their meanings are defined by the registration process defined in Section 3.5.

2. Variant subtags MUST follow all of the other defined subtags, but precede any extension or private use subtag sequences.

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.

Variant subtag records in the language subtag registry MAY include one or more 'Prefix' fields, which indicate the language tag or tags that would make a suitable prefix (with other subtags, as appropriate) in forming a language tag with the variant. For example, the subtag 'nedis' has a Prefix of "sl", making it suitable to form language tags such as "sl-nedis" and "sl-IT-nedis", but not suitable for use in a tag such as "zh-nedis" or "it-IT-nedis".

"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.

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".

2.2.6. Extension Subtags

Extensions provide a mechanism for extending language tags for use in various applications. See Section 3.7. The following rules apply to extensions:

1. Extension subtags are separated from the other subtags defined in this document by a single-character subtag ("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.

2. Note: Private use subtag sequences starting with the singleton subtag ’x’ are described in Section 2.2.7 below.

3. 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.

4. 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.

5. Extension subtags MUST meet all of the requirements for the content and format of subtags defined in this document.

6. 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.

7. 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 ’-’.

8. 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’.

9. Extension subtags MUST follow all language, extended language, script, region, and variant subtags in a tag.

10. 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’.

11. In the event that more than one extension appears in a single tag, the tag SHOULD be canonicalized as described in Section 4.4.

For example, if the prefix singleton 'r' and the shown subtags were defined, 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 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.

3. Private use subtags MUST follow all 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 4.5 for more information on private use subtag choice.

For example: Users who wished to utilize codes from the Ethnologue publication of SIL International for language identification might agree to exchange tags such as "az-Arab-x-AZE-derbend". This example contains two private use subtags. The first is 'AZE' and the second is 'derbend'.

2.2.8. Preexisting RFC 3066 Registrations

Existing IANA-registered language tags from RFC 1766 and/or RFC 3066 maintain their validity. These tags will be maintained in the registry in records of either the "grandfathered" or "redundant" type. Grandfathered tags contain one or more subtags that are not defined in the Language Subtag Registry (see Section 3). Redundant tags consist entirely of subtags defined above and whose independent registration is superseded by this document. For more information, see Section 3.8.

It is important to note that all language tags formed under the guidelines in this document were either legal, well-formed tags or could have been registered under RFC 3066.

2.2.9. Classes of Conformance

Implementations sometimes need to describe their capabilities with regard to the rules and practices described in this document. There are two classes of conforming implementations described by this document: "well-formed" processors and "validating" processors. Claims of conformance SHOULD explicitly reference one of these definitions.

An implementation that claims to check for well-formed language tags MUST:

- Check that the tag and all of its subtags, including extension and private use subtags, conform to the ABNF or that the tag is on the list of grandfathered tags.

- Check that singleton subtags that identify extensions do not repeat. For example, the tag "en-a-xx-b-yy-a-zz" is not well-formed.

Well-formed processors are strongly encouraged to implement the canonicalization rules contained in Section 4.4.

An implementation that claims to be validating MUST:

- Check that the tag is well-formed.

- Specify the particular registry date for which the implementation performs validation of subtags.

- Check that either the tag is a grandfathered tag, or that all language, script, region, and variant subtags consist of valid codes for use in language tags according to the IANA registry as of the particular date specified by the implementation.

- Specify which, if any, extension RFCs as defined in Section 3.7 are supported, including version, revision, and date.

- For any such extensions supported, check that all subtags used in that extension are valid.

- For variant and extended language subtags, if the registry contains one or more ’Prefix’ fields for that subtag, check that the tag matches at least one prefix. The tag matches if all the

subtags in the 'Prefix' also appear in the tag. For example, the prefix "es-CO" matches the tag "es-Latn-CO-x-private" because both the 'es' language subtag and 'CO' region subtag appear in the tag.

3. Registry Format and Maintenance

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

The Language Subtag 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 Language Subtag 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 IANA registry.)

3.1. Format of the IANA Language Subtag Registry

The IANA Language Subtag Registry ("the registry") consists of a text file that is machine readable 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 will be maintained as part of the obsolete RFC 3066 registry. The remaining set of initial subtags will not have registration forms created for them.

The registry is in the text format described below. This format was based on the record-jar format described in [record-jar].

Each line of text is limited to 72 characters, including all whitespace. Records are separated by lines containing only the sequence "%%" (%x25.25).

