

## **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 at least partially internationalized. It is important for domain names to be similarly internationalized or for an equivalent solution to be found. This document assumes that the most effective solution involves putting non-ASCII names inside some parts of the overall DNS system.

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\\*](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](#)].

Characters mentioned in this document are identified by their position in the Unicode [[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 [[RFC2279](#)] 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 [[RFC2278](#)].

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.

## **[1.2](#) Description of the Domain Name System**

The Domain Name System is defined by [[RFC1034](#)] and [[RFC1035](#)], with clarifications, extensions and modifications given in [[RFC1123](#)], [[RFC1996](#)], [[RFC2181](#)], and others. Of special importance here is the security extensions described in [[RFC2535](#)] and companions.

Over the years, many different words have been used to describe the components of resource naming on the Internet (e.g., [URI], [URN]); to make certain that the set of terms used in this document are well-defined and

non-ambiguous, the definitions are given here.

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. Slave servers MAY be either authorized by the zone owner (secondary servers) or unauthorized (so-called "stealth secondaries"). Master and authorized slave servers are listed in the NS records for the zone, and are termed "authoritative" servers. In many contexts, outside this document the term "primary" is used interchangeably with "master" and "secondary" is used interchangeably with "slave".

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](#)].

DNS names can be represented in multiple forms, with different properties for internationalization. The most important ones are:

- Domain name: The binary representation of a name used internally in the DNS protocol. This consists of a series of components of 1-63 octets, with an overall length limited to 255 octets (including the length fields).
- Master file format domain name: This is a representation of the name as a sequence of characters in some character sets; the common convention (derived from [[RFC1035](#)] [section 5.1](#)) is to represent the octets of the name as ASCII characters where the octet is in the set corresponding to the ASCII values for [a-zA-Z0-9-], using an escape mechanism (\x or \NNN) where not, and separating the components of the name by the dot character (".").

The form specified for most protocols using the DNS is a limited form of the master file format domain name. This limited form is defined in [[RFC1034](#)] [Section 3.5](#) and [[RFC1123](#)]. In most implementations of applications today, domain names in the Internet have been limited to the much more restricted forms used, e.g., in email. Those names are limited to the ASCII upper and lower-case characters (interpreted in a case-independent fashion), the digits, and the hyphen.

### **[1.3](#) Definition of "hostname" and "Internationalized Domain Name"**

In the DNS protocols, a name is referred to as a sequence of octets. However, when discussing requirements for internationalized domain names, what we are looking for are ways to represent characters that are meaningful for humans.

In this document, this is referred to as a "hostname". While this term has been used for many different purposes over the years, it is used here in the sense of "sequence of characters (not octets) representing a domain name conforming to the limited hostname syntax".

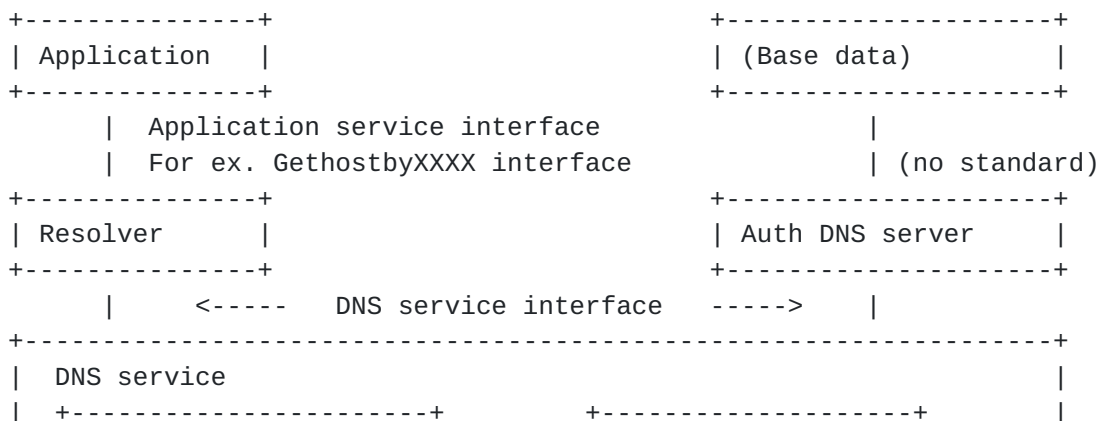
This document attempts to define the requirements for an "Internationalized Domain Name" (IDN). This is defined as a sequence of characters that can be used in the context of functions where a hostname is used today, but contains one or more characters that are outside the set of characters specified as legal characters for host names.

