IETF IDN Working Group
Internet Draft
10th Mar 2000

James Seng

draft-ietf-idn-requirements-01.txt
Expires 10th Mar 2000

Requirements of Internationalized Domain Names

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

Abstract

This document describes the requirement for encoding international characters into DNS names and records. This document is guidance for developing protocols for internationalized domain names.

1. Introduction

At present, the encoding of Internet domain names is restricted to a subset of 7-bit ASCII (ISO/IEC 646). HTML, XML, IMAP, FTP, and many other text based items on the Internet have already been internationalized. It is important for domain names to be similarly internationalized.

This document is being discussed on the "idn" mailing list. To join the list, send a message to <majordomo@ops.ietf.org> with the words "subscribe idn" in the body of the message. Archives of the mailing list can also be found at ftp://ops.ietf.org/pub/lists/idn*.

1.1 Definitions and Conventions

The key words "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].

"IDN" is used in this document as an abbreviation for "internationalized domain name". This is defined as a domain name that contains one or more characters that are outside the set of characters specified as legal

characters for domain names in [RFC1034] Section 3.5 and [RFC1123].

It is important to note the difference between domain name and host name. Current domain names has no restriction on what is legal character (8bit). The only restrictions are the total and label lengths. Host name on the other hand are restricted to alphanumeric and '-' case insensitive with "." only allowed between labels.

A master server for a zone holds the main copy of that zone. This copy is sometimes stored in a zone file. A slave server for a zone holds a complete copy of the records for that zone. A caching server holds temporary copies of DNS records; it uses records to answer queries about domain names. Further explanation of these terms can be found in [RFC1034] and [RFC1996].

Characters mentioned in this document are identified by their position in the Unicode character set. The notation U+12AB, for example, indicates the character at position 12AB (hexadecimal) in the Unicode character set. Note that the use of this notation is not an indication of a requirement to use Unicode.

Examples quoted in this document should be considered as a method to further explain the meanings and principles adopted by the document. It is not a requirement for the protocol to satisfy the examples.

A character is a member of a set of elements used for organization, control, or representation of data.

A coded character is a character with its coded representation.

A coded character set ("CCS") is a set of unambiguous rules that establishes a character set and the relationship between the characters of the set and their coded representation.

A graphic character or glyph is a character, other than a control function, that has a visual representation normally handwritten, printed, or displayed.

A character encoding scheme or "CES" is a mapping from one or more coded character sets to a set of octets. Some CESs are associated with a single CCS; for example, UTF-8 applies only to ISO 10646. Other CESs, such as ISO 2022, are associated with many CCSs.

A charset is a method of mapping a sequence of octets to a sequence of abstract characters. A charset is, in effect, a combination of one or more CCS with a CES. Charset names are registered by the IANA according to procedures documented in RFC 2278.

A language is a way that humans interact. In written form, a language is expressed in characters. The same set of characters can often be used in many languages, and many languages can be expressed using different scripts. A particular charset may have different glyphs (shapes) depending on the language being used.

2. General Requirements

2.1 Compatibility and Interoperability

The DNS is essential to the entire Internet. Therefore, the protocol must not damage present DNS protocol interoperability. It must make the minimum number of changes to existing protocols on all layers of the stack. It must continue to allow any system anywhere to resolve any Internet internationalized domain name.

The protocol must preserve the basic concept and facilities of domain names as described in [RFC1034]. It must maintain a single, global, universal, and consistent hierarchical namespace.

The same name resolution request must generate the same response, regardless of the location or localization settings in the resolver, in the master server, and in any slave servers involved in the resolution process.

If the protocol allows more than one charset, it should also allow creation of caching servers that do not understand the charset in which a request or response is encoded. Such caching servers should work as well for IDNs as they do for current domain names. The caching server performs correctly if it gives the essentially the same answer (without the authoritative bit) as the master server would have if presented with the same request.

A caching server must not return data in response to a query that would not have been returned if the same query had been presented to an authoritative server. This applies fully for the cases when:

- The caching server does not know about IDN
- The caching server implements the whole specification
- The caching server implements a legal subset of the specification

The protocol should be able to be upgraded at any time with new features

and retain backwards compatibility with the current specification.

The protocol may modify the DNS protocol [RFC1035] and other related work undertaken by the DNSEXT WG. However, these changes should be as small as possible and any changes must be approved by the DNSEXT WG.