Each field can be viewed as a single, logical line of ASCII characters, comprising a field-name and a field-body separated by a COLON character (%x3A). For convenience, the field-body portion of this conceptual entity can be split into a multiple-line representation; this is called "folding". The format of the registry is described by the following ABNF (per [RFC4234]):

```
registry   = record *("%%" CRLF record)
record     = 1*( field-name *SP ":" *SP field-body CRLF )
field-name = (ALPHA / DIGIT) [*(ALPHA / DIGIT / "-") (ALPHA / DIGIT)]
field-body = *(ASCCHAR/LWSP)
ASCCHAR    = %x21-25 / %x27-7E / UNICHAR ; Note: AMPERSAND is %x26
UNICHAR    = "&#x" 2*6HEXDIG ";"
```

Figure 2: Registry Format ABNF

The sequence '..' (%x2E.2E) 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'.

Characters from outside the US-ASCII [ISO646] repertoire, as well as the AMPERSAND character ("&", %x26) when it occurs in a field-body, are represented by a "Numeric Character Reference" using hexadecimal notation in the style used by [XML10] (see <http://www.w3.org/TR/REC-xml/#dt-charref>). This consists of the sequence "&#x" (%x26.23.78) followed by a hexadecimal representation of the character's code point in [ISO10646] followed by a closing semicolon (%x3B). For example, the EURO SIGN, U+20AC, would be represented by the sequence "€". Note that the hexadecimal notation MAY have between two and six digits.

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.

The first record in the file contains the single field whose field-name is "File-Date" (see Figure 3). The field-body of this record contains the last modification date of this copy of the registry, making it possible to compare different versions of the registry. The registry on the IANA website is the most current. Versions with an older date than that one are not up-to-date.

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

Figure 3: Example of the File-Date Record

Subsequent records represent subtags in the registry. Each of the fields in each record MUST occur no more than once, unless otherwise noted below. Each record MUST contain the following fields:

- 'Type'
  - Type's field-value MUST consist of one of the following strings: "language", "extlang", "script", "region", "variant", "grandfathered", and "redundant" and denotes the type of tag or subtag.
- Either 'Subtag' or 'Tag'
  - Subtag's field-value contains the subtag being defined. This field MUST only appear in records of whose 'Type' has one of these values: "language", "extlang", "script", "region", or "variant".
  - Tag's field-value contains a complete language tag. This field MUST only appear in records whose 'Type' has one of these values: "grandfathered" or "redundant". Note that the field-value will always follow the 'grandfathered' production in the ABNF in Section 2.1
- Description
  - Description's field-value contains a non-normative description of the subtag or tag.
- Added
  - Added's field-value contains the date the record was added to the registry.

The 'Subtag' or 'Tag' field MUST use lowercase letters to form the subtag or tag, 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, subtags defined by ISO 3166) MUST use uppercase. These exceptions mirror the use of case in the underlying standards.

The field 'Description' MAY appear more than one time and contains a description of the tag or subtag in the record. At least one of the 'Description' fields MUST be written or transcribed into the Latin script; the same or additional fields MAY also include a description in a non-Latin script. The 'Description' field is used for identification purposes and SHOULD NOT be taken to represent the actual native name of the language or variation or to be in any particular language. Most descriptions are taken directly from source standards such as ISO 639 or ISO 3166.

Note: Descriptions in registry entries that correspond to ISO 639, ISO 15924, ISO 3166, 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. The description does not replace the content of the source standard itself. The descriptions are not intended to be the English localized names for the subtags. Localization or translation of language tag and subtag descriptions is out of scope of this document.

Each record MAY also contain the following fields:

- Preferred-Value
  - For fields of type 'language', 'extlang', 'script', 'region', and 'variant', 'Preferred-Value' contains the subtag of the same 'Type' that is preferred for forming the language tag.
  - For fields of type 'grandfathered' and 'redundant', a canonical mapping to a complete language tag.
- Deprecated
  - Deprecated's field-value contains the date the record was deprecated.
- Prefix
  - Prefix's field-value contains a language tag with which this subtag MAY be used to form a new language tag, perhaps with other subtags as well. This field MUST only appear in records whose 'Type' field-value is 'variant' or 'extlang'. For example, the 'Prefix' for the variant 'nedis' is 'sl', meaning that the tags "sl-nedis" and "sl-IT-nedis" might be appropriate while the tag "is-nedis" is not.
- Comments
  - Comments contains additional information about the subtag, as deemed appropriate for understanding the registry and implementing language tags using the subtag or tag.
- Suppress-Script
  - Suppress-Script contains a script subtag that SHOULD NOT be used to form language tags with the associated primary language subtag. This field MUST only appear in records whose 'Type' field-value is 'language'. See Section 4.1.

The field 'Deprecated' MAY be added to 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. 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. 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.

The field 'Preferred-Value' contains a mapping between the record in which it appears and another tag or subtag. The value in this field is STRONGLY RECOMMENDED as the best choice to represent the value of this record when selecting a language tag. These values form three groups:

1. ISO 639 language codes that were later withdrawn in favor of other codes. These values are mostly a historical curiosity.

