Internet Architecture Board (IAB)
Request for Comments: 6912
Category: Informational
ISSN: 2070-1721

A. Sullivan
Dyn, Inc.
D. Thaler
Microsoft
J. Klensin

O. Kolkman
NLnet Labs

April 2013

Principles for Unicode Code Point Inclusion in Labels in the DNS

Abstract

Internationalized Domain Names in Applications (IDNA) makes available to DNS zone administrators a very wide range of Unicode code points. Most operators of zones should probably not permit registration of U-labels using the entire range. This is especially true of zones that accept registrations across organizational boundaries, such as top-level domains and, most importantly, the root. It is unfortunately not possible to generate algorithms to determine whether permitting a code point presents a low risk. This memo presents a set of principles that can be used to guide the decision of whether a Unicode code point may be wisely included in the repertoire of permissible code points in a U-label in a zone.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Architecture Board (IAB) and represents information that the IAB has deemed valuable to provide for permanent record. It represents the consensus of the Internet Architecture Board (IAB). Documents approved for publication by the IAB are not a candidate for any level of Internet Standard; see Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc6912.

Table of Contents

1. Introduction

Operators of a DNS zone need to set policies around what Unicode code points are allowed in labels in that zone. Typically there are a number of important goals to consider when constructing such policies. These include, for instance, avoiding possible visual confusability between two labels, avoiding possible confusion between Fully Qualified Domain Names (FQDNs) and IP address literals, accessibility to the disabled (see "Web Content Accessibility Guidelines (WCAG) 2.0" [WCAG20] for some discussion in a web context), and other usability issues.

This document provides a set of principles that zone operators can use to construct their code point policies in order to improve usability and clarity and thereby reduce confusion.

1.1. Terminology

This document uses the following terms.

A-label: an LDH label that starts with "xn--" and meets all the IDNA requirements, with additional restrictions as explained in Section 2.3.2.1 of the IDNA Definitions document [RFC5890].

Character: a member of a set of elements used for the organization, control, or representation of data. See Section 2 of the Internationalization Terminology document [RFC6365] for more details.

Language: a way that humans communicate. The use of language occurs in many forms, the most common of which are speech, writing, and signing. See Section 2 of RFC 6365 for more details.

LDH label: a string consisting of ASCII letters, digits, and the hyphen, with additional restrictions as explained in Section 2.3.1 of RFC 5890.

Public zone: in this document, a DNS zone that accepts registration requests from organizations outside the zone administrator’s own organization. (Whether the zone performs delegation is a separate question. What is important is the diversity of the registration-requesting community.) Note that under this definition, the root zone is a public zone, though one that has a unique function in the DNS.

Rendering: the display of a string of text. See Section 5 of RFC 6365 for more details.

Script: a set of graphic characters used for the written form of one or more languages. See Section 2 of RFC 6365 for more details.

U-label: a string of Unicode characters that meets all the IDNA requirements and includes at least one non-ASCII character, with additional restrictions as explained in Section 2.3.2.1 of RFC 5890.

Writing system: a set of rules for using one or more scripts to write a particular language. See Section 2 of RFC 6365 for more details.

This memo does not propose a protocol standard, and the use of words such as "should" follow the ordinary English meaning, and not that laid out in [RFC2119].

2. Background

In recent communications [IABCOMM1] [IABCOMM2], the IAB has emphasized the importance of conservatism in allocating labels conforming to IDNA2008 [RFC5890] [RFC5891] [RFC5892] [RFC5893] [RFC5894] [RFC5895] in DNS zones, and especially in the root zone. Traditional LDH labels in the root zone used only alphabetic characters (i.e., ASCII a-z, which under the DNS also match A-Z). Matters are more complicated with U-labels, however. The IAB communications recommended that U-labels permit only code points with a General_Category (gc) of Ll (Lowercase_Letter), Lo (Other_Letter), or Lm (Modifier_Letter), but noted that for practical considerations other code points might be permitted on a case-by-case basis.