The protocol should be as simple as possible from the user's perspective. Ideally, users should not realize that IDN was added on to the existing DNS.

A fall-back strategy or mechanism based upon ASCII may be needed during a transition period during deployment and adoption of IDN. Therefore, if an encoding is not mapped into ASCII, then there should be an ASCII-only representation compatible with the current DNS and there should be a way for a program to find the ASCII-only representation for IDN.

The best solution is one that maintains maximum feasible compatibility with current DNS standards as long as it meets the other requirements

in this document.

2.2 Internationalization

Internationalized characters must be allowed to be represented and used in DNS names and records. The protocol must specify what charset is used when resolving domain names and how characters are encoded in DNS records.

This document does not recommend any charset for I18N. If more than one charset is used in the protocol, then the protocol must specify all the charsets being used and for what purpose. A CCS(s) chosen must at least cover the range of characters as currently defined (and as being added) by ISO 10646/Unicode.

CES(s) chosen should not encode ASCII characters differently depending on the other characters in the string. In other words, ASCII character should remain as specified in [US-ASCII].

The protocol must not invent a new CCS for the purpose of IDN only and should use existing CES. The charset(s) chosen should also be non-ambiguous.

The protocol should not make any assumptions where in a domain name that internationalization might appear. In other words, it should not differentiate between any part of a domain name because this may impose a restriction on future internationalization efforts.

The protocol should also not make any localized restrictions in the

protocol. For example, an IDN implementation which only allows domain names to use a single local script would immediately restrict multinational organization.

Because of the wide range of devices that use the DNS and the wide range of characteristics of international scripts, the protocol should allow more than one method of domain name input and display. However, there has to be a single way of encoding an internationalized domain name within the core of the DNS.

2.3 Localization

The protocol must be able to handle localized requirement of different languages. For example, IDN must be able to handle bi-directional writing for scripts such as Arabic.

Historically, "." has been the separator of labels in the host names. The protocol should not use different separators for different languages.

Most localization can be handled by the user interface. It should not matter how the domain names are input or presented, such as in a reverse order or bi-directional, or with the introduction of a new separator. However, the final wire format must be in canonical order.

2.4 Canonicalization

Matching rules are a complicated process for IDN. Canonicalization of characters must follow precise and predictable rules to ensure consistency. [CHARREQ] is a recommended as a guide on canonicalization.

The DNS has to match a host name in a request with a host name held in one or more zones. It also needs to sort names into order. It is expected that some sort of canonicalization algorithm will be used as the first step of this process. This section discusses some of the properties which will be required of that algorithm.

The canonicalization algorithm might specify operations for case, ligature, and punctuation folding.

In order to retain backwards compatibility with the current DNS, the protocol must retain the case-insensitive comparison for US-ASCII as specified in [RFC1035]. For example, Latin capital letter A (U+0041) must match Latin small letter A (U+0061). [UTR-21] describes some of the issues with case mapping.

Case folding must not be locale dependent. For example, Latin capital

letter I (U+0049) case folded to lower case in the Turkish context will become Latin small letter dotless I (U+0131). But in the English context, it will become Latin small letter I (U+0069).

If other canonicalization is done, then it must be done before the domain name is resolved. Further, the canonicalization must be easily upgrade able as new languages and writing systems are added.

Any conversion (case, ligature folding, punctuation folding, ...) from what the user enters into a client to what the client asks for resolution must be done identically on all requests from any client.

If the protocol specifies a canonicalization algorithm, a caching server should perform correctly regardless of how much (or how little) of that algorithm it has implemented. [1 request to remove]

If the protocol requires a canonicalization algorithm, all requests sent to a caching server must already be in the canonical form.

If the charset can be normalized, then it should be normalized before it is used in IDN. (conflict)

The protocol should avoid inventing a new normalization form provided a technically sufficient one is available (such as in an ISO standard).

2.5 Operational Issues

Zone files should remain easily editable.

An IDN-capable resolver or server should not generate more traffic than a non-IDN-capable resolver or server would when resolving an ASCII-only domain name. The amount of traffic generated when resolving an IDN

should be similar to that generated when resolving an ASCII-only name.

The protocol should add no new centralized administration for the DNS. A domain administrator should be able to create internationalized names as easily as adding current domain names.