2. ISO 3166 region codes that have been withdrawn in favor of a new code. This sometimes happens when a country changes its name or administration in such a way that warrants a new region code.

3. Tags grandfathered from RFC 3066. In many cases, these tags have become obsolete because the values they represent were later encoded by ISO 639.

Records that contain a 'Preferred-Value' field MUST also have a 'Deprecated' field. This field contains a date of deprecation. Thus, a language tag processor can use the registry to construct the valid, non-deprecated set of subtags for a given date. In addition, for any given tag, a processor can construct the set of valid language tags that correspond to that tag for all dates up to the date of the registry. The ability to do these mappings MAY be beneficial to applications that are matching, selecting, for filtering content based on its language tags.

Note that '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.

In particular, the 'Preferred-Value' field does not imply retagging content that uses the affected subtag.

The field 'Preferred-Value' MUST NOT be modified once created in the registry. The field MAY be added to records of type "grandfathered" and "region" according to the rules in Section 3.3. Otherwise the field MUST NOT be added to any record already in the registry.

The 'Preferred-Value' field in records of type "grandfathered" and "redundant" contains whole language tags that are strongly RECOMMENDED for use in place of the record's value. In many cases, the mappings were created by deprecation of the tags during the period before this document was adopted. For example, the tag "no-nyn" was deprecated in favor of the ISO 639-1-defined language code 'nn'.

Records of type 'variant' MAY have more than one field of type 'Prefix'. Additional fields of this type MAY be added to a 'variant' record via the registration process.

Records of type 'extlang' MUST have _exactly_ one 'Prefix' field.

The field-value of the 'Prefix' field consists of a language tag whose subtags are appropriate to use with this subtag. For example, the variant subtag '1996' has a 'Prefix' field of "de". This means that tags starting with the sequence "de-" are appropriate with this subtag, so "de-Latg-1996" and "de-CH-1996" are both acceptable, while the tag "fr-1996" is an inappropriate choice.

The field of type 'Prefix' MUST NOT be removed from any record. The field-value for this type of field MUST NOT be modified.

The field 'Comments' MAY appear more than once per record. 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 field 'Suppress-Script' MUST only appear in records whose 'Type' field-value is 'language'. 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 and that therefore adds no distinguishing information to a language tag. It 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".

3.2. Language Subtag Reviewer

The Language Subtag Reviewer is appointed by the IESG for an indefinite term, subject to removal or replacement at the IESG's discretion. The Language Subtag Reviewer moderates the ietf-languages 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 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 decision 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, determine whether it conflicts with existing registry entries, and submit the information to IANA for inclusion in the registry. If a change takes place and the Language Subtag Reviewer does not do this in a timely manner, then any interested party MAY use the procedure in Section 3.5 to register the appropriate update.

Note: The redundant and grandfathered entries together are the complete list of tags registered under [RFC3066]. The redundant tags are those that can now be formed using the subtags defined in the registry together with the rules of Section 2.2. The grandfathered entries include those that can never be legal under those same provisions.

The set of redundant and grandfathered tags is permanent and stable: new entries in this section MUST NOT be added and existing entries MUST NOT be removed. Records of type 'grandfathered' MAY have their type converted to 'redundant'; see item 12 in Section 3.6 for more information. The decision-making process about which tags were initially grandfathered and which were made redundant is described in [RFC4645].

RFC 3066 tags that were deprecated prior to the adoption of this document are part of the list of grandfathered tags, and their component subtags were not included as registered variants (although they remain eligible for registration). For example, the tag "art-lojban" was deprecated in favor of the language subtag 'jbo'.

The Language Subtag Reviewer MUST ensure that new subtags meet the requirements in Section 4.1 or submit an appropriate alternate subtag as described in that section. When either a change or addition to the registry is needed, the Language Subtag Reviewer MUST prepare the complete record, including all fields, and forward it to IANA for insertion into the registry. Each record being modified or inserted MUST be forwarded in a separate message.

If a record represents a new subtag that does not currently exist in the registry, then the message's subject line MUST include the word "INSERT". If the record represents a change to an existing subtag, then the subject line of the message MUST include the word "MODIFY". The message MUST contain both the record for the subtag being inserted or modified and the new File-Date record. Here is an example of what the body of the message might contain:

```
LANGUAGE SUBTAG MODIFICATION
File-Date: 2005-01-02
%%
Type: variant
Subtag: nedis
Description: Natisone dialect
Description: Nadiza dialect
Added: 2003-10-09
Prefix: sl
Comments: This is a comment shown
  as an example.
%%
```

Figure 4: Example of a Language Subtag Modification Form

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 in the [RFC3339] "full-date" format.

