DRAFT Status of work on IDNA2008

3/21/2009 0523 PDT

Vint Cerf

This brief summary is intended to provide some focus for the IDNABIS WG meetings scheduled for Monday and Tuesday, March 23 (1740-1940) and March 24 (0900-1130).

One goal is to try to assess rough consensus about the present documentation on the presumption that we are abiding by the ground-rules set forth in the charter of the WG. Another is to assess what the implications are for users, registries, registrars if IDNA2008 is adopted as it presently stands. A third goal is to examine the implications of the IDNAV2 proposal from Paul Hoffman and contrast with adoption of IDNA2008.

I fully recognize that consensus has to be assessed from mailing list exchanges, not merely from appearances at our face to face meetings.

The material presented below is by no means intended to be more than a basis for discussion, and is not intended as a penultimate recommendation.

Background

Consistent with the IDNABIS charter, the IDNA2008 design as it now stands makes several specific assumptions or makes specific propositions to achieve a number of goals:

0. Avoid dependence on any specific version of Unicode through the use of rules for determining PVALID characters based on Unicode character properties
1. No change to the deployed DNS server functionality (domain name labels limited to ASCII and case-insensitive matching only)
2. Esszet, Final Sigma, ZWJ and ZWNJ, geresh and gershayim are PVALID characters some of which are treated through contextual rules (there is still ongoing discussion about the implications of these choices)
3. Unassigned Unicode characters will not be looked up
4. No mapping of characters at least within the protocol specification
5. No modification of or dependence on Nameprep (and thus no impact on other protocols relying on Nameprep or Stringprep.)
6. Clear specification of valid "dot" form in a way that is consistent with DNS protocol requirements.
7. Symmetry between native-character ("Unicode") and ACE ("Punycode") forms of a label.
8. Conversion to an inclusion list of PVALID characters (as distinct from the IDNA2003 posture that excluded only a few Unicode characters)
9. Improved terminology to make categories and types of labels more clear. (Definitions)
10. Provide explanation for decisions and their motivations (Rationale) to aid implementors, registries, registrants and users in understanding IDNA.
11. Separately describe registration and lookup procedures to improve clarity

12. Specify tests to be applied at lookup time in an attempt to limit abuse of IDNA at all levels of registration
13. Clarify what is expected of IDNA-aware applications and domain name "slots" with regard to invalid labels and future extensibility

Chartering and Re-Chartering

(1) A Re-charter is needed if we abandon a significant fraction of the IDNA2008 goals and methods. IDNAv2, as described by Paul Hoffman requires a re-charter.

(2) A Re-charter is needed if the WG decides to introduce mappings into the IDNA2008 specifications since the basic assumption in IDNA2008 was that mapping would not be part of the specification.

(3) It is possible that re-charter might not be needed if IDNA2008 adopts some IDNA2003 operations under a restricted set of conditions and only at lookup time for purposes of easing the transition to IDNA2008. This would be up to the AD and IESG presumably to decide.

Basics for IDNA2003 and IDNA2008

Both of these specifications use the Punycode algorithm to generate what IDNA2008 would call an A-label (ie. "xn-- <LDH compliant string>") from labels expressed as a string of characters drawn from a subset of Unicode defined characters.

DNS matching is done in the servers by comparing the query string to the registered string in a case-independent fashion. For IDNs, these comparisons are done after conversion into the "xn--" prefix form. For IDNs the case insensitive matching of the DNS servers applies only to the A-label form and not to the Unicode form. This means that the case-insensitive matching behavior of in traditional ASCII labels is not conferred on IDNs in their Unicode form.

The case-insensitive comparisons between traditional LDH domain names is approximated under IDNA2003 by using CaseFold as a mapping guide on the Unicode strings being looked up. In addition, IDNA2003 also maps the so-called "compatibility" characters of Unicode into their counterparts. The same actions precede the registration of new domain names under IDNA2003.

Unicode CaseFold maps to upper case and then map back to lower case. Prior to Unicode 5.0, Ezsett became "SS" because there was no upper case, then became "ss" in the lower case mapping. Under Unicode 5.0 CaseFold was unchanged for stability reasons. Consequently CaseFold (ESSZETT) is "ss" rather than lower case esszett even after the introduction of upper case ESSZETT in Unicode 5.0.

Under IDNA2003, UNASSIGNED characters are looked up. If abusive registrations are made using UNASSIGNED characters, these registered domain names may be be found on lookup by IDNA2003-compliant clients.

Under IDNA2008, UNASSIGNED and DISALLOWED characters are not looked up.

