Internationalized Domain Names for Applications (IDNA): Definitions and
                          Document Framework
                    draft-ietf-idnabis-defs-13.txt

Abstract

   This document is one of a collection that, together, describe the
   protocol and usage context for a revision of Internationalized Domain
   Names for Applications (IDNA), superseding the earlier version. It
   describes the document collection and provides definitions and other
   material that are common to the set.

Status of this Memo

   This Internet-Draft is submitted to IETF in full conformance with the
   provisions of BCP 78 and BCP 79.

   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.

   This Internet-Draft will expire on July 11, 2010.

Table of Contents

1. Introduction

1.1. IDNA2008

   This document is one of a collection that, together, describe the
   protocol and usage context for a revision of Internationalized Domain
   Names for Applications (IDNA) that was largely completed in 2008,
   known within the series and elsewhere as IDNA2008. The series
   replaces an earlier version of IDNA, described in [RFC3490] and
   [RFC3491]. For convenience, that version of IDNA is referred to in
   this documents as "IDNA2003". The newer version continues to use the
   Punycode algorithm [RFC3492] and ACE (ASCII-compatible encoding)
   prefix from that earlier version. The document collection is
   described in Section 1.3. As indicated there, this document provides
   definitions and other material that are common to the set.

1.1.1. Audiences

   While many IETF specifications are directed exclusively to protocol
   implementers, the character of IDNA requires that it be understood
   and properly used by those whose responsibilities include making
   decisions about

   o what names are permitted in DNS zone files,

   o policies related to names and naming, and

   o the handling of domain name strings in files and systems, even
      with no immediate intention of looking them up.

   This document and those concerned with the protocol definition, rules
   for handling strings that include characters written right-to-left,
   and the actual list of characters and categories will be of primary
   interest to protocol implementers. This document and the one
   containing explanatory material will be of primary interest to
   others, although they may have to fill in some details by reference
   to other documents in the set.

   This document and the associated ones are written from the
   perspective of an IDNA-aware user, application, or implementation.
   While they may reiterate fundamental DNS rules and requirements for
   the convenience of the reader, they make no attempt to be
   comprehensive about DNS principles and should not be considered as a
   substitute for a thorough understanding of the DNS protocols and
   specifications.

1.1.2. Normative Language

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 RFC 2119 [RFC2119].

1.2. Discussion Forum

[[RFC Editor: please remove this section.]]

IDNA2008 is being discussed in the IETF "idnabis" Working Group and
on the mailing list idna-update@alvestrand.no

1.3. Roadmap of IDNA2008 Documents

IDNA2008 consists of the following documents:

o This document, containing definitions and other material that are
   needed for understanding other documents in the set. It is
   referred to informally in other documents in the set as "Defs" or
   "Definitions".

o A document [IDNA2008-Rationale] that provides an overview of the
   protocol and associated tables together with explanatory material
   and some rationale for the decisions that led to IDNA2008. That
   document also contains advice for registry operations and those
   who use internationalized domain names. It is referred to
   informally in other documents in the set as "Rationale". It is
   not normative.

o A document [IDNA2008-Protocol] that describes the core IDNA2008
   protocol and its operations. In combination with the "Bidi"
   document described immediately below, it explicitly updates and
   replaces RFC 3490. It is referred to informally in other
   documents in the set as "Protocol".

o A document [IDNA2008-Bidi] that specifies special rules ("Bidi")
   for labels that contain characters that are written from right to
   left.

o A specification [IDNA2008-Tables] of the categories and rules that
   identify the code points allowed in a label written in native
   character form (defined more specifically as a "U-label" in
   Section 2.3.2.1 below), based on Unicode 5.2 [Unicode52] code
   point assignments and additional rules unique to IDNA2008. The
   Unicode-based rules are expected to be stable across Unicode
   updates and hence independent of Unicode versions. That
   specification obsoletes RFC 3941 and IDN use of the tables to

which it refers. It is referred to informally in other documents
in the set as "Tables".

o A document [IDNA2008-Mapping] that discusses the issue of mapping
   characters into other characters and that provides guidance for
   doing so when that is appropriate. This document provides advice;
   it is not a required part of IDNA.

2. Definitions and Terminology

2.1. Characters and Character Sets

A code point is an integer value in the codespace of a coded
character set. In Unicode, these are integers from 0 to 0x10FFFF.

Unicode [Unicode52] is a coded character set with about 100,000
characters assigned to code points as of version 5.1. A single
Unicode code point is denoted in these documents by "U+" followed by
four to six hexadecimal digits, while a range of Unicode code points
is denoted by two four to six digit hexadecimal numbers separated by
"..", with no prefixes.

ASCII means US-ASCII [ASCII], a coded character set containing 128
characters associated with code points in the range 0000..007F.
Unicode is a superset of ASCII and may be thought of as a
generalization of it; it includes all the ASCII characters and
associates them with equivalent code points.

"Letters" are, informally, generalizations from the ASCII and common-
sense understanding of that term, i.e., characters that are used to
write text that are not digits, symbols, or punctuation. Formally,
they are characters with a Unicode General Category value starting in
"L" (see Section 4.5 of [Unicode52]).