Before forwarding a new registration to IANA, the Language Subtag Reviewer MUST ensure that values in the 'Subtag' field match case according to the description in Section 3.1.

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', 'Added', 'Deprecated' and 'Preferred-Value' MUST NOT be changed and are guaranteed to be stable over time.

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

3. Values in the field 'Prefix' MAY be added to records of type 'variant' via the registration process.

4. Values in the field 'Prefix' MAY 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 prefixes were needed, a new Prefix SHOULD be registered.

5. Values in the field 'Prefix' MUST NOT be removed.

6. 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.

7. The field 'Suppress-Script' MAY be added or removed via the registration process.

8. Codes assigned by ISO 639, ISO 15924, and ISO 3166 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.

9. Codes assigned by ISO 639, ISO 15924, or ISO 3166 that are withdrawn by their respective maintenance or registration authority remain valid in language tags. A 'Deprecated' field containing the date of withdrawal is 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.

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 a 'Preferred-Value' of 'TL' and its field 'Deprecated' contains the date the new code was assigned ('2004-07-06').

10. Codes assigned by ISO 639, ISO 15924, or ISO 3166 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 (i.e., ISO 639, ISO 15924, ISO 3166, 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: ISO 639 maintenance agency/registration authority (MA/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 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 code.

E. For ISO 3166 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 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 ninety 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.

11. 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 code SHOULD NOT be registered, except as a surrogate for an ISO 3166 code that is blocked from registration by an existing subtag. If such a code becomes necessary, then the registration authority for ISO 3166 SHOULD first be petitioned to assign a code to the region. If the petition for a code assignment by ISO 3166 is refused or not acted on in a timely manner, the registration process described in Section 3.5 MAY then be used to register the corresponding UN M.49 code. At the time this document was written, there were only four such codes: 830 (Channel Islands), 831 (Guernsey), 832 (Jersey), and 833 (Isle of Man). This way, UN M.49 codes remain available as the value of last resort in cases where ISO 3166 reassigns a deprecated value in the registry.

12. Stability provisions apply to grandfathered tags with this exception: should all of the subtags in a grandfathered tag become valid subtags in the IANA registry, then the field 'Type' in that record is changed from 'grandfathered' to 'redundant'. Note that this will not affect language tags that match the grandfathered tag, since these tags will now match valid generative subtag sequences. For example, if the subtag 'gan' in the language tag "zh-gan" were to be registered as an extended language subtag, then the grandfathered tag "zh-gan" would be deprecated (but existing content or implementations that use "zh-gan" would remain valid).

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.

Only subtags of type 'language' and 'variant' will be considered for independent registration of new subtags. Handling of 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 are described in Section 3.3. Stability provisions are described in Section 3.4.

