

Internet Draft  
[draft-ietf-idn-idne-01.txt](#)  
July 8, 2000  
Expires in six months

Marc Blanchet  
Viagenie  
Paul Hoffman  
IMC & VPNC

## Internationalized domain names using EDNS (IDNE)

### 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

The current DNS infrastructure does not provide a way to use internationalized domain names (IDN). This document describes an extension mechanism based on EDNS which enables the use of IDN without causing harm to the current DNS. IDNE enables IDN host names with as many characters as current ASCII-only host names. It fully supports UTF-8 and conforms to the IDN requirements.

## **[1. Introduction](#)**

Various proposals for IDN have tried to integrate IDN into the current limited ASCII DNS. However, the compatibility issues make too many constraints on the architecture. Many of these proposals require modifications to the applications or to the DNS protocol or to the servers. This proposal take a different approach: it uses the standardized extension mechanism for DNS (EDNS) and uses UTF-8 as the mandatory charset. It causes no harm to the current DNS because it uses the EDNS extension mechanism. The major drawback of this proposal is that all protocols, applications and DNS servers will have to be upgraded to support this proposal.

### **[1.1 Terminology](#)**

The key words "MUST", "SHALL", "REQUIRED", "SHOULD", "RECOMMENDED", and "MAY" in this document are to be interpreted as described in [RFC 2119](#) [[RFC2119](#)].

Hexadecimal values are shown preceded with an "0x". For example, "0xa1b5" indicates two octets, 0xa1 followed by 0xb5. Binary values are shown preceded with an "0b". For example, a nine-bit value might be shown as "0b101101111".

Examples in this document use the notation from the Unicode Standard [UNICODE3] as well as the ISO 10646 [ISO10646] names. For example, the letter "a" may be represented as either "U+0061" or "LATIN SMALL LETTER A". In the lists of prohibited characters, the "U+" is left off to make the lists easier to read.

## 1.2 IDN summary

Using the terminology in [[IDNCOMP](#)], this protocol specifies an IDN architecture of arch-2 (send binary or ACE). The binary format is bin-1.1 (UTF-8), and the method for distinguishing binary from current names is bin-2.4 (mark binary with EDNS0). The transition period is not specified.

## 2. Functional Description

DNS query and responses containing IDNE labels have the following properties:

- The string in the label MUST be pre-processed as described in [\[NAMEPREP\]](#) before the query or response is prepared.
- The characters in the label MUST be encoded using UTF-8 [\[RFC2279\]](#).
- The entire label MUST be encoded EDNS [\[RFC2671\]](#).
- The version of the IDN protocol MUST be identified.

### 3. Encoding

An IDNE label uses the EDNS extended label type prefix (0b01), as described in [RFC2671]. (A normal label type always begin with 0b00). A new extended label type for IDNE is used to identify an IDNE label. This document uses 0b0000010 as the extended label type; however, the label type will be assigned by IANA and it may not be 0b0000010.

```

bits 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 . . .
      +-+ +-+ +-+ +-+ +-+ +-+ +-+ +-+ +-+ +-+ +-+ +-+ +-+ +-+ +-+ +-+ +-+ +-+ +-+ +-+ +-+ +-+ // +-+ +-+ +-+ +-+ +-+ +-+
      |0 1|   ELT   |   Size   |   IDN label ...   |

```



Using the previous labels, one datagram might contain "www.m\*.com" and "m\*.com" (where the "\*" is <LATIN CAPITAL LETTER E WITH ACUTE>).

