

Network Working Group
Internet-Draft
Intended status: Informational
Expires: December 7, 2012

A. Sullivan
Dyn, Inc.
D. Thaler
Microsoft
O. Kolkman
NLnet Labs
June 5, 2012

Principles for Unicode Code Point Inclusion in Labels in the DNS Root
draft-sullivan-dns-zone-codepoint-pples-00

Abstract

Traditionally, the management of the DNS root zone permitted only "alphabetic" labels. As long as the root zone included only ASCII characters, and as long as there was only one form of a label, the restriction plainly meant that only the letters A-Z and a-z were permitted. The advent of internationalized labels using IDNA2008 presents some complications for the restriction. One of the complications is the meaning of the term "alphabetic" when applied to the Unicode code points in U-labels. This memo presents a set of principles that can be used to determine 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 Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on December 7, 2012.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Background and Introduction	3
1.1.	Terminology	4
2.	Conservatism Principle	4
3.	Inclusion Principle	4
4.	Simplicity Principle	4
5.	Predictability Principle	5
6.	Stability Principle	5
7.	Letter Principle	6
8.	Conclusion	6
9.	Security Considerations	6
10.	IANA Considerations	7
11.	Acknowledgements	7
12.	Informative References	7
	Authors' Addresses	8

1. Background and Introduction

In recent communications ([[IABCOMM1](#)] and [[IABCOMM2](#)]), the IAB has emphasized the importance of conservatism in allocating labels conforming to IDNA2008 ([[RFC5890](#)], [[RFC5891](#)], [[RFC5892](#)], [[RFC5893](#)], [[RFC5894](#)], [[RFC5895](#)]) inside the root zone. Traditional LDH-labels (see [[RFC5890](#)] for definitions of IDNA terms) in the root zone used only alphabetic characters (i.e., ASCII a-z or 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. In what follows we will use the Unicode notation; e.g., gc=Ll.

The IAB recommendation does, however, present some problems that need to be addressed. First, it is by no means clear that all of the code points with gc=Lo or gc=Lm and which are permitted under IDNA2008 are appropriate for the root zone. To take but one example, the code point U+02BC MODIFIER LETTER APOSTROPHE has gc=Lm. In practically every rendering (we are unaware of an exception), U+02BC is indistinguishable from U+2019 RIGHT SINGLE QUOTATION MARK, which has gc=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 gc=Po (Other_Punctuation). U+02BC is PROTOCOL VALID (PVALID) under IDNA2008 (see [[RFC5892](#)]), whereas both other code points are DISALLOWED. So, to begin with, it is plain that not every code point with gc in {Ll, Lo, Lm} is consistent with any conservatism principle.

To make matters worse, some languages are dependent on code points with gc=Mc (Spacing_Mark) or gc=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 only to code points with gc in {Ll, Lo, Lm} would be extremely limiting. While DNS labels are not words, or sentences, or phrases (as noted in [\[RFC4690\]](#)), they are intended as useful mnemonics. Mnemonics that diverge wildly from the usual conventions in a language are likely to attract strong objections, particularly in the root. The objections might drag the discussion away from sound management of the shared DNS root zone and towards discussions of cultural hegemony. That sort of discussion itself might present risks for the operation of the root zone.

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 come up with a set of principles for guiding decisions about code points. These principles can then function as meta-rules, determining the rules for inclusion of any code point (from those permitted by IDNA) in labels in the root. The principles might also be adopted by other zones that are shared by much of the Internet. Such a set of principles follows in the sections below. Each section includes remarks on the extent to which the principle could be wisely adopted by zones other than the root.

[1.1.](#) Terminology

Terms relevant to IDNA2008 can be found in [\[RFC5890\]](#). Other relevant internationalization terms are defined in [\[RFC6365\]](#).

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

[2.](#) Conservatism Principle

The root zone is, by definition, the one DNS zone that must be shared by everybody. Therefore, any decision to permit a code point in the root zone 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.

This principle is easily (and wisely) adoptable by any zone. It is also the one that is most likely to yield the safest result.

[3.](#) 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 the root zone should only permit inclusion of a code point if it is known to be safe. The default treatment of a code point should be that it is excluded.

This principle is easily (and wisely) adoptable by any zone.

[4.](#) 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 why a code point is included, but it does mean that the reason for inclusion of very peculiar code points, even if the code points are safe in themselves, will be too difficult to understand and will therefore be rejected.

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 very small subset of Unicode characters (for instance, only those needed to write a single 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.

[5.](#) 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 would all generally reach the same conclusion. This is not a requirement for algorithmic treatment of code points (the difficulties with the Unicode Letter and Mark categories illustrate why that would be too difficult). It is rather to say that the consistent application of professional judgment is likely to yield the same results; combined with the principle in [Section 2](#), when results are not predictable the anomalous code point would not be included.

Just as in [Section 4](#), this principle is not easily extended to zones lower than the root because what is predictable within a given language community is possibly very surprising across languages.

[6.](#) Stability Principle

Once a code point is permitted, it is at least very hard to stop permitting that code point. In general, 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 indicates that inclusion of such a code point is both safe and consistent with these principles.

This principle likely extends to every delegation-centric domain: if one delegation is permitted to use a code point, it is very hard to

see why others might not.

[7.](#) Letter Principle

In keeping with the spirit of the note in [\[RFC1123\]](#) that top-level labels "will be alphabetic", the rules should not include 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. But 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 the way letters are. Under this principle, code points with (for example) gc=Mn might be included -- but only those that are used to write words and not (for

instance) musical symbols. This principle should be applied as narrowly as possible; as [\[RFC4690\]](#) 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."

Because the root zone must be shared by everyone, this principle is more important in it than in zones that are intended for use by clearly-defined linguistic communities.

[8.](#) Conclusion

The foregoing principles could be applied generally when considering any range of Unicode code points for possible inclusion in the root zone. It is worth observing that doing anything (especially in light of [Section 6](#)) 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.

[9.](#) Security Considerations

The principles outlined in this memo are partly intended to reduce the possibility of 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.

[10.](#) IANA Considerations

None. RFC Editor: this section may be removed on publication.

[11.](#) Acknowledgements

The authors thank the participants in the IAB Internationalization

programme for the discussion of the ideas in this memo.

12. Informative References

[IABCOMM1]

Internet Architecture Board, "IAB Statement: 'The interpretation of rules in the ICANN gTLD Applicant Guidebook.'", February 2012.

[IABCOMM2]

Internet Architecture Board, "Response to ICANN questions concerning 'The interpretation of rules in the ICANN gTLD Applicant Guidebook'", March 2012.

[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.

Authors' Addresses

Andrew Sullivan
Dyn, Inc.
150 Dow St
Manchester, NH 03101
U.S.A.

Email: asullivan@dyn.com

Dave Thaler
Microsoft
One Microsoft Way
Redmond, WA 98052
U.S.A.

Email: dthaler@microsoft.com

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

Email: olaf@NLnetLabs.nl