This procedure MAY also be used to register or alter the information for the 'Description', 'Comments', 'Deprecated', or 'Prefix' 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 SHOULD follow the requirements in Section 3.1.

```
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:
   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

The subtag registration form MUST be sent to <ietf-languages@iana.org> for a two-week review period before it can be submitted to IANA. (This is an open list and can be joined by sending a request to <ietf-languages-request@iana.org>.)

Variant subtags are usually registered for use with a particular range of language tags. For example, the subtag 'rozaj' is intended for use with language tags that start with the primary language subtag "sl", 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.

Extended language subtags are reserved for future standardization. These subtags will be REQUIRED to include exactly one 'Prefix' field once they are allowed for registration.

The 'Prefix' field for a given registered subtag exists in the IANA registry as a guide to usage. Additional prefixes 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 to a variant subtag that imply a different semantic meaning will probably be rejected. For example, a request to add the prefix "de" to the subtag 'nedis' so that the tag

"de-nedis" represented some German dialect would be rejected. The ’nedis’ subtag represents a particular Slovenian dialect and the additional registration would change the semantic meaning assigned to the subtag. A separate subtag SHOULD be proposed instead.

The ’Description’ field MUST contain a description of the tag being registered written or transcribed into the Latin script; it MAY also include a description in a non-Latin script. Non-ASCII characters MUST be escaped using the syntax described in Section 3.1. The ’Description’ field is used for identification purposes and doesn’t necessarily represent the actual native name of the language or variation or to be in any particular language.

While the ’Description’ field itself is not guaranteed to be stable and errata corrections MAY be undertaken from time to time, attempts to provide translations or transcriptions of entries in the registry itself will probably be frowned upon by the community or rejected outright, as changes of this nature have an impact on the provisions in Section 3.4.

When the two-week period has passed, the Language Subtag Reviewer either forwards the record to be inserted or modified to iana@iana.org according to the procedure described in Section 3.3, or rejects the request because of significant objections raised on the list or due to problems with constraints in this document (which MUST be explicitly cited). The Language Subtag Reviewer MAY also extend the review period in two-week increments to permit further discussion. The Language Subtag Reviewer MUST indicate on the list whether the registration has been accepted, rejected, or extended following each two-week period.

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.

The applicant is free to modify a rejected application with additional information and submit it again; this restarts the two-week comment period.

Decisions made by the Language Subtag Reviewer MAY be appealed to the IESG [RFC2028] under the same rules as other IETF decisions [RFC2026].

All approved registration forms are available online in the directory http://www.iana.org/numbers.html under "languages".

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 "Description" in the registration form is to aid people trying to verify whether a language is registered or what language or language variation a particular subtag refers to. 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 codes that exist in ISO 639-1 or ISO 639-2, that are under consideration by the ISO 639 maintenance or 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 in the IANA registry (to the extent that it is very unlikely that any subtags will be registered of this type).

- 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).

- The addition or maintenance of fields (generally of an informational nature) in Tag or Subtag records as described in Section 3.1 and subject to the stability provisions in Section 3.4. This includes descriptions, comments, deprecation and preferred values for obsolete or withdrawn codes, or the addition of script or extlang information to primary language subtags.

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

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 maintenance agency 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 maintenance agency 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, D.C. 20540 USA
Phone: +1 202 707 6237 Fax: +1 202 707 0115
URL: http://www.loc.gov/standards/iso639-2

The maintenance agency for ISO 3166 (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
E-mail: statistics@un.org
URL: http://unstats.un.org/unsd/methods/m49/m49alpha.htm

3.7. Extensions and 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 Consensus" policy defined by [RFC2434]. 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. 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 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. Initialization of the Registries

Upon adoption of this document, an initial version of the Language Subtag Registry containing the various subtags initially valid in a language tag is necessary. This collection of subtags, along with a description of the process used to create it, is described by [RFC4645]. IANA SHALL publish the initial version of the registry described by this document from the content of [RFC4645]. Once published by IANA, the maintenance procedures, rules, and registration processes described in this document will be available for new registrations or updates.

Registrations that are in process under the rules defined in [RFC3066] when this document is adopted MAY be completed under the former rules, at the discretion of the Language Tag Reviewer (as described in [RFC3066]). Until the IESG officially appoints a Language Subtag Reviewer, the existing Language Tag Reviewer SHALL serve as the Language Subtag Reviewer.

Any new registrations submitted using the RFC 3066 forms or format after the adoption of this document and publication of the registry by IANA MUST be rejected.

An initial version of the Language Tag Extensions Registry described in Section 3.7 is also needed. The Language Tag Extensions Registry SHALL be initialized with a single record containing a single field of type "File-Date" as a placeholder for future assignments.

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

One is sometimes faced with the choice between several possible tags for the same body of text.

Interoperability is best served when all users use the same language tag in order 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.

Subtags SHOULD only be used where they add useful distinguishing information; 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 (and SHOULD NOT) be used in a language tag.

Of particular note, many 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 were not formally defined in RFC 3066 and their use can affect matching and subtag identification by implementations of RFC 3066, as these subtags appear between the primary language and region subtags. For example, if a user requests content in an implementation of Section 2.5 of [RFC3066] using the language range "en-US", content labeled "en-Latn-US" will not match the request. Therefore, it is important to know when script subtags will customarily be used and when they ought not be used. In the registry, the Suppress-Script 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 by defining when users SHOULD NOT include a script subtag with a particular primary language subtag.

Extended language subtags (type 'extlang' in the registry; see Section 3.1) also appear between the primary language and region subtags and are reserved for future standardization. Applications might benefit from their judicious use in forming language tags in the future. Similar recommendations are expected to apply to their use as apply to script subtags.

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

The choice of subtags used to form a language tag SHOULD be guided by the following rules:

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.

2. The script subtag SHOULD NOT be used to form language tags unless the script adds some distinguishing information to the tag. The field 'Suppress-Script' in the primary language record in the registry indicates which script subtags do not add distinguishing information for most applications.

   * 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.

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 'he' for Hebrew in preference to 'iw'.

4. The 'und' (Undetermined) primary language subtag SHOULD NOT be used to label content, even if the language is unknown. Omitting the language tag altogether is preferred to using a tag with a primary language subtag of 'und'. The 'und' subtag MAY be useful for protocols that require a language tag to be provided. The 'und' subtag MAY also be useful when matching language tags in certain situations.

5. The 'mul' (Multiple) primary language subtag SHOULD NOT be used whenever the protocol allows the separate tags for multiple languages, as is the case for the Content-Language header in HTTP. The 'mul' subtag conveys little useful information: content in multiple languages SHOULD individually tag the languages where they appear or otherwise indicate the actual language in preference to the 'mul' subtag.

6. The same variant subtag SHOULD NOT be used more than once within a language tag.

   * For example, do not use "de-DE-1901-1901".

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 language tag created under the rules in this document will become obsolete.

4.2. Meaning of the Language Tag

The relationship between the tag and the information it relates to is defined by the context in which the tag appears. Accordingly, this section gives only possible examples of its usage.

- 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 multi-lingual to get complete

understanding of the document. Example: MIME multipart/alternative.

- In 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 Norwegian document; the Norwegian-speaking user could then access a French-Norwegian 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 Norwegian rules.

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 text 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.

4.3. Length Considerations

[RFC3066] did not provide an upper limit on the size of language tags. While RFC 3066 did define the semantics of particular subtags in such a way that most language tags consisted of language and region subtags with a combined total length of up to six characters, larger registered tags were not only possible but were actually registered.

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.

4.3.1. 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 for causing 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 up to 33 characters.

Protocols or specifications that specify limited buffer sizes for language tags SHOULD allow for language tags of at least 42 characters.

The following illustration shows how the 42-character recommendation was derived. The combination of language and extended language subtags was chosen for future compatibility. At up to 15 characters, this combination is longer than the longest possible primary language subtag (8 characters):