2.2. DNS-related Terminology

When discussing the DNS, this document generally assumes the
terminology used in the DNS specifications [RFC1034] [RFC1035] as
modified by [RFC1123] and [RFC2181]. The term "lookup" is used to
describe the combination of operations performed by the IDNA2008
protocol and those actually performed by a DNS resolver. The process
of placing an entry into the DNS is referred to as "registration",
similar to common contemporary usage in other contexts.
Consequently, any DNS zone administration is described as a
"registry", and the terms "registry" and "zone administrator" are
used interchangeably, regardless of the actual administrative
arrangements or level in the DNS tree. More detail about that

relationship is included in the "Rationale" document.

The term "LDH code point" is defined in this document to refer to the code points associated with ASCII letters (Unicode code points 0041..005A and 0061..007A), digits (0030..0039), and the hyphen-minus (U+002D). "LDH" is an abbreviation for "letters, digits, hyphen".

The base DNS specifications [RFC1034] [RFC1035] discuss "domain names" and "host names", but many people use the terms interchangeably, as do sections of these specifications. Lack of clarity about that terminology has contributed to confusion about intent in some cases. These documents generally use the term "domain name". When they refer to, e.g., host name syntax restrictions, they explicitly cite the relevant defining documents. The remaining definitions in this subsection are essentially a review: if there is any perceived difference between those definitions and the definitions in the base DNS documents or those cited below, the definitions in the other documents take precedence.

A label is an individual component of a domain name. Labels are usually shown separated by dots; for example, the domain name "www.example.com" is composed of three labels: "www", "example", and "com". (The zero-length root label described in RFC 1123 [RFC1123], which can be explicit as in "www.example.com." or implicit as in "www.example.com", is not considered in this specification.) IDNA extends the set of usable characters in labels that are treated as text (as distinct from the binary string labels discussed in RFC 1035 and RFC 2181 [RFC2181] and the bitstring ones described in RFC 2673 [RFC2673]), but only in certain contexts. The different contexts for different sets of usable characters are outlined in the next section. For the rest of this document and in the related ones, the term "label" is shorthand for "text label", and "every label" means "every text label", including the expanded context.

2.3. Terminology Specific to IDNA

This section defines some terminology to reduce dependence on terms and definitions that have been problematic in the past. The relationships among these definitions are illustrated in Figure 1 and Figure 2. In the first of those figures, the parenthesized numbers refer to the notes below the figure.

2.3.1. LDH-label

This is the classical label form used in host names [RFC0952] and described as the preferred form in RFC 1035 [RFC1035]. It is a string consisting of ASCII letters, digits, and the hyphen with the further restriction that the hyphen cannot appear at the beginning or

end of the string. Like all DNS labels, its total length must not
exceed 63 octets.

LDH-labels include the specialized labels used by IDNA (described as
"A-labels" below) and some additional restricted forms (also
described below).

To facilitate clear description, two new subsets of LDH-labels are
created by the introduction of IDNA. These are called Reserved LDH
labels (R-LDH labels) and Non-Reserved LDH labels (NR-LDH labels).
Reserved LDH labels, known as "tagged domain names" in some other
contexts, have the property that they contain "--" in the third and
fourth characters but which otherwise conform to LDH-label rules.
Only a subset of the R-LDH labels can be used in IDNA-aware
applications. That subset consists of the class of labels that begin
with the prefix "xn--" (case independent), but otherwise conform to
the LDH-label rules. That subset is called "XN-labels" in this set
of documents. XN-labels are further divided into those whose
remaining characters (after the "xn--") are valid output of the
Punycode algorithm RFC 3492 [RFC3492] and those that are not (see
below). The XN-labels that are valid Punycode output are known as
"A-labels" if they also meet the other criteria for IDNA-validity
described below. Because LDH-labels (and, indeed, any DNS label)
must not be more than 63 octets in length, the Punycode-algorithm-
derived portion of XN-labels is limited to no more than 59 ASCII
characters. Non-reserved LDH labels are the set of valid LDH labels
that do not have "--" in the third and fourth positions.

A consequence of the restrictions on valid characters in the native
Unicode character form (see U-labels) turns out to be that mixed-case
annotation, of the sort outlined in RFC 3492 Appendix A [RFC3492], is
never useful. Therefore, since a valid A-label is the result of
Punycode encoding of a U-label, A-labels should be produced only in
lower case, despite matching other (mixed- or upper-case) potential
labels in the DNS.

Some labels that are prefixed with "xn--" may not be the output of
the Punycode algorithm, or may fail the other tests outlined below or
violate other IDNA restrictions and thus are also not valid IDNA-
labels. They are called "Fake A-Labels" for convenience.

Labels within the class of R-LDH labels that are not prefixed with
"xn--" are also not valid IDNA-labels. To allow for future use of
mechanisms similar to IDNA, those labels MUST NOT be processed as
ordinary LDH-labels by IDNA-conforming programs and SHOULD NOT be
mixed with IDNA-labels in the same zone.

These distinctions among possible LDH labels are only of significance

for software that is "IDNA-aware" or for future extensions that use
extensions based on the same "prefix and encoding" model. For IDNA-
aware systems, the valid label types are: A-labels, U-labels and NR-
LDH labels.