If new characters become defined under a new version of Unicode an old client will not look them up until it is updated. Abusive registrations using UNASSIGNED characters will not be looked up.

Script mixing is permitted under IDNA2003. Under IDNA2008, BiDi bans mixing of European and Extended Arabic-Indic numbers with Arabic numbers. That is AN and EN characters may not be present in the same label. Otherwise, mixing is permitted in IDNA2008.

IMPLICATIONS OF ADOPTING IDNA2008 AS CURRENTLY SPECIFIED

1. IDNA2008 is case sensitive for labels with non-LDH characters in them but is case-insensitive for LDH characters

for example" buecher "is all ASCII and could be matched with "Buecher" or "bUecher" under IDNA2008

however "B<u-umlaut>cher" would not be allowed because Tables (see 4.2.2) would disallow Latin Capital letters. Some users accustomed to LDH-label behavior may be surprised that "B<u-umlaut>cher" and "b<u-umlaut>cher" do not match.

On the other hand, the symmetric relationship between the IDNA2008-defined A-Label and U-Label has the benefit one can use exact match for either U-label form or A-label forms since they are directly and unambiguously transformable into each other. However, this symmetry will not exist for cases where the IDNA2003 A-Label and IDNA2008 A-label for the same U-Label differ. [Query: will this be a material problem only for actual registrations under IDNA2003 that differ in A-label form from IDNA2008?]

2. IDNA2008 does not ban script mixing even within labels.

Attempts to fashion rules along these lines have run into problems in which characters that may be confused for others are needed to express strings in particular languages. The International Phonetic Alphabet (IPA) characters are a case in point. Some are used for certain (e.g. African) languages but some of these characters can be confused for others in the Latin alphabet. Other examples exist in Arabic, Cyrillic, Greek among others.

Even in the absence of intra-label script mixing, inter-script confusion such as the Russian word for "restaurant" looking like "pectopah" in Latin characters is quite possible.

Despite the apparent desirability of such a ban at protocol level, there are simply too many combinations of confusion within-scripts and between scripts to benefit significantly from a protocol-level ban. On the other hand, registry level constraints that may be more script-aware appear to be the most effective tool we have.

3. Esszet is permitted and its usage appears to be geographically and language specific. Under IDNA2003, this character is mapped into "ss". To deal with the potential conflict with previously mapped registrations in which Esszet is mapped to "ss" registries would need to appeal to Rationale 7.2 options, for example, to deal with this. Note that not all collisions may be a consequence of mapping, i.e., many occurrences of "ss" in German text are not typographic variations of Esszett and very few occurrences in Latin script, without consideration of language, are variations of Esszett either.

4. Final Sigma is permitted and raises similar issues to Esszet with regard to collisions and the same remedies would apply.

5. ZWJ/ZWNJ

In IDNA2003, these characters were mapped to "nothing". These characters and others that are mapping to nothing, are required for various scripts and languages. Persian registries currently reject registration of labels including ZWJ/ZWNJ. ZWNJ is used in writing Persian languages.

Arabic languages do not need ZWJ/ZWNJ.

Mapping to "nothing" in IDNA2003 has the side-effect of creating homonyms in some scripts and languages (eg. Tamil and Devanagari) where the same string with and without the mapped characters(s) have two distinct meanings. When converting a DNS label that has characters that map to "nothing", and/or characters that map to other strings, one cannot tell whether then label, when converted back to native character form, was intended to be written with ZWJ, ZWNJ or neither.

Elaboration: Suppose that "ab" is a string in one of the scripts in which we now propose to permit ZWNJ. All we have in the DNS is the A-label equivalent of "ab". We can't tell from looking at it whether the starting string, as seen/preferred by the registrant, was
  ab   or
  aZWJNb
since both map to the same A-label.

Under IDNA2008, if the user enters "ab", she gets one A-label while, if she enters "aZWNJb", she gets a different A-label. That is exactly the same as the Eszett problem -- you can't tell from the IDNA2003 A-label what the original intention was and use of the string under IDNA2008 gets you a different A-label than it does under IDNA2003.

Joiner characters become invisible if inserted in strings written in scripts that do not use them. Unicode classifies these characters as "COMMON" so they also end up passing any plausible tests to prevent mixing of scripts in a label. IDNA2008 uses contextual rules to restrict their use to strings in scripts where they have some effect. IDNA2003 maps these characters to "nothing." Under IDNA2008 we end up relying on registries to adopt their use judiciously within those scripts. See also the Rationale document for further commentary.