```
language      =  3 (ISO 639-2; ISO 639-1 requires 2)
extlang1      =  4 (each subsequent subtag includes '-')
extlang2      =  4 (unlikely: needs prefix="language-extlang1")
extlang3      =  4 (extremely unlikely)
script        =  5 (if not suppressed: see Section 4.1)
region        =  4 (UN M.49; ISO 3166 requires 3)
variant1      =  9 (MUST have language as a prefix)
variant2      =  9 (MUST have language-variant1 as a prefix)

total         = 42 characters
```

Figure 7: Derivation of the Limit on Tag Length

4.3.2. 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

4.4. Canonicalization of Language Tags

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

A language tag is in canonical form when:

1. The tag is well-formed according the rules in Section 2.1 and Section 2.2.

2. Subtags of type 'Region' that have a Preferred-Value mapping in the IANA registry (see Section 3.1) SHOULD be replaced with their mapped value. Note: In rare cases, the mapped value will also have a Preferred-Value.

3. Redundant or grandfathered tags that have a Preferred-Value mapping in the IANA registry (see Section 3.1) MUST be replaced with their mapped value. These items either are deprecated mappings created before the adoption of this document (such as the mapping of "no-nyn" to "nn" or "i-klingon" to "tlh") or are the result of later registrations or additions to this document (for example, "zh-guoyu" might be mapped to a language-extlang combination such as "zh-cmn" by some future update of this document).

4. Other subtags that have a Preferred-Value mapping in the IANA registry (see Section 3.1) MUST be replaced with their mapped value. These items consist entirely of clerical corrections to ISO 639-1 in which the deprecated subtags have been maintained for compatibility purposes.

5. If more than one extension subtag sequence exists, the extension sequences are ordered into case-insensitive ASCII order by singleton subtag.

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 but not in canonical form.

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

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. Note that this corresponds to the following casing rules: uppercase all non-initial two-letter subtags; titlecase all non-initial four-letter subtags; lowercase everything else.

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" 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'.

Note: 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. Validating processors SHOULD NOT generate tags that include these values, although the values are canonical when they appear in a language tag.

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 are 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.

4.5. 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.

Subtags defined in the IANA registry as having a specific private use meaning convey more information that a purely private use tag prefixed by the singleton subtag 'x'. For applications, this additional information MAY be useful.

For example, the region subtags 'AA', 'ZZ', and in the ranges 'QM'-'QZ' and 'XA'-'XZ' (derived from ISO 3166 private use codes) MAY 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", which contains no information about the language subtag or script subtag outside of the private agreement.

However, in some cases content tagged with private use subtags MAY 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 undertake to maintain the subtag and extension registries as defined by this document and in accordance with the requirements of [RFC2434].

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.

5.1. Language Subtag Registry

Upon adoption of this document, the registry will be initialized by a companion document: [RFC4645]. The criteria and process for selecting the initial set of records are described in that document. The initial set of records represents no impact on IANA, since the work to create it will be performed externally.

The new registry MUST be listed under "Language Tags" at <http://www.iana.org/numbers.html>, replacing the existing registrations defined by [RFC3066]. The existing set of registration forms and RFC 3066 registrations MUST be relabeled as "Language Tags (Obsolete)" and maintained (but not added to or modified).

Future work on the Language Subtag Registry SHALL be limited to inserting or replacing whole records preformatted for IANA by the Language Subtag Reviewer as described in Section 3.3 of this document and archiving the forwarded registration form.

Each record MUST be sent to iana@iana.org with a subject line indicating whether the enclosed record is 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). Records MUST NOT be deleted from the registry. IANA MUST place any inserted or modified records into the appropriate section of the language subtag registry, grouping the records by their 'Type' field. Inserted records MAY be placed anywhere in the appropriate section; there is no guarantee of the order of the records beyond grouping them together by 'Type'. Modified records MUST overwrite the record they replace.

Included in any request to insert or modify records MUST be a new File-Date record. This record MUST be placed first in the registry. In the event that the File-Date record present in the registry has a later date than the record being inserted or modified, the existing record MUST be preserved.

5.2. Extensions Registry

The Language Tag Extensions Registry will also be generated and sent to IANA as described in Section 3.7. This 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. It should reasonably be seen to come 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 4.3 for details on language tag truncation, which can occur as a consequence of defenses against buffer overflow.

