

Language Tags

W3C Project Review

A thick, dark blue horizontal bar with rounded ends, positioned below the text 'W3C Project Review'.

Presenter and Agenda

- Addison Phillips
- *Internationalization Architect, Yahoo!*
- *Co-Editor, Language Tag Registry Update (LTRU) Working Group*
 - *RFC 4646 “Tags for Identification of Languages”*
 - *RFC 4647 “Matching of Language Tags”*

Language Tags

What's a language tag?

Why the #@&%\$ are they changing them (again)?

What do we need to do?



Language Tags

- Enable presentation, selection, and negotiation of content
- Defined by BCP 47
 - Widely used! XML, HTML, RSS, MIME, SOAP, SMTP, LDAP, CSS, XSL, CCXML, Java, C#, ASP, perl.....
 - Well understood (?)

Locale Identifiers

- Different ideas:
 - Accept-Locale vs. Accept-Language
 - URIs/URNs, etc.
 - CLDR/LDML
- And Requirements:
 - Operating environments and harmonization
 - App Servers
 - Web Services

In the Beginning

Received Wisdom from the Dark Ages

- Locales:

- japanese, french, german, C
- ENU, FRA, JPN
- ja_JP.PCK
- AMERICAN_AMERICA.WE8ISO8859P1

- Languages...

... looked a lot like locales (and vice versa)



Locales and Language Tags meet

- Conversations in Prague...
 - Language tags are being locale identifiers anyway...
 - Not going to need a big new thing...
 - Just a few things to fix...
... we can do this really fast



BCP 47 Basic Structure

- Alphanumeric (ASCII only) subtags
- Up to eight characters long
- Separated by hyphens
- Case not important (i.e. zh = ZH = zH = Zh)

1*8alphanum * ["-" 1*8 alphanum]

RFC 1766

zh-TW

ISO 639-1 (alpha2)

ISO 3166 (alpha2)

i-klíngon

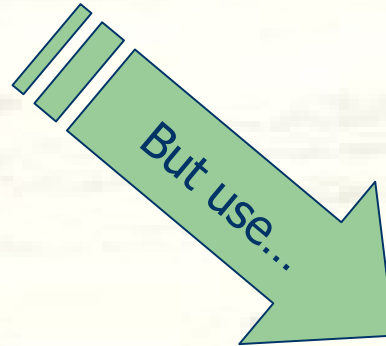
Registered value



RFC 3066

SCO-GB

ISO 639-2 (alpha 3 codes)



~~en~~-GB

alpha 2 codes when they exist

Problems

- Script Variation:
 - zh-Hant/zh-Hans
 - (sr-Cyrl/sr-Latn, az-Arab/az-Latn/az-Cyrl, etc.)
- Obsolence of registrations:
 - art-lojban (now jbo), i-klingon (now tlh)
- Instability in underlying standards:
 - sr-CS (CS used to be Czechoslovakia...

And More Problems

- Lack of scripts
- Little support for registered values in software
- Reassignment of values by ISO 3166
- Lack of consistent tag formation (Chinese dialects?)
- Standards not readily available, bad references
- Bad implementation assumptions
 - 1*8 alphanum *[“-” 1*8 alphanum]
 - 2*3 ALPHA [“-” 2ALPHA]
- Many registrations to cover small variations
 - 8 German registrations to cover two variations

RFC 4646 (“3066bis”)

- Defines a generative syntax
 - machine readable
 - future proof, extensible
- Defines a single source (IANA Language Subtag Registry)
 - Stable subtags, no conflicts
 - Machine readable
- Defines when to use subtags
 - (sometimes)

RFC 3066bis and LTRU

sl-Latn-IT-rozaj-x-mine

Private Use and Extension

Registered variants (any number)

ISO 3166 (alpha2) or UN M49

ISO 15924 script codes (alpha 4)

ISO 639-1/2 (alpha2/3)

More Examples

- es-419 (Spanish for Americas)
- en-US (English for USA)
- de-CH-1996 (Old tags are all valid)
- sl-rozaj-nedis (Multiple variants)
- zh-t-wadegile (Extensions)
- x-tim-b-lee (Private Use, opaque)
- en-US-x-twain (Private Use, composed)

Benefits

- Subtag registry in one place: one source.
- Subtags identified by length/content
- Extensible
- Compatible with RFC 3066 tags
- Stable: subtags are forever