#### **1.4 A multilayer model of the DNS function**

The DNS can be seen as a multilayer function:

- The bottom layer is where the packets are passed across the Internet in a DNS query and a DNS response. At this level, what matters is the format and meaning of bits and octets in a DNS packet.
- Above that is the "DNS service", created by an infrastructure of DNS servers, NS records that point to those DNS servers, that is pointed to by the root servers (listed in the "root cache file" on each DNS server, often called "named.cache". It is at this level that the statement "the DNS has a single root" [[RFC2826](#)] makes sense, but still, what are being transferred are octets, not characters.
- Interfacing to the user is a service layer, often called "the resolver library", and often embedded in the operating system or system libraries of the client machines. It is at the top of this layer that the API calls commonly known as "gethostbyname" and "gethostbyaddress" reside. These calls are modified to support IPv6 [[RFC2553](#)]. A conceptually similar layer exists in authoritative DNS servers, comprising the parts that generate "meaningful" strings in DNS files. Due to the popularity of the "master file" format, this layer often exists only in the administrative routines of the service maintainers.
- The user of this layer (resolver library) is the application programs that use the DNS, such as mailers, mail servers, Web clients, Web servers, Web caches, IRC clients, FTP clients, distributed file systems, distributed databases, and almost all other applications on TCP/IP.

Graphically, one can illustrate it like this:





users (the application service), and the protocol extensions, if needed, added to support this service.

In the requirements, we attempt to use the term "service" whenever a requirement concerns the service, and "protocol" whenever a requirement is believed to constrain the possible implementation.

## **2.1 Compatibility and Interoperability**

[1] The DNS is essential to the entire Internet. Therefore, the service 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 internationalized domain name.

[2] The service 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.

[2.5] The DNS service layer (the packet formats that go on the wire) MUST NOT limit the codepoints that can be used. This interface SHOULD NOT assign meaning to name strings; the application service layer, where "gethostbyname" et al reside, MAY constrain the name strings to be used in certain services. (conflict)

[3] 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.

[4] The protocol SHOULD allow creation of caching servers that do not understand the charset in which a request or response is encoded. The caching server SHOULD perform correctly for IDN as well as for current domain names (without the authoritative bit) as the master server would have if presented with the same request.

[5] 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 valid subset of the specification

[7] The service 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 SHOULD be coordinated with the [[DNSEXT](#)] WG.

[8] The protocol supporting the service 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.

[10] 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](#)**

[11] 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.

[12] This document RECOMMENDS Unicode only. If multiple charsets are allowed, each charset MUST be tagged and conform to [[RFC2277](#)].

[12.5] IDN MUST NOT return illegal code points in responses, SHOULD reject queries with illegal codepoints. (one request to add; one request to remove)

[13] CES(s) chosen SHOULD NOT encode ASCII characters differently depending on the other characters in the string. In other words, unless IDN names are identified and coded differently from ASCII-only ones, characters in the ASCII set SHOULD remain as specified in [[US-ASCII](#)] (one request to remove).

[14] The protocol SHOULD 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.

[15] The protocol SHOULD NOT make any assumptions about the location in a domain name where internationalization might appear. In other words, it SHOULD NOT differentiate between any part of a domain name because this MAY impose restrictions on future internationalization efforts.

[16] The protocol also SHOULD 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.

[17] While there are a wide range of devices that use the DNS and a wide range of characteristics of international scripts and methods of domain name input and display, IDN is only concerned with the protocol. Therefore, there MUST be a single way of encoding an internationalized domain name within the DNS.

[18] The protocol SHOULD NOT place any restrictions on the application service layer. It SHOULD only specify changes in the DNS service layer and within the DNS itself.

## **[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 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.

[22] To achieve interoperability, canonicalization MUST be done at a single well-defined place in the DNS resolution process. The protocol MUST specify canonicalization; it MUST specify exactly where in the DNS that canonicalization happens and does not happen; it MUST specify how additions to ISO 10646 will affect the stability of the DNS and the amount of work done on the root DNS servers.

[23] The canonicalization algorithm MAY specify operations for case, ligature, and punctuation folding.

[24] In order to retain backwards compatibility with the current DNS, the service 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). [[UTR21](#)] describes some of the issues with case mapping. Case-insensitivity for non [[US-ASCII](#)] MUST be discussed in the protocol proposal.

[25] Case folding MUST be locale independent. 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).