IDNA-labels come in two flavors: An ACE-encoded form and a Unicode
(native character) form. These are referred to as A-labels and
U-labels respectively and are described in detail in the next
section.

```
                            ASCII-LABEL

 _____
|                      LDH-LABEL (1) (4)                               |
|                                                                      |
|        _____                |
|    |  _____             |  |              |
|    | |IDN Reserved LDH Labels          |            |  |             |
|    | | ("??--") or R-LDH LABELS         |  _____   |  |    |
|    | | _____  | |NON-RESERVED |   |  |      |
|    | | |      XN LABELS              |  | | LDH LABELS  |   |  |      |
|    | | | _____  _____ | | | (NR-LDH     |   |  |      |
|    | | | | A-labels  |   | Fake (3) || | | |  LABELS)   |   |  |     |
|    | | | | "xn--"(2) |   | A-labels ||| | |_____|   |  |      |
|    | | | |_____|   |_____|| |  |                   |  |    |
|    | | |_____| |  |                   |  |    |
|    | |_____|               |  |     |
|    |_____|  | |
|                                                                      |
|             _____NON-LDH-LABEL_____                    |
|        |      _____    |                           |
|        |     | Underscore labels  |    |                            |
|        |     |  e.g. _tcp         |    |                            |
|        |     |_____|    |                            |
|        |     | Labels with leading|    |                            |
|        |     | or trailing        |    |                            |
|        |     | hyphens "-abcd"    |    |                            |
|        |     | or "xyz-"          |    |                            |
|        |     | or "-uvw-"         |    |                            |
|        |     |_____|    |                            |
|        |     | Labels with other  |    |                            |
|        |     | non-LDH ASCII chars|    |                            |
|        |     | e.g. #$%_          |    |                            |
|        |     |_____|    |                            |
|        |_____|                            |
|_____|
```
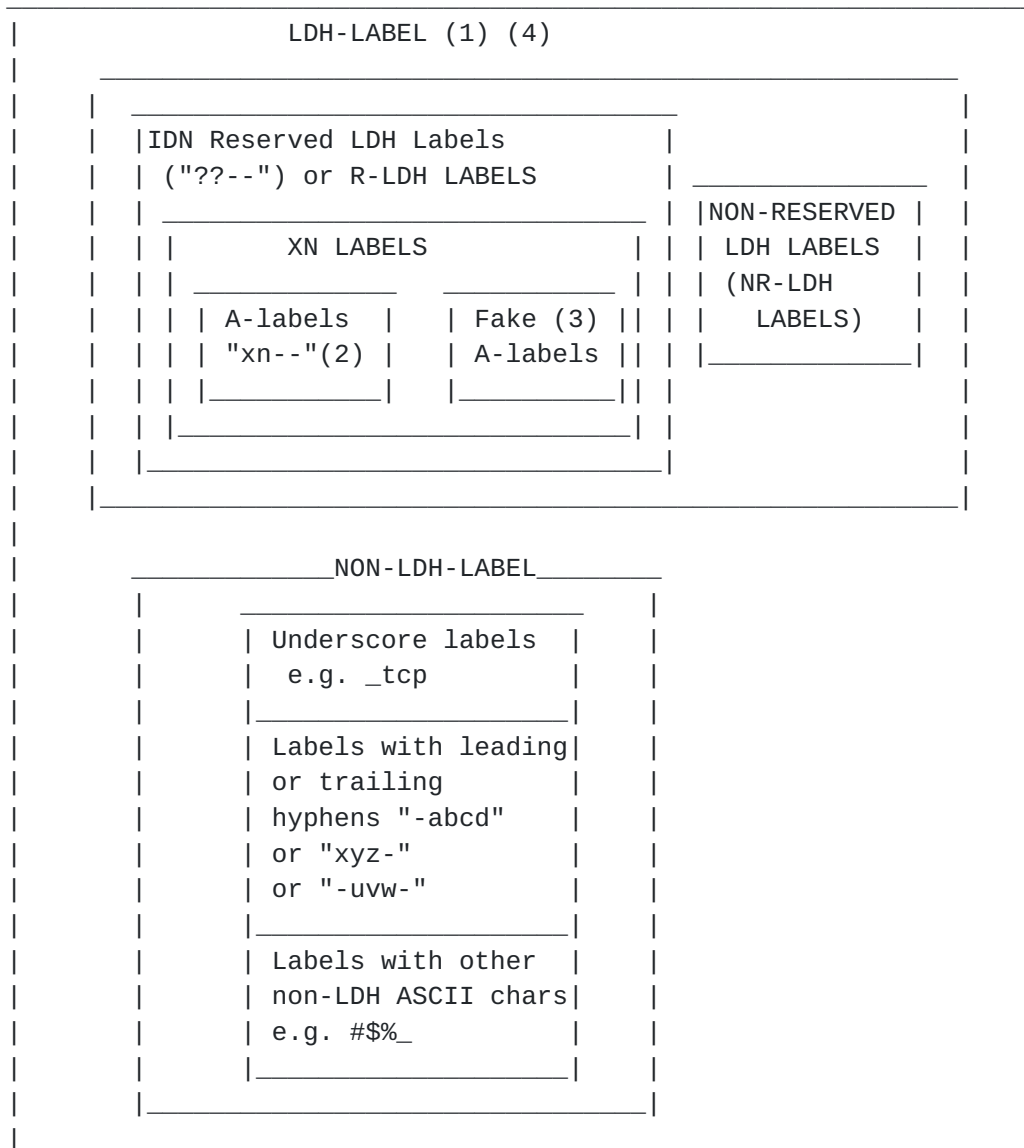
(1) ASCII letters (upper and lower case), digits,
        hyphen. Hyphen may not appear in first or last
        position. No more than 63 octets.
(2) Note that the string following "xn--" must
        be the valid output of the Punycode algorithm
        and must be convertible into valid U-label form.
(3) Note that a Fake A-Label has a prefix "xn--"
        but the remainder of the label is NOT the valid
        output of the Punycode algorithm.
(4) LDH-LABEL subtypes are indistinguishable to
        IDNA-unaware applications.