The IAB recommendations do, however, leave some issues open that need to be addressed. It is not clear that all code points permitted under IDNA2008 that have a General_Category of Lo or Lm are appropriate for a zone such as the root zone. To take but one example, the code point U+02BC (MODIFIER LETTER APOSTROPHE) has a General_Category of Lm. In practically every rendering (and we are unaware of an exception), U+02BC is indistinguishable from U+2019 (RIGHT SINGLE QUOTATION MARK), which has a General_Category of Pf (Final_Punctuation). U+02BC will also be read by large numbers of people as being the same character as U+0027 (APOSTROPHE), which has a General_Category of Po (Other_Punctuation), and some computer systems may treat U+02BC as U+0027. U+02BC is PROTOCOL VALID (PVALID) under IDNA2008 (see the IDNA Code Points document [RFC5892]), whereas both other code points are DISALLOWED. So, to begin with, it is plain that not every code point with a

General_Category of Ll, Lo, or Lm is consistent with the type of conservatism principle discussed in Section 4.1 below or the previous IAB recommendations.

To make matters worse, some languages are dependent on code points with General_Category Mc (Spacing_Mark) or General_Category Mn (Nonspacing_Mark). This dependency is particularly common in Indic languages, though not exclusive to them. (At the risk of vastly oversimplifying, the overarching issue is mostly the interaction of complex writing systems and the way Unicode works.) To restrict users of those languages to only code points with General_Category of Ll, Lo, or Lm would be extremely limiting. While DNS labels are not words, or sentences, or phrases (as noted in the next steps for IDN [RFC4690]), they are intended to support useful mnemonics. Mnemonics that diverge wildly from the usual conventions are poor ones, because in not following the usual conventions they are not easy to remember. Also, wide divergence from usual conventions, if not well-justified (and especially in a shared namespace like the root), invites political controversy.

Many of the issues above turn out to be relevant to all public zones. Moreover, the overall issue of developing a policy for code point permission is common to all zones that accept A-labels or U-labels for registration. As Section 4.3 of the IDNA Protocol document [RFC5891] says, every registry at every level of the DNS is "expected to establish policies about label registrations".

For reasons of sound management, it is not desirable to decide whether to permit a given code point only when an application containing that code point is pending. That approach reduces predictability and is bound to appear subject to special pleas. It is better instead to produce the rules governing acceptance of code points in advance.

As is evident from the foregoing discussion about the Letter and Mark categories, it is simply not possible to make code point decisions algorithmically. If it were possible to develop such an algorithm, it would already exist: the DNS is hardly unique in needing to impose restrictions on code points while accommodating many different linguistic communities. Nevertheless, new guidelines can be made by starting from overarching principles. These guidelines act more as meta-rules, leading to the establishment of other rules about the inclusion and exclusion of particular code points in labels in a given zone, always based on the list of code points permitted by IDNA.

2.1. More-Restrictive Rules Going Up the DNS Tree

A set of principles derived from the above ideas follows in Sections 3 through 5 below. Such principles fall into three categories. Some principles apply to every DNS zone. Some additional principles apply to all public zones, including the root zone. Finally, other principles apply only to the root zone. This means that zones higher in the DNS tree tend to have more restrictive rules (since additional principles apply), and zones lower in the DNS tree tend to have less restrictive rules, since they are used within a more narrow context. In general, the relevant context for a principle is that of the zone, not that of a given subset of the user community; for the root zone, for example, the context is "the entire Internet population".

3. Principles Applicable to All Zones

3.1. Longevity Principle

Unicode properties of a code point ought to be stable across the versions of Unicode that users of the zone are likely to have installed. Because it is possible for the properties of a code point to change between Unicode versions, a good way to predict such stability is to ensure that a code point has in fact been stable for multiple successive versions of Unicode. This principle is related to the Stability Principle in Section 4.5.

The more diverse the community using the zone, the greater the importance of following this principle. The policy for a leaf zone in the DNS might only require stability across two Unicode versions, whereas a more public zone might require stability across four or more releases before the code point’s properties are considered long-lived and stable.

3.2. Least Astonishment Principle

Every zone administrator should be sensitive to the likely use of a code point to be permitted, particularly taking into account the population likely to use the zone. Zone administrators should especially consider whether a candidate code point could present difficulty if the code point is encountered outside the usual linguistic circumstances. By the same token, the failure to support a code point that is normal in some linguistic circumstances could be very surprising for users likely to encounter the names in that circumstance.