Ignoring the other fields of the message, the domain name portions of the datagram could look like:

```

+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
20 | 0 1 0 0 0 0 1 0 | 0 0 0 0 0 0 1 1 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
22 |           0x6D (m)           |           0xC3 (e'(1))           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
24 |           0xA9 (e'(2))       |           3                       |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
26 |           0x63 (c)           |           0x6F (o)           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
28 |           0x6D (m)           |           0x00                       |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
.   .   .
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
40 |           3                   |           0x77 (w)           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
42 |           0x77 (w)           |           0x77 (w)           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
44 | 1 1 |           20                       |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

The domain name "m\*.com" is shown at offset 20. The domain name "www.m\*.com" is shown at offset 40; this definition uses a pointer to concatenate a label for www to the previously defined "m\*.com".

#### 4. Label Size

In IDNE, the maximum length of a label is 255 octets, and the maximum size for a domain name is 1023 octets. The reason for using these values is so that IDNE labels can have the same number of characters as the ASCII-based labels in [\[STD13\]](#). Because character encoding in UTF-8 is variable length, the maximum octet length for characters expected in the foreseeable future (that is, 4 octets for a single character) was used. Note that this extension allows some IDNE labels to be longer than 63 characters and some IDNE names to be longer than 255 octets.

Software creating DNS queries or responses using IDNE MUST verify that, after IDN preparation and transformation to UTF8, that no labels are longer than 255 octets and that no names are longer than 1023 octets. If there is a user interface associated with the process creating the query or response, that interface SHOULD give the user an error message.

Software MUST NOT transmit DNS queries or responses which contain labels that are longer than 255 octets or names that are longer than 1023

octets. Servers MUST NOT accept DNS queries or responses which contain labels that are longer than 255 octets or names that are longer than **1023 octets**, and MUST send the **NOTIMPL RCODE error message** if such queries or responses are received.

## **5. UDP Packet Size**

IDNE-capable senders and receivers MUST support UDP packet sizes of 1220 octets, not including IP and UDP headers (note that the minimum MTU for IPv6 is 1280 [[RFC2460](#)]). A sender MUST announce its capability in the OPT pseudo-RR described in [section 4.3 of \[RFC2671\]](#) by having the CLASS sender's UDP payload size be greater than or equal to 1220.

## **6. Canonicalization, Prohibited Characters, and Case Folding**

The string in the label MUST be pre-processed as described in [[NAMEPREP](#)] before the query or response is prepared. A query or response MUST NOT contain a label that does not conform to [[NAMEPREP](#)].

## **7. Versions of IDNE**

The IDN protocol version number MUST be included in the OPT RR RDATA of EDNS (described in [Section 4.4 of \[RFC2671\]](#)). An OPTION-CODE will be assigned by IANA for storing the IDNE protocol version number; this document uses 0x0001 for the OPTION-CODE. The value (that is, the OPTION-DATA) is the version number coded in 8 bits.

All requesters MUST send this information as part of the OPT RR included in the EDNS packet.

### **7.1 This version of IDNE**

This document describes version 1 of IDNE. This version is a combination of the protocol in this document and the rules as described in [[NAMEPREP](#)]. Note that [[NAMEPREP](#)] describes a single version of the list of canonicalization, case folding, and prohibited characters, and that this document is linked to that single version of [[NAMEPREP](#)].

The identifiers for this specification are:

OPTION-CODE = 0x0001 (IDNE protocol version)  
OPTION-LENGTH = 0x0001 (1 octet following)  
OPTION-DATA = 0x01 (IDNE protocol version 1)

### **7.2 Creating new versions of IDNE**

A new version of IDNE is created by a standards-track RFC that specifies:

- a normative reference to [[NAMEPREP](#)] or a successor document to

## [\[NAMEPREP\]](#)

- an IDNE version number that is 1 greater than the highest IDNE version number at the time the RFC is published

If there are any changes to the encoding or interpretation of the protocol, they must also be specified in the same standards-track RFC.

### **[7.3 Prohibited characters and versions of IDNE](#)**

If a server receives a request containing an illegal or unknown character (as described in the version number in the request), it MUST send a NOTIMPL RCODE to the client. For example, if a server that understands both version 1 and version 2 receives a request that is marked as version 1, but contains a label that includes a character that is prohibited in version 1 but allowed in version 2, that server must still send a NOTIMPL RCODE to the client.

## **[8. API Specifications](#)**

The current API for TCP/IP uses `gethostbyname` and `gethostbyaddr` for IPv4 and `getnodeipbyname` and `getnodeipbyaddr` (specified in [\[RFC 2671\]](#)) for both IPv4 and IPv6. These function calls return `hostent` structs, where the `h_name` field contains a pointer to a char. In this context, receiving a UTF-8 string means that the application should know that UTF-8 uses more than one octet per char.