Figure 1: IDNA and Related DNS Terminology Space -- ASCII labels

```
             _____
            |   Non-ASCII               |
            |                           |
            |    _____   |
            |   | U-label (5)      |  |
            |   |_____|  |
            |   |                  |  |
            |   |   Binary Label   |  |
            |   | (including       |  |
            |   |   high bit on)   |  |
            |   |_____|  |
            |   |                  |  |
            |   | Bit String       |  |
            |   |    Label         |  |
            |   |_____|  |
            |_____|
```
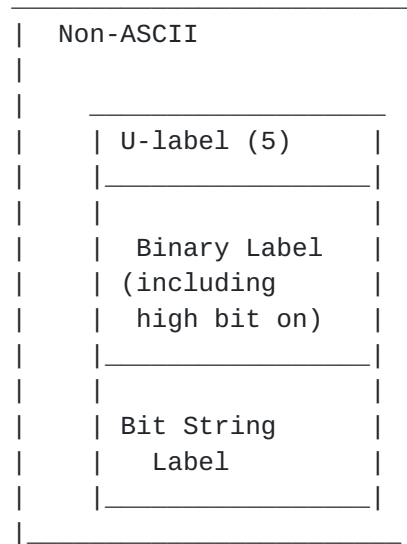
         (5) To IDNA-unaware applications, U-labels are
                indistinguishable from Binary ones.

                  Figure 2: Non-ASCII labels

2.3.2. Terms for IDN Label Codings

2.3.2.1. IDNA-valid strings, A-label, and U-label

   For IDNA-aware applications, the three types of valid labels are
   "A-labels", "U-labels", and "NR-LDH-labels", each of which is defined
   below. The relationships among them are illustrated in Figure 1 and
   Figure 2.

   o A string is "IDNA-valid" if it meets all of the requirements of
      these specifications for an IDNA label. IDNA-valid strings may
      appear in either of the two forms, defined immediately below, or
      may be drawn from the NR-LDH-label subset. IDNA-valid strings
      must also conform to all basic DNS requirements for labels. These
      documents make specific reference to the form appropriate to any
      context in which the distinction is important.

   o An "A-label" is the ASCII-Compatible Encoding (ACE, see
      Section 2.3.2.5) form of an IDNA-valid string. It must be a
      complete label: IDNA is defined for labels, not for parts of them
      and not for complete domain names. This means, by definition,
      that every A-label will begin with the IDNA ACE prefix, "xn--"
      (see Section 2.3.2.5), followed by a string that is a valid output
      of the Punycode algorithm [RFC3492] and hence a maximum of 59
      ASCII characters in length. The prefix and string together must
      conform to all requirements for a label that can be stored in the

DNS including conformance to the rules for the preferred form
described in RFC 1034, RFC 1035, and RFC 1123. If and only if a
string meeting the above requirements can be decoded into a
U-label, then it is an A-label.

o A "U-label" is an IDNA-valid string of Unicode characters, in
   normalization form NFC and including at least one non-ASCII
   character, expressed in a standard Unicode Encoding Form (such as
   UTF-8). It is also subject to the constraints about permitted
   characters that are specified in Section 4.2 of the Protocol
   document and the rules in the Sections 2 and 3 of the Tables
   document, the Bidi constraints in that document if it contains any
   character from scripts that are written right to left, and the
   symmetry constraint described immediately below. Conversions
   between U-labels and A-labels are performed according to the
   "Punycode" specification [RFC3492], adding or removing the ACE
   prefix as needed.

To be valid, U-labels and A-labels must obey an important symmetry
constraint. While that constraint may be tested in any of several
ways, an A-label A1 must be capable of being produced by conversion
from a U-label U1, and that U-label U1 must be capable of being
produced by conversion from A-label A1. Among other things, this
implies that both U-labels and A-labels must be strings in Unicode
NFC [Unicode-UAX15] normalized form. These strings MUST contain only
characters specified elsewhere in this document series, and only in
the contexts indicated as appropriate.

Any rules or conventions that apply to DNS labels in general, apply
to whichever of the U-label or A-label would be more restrictive.
There are two exceptions to this principle. First, the restriction
to ASCII characters does not apply to the U-label. Second, expansion
of the A-label form to a U-label may produce strings that are much
longer than the normal 63 octet DNS limit (potentially up to 252
characters) due to the compression efficiency of the Punycode
algorithm. Such extended-length U-labels are valid from the
standpoint of IDNA, but caution should be exercised as shorter limits
may be imposed by some applications.

For context, IDNA-unaware applications treat all LDH-labels as valid
for appearance in DNS zone files and queries. IDNA-aware
applications permit only A-labels and NR-LDH-labels to appear in zone
files and queries. U-labels can appear, along with the other two, in
presentation and user interface forms, and in protocols that use IDNA
forms but that do not involve the DNS itself.

Specifically, for IDNA-aware applications, the three allowed
categories are A-label, U-label, and NR-LDH-label. Of the reserved

LDH labels (R-LDH-labels) only A-labels are valid for IDNA use.

Strings that appear to be A-labels or U-labels are processed in
various of the operations of [IDNA2008-Protocol]. Those strings are
not yet demonstrably conformant with the conditions outlined above,
because they are in the process of validation. Such strings may be
referred to as "unvalidated", "putative", or "apparent", or as being
"in the form of" one of the label types to indicate that they have
not been verified to meet the specified conformance requirements.