3.3. Contextual Safety Principle

Every zone administrator should be sensitive to ways in which a code point that is permitted could be used in support of malicious activity. This is not a completely new problem: the digit 1 and the lowercase letter l are, for instance, easily confused in many contexts. The very large repertoire of code points in Unicode (even just the subset permitted for IDNs) makes the problem somewhat worse, just because of the scale.

4. Principles Applicable to All Public Zones

4.1. Conservatism Principle

Public zones are, by definition, zones that are shared by different groups of people. Therefore, any decision to permit a code point in a public zone (including the root) should be as conservative as practicable. Doubts should always be resolved in favor of rejecting a code point for inclusion rather than in favor of including it, in order to minimize risk.

4.2. Inclusion Principle

Just as IDNA2008 starts from the principle that the Unicode range is excluded, and then adds code points according to derived properties of the code points, so a public zone should only permit inclusion of a code point if it is known to be "safe" in terms of usability and confusability within the context of that zone. The default treatment of a code point should be that it is excluded.

4.3. Simplicity Principle

The rules for determining whether a code point is to be included should be simple enough that they are readily understood by someone with a moderate background in the DNS and Unicode issues. This principle does not mean that a completely naive person needs to be able to understand the rationale for including a code point, but it does mean that if the reason for inclusion of a very peculiar code point, even a safe one, is too difficult to understand, the code point would not be permitted.

The meaning of "simple" or "readily understood" is context-dependent. For instance, the root zone has to serve everyone in the world; for practical purposes, this means that the reasons for including a code point need to be comprehensible even to people who cannot use the script where the code point is found. In a zone that permits a constrained subset of Unicode characters (for instance, only those needed to write a single alphabetic language) and that supports a

clearly delineated linguistic community (for instance, the speakers of a single language with well-understood written conventions), more complicated rules might be acceptable. Compare this principle with the Least Astonishment Principle in Section 3.2.

4.4. Predictability Principle

The rules for determining whether a code point is to be included should be predictable enough that those with the requisite understanding of DNS, IDNA, and Unicode will usually reach the same conclusion. This is not a requirement for algorithmic treatment of code points; as previously noted, that is not possible. Rather, it is to say that the consistent application of professional judgment is likely to yield the same results; combined with the principle in Section 4.1, when results are not predictable, the anomalous code point would not be permitted.

Just as in Section 4.3, this principle tends to cause more restriction the more diverse the community using the zone; it is most restrictive for the root zone. This is because what is predictable within a given language community is possibly very surprising across languages.

4.5. Stability Principle

Once a code point is permitted, it is at least very hard to stop permitting that code point. In public zones (including the root), the list of code points to be permitted should change very slowly, if at all, and usually only in the direction of permitting an addition as time and experience indicate that inclusion of such a code point is both safe and consistent with these principles.

5. Principle Specific to the Root Zone

5.1. Letter Principle

"Requirements for Internet Hosts - Application and Support" [RFC1123] notes that top-level labels "will be alphabetic". In the absence of widespread agreement about the force of that note, prudence suggests that U-labels in the root zone should exclude code points that are not normally used to write words, or that are in some cases normally used for purposes other than writing words. This is not the same as using Unicode's General_Category to include only letters. It is a restriction that expands the possible class of included code points beyond the Unicode letters, but only expands so far as to include the things that are normally used to create words. Under this principle, code points with (for example) General_Category Mn (Nonspacing_Mark) might be included -- but only those that are used to write words and

not (for instance) musical symbols. In addition, such marks should only be used within a label in ways that they would be used when making a word: combinations that would be nonsense when used in a word should also be rejected when tried in DNS labels. This principle should be applied as narrowly as possible; as the next steps for IDN document [RFC4690] says, "While DNS labels may conveniently be used to express words in many circumstances, the goal is not to express words (or sentences or phrases), but to permit the creation of unambiguous labels with good mnemonic value".

6. Confusion and Context

While many discussions of confusion have focused on characters, e.g., whether two characters are confusable with each other (and under what circumstances), a focus on characters alone could lead to the prohibition of very large numbers of labels, including many that present little risk. Instead, the focus should be on whether one label is confusable with another. For example, if a label contains several characters that are distinct to a particular script, and all of its characters are from that script, it is inherently not confusable with a label from any other script no matter what other characters might appear in it. Another label that lacks those distinguishing characters might be a problem. The notion extends from labels to domain names, in the sense that distinguishing characters used in a higher-level label may set expectations with respect to the characters in the lower-level labels. This expectation might be regarded as a benefit, but it is also a problem, since there is no technical way to require consistent policies in delegated namespaces.