ABNF

```
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*8alphanumeric       ; registered variants
               / (DIGIT 3alphanumeric)


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

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*8alphanumeric))

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

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



Registry

- Stability guarantees on normative information, especially subtags
- Fixed registration rules (“no junk”)
- Deprecation
- Preferred Values
- File and Subtag dates, deprecation dates
- Prefixes (what subtags go together)
- Descriptions and Comments

Example: Language

%%

Type: language

Subtag: in

Description: Indonesian

Added: 2005-10-16

Preferred-Value: id

Deprecated: 1989-01-01

Suppress-Script: Latn

%%

Example: Variant

%%

Type: variant

Subtag: nedis

Description:

Natisone dialect

Description: Nadiza dialect

Added: 2005-10-16

Prefix: sl

%%

Example: Grandfathered

%%

Type: grandfathered

Tag: art-lojban

Description: Lojban

Added: 2001-11-11

Preferred-value: jbo

Deprecated: 2003-09-02

Comments: replaced by ISO code jbo

%%

Problems

- Matching
 - Does “en-US” match “en-Latn-US”?
- Tag Choices
 - Users have more to choose from.
- Implementations
 - More to do, more to think about
 - (easier to parse, process, support the good stuff)

Tag Matching

- Uses “Language Ranges” in a “Language Priority List” to select sets of content according to the language tag.
- Basically what we already had, but in one place.
- Three Schemes
 - Basic Filtering
 - Extended Filtering
 - Lookup

Filtering

- Ranges specify the *least* specific item
 - “en” matches:
“en”, “en-US”, “en-Brai”, “en-boont”
- Can select zero or more items (selects a set, including empty set)

Basic Filtering

- Basic matching uses plain prefixes
 - en-US matches:
 - “en-us”, “en-us-boont”
 - en-US does NOT match:
 - “en-Latn-US”, “en-boont”, “en-x-US”

Extended Filtering

- Extended matching can match “inside bits”
 - “en-*-US” matches:
“en-Brai-US”, “en-us”, “en-us-boont”
 - Does NOT match:
“en-x-US”, “en-Brai”
- Wildcard only has “meaning” in first position
 - for example: “*-DE”
 - en-US equivalent to en-*-US
matches “en-Brai-US”!!!

Lookup

- Range specifies the *most* specific tag in a match.
 - “en-US” matches “en” and “en-US” but *not* “en-US-boont”
- Mirrors the locale fallback mechanism and many language negotiation schemes.
- Implementations **MUST** specify defaulting behavior.

Fallback

Range to match: zh-Hant-CN-x-private1-private2

1. zh-Hant-CN-x-private1-private2
2. zh-Hant-CN-x-private1
3. zh-Hant-CN
4. zh-Hant
5. zh
6. (default)

Defaulting

Language Preference List: “fr-fr,zh-hant”

1. fr-FR
2. fr
3. zh-Hant // next language
4. zh
5. ja-JP // now searching for the default content
6. ja
7. (implementation defined default)

Filtering vs. Lookup

- Filtering can produce “zero or more matches”
 - example: CSS :lang pseudo-attribute
 - ... but can produce “exactly one” behavior
- Lookup produces “exactly one” match
 - example: resource lookup

What to Reference

- BCP 47 (urn:ietf:bcp:47)
- Tags: RFC 4646 or successor
 - tags
- Matching: RFC 4647 or successor
 - language ranges, language preferences (“language priority list”), matching schemes

References to Replace

- RFC 1766, RFC 3066 (tags)
- ISO 639, ISO 3166 (XML 1.0 4e!) [reference IANA Language Subtag Registry]
- RFC 2616 (HTTP 1.1, §14.4: language ranges, basic matching)

Approach Changes, Issues

- Reference the registry
- Specify “well-formed” or “validating”
- Choose matching schemes carefully
 - consider using Extended Filtering, e.g. in XPath
 - use Lookup for locale-like operations
- `xml:lang=""` matching

What Do I Do (Content Author)?

- Not much.
 - Existing tags are all still valid: tagging is mostly unchanged.
 - Resist temptation to (ab)use the private use subtags.
- If your language typically has script variations (or if you content exhibits it):
 - ONLY THEN tag content with script subtag(s)
 - Script subtags only apply to a small number of languages: “zh”, “sr”, “uz”, “az”, “mn”, and a very small number of others.

What Do I Do (Programmer)?

- Check code for compliance with 4646
 - Decide on well-formed or validating implementation (note requirements well)
 - Implement suppress-script
 - Change to using the registry
 - Bother infrastructure folks (Java, MS, Mozilla, etc) to implement the standard

What Do I Do (End-User)?

- Check and update your language ranges.
- Tag content wisely.

LTRU Milestone Dates

- RFC 4645, 4646, 4647 published
- Coming: RFC 4646bis (3066ter)
 - This includes ISO 639-3 support and extended language support

RFC 4646bis: What, more changes?!?

- Adds support for ISO 639-3 (about 7000 additional alpha3 language codes)
 - Two flavors: language subtags and extlangs
 - sgn-ase [sign language, ASL]
 - zh-cmn [Chinese, Mandarin]
zh-cms [Chinese, Cantonese]
 - azz [Highland Puebla Nahuatl]
- Nothing else??

W3C and Unicode Activities

- W3C
 - LTLI (Language Tags and Locale Identifiers)
 - Web services (WS-I18N)
 - XML, HTML
 - Notes and Best Practices (I18N GEO WG)
- Unicode Consortium
 - LDML
 - CLDR

Questions/Discussion