[26] If other canonicalization is done, it MUST be done before the domain name is resolved. Further, the canonicalization MUST be easily upgradable as new languages and writing systems are added.

[27] Any conversion (case, ligature folding, punctuation folding, etc) from what the user enters into a client to what the client asks for resolution MUST be done identically on any request from any client.

[30] If the charset can be normalized, then it SHOULD be normalized before it is used in IDN. Normalization SHOULD follow [[UTR15](#)].  
(conflict)

[31] The protocol SHOULD avoid inventing a new normalization form provided a technically sufficient one is available.

## **[2.5 Operational Issues](#)**

[32] Zone files SHOULD remain easily editable.

[33] An IDN-capable resolver or server SHALL 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 SHALL be similar to that generated when resolving an ASCII-only name.

[34] The service SHOULD NOT add new centralized administration for the DNS. A domain administrator SHOULD be able to create internationalized names as easily as adding current domain names.

[35] 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 (if Unicode is chosen), Cyrillic/Greek/Latin folding, or traditional/simplified Chinese equivalence. Such defined equivalences MUST NOT remove equivalences that are assumed by (old or local-rule-ignorant) caches.

[36] The protocol MUST work with DNSSEC.

[37] The protocol MUST work for all features of DNS, IPv4, and IPv6.

#### **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 and, if multiple charsets or representation forms are permitted, the implications of this name-spoof MUST be thoroughly understood.

#### **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](mailto: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.
- [RFC2181] "Clarifications to the DNS Specification", [rfc2181.txt](#), July 1997, R. Elz, R. Bush.
- [RFC2277] "IETF Policy on Character Sets and Languages", [rfc2277.txt](#), January 1998, H. Alvestrand.
- [RFC2278] "IANA Charset Registration Procedures", [rfc2278.txt](#), January 1998, N. Freed and J. Postel.
- [RFC2279] "UTF-8, a transformation format of ISO 10646", [rfc2279.txt](#), F. Yergeau, January 1998.
- [RFC2535] "Domain Name System Security Extensions", [rfc2535.txt](#), March 1999, D. Eastlake.
- [RFC2553] "Basic Socket Interface Extensions for IPv6", [rfc2553.txt](#), March 1999, R. Gilligan et al.
- [RFC2825] "A Tangled Web: Issues of I18N, Domain Names, and the Other Internet protocols", [rfc2825.txt](#), May 2000, L. Daigle et al.
- [RFC2826] "IAB Technical Comment on the Unique DNS Root", [rfc2826.txt](#), May 2000, Internet Architecture Board.
- [IDNCOMP] "Comparison of Internationalized Domain Name Proposals", [draft-ietf-idn-compare-00.txt](#), June 2000, P. Hoffman.
- [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. Approved status.

## **6. Editors' Contact**

Zita Wenzel, Ph.D.  
Information Sciences Institute  
University of Southern California  
**4676 Admiralty Way**  
Marina del Rey, CA  
**90292 USA**  
Tel: +1 310 448 8462  
Fax: +1 310 823 6714  
zita@isi.edu

James Seng  
**8 Temesek Boulevard**  
#24-02 Suntec Tower 3  
Singapore 038988  
Tel: +65 248 6208  
Fax: +65 248 6198  
Email: jseng@pobox.org.sg

## **7. Acknowledgements**

The editors gratefully acknowledge the contributions of:

Harald Tveit Alvestrand <Harald@Alvestrand.no>  
Mark Andrews <Mark.Andrews@nominum.com>  
RJ Atkinson <request not to have email>  
Alan Barret <apb@cequrux.com>  
Randy Bush <randy@psg.com>  
Andrew Draper <ADRAPER@altera.com>  
Martin Duerst <duerst@w3.org>  
Patrik Faltstrom <paf@swip.net>  
Ned Freed <ned.freed@innosoft.com>  
Olafur Gudmundsson <ogud@tislabs.com>  
Paul Hoffman <phoffman@imc.org>  
Simon Josefsson <jas+idn@pdc.kth.se>  
Karlsson Kent <keka@im.se>  
John Klensin <klensin+idn@jck.com>  
Tan Juay Kwang <tanj@i-dns.net>  
Dongman Lee <dlee@icu.ac.kr>  
Bill Manning <bmann@ISI.EDU>  
Dan Oscarsson <Dan.Oscarsson@trab.se>  
**J. William Semich** <bill@mail.nic.nu>  
James Seng <jseng@pobox.org.sg>