7. Conclusion

The principles outlined in this document can be applied when considering any range of Unicode code points for possible inclusion in a DNS zone. It is worth observing that doing anything (especially in light of Section 4.5) implicitly disadvantages communities with a writing system not yet well understood and not represented in the technical and policy communities involved in the discussion. That disadvantage is to be guarded against as much as practical, but is effectively impossible to prevent (while still taking action) in light of imperfect human knowledge.

8. Security Considerations

The principles outlined in this memo are intended to improve usability and clarity and thereby reduce confusion among different labels. While these principles may contribute to reduction of risk, they are not sufficient to provide a comprehensive internationalization policy for zone management.

Additional discussion of security considerations can be found in the Unicode Security Considerations [UTR36].

9. Acknowledgements

The authors thank the participants in the IAB Internationalization program for the discussion of the ideas in this memo, particularly Marc Blanchet. In addition, Stephane Bortzmeyer, Paul Hoffman, Daniel Kalchev, Panagiotis Papaspiliopoulos, and Vaggelis Segredakis made specific comments.

10. IAB Members at the Time of Approval

Bernard Aboba
Jari Arkko
Marc Blanchet
Ross Callon
Alissa Cooper
Spencer Dawkins
Joel Halpern
Russ Housley
David Kessens
Danny McPherson
Jon Peterson
Dave Thaler
Hannes Tschofenig

11. Informative References

[IABCOMM1] Internet Architecture Board, "IAB Statement: 'The interpretation of rules in the ICANN gTLD Applicant Guidebook'", February 2012, <http://www.iab.org/documents/correspondence-reports-documents/201/>.

[IABCOMM2] Internet Architecture Board, "Response to ICANN questions concerning 'The interpretation of rules in the ICANN gTLD Applicant Guidebook'", March 2012, <http://www.iab.org/documents/correspondence-reports-documents/201/>.

[RFC1123] Braden, R., "Requirements for Internet Hosts - Application and Support", STD 3, RFC 1123, October 1989.

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

[RFC4690] Klensin, J., Faltstrom, P., Karp, C., and IAB, "Review and Recommendations for Internationalized Domain Names (IDNs)", RFC 4690, September 2006.

[RFC5890] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework", RFC 5890, August 2010.

[RFC5891] Klensin, J., "Internationalized Domain Names in Applications (IDNA): Protocol", RFC 5891, August 2010.

[RFC5892] Faltstrom, P., "The Unicode Code Points and Internationalized Domain Names for Applications (IDNA)", RFC 5892, August 2010.

[RFC5893] Alvestrand, H. and C. Karp, "Right-to-Left Scripts for Internationalized Domain Names for Applications (IDNA)", RFC 5893, August 2010.

[RFC5894] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Background, Explanation, and Rationale", RFC 5894, August 2010.

[RFC5895] Resnick, P. and P. Hoffman, "Mapping Characters for Internationalized Domain Names in Applications (IDNA) 2008", RFC 5895, September 2010.

[RFC6365] Hoffman, P. and J. Klensin, "Terminology Used in Internationalization in the IETF", BCP 166, RFC 6365, September 2011.

[UTR36] Davis, M. and M. Suignard, "Unicode Security Considerations", Unicode Technical Report #36, July 2012.

[WCAG20] W3C, "Web Content Accessibility Guidelines (WCAG) 2.0", W3C Recommendation, December 2008, <http://www.w3.org/TR/2008/REC-WCAG20-20081211/>.

Authors' Addresses

Andrew Sullivan
Dyn, Inc.
150 Dow St
Manchester, NH 03101
USA

EMail: asullivan@dyn.com

Dave Thaler
Microsoft
One Microsoft Way
Redmond, WA 98052
USA

EMail: dthaler@microsoft.com

John C Klensin
1770 Massachusetts Ave, Ste 322
Cambridge, MA 02140
USA

Phone: +1 617 491 5735
EMail: john-ietf@jck.com

Olaf Kolkman
NLnet Labs
Science Park 400
Amsterdam 1098 XH
The Netherlands

EMail: olaf@NLnetLabs.nl