6. Symbols and punctuation are NOT PVALID under IDNA2008 but are valid under IDNA2003 leading to a variety of potential confusions with "slash-like" symbols or other symbols used in URIs for example. IDNA2008 rules reduce confusion potential by making all characters with these Unicode properties invalid for use with Domain labels. These symbols are not needed for domain names.

Another reason for banning these characters is that they complicate references, discussions and databases (such as WHOIS) because it is not clear how to describe them in common, informal usage.
Many symbols have multiple characters matching informal usage. For example, there are many symbol characters that one would describe as "heart" or "bullet point."

This problem, which exists in both IDNA2003 and IDNA2008 can be ameliorated by using the "U+" form but for most users of the Internet, who are not familiar with Unicode conventions, such references are not likely to be meaningful.

This restriction does not completely eliminate all forms of confusion as both IDNA2008 and IDNA2003 allow some characters that can be confused owing to fonts used, etc.

7. JAMO characters in Korean have been made Protocol Invalid (DISALLOWED) for reasons similar to (6) above. They introduce a combinatorial explosion of different string representations built from JAMO primitive characters. They are valid under IDNA2003.

8. Under IDNA2008, when a new version of Unicode is released the following steps can be taken:

a. review of changes that might require new rules in the IDNA2008 framework. Such a conclusion would assuredly require formation of a WG to facilitate new RFC production. Unicode experts believe this to be extremely unlikely to happen.

b. A review of changes might only require exception rules to preserve compatibility. It is possible that the required changes might be delegated to an IANA action possibly in consultation with an expert committee to generate new tables.

c. Generate new tables for IANA registry (suitable for downloading as needed

After the first new version of Unicode, after IDNA2008 is standardized, some clients and some registries will have tables that are not current.
Lookups of Domain Names containing new PVALID characters by clients using out of date tables will fail under IDN2008 because the client will reject UNASSIGNED characters until the clients are updated with the new PVALID characters.

9. Applications that allow entry of combining characters may need revision

In IDNA2003, the label "<e-acute>xample" could be entered in an application either as "<e-acute>xample" of <combining-acute>example" and would resolve to the same label: "xn--xample-9ua." Adoption of IDNA2008 would require such applications to pre-process the second entry or the APIs and GUI elements of the operating system that process string entry would need to be altered to perform the mapping, if it is concluded that the behavior under IDNA2003 needs to be preserved.

"IDNAV2" - cf: draft-hoffman-idna2-01.txt

In this proposal, IDNA2003 would be updated by adding new characters added in versions of the Unicode Standard between version 3.2 and the current version.
Under IDNA2003 and IDNAv2, mapping based on CaseFold and mapping of compatibility characters is carried out prior to registration and lookup.

All the properties of IDNA2003 apply including the Nameprep profile of Stringprep.

1. To pursue this proposal formally, the proposed charter change would have to be shown to have community consensus and then approved by the AD and IESG because it diverges from assumptions in the IDNA2008 charter.

2. New Unicode versions require new standards-track RFCs to adopt new specifications because the tables in IDNAV2 make references to specific Unicode versions.

3. As a practical matter, the proposal means that IDNA needs to be revised whenever new versions of the Unicode Standard add characters that are deemed to be needed in domain names. Each release of Unicode would need to be evaluated to determine whether IDNA revision is required.

4. A sequence of changes/additions to allowed characters would require examination of NamePrep and StringPrep which are currently defined in terms of Unicode 3.2). Since many other protocols (including security) rely on Stringprep and possibly on Nameprep, changes could have significant ripple effects.

A different view:

"The other protocols are based on a particular version of Stringprep, and therefore the is no ripple effect in updating Stringprep or Nameprep unless those protocols want them. "Changes in IDNAv2, and future versions of IDNA, will change Stringprep and Nameprep. Developers of other protocols that rely on these two standards will need to decide whether or not they want to update their standards to use the new versions."

5. JAMO are allowed under IDNA2003. A strong recommendation has been made by Korean language experts to disallow these characters.

------------------ Questions for discussion ----------------

A. Multiple characters are allowed as "dots" in domain names under IDNA2003 and presumably under IDNAV2. This is a general problem for all versions of IDNA but may be exacerbated by the variants for "dots" that are permitted under IDNA2003 and IDNAv2. What is the WG view?

B. There are few if any restrictions on the lookup phase of IDNAv2 (and IDNA2003). The consequences are that lookup will match domain names injected into DNS by registries that are non-conformant with registration restrictions intended by the protocol specification. This condition arises from permitting the looking up of DISALLOWED or UNASSIGNED characters. How serious a problem is this in the view of the WG?