Unvalidated A-labels are known only to be XN-Labels, while Fake
A-labels have been demonstrated to fail some of the A-label tests.
Similarly, unvalidated U-labels are simply Non-ASCII labels that may
or may not meet the requirements for U-labels.

2.3.2.2. NR-LDH-label and Internationalized Label

These specifications use the term "NR-LDH-label" strictly to refer to
an all-ASCII label that obeys the preferred syntax (often known as
"hostname" (from RFC 952 [RFC0952]) or "LDH") conventions and that is
neither an IDN nor a label form reserved by IDNA (R-LDH-label). It
should be stressed that an A-label obeys the "hostname" rules and is
sometimes described as "LDH-conformant".

2.3.2.3. Internationalized Domain Name

An "internationalized domain name" (IDN) is a domain name that
contains at least one A-label or U-label, but that otherwise may
contain any mixture of NR-LDH-labels, A-labels, or U-labels. Just as
has been the case with ASCII names, some DNS zone administrators may
impose restrictions, beyond those imposed by DNS or IDNA, on the
characters or strings that may be registered as labels in their
zones. Because of the diversity of characters that can be used in a
U-label and the confusion they might cause, such restrictions are
mandatory for IDN registries and zones even though the particular
restrictions are not part of these specifications (the issue is
discussed in more detail in Section 4.3 of [IDNA2008-Protocol] .
Because these restrictions, commonly known as "registry
restrictions", only affect what can be registered and not lookup
processing, they have no effect on the syntax or semantics of DNS
protocol messages; a query for a name that matches no records will
yield the same response regardless of the reason why it is not in the
zone. Clients issuing queries or interpreting responses cannot be
assumed to have any knowledge of zone-specific restrictions or
conventions. See the section on registration policy in
[IDNA2008-Rationale] for additional discussion.

"Internationalized label" is used when a term is needed to refer to a

single label of an IDN, i.e., one that might be any of an NR-LDH-
label, A-label, or U-label. There are some standardized DNS label
formats, such as the "underscore labels" used for service location
(SRV) records [RFC2782], that do not fall into any of the three
categories and hence are not internationalized labels.

2.3.2.4. Label Equivalence

In IDNA, equivalence of labels is defined in terms of the A-labels.
If the A-labels are equal in a case-independent comparison, then the
labels are considered equivalent, no matter how they are represented.
Because of the isomorphism of A-labels and U-labels in IDNA2008, it
is possible to compare U-labels directly; see [IDNA2008-Protocol] for
details. Traditional LDH labels already have a notion of
equivalence: within that list of characters, upper case and lower
case are considered equivalent. The IDNA notion of equivalence is an
extension of that older notion but, because the protocol does not
specify any mandatory mapping and only those isomorphic forms are
considered, the only equivalents are:

o Exact (bit-string identity) matches between a pair of U-labels.

o Matches between a pair of A-labels, using normal DNS matching
   rules.

o Equivalence between a U-label and an A-label determined by
   translating the U-label form into an A-label form and then testing
   for a match between the A-labels using normal DNS case-insensitive
   matching rules.

2.3.2.5. ACE Prefix

The "ACE prefix" is defined in this document to be a string of ASCII
characters "xn--" that appears at the beginning of every A-label.
"ACE" stands for "ASCII-Compatible Encoding".

2.3.2.6. Domain Name Slot

A "domain name slot" is defined in this document to be a protocol
element or a function argument or a return value (and so on)
explicitly designated for carrying a domain name. Examples of domain
name slots include the QNAME field of a DNS query; the name argument
of the gethostbyname() or getaddrinfo() standard C library functions;
the part of an email address following the at-sign (@) in the
parameter to the SMTP MAIL or RCPT commands or the "From:" field of
an email message header; and the host portion of the URI in the src
attribute of an HTML <IMG> tag. A string that has the syntax of a
domain name but that appears in general text is not in a domain name

slot. For example, a domain name appearing in the plain text body of
an email message is not occupying a domain name slot.

An "IDN-aware domain name slot" is defined for this set of documents
to be a domain name slot explicitly designated for carrying an
internationalized domain name as defined in this document. The
designation may be static (for example, in the specification of the
protocol or interface) or dynamic (for example, as a result of
negotiation in an interactive session).

An "IDN-unaware domain name slot" is defined for this set of
documents to be any domain name slot that is not an IDN-aware domain
name slot. Obviously, this includes any domain name slot whose
specification predates IDNA. Note that the requirements of some
protocols that use the DNS for data storage prevent the use of IDNs.
For example, the format required for the underscore labels used by
the service location protocol [RFC2782] precludes representation of a
non-ASCII label in the DNS using A-labels because those SRV-related
labels must start with underscores. Of course, non-ASCII IDN labels
may be part of a domain name that also includes underscore labels.

2.3.3. Order of Characters in Labels

Because IDN labels may contain characters that are read, and
preferentially displayed, from right to left, there is a potential
ambiguity about which character in a label is "first". For the
purposes of these specifications, labels are considered, and
characters numbered, strictly in the order in which they appear "on
the wire". That order is equivalent to the leftmost character being
treated as first in a label that is read left-to-right and to the
rightmost character being first in a label that is read right-to-
left. The "Bidi" specification contains additional discussion of the
conditions that influence reading order.