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 it being always accessible, to prevent denial-of-service attacks.

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 should not have any character set issues.

Rendering of characters based on the content of a language tag is not addressed in this memo. Historically, some languages have relied on the use of specific character sets or other information in order to infer how a specific character should be rendered (notably this applies to language- and culture-specific variations of Han ideographs as used in Japanese, Chinese, and Korean). When language

tags are applied to spans of text, rendering engines sometimes use that information in deciding which font to use in the absence of other information, particularly where languages with distinct writing traditions use the same characters.

8. Changes from RFC 3066

The main goals for this revision of language tags were the following:

Compatibility. All RFC 3066 language tags (including those in the IANA registry) remain valid in this specification. The changes in this document represent additional constraints on language tags. That is, in no case is the syntax more permissive and processors based on the ABNF and other provisions of RFC 3066 (such as those described in [XMLSchema]) will be able to process the tags described by this document. In addition, this document defines language tags in such as way as to ensure future compatibility.

Stability. Because of changes in the past in the underlying ISO standards, a valid RFC 3066 language tag could become invalid or have its meaning change. This has the potential of invalidating content that may have an extensive shelf-life. In this specification, once a language tag is valid, it remains valid forever.

Validity. The structure of language tags defined by this document makes it possible to determine if a particular tag is well-formed without regard for the actual content or "meaning" of the tag as a whole. This is important because the registry grows and underlying standards change over time. In addition, it must be possible to determine if a tag is valid (or not) for a given point in time in order to provide reproducible, testable results. This process must not be error-prone; otherwise implementations might give different results. By having an authoritative registry with specific versioning information, the validity of language tags at any point in time can be precisely determined (instead of interpolating values from many separate sources).

Utility. It is sometimes important to be able to differentiate between written forms of a language -- for many implementations this is more important than distinguishing between the spoken variants of a language. Languages are written in a wide variety of different scripts, so this document provides for the generative use of ISO 15924 script codes. Like the generative use of ISO language and country codes in RFC 3066, this allows combinations to be produced without resorting to the registration process. The addition of UN M.49 codes provides for the generation of language tags with regional scope, which is also required by some applications.

The recast of the registry from containing whole language tags to subtags is a key part of this. An important feature of RFC 3066 was that it allowed generative use of subtags. This allows people to meaningfully use generated tags, without the delays in registering whole tags or the need to register all of the combinations that might be useful.

The choice of placing the extended language and script subtags between the primary language and region subtags was widely debated. This design was chosen because the prevalent matching and content negotiation schemes rely on the subtags being arranged in order of increasing specificity. That is, the subtags that mark a greater barrier to mutual intelligibility appear left-most in a tag. For example, when selecting content written in Azerbaijani, the script (Arabic, Cyrillic, or Latin) represents a greater barrier to understanding than any regional variations (those associated with Azerbaijan or Iran, for example). Individuals who prefer documents in a particular script, but can deal with the minor regional differences, can therefore select appropriate content. Applications that do not deal with written content will continue to omit these subtags.

Extensibility. Because of the widespread use of language tags, it is disruptive to have periodic revisions of the core specification, even in the face of demonstrated need. The extension mechanism provides for a way for independent RFCs to define extensions to language tags. These extensions have a very constrained, well-defined structure that prevents extensions from interfering with implementations of language tags defined in this document.

The document also anticipates features of ISO 639-3 with the addition of the extended language subtags, as well as the possibility of other ISO 639 parts becoming useful for the formation of language tags in the future.

The use and definition of private use tags have also been modified, to allow people to use private use subtags to extend or modify defined tags and to move as much information as possible out of private use and into the regular structure.

The goal for each of these modifications is to reduce or eliminate the need for future revisions of this document.

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

- Defines the ABNF and rules for subtags so that the category of all subtags can be determined without reference to the registry.