A new flag "IDN" (to appear in `netdb.h`) is defined to be passed in the `flags` argument of `getnodeipbyname` and `getnodeipbyaddr`. This flag tells the resolver to request an IDNE-encoded name. No new return code is defined since the returned codes in [RFC 2671](#) are meaningful in the IDNE context.

If one has not yet converted his code to IPv6 and still wants to enable IDNs with this API, one can do a macro of the `getnodeipby*` functions mapped to the IPv4 `gethostby*` ones, including the "IDN" flag, and then process differently based on the presence of the flag.

## **[9. Transition and Deployment](#)**

Deployment of this proposal means updating clients and servers, as well as applications and protocols, and therefore a transition strategy is proposed. Because many DNS servers do not yet handle IDNE and may take years or decades to do so, an ASCII-compatible encoding (ACE) format for IDN names is also needed as a transition to an all-IDNE DNS. Note that IDNE and an ACE are not related, and do not interact in the DNS. If the IETF chooses to have an ACE mechanism in use at the same time as IDNE, it would be wise to choose an ACE that allows as many characters as possible in the name parts and full names.

IDNE allows names with as many characters as current names. This means that it is possible to create names in IDNE that are longer than those that can be created in the ACE protocols that have been described so far. Although not prohibited, it is unwise to create a name that can be legally represented in IDNE but not in the ACE, or a name that can be legally represented in the ACE but not in IDNE.

The IETF should periodically evaluate the benefits and problems associated with having three different formats for names (STD13, IDNE, and ACE). If at some point it is decided that the problems outweigh the benefits, the IETF can state a time when one or more of the services should not be used on the Internet.

## **10. Root Server Considerations**

Because this specification uses EDNS, root servers should be prepared to receive EDNS requests. This specification handles IDN top-level domains in exactly the same fashion as it does every other domain. Considerations about IDN top-level domains are outside of this work, but the first IDN top-level domains would require all root servers to be ready for IDNE requests.

## **11. IANA Considerations**

[[ TBD. This section will have two parts. The first will request an EDNS option code. The second will specify how IDNE version numbers are allocated (namely, standards-track RFC only). ]]

## **12. Security Considerations**

Because IDNE uses EDNS, it inherits the same security considerations as EDNS.

Much of the security of the Internet relies on the DNS. Thus, any change to the characteristics of the DNS can change the security of much of the Internet.

Host names are used by users to connect to Internet servers. The security of the Internet would be compromised if a user entering a single internationalized name could be connected to different servers based on different interpretations of the internationalized host name.

Because this document normatively refers to [\[NAMEPREP\]](#) and [\[RFC2671\]](#), it includes the security considerations from those documents as well.

## **13. References**

[IDNCOMP] Paul Hoffman, "Comparison of Internationalized Domain Name

Proposals", [draft-ietf-idn-compare](#).

[ISO10646] ISO/IEC 10646-1:1993. International Standard -- Information technology -- Universal Multiple-Octet Coded Character Set (UCS) -- Part 1: Architecture and Basic Multilingual Plane. Five amendments and a technical corrigendum have been published up to now. UTF-16 is described in Annex Q, published as Amendment 1. 17 other amendments are currently at various stages of standardization. [[[ THIS REFERENCE NEEDS TO BE UPDATED AFTER DETERMINING ACCEPTABLE WORDING ]]]

[NAMEPREP] Paul Hoffman & Marc Blanchet, "Preparation of Internationalized Host Names", [draft-ietf-idn-nameprep](#).

[RFC2119] Scott Bradner, "Key words for use in RFCs to Indicate Requirement Levels", March 1997, [RFC 2119](#).

[RFC2279] Francois Yergeau, "UTF-8, a transformation format of ISO 10646", January 1998, [RFC 2279](#).

[RFC2460] Steve Deering & Bob Hinden, "Internet Protocol, Version 6 (IPv6) Specification", December 1998, [RFC 2460](#).

[RFC2671] Paul Vixie, "Extension Mechanisms for DNS (EDNS0)", August 1999, [RFC 2671](#).

[STD13] Paul Mockapetris, "Domain names - implementation and specification", November 1987, STD 13 ([RFC 1035](#)).

[UNICODE3] 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>.

## **A. Acknowledgements**

This document is the result of the thinking of many people. The following people made significant comments on the early drafts:

Andre Cormier  
Andrew Draper  
Bill Sommerfeld  
Francois Yergeau

## **B. Changes from -00 to -01**

1.1: Added reference to Unicode names.

3: Clarified that a size of zero is not allowed.

**3.1.1** and **3.1.2**: Fixed two very serious errors in the examples.



6: Removed second paragraph, which was redundant with 7.3.

12: Beefed up the security considerations.

13: Added [[ISO10646](#)] and [[UNICODE3](#)].

Added [Appendix A](#).

Added Appendix B.

### **[C](#). Authors' Addresses**

Marc Blanchet  
Viagenie  
**[2875](#) boul. Laurier, bureau 300**  
Sainte-Foy, QC G1V 2M2 Canada  
Marc.Blanchet@viagenie.qc.ca

Paul Hoffman  
Internet Mail Consortium and VPN Consortium  
**[127](#) Segre Place**  
Santa Cruz, CA 95060 USA  
phoffman@imc.org