2.3.4. Punycode is an Algorithm, not a Name or Adjective

There has been some confusion about whether a "Punycode string" does
or does not include the ACE prefix and about whether it is required
that such strings could have been the output of the ToASCII operation
(see RFC 3490, Section 4 [RFC3490]). This specification discourages
the use of the term "Punycode" to describe anything but the encoding
method and algorithm of [RFC3492]. The terms defined above are
preferred as much more clear than the term "Punycode string".

3. IANA Considerations

Actions for IANA are specified in other documents in this series

[IDNA2008-Protocol] [IDNA2008-Tables]. An overview of the
relationships among the various IANA registries appears in
[IDNA2008-Rationale]. This document does not specify any actions for
IANA.

4. Security Considerations

4.1. General Issues

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

Domain names are used by users to identify and connect to Internet
hosts and other network resources. The security of the Internet is
compromised if a user entering a single internationalized name is
connected to different servers based on different interpretations of
the internationalized domain name. In addition to characters that
are permitted by IDNA2003 and its mapping conventions (See
Section 4.6), the current specification changes the interpretation of
a few characters that were mapped to others in the earlier version;
zone administrators should be aware of the problems that might raise
and take appropriate measures. The context for this issue is
discussed in more detail in [IDNA2008-Rationale]).

In addition to the Security Considerations material that appears in
this document, [IDNA2008-Bidi] contains a discussion of security
issues specific to labels containing characters from scripts that are
normally written right to left.

4.2. U-label Lengths

Labels associated with the DNS have traditionally been limited to 63
octets by the general restrictions in RFC 1035 and by the need to
treat them as a six-bit string length followed by the string in
actual calls to the DNS. That format is used in some other
applications and, in general, that representations of domain names as
dot-separated labels and as length-string pair have been treated as
interchangeable. Because A-labels (the form actually used in the
DNS) are potentially much more compressed than UTF-8 (and UTF-8 is,
in general, more compressed that UTF-16 or UTF-32), U-labels that
obey all of the relevant symmetry (and other) constraints of these
documents may be quite a bit longer, potentially up to 252 characters
(Unicode code points). A fully-qualified domain name containing
several such labels can obviously also exceed the nominal 255 octet
limit for such names. Application authors using U-labels must exert
due caution to avoid buffer overflow and truncation errors and

   attacks in contexts where shorter strings are expected.

4.3. Local Character Set Issues

   When systems use local character sets other than ASCII and Unicode,
   these specifications leave the problem of converting between the
   local character set and Unicode up to the application or local
   system. If different applications (or different versions of one
   application) implement different rules for conversions among coded
   character sets, they could interpret the same name differently and
   contact different servers. This problem is not solved by security
   protocols, such as Transport Layer Security (TLS) [RFC5246], that do
   not take local character sets into account.

4.4. Visually Similar Characters

   To help prevent confusion between characters that are visually
   similar (sometimes called "confusables"), it is suggested that
   implementations provide visual indications where a domain name
   contains multiple scripts, especially when the scripts contain
   characters that are easily confused visually, such as an omicron in
   Greek mixed with Latin text. Such mechanisms can also be used to
   show when a name contains a mixture of simplified and traditional
   Chinese characters, or to distinguish zero and one from upper-case
   "O" and lower-case "L". DNS zone administrators may impose
   restrictions (subject to the limitations identified elsewhere in
   these documents) that try to minimize characters that have similar
   appearance or similar interpretations.

   If multiple characters appear in a label and the label consists only
   of characters in one script, individual characters that might be
   confused with others if compared separately may be unambiguous and
   non-confusing. On the other hand, that observation makes mixed-
   script labels even more risky -- users will tend to see what they
   expect to see and context is a powerful reinforcement to perception.
   At the same time, while the risks associated with mixed-script labels
   are clear, simply prohibiting them will not eliminate problems,
   especially where closely-related scripts are involved. For example,
   there are many strings that are entirely in Greek or Cyrillic scripts
   that can be confused with each other or with Latin-script strings.

   It is worth noting that there are no comprehensive technical
   solutions to the problems of confusable characters. One can reduce
   the extent of the problems in various ways, but probably never
   eliminate it. Some specific suggestions about identification and
   handling of confusable characters appear in a Unicode Consortium
   publication [Unicode-UTR36].

4.5. **IDNA Lookup, Registration, and the Base DNS Specifications**

   The Protocol specification [IDNA2008-Protocol] describes procedures
   for registering and looking up labels that are not compatible with
   the preferred syntax described in the base DNS specifications (STD13
   [RFC1034] [RFC1035] and Host Requirements [RFC1123]) because they
   contain non-ASCII characters. These procedures depend on the use of
   a special ASCII-compatible encoding form that contains only
   characters permitted in host names by those earlier specifications.
   The encoding used is Punycode [RFC3492]. No security issues such as
   string length increases or new allowed values are introduced by the
   encoding process or the use of these encoded values, apart from those
   introduced by the ACE encoding itself.

   Domain names (or portions of them) are sometimes compared against a
   set of domains to be given special treatment if a match occurs, e.g.,
   treated as more privileged than others or blocked in some way. In
   such situations, it is especially important that the comparisons be
   done properly, as specified in the Requirements section of
   [IDNA2008-Protocol]. For labels already in ASCII form, the proper
   comparison reduces to the same case-insensitive ASCII comparison that
   has always been used for ASCII labels although IDNA-aware
   applications are expected to look up only A-labels and NR-LDH-labels,
   i.e., to avoid looking up R-LDH-labels that are not A-labels.