- Adds the concept of well-formed vs. validating processors, defining the rules by which an implementation can claim to be one or the other.

- Replaces the IANA language tag registry with a language subtag registry that provides a complete list of valid subtags in the IANA registry. This allows for robust implementation and ease of maintenance. The language subtag registry becomes the canonical source for forming language tags.

- Provides a process that guarantees stability of language tags, by handling reuse of values by ISO 639, ISO 15924, and ISO 3166 in the event that they register a previously used value for a new purpose.

- Allows ISO 15924 script code subtags and allows them to be used generatively. Defines a method for indicating in the registry when script subtags are necessary for a given language tag.

- Adds the concept of a variant subtag and allows variants to be used generatively.

- Adds the ability to use a class of UN M.49 tags for supra-national regions and to resolve conflicts in the assignment of ISO 3166 codes.

- Defines the private use tags in ISO 639, ISO 15924, and ISO 3166 as the mechanism for creating private use language, script, and region subtags, respectively.

- Adds a well-defined extension mechanism.

- Defines an extended language subtag, possibly for use with certain anticipated features of ISO 639-3.

9. References

9.1. Normative References

[ISO10646] International Organization for Standardization, "ISO/IEC 10646:2003. Information technology -- Universal Multiple-Octet Coded Character Set (UCS)", 2003.

[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 3166-1:1997. Codes for the representation of names of countries and their subdivisions -- Part 1: Country codes", 1997.

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

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

[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.

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

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

[RFC2434] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.

[RFC2860] Carpenter, B., Baker, F., and M. Roberts, "Memorandum of Understanding Concerning the Technical Work of the Internet Assigned Numbers Authority", RFC 2860, June 2000.

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

[RFC4234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 4234, October 2005.

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

9.2. Informative References

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

[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.

[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", BCP 47, RFC 3066, January 2001.

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

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

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

[Unicode] Unicode Consortium, "The Unicode Standard, Version 5.0", Boston, MA, Addison-Wesley, 2007. ISBN 0-321-48091-0.

[XML10] Bray (et al), T., "Extensible Markup Language (XML) 1.0", 02 2004.

[XMLSchema] Biron, P., Ed. and A. Malhotra, Ed., "XML Schema Part 2: Datatypes Second Edition", 10 2004, <http://www.w3.org/TR/xmlschema-2/>.

[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. 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 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 (in alphabetical order) contributed to this document or to RFCs 1766 and 3066:

Glenn Adams, Harald Tveit Alvestrand, Tim Berners-Lee, Marc Blanchet, Nathaniel Borenstein, Karen Broome, Eric Brunner, Sean M. Burke, M.T. Carrasco Benitez, Jeremy Carroll, John Clews, Jim Conklin, Peter Constable, John Cowan, Mark Crispin, Dave Crocker, Elwyn Davies, Martin Duerst, Frank Ellerman, Michael Everson, Doug Ewell, Ned Freed, Tim Goodwin, Dirk-Willem van Gulik, Marion Gunn, Joel Halpren, Elliotte Rusty Harold, Paul Hoffman, Scott Hollenbeck, Richard Ishida, Olle Jarnefors, Kent Karlsson, John Klensin, Erkki Kolehmainen, Alain LaBonte, Eric Mader, Ira McDonald, Keith Moore, Chris Newman, Masataka Ohta, Dylan Pierce, Randy Presuhn, George Rhoten, Felix Sasaki, Markus Scherer, Keld Jorn Simonsen, Thierry Sourbier, Otto Stolz, Tex Texin, Andrea Vine, Rhys Weatherley, Misha Wolf, Francois Yergeau 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 must go to Michael Everson, who has served as Language Tag Reviewer for almost the complete period since the publication of RFC 1766. Special thanks to Doug Ewell, for his production of the first complete subtag registry, and his work in producing a test parser for verifying language tags.

Appendix B. 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)

Language-Script-Region:

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

sr-Latn-CS (Serbian written using the Latin script as used in Serbia and Montenegro)

Language-Variant:

sl-rozaj (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:

sl-Latn-IT-nedis (Nadiza dialect of Slovenian written using the Latin script as used in Italy. Note that this tag is NOT RECOMMENDED because subtag 'sl' has a Suppress-Script value of 'Latn')

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

Extended language subtags (examples ONLY: extended languages MUST be defined by revision or update to this document):

zh-min

zh-min-nan-Hant-CN

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-CS (Serbian, private script, for Serbia and Montenegro)

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)

Authors' Addresses

Addison Phillips (Editor)
Yahoo! Inc.

EMail: addison@inter-locale.com

Mark Davis (Editor)
Google

EMail: mark.davis@macchiato.com or mark.davis@google.com

Full Copyright Statement

Intellectual Property

Acknowledgement

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).