Within a single zone, the zone manager must be able to define equivalence rules that suit the purpose of the zone, such as, but not limited to, and not necessarily, non-ASCII case folding, Unicode normalizations, Cyrillic/Latin folding, or traditional/simplified Chinese equivalence. Such defined equivalences must not remove equivalences that are assumed by (old or local-rule-ignorant) caches.

The character set of a signed zone file should be capable of being the same as the character set of the unsigned zone file. The protocol must

allow offline DNSSEC signing. It should be possible to look at the signed file and see that it is the same as the unsigned one.

2.6 Others

The protocol may provide the same DNS resources using internationalized text as it currently provides using ASCII text.

To get full semantics for IDN, an upgrade of the DNS and related software may be needed.

The protocol should consider new features of DNS such as DNSSEC and DNAME. For example, DNAME might be useful to simplify canonicalization for IDN.

3. Technical Analysis

There are many standard protocols and RFCs which are depend on domain names and have make various assumptions about the characters in them always conforming to [RFC-1034]. We expect that the protocols listed below to be affected:

<...list the sets of RFCs which we would like to have an summary...>
RFC821, RFC822, ...

All idn protocol documents must fully detail the expected effects of leaking of the specified encoding to protocols other than the DNS resolution protocol. They must also contain a summary of the technical opinions of the IDN Working Group.

4. Security Considerations

Any solution that meets the requirements in this document must not be less secure than the current DNS. Specifically, the mapping of internationalized host names to and from IP addresses must have the same characteristics as the mapping of today's host names.

Specifying requirements for internationalized domain names does not itself raise any new security issues. However, any change to the DNS

may affect the security of any protocol that relies on the DNS or on DNS names. A thorough evaluation of those protocols for security concerns will be needed when they are developed. In particular, IDNs must be compatible with DNSSEC.

5. References

[CHARREQ] "Requirements for string identity matching and String

Indexing", http://www.w3.org/TR/WD-charreq, July 1998, World Wide Web Consortium.

[DNSEXT] "IETF DNS Extensions Working Group", namedroppers@internic.net, Olafur Gudmundson, Randy Bush.

[RFC1034] "Domain Names - Concepts and Facilities", rfc1034.txt, November 1987, P. Mockapetris.

[RFC1035] "Domain Names - Implementation and Specification", rfc1035.txt, November 1987, P. Mockapetris.

[RFC1123] "Requirements for Internet Hosts -- Application and Support", rfc1123.txt, October 1989, R. Braden.

[RFC1996] "A Mechanism for Prompt Notification of Zone Changes (DNS NOTIFY)", rfc1996.txt, August 1996, P. Vixie.

[RFC2119] "Key words for use in RFCs to Indicate Requirement Levels", rfc2119.txt, March 1997, S. Bradner.

[UNICODE] The Unicode Consortium, "The Unicode Standard -- Version 3.0", ISBN 0-201-61633-5. Described at http://www.unicode.org/unicode/standard/versions/Unicode3.0.html

[US-ASCII] Coded Character Set -- 7-bit American Standard Code for Information Interchange, ANSI X3.4-1986.

[UTR15] "Unicode Normalization Forms", Unicode Technical Report #15, http://www.unicode.org/unicode/reports/tr15/, Nov 1999, M. Davis & M. Duerst, Unicode Consortium.

[UTR21] "Case Mappings", Unicode Technical Report #21, http://www.unicode.org/unicode/reports/tr21/, Dec 1999, M. Davis, Unicode Consortium.

Appendix A. Acknowledgements

The editor gratefully acknowledges the contributions of:

Harald Tveit Alvestrand <Harald@Alvestrand.no>
Martin Duerst <duerst@w3.org>
Patrik Faltstrom <paf@swip.net>
Andrew Draper <ADRAPER@altera.com>
Bill Manning <bmanning@ISI.EDU>

Paul Hoffman <phoffman@imc.org>

James Seng <jseng@pobox.org.sg>
Randy Bush <randy@psg.com>
Alan Barret <apb@cequrux.com>
Olafur Gudmundsson <ogud@tislabs.com>
Karlsson Kent <keka@im.se>
Dan Oscarsson <Dan.Oscarsson@trab.se>
J. William Semich <bill@mail.nic.nu>
RJ Atkinson <request not to have email>
Simon Josefsson <jas+idn@pdc.kth.se>
Ned Freed <ned.freed@innosoft.com>
Dongman Lee <dlee@icu.ac.kr>
Mark Andrews <Mark.Andrews@nominum.com>