   The introduction of IDNA meant that any existing labels that start
   with the ACE prefix would be construed as A-labels, at least until
   they failed one of the relevant tests, whether or not that was the
   intent of the zone administrator or registrant. There is no evidence
   that this has caused any practical problems since RFC 3490 was
   adopted, but the risk still exists in principle.

4.6. **Legacy IDN Label Strings**

   The URI Standard [RFC3986] and a number of application specifications
   (e.g., [RFC5321], [RFC2616]) do not permit non-ASCII labels in DNS
   names used with those protocols, i.e., only the A-label form of IDNs
   is permitted in those contexts. If only A-labels are used,
   differences in interpretation between IDNA2003 and this version arise
   only for characters whose interpretation have actually changed (e.g.,
   characters, such as ZWJ and ZWNJ, that were mapped to nothing in
   IDNA2003 and that are considered legitimate in some contexts by these
   specifications). Despite that prohibition, there are a significant
   number of files and databases on the Internet in which domain name
   strings appear in native-character form; a subset of those strings
   use native-character labels that require IDNA2003 mapping to produce
   valid A-labels. The treatment of such labels will vary by types of
   applications and application-designer preference: in some situations,

warnings to the user or outright rejection may be appropriate; in
others, it may be preferable to attempt to apply the earlier mappings
if lookup strictly conformant to these specifications fails or even
to do lookups under both sets of rules. This general situation is
discussed in more detail in [IDNA2008-Rationale]. However, in the
absence of care by registries about how strings that could have
different interpretations under IDNA2003 and the current
specification are handled, it is possible that the differences could
be used as a component of name-matching or name-confusion attacks.
Such care is therefore appropriate.

4.7. Security Differences from IDNA2003

The registration and lookup models described in this set of documents
change the mechanisms available for lookup applications to determine
the validity of labels they encounter. In some respects, the ability
to test is strengthened. For example, putative labels that contain
unassigned code points will now be rejected, while IDNA2003 permitted
them (see [IDNA2008-Rationale] for a discussion of the reasons for
this). On the other hand, the protocol specification no longer
assumes that the application that looks up a name will be able to
determine, and apply, information about the protocol version used in
registration. In theory, that may increase risk since the
application will be able to do less pre-lookup validation. In
practice, the protection afforded by that test has been largely
illusory for reasons explained in RFC 4690 [RFC4690] and elsewhere in
these documents.

Any change to the Stringprep [RFC3454] procedure that is profiled and
used in IDNA2003, or, more broadly, the IETF's model of the use of
internationalized character strings in different protocols, creates
some risk of inadvertent changes to those protocols, invalidating
deployed applications or databases, and so on. But these
specifications do not change Stringprep at all; they merely bypass
it. Because these documents do not depend on Stringprep, the
question of upgrading other protocols that do have that dependency
can be left to experts on those protocols: the IDNA changes and
possible upgrades to security protocols or conventions are
independent issues.

4.8. Summary

No mechanism involving names or identifiers alone can protect against
a wide variety of security threats and attacks that are largely
independent of the naming or identification system. These attacks
include spoofed pages, DNS query trapping and diversion, and so on.

5. Acknowledgments

   The initial version of this document was created largely by
   extracting text from the "rationale" document [IDNA2008-Rationale].
   See the section of this name, and the one entitled "Contributors", in
   it.

   Specific textual suggestions after the extraction process came from
   Vint Cerf, Lisa Dusseault, Bill McQuillan, Andrew Sullivan, and Ken
   Whistler. Other changes were made in response to more general
   comments, lists of concerns or specific errors from participants in
   the Working Group and other observers, including Lyman Chapin, James
   Mitchell, Subramanian Moonesamy, and Dan Winship.

6. References

6.1. Normative References

   [ASCII]   American National Standards Institute (formerly United
                 States of America Standards Institute), "USA Code for
                 Information Interchange", ANSI X3.4-1968, 1968.

                 ANSI X3.4-1968 has been replaced by newer versions with
                 slight modifications, but the 1968 version remains
                 definitive for the Internet.

   [RFC1034] Mockapetris, P., "Domain names - concepts and facilities",
                 STD 13, RFC 1034, November 1987.

   [RFC1035] Mockapetris, P., "Domain names - implementation and
                 specification", STD 13, RFC 1035, November 1987.

   [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.

   [Unicode-UAX15]
                 The Unicode Consortium, "Unicode Standard Annex #15:
                 Unicode Normalization Forms", March 2008,
                 <http://www.unicode.org/reports/tr15/>.

   [Unicode52]
                 The Unicode Consortium, "The Unicode Standard, Version
                 5.2.0", 2009.

defined by: The Unicode Standard, Version 5.0, Boston, MA,
Addison-Wesley, 2007, ISBN 0-321-48091-0, as amended by
Unicode 5.1.0 (2008)
(http://www.unicode.org/versions/Unicode5.1.0/) and
Unicode 5.2.0 (2009)
(http://www.unicode.org/versions/Unicode5.2.0/).

6.2. Informative References

[IDNA2008-Bidi]
           Alvestrand, H. and C. Karp, "An updated IDNA criterion for
           right to left scripts", August 2009, <https://
           datatracker.ietf.org/drafts/draft-ietf-idnabis-bidi/>.

[IDNA2008-Mapping]
           Resnick, P. and P. Hoffman, "Mapping Characters in IDNA",
           September 2009, <https://datatracker.ietf.org/drafts/
           draft-ietf-idnabis-mapping/>.

[IDNA2008-Protocol]
           Klensin, J., "Internationalized Domain Names in
           Applications (IDNA): Protocol", September 2009, <https://
           datatracker.ietf.org/drafts/draft-ietf-idnabis-protocol/>.

[IDNA2008-Rationale]
           Klensin, J., "Internationalized Domain Names for
           Applications (IDNA): Background, Explanation, and
           Rationale", August 2009, <https://datatracker.ietf.org/
           drafts/draft-ietf-idnabis-rationale/>.

[IDNA2008-Tables]
           Faltstrom, P., "The Unicode Code Points and IDNA",
           August 2009, <https://datatracker.ietf.org/drafts/
           draft-ietf-idnabis-tables/>.

           A version of this document is available in HTML format at
           http://stupid.domain.name/idnabis/
           draft-ietf-idnabis-tables-02.html

[RFC0952] Harrenstien, K., Stahl, M., and E. Feinler, "DoD Internet
           host table specification", RFC 952, October 1985.

[RFC2181] Elz, R. and R. Bush, "Clarifications to the DNS
           Specification", RFC 2181, July 1997.

[RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H.,
           Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext
           Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

[RFC2673]   Crawford, M., "Binary Labels in the Domain Name System",
            RFC 2673, August 1999.

[RFC2782]   Gulbrandsen, A., Vixie, P., and L. Esibov, "A DNS RR for
            specifying the location of services (DNS SRV)", RFC 2782,
            February 2000.

[RFC3454]   Hoffman, P. and M. Blanchet, "Preparation of
            Internationalized Strings ("stringprep")", RFC 3454,
            December 2002.

[RFC3490]   Faltstrom, P., Hoffman, P., and A. Costello,
            "Internationalizing Domain Names in Applications (IDNA)",
            RFC 3490, March 2003.

[RFC3491]   Hoffman, P. and M. Blanchet, "Nameprep: A Stringprep
            Profile for Internationalized Domain Names (IDN)",
            RFC 3491, March 2003.

[RFC3492]   Costello, A., "Punycode: A Bootstring encoding of Unicode
            for Internationalized Domain Names in Applications
            (IDNA)", RFC 3492, March 2003.

[RFC3986]   Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform
            Resource Identifier (URI): Generic Syntax", STD 66,
            RFC 3986, January 2005.

[RFC4690]   Klensin, J., Faltstrom, P., Karp, C., and IAB, "Review and
            Recommendations for Internationalized Domain Names
            (IDNs)", RFC 4690, September 2006.

[RFC5246]   Dierks, T. and E. Rescorla, "The Transport Layer Security
            (TLS) Protocol Version 1.2", RFC 5246, August 2008.

[RFC5321]   Klensin, J., "Simple Mail Transfer Protocol", RFC 5321,
            October 2008.

[Unicode-UTR36]
            The Unicode Consortium, "Unicode Technical Report #36:
            Unicode Security Considerations", July 2008,
            <http://www.unicode.org/reports/tr36/>.

Appendix A. Change Log

[[RFC Editor: Please remove this appendix]]

A.1. **Version -00**

   This document was created by pulling selected material out of
   draft-ietf-idnabis-rationale-03 ("Rationale") after a WG consensus
   call indicated that the rearrangement was appropriate. Mark Davis
   made the major contribution of getting the process started by
   identifying particular sections to be moved, even though this draft
   does not completely reflect his list.

   For Version -00 only, each section is identified with the associated
   former section of Rationale-03. Those sections were edited after
   incorporation into this document, so "Formerly" should be interpreted
   very loosely.

A.2. **Version -01**

   o Typographical errors corrected and some sections slightly renamed
      for clarity.

   o Other adjustments made to synchronize with current versions of
      "Rationale" and "Protocol".

A.3. **Version -02**

   o All back pointers to section numbers in Rationale have been
      removed.

   o Some definitions clarified. Added one about string order.

   o Usual small editorial tuning.

A.4. **Version -03**

   o Additional fine tuning based on discussions during and immediately
      before IETF 72.

A.5. **Version -04**

   o Corrections of text and improvement of definitions based on
      discussions after -03 was released.

   o Discussion of label comparisons tightened and made more consistent
      with Protocol.

   o Definitions of categories of labels supplemented with a picture
      (Figure 1).

    o Explicit text added to define strings that look like A-labels or
       U-labels but are not. This section was the original Section 2.3.3
       which was then removed due to another rewrite (See Appendix A.11).

A.6. Version -05

    o Consolidated Security Considerations sections, moving material
       from Protocol and Rationale here.

A.7. Version -06

    o Added pointer to the discussion, in Rationale, of looking up
       unassigned code points.

    o Clarified relationship to base DNS specifications.

    o Made several clarifications suggested by Mark Davis.

    o Added a security considerations stub to more explicitly mention
       issues with IDNA2003 labels (Section 4.6).

    o Rewrote definitions and terminology using suggestions (and
       considerable text and revised figures) from Vint Cerf. Relocated
       the figures for easier accessibility.

    o Small editorial corrections and new copyright material.

A.8. Version -07

    o Modified Figure 1 to put an additional box around NR-LDH Labels
       (per Andrew Sullivan) and rationalized spelling of "non-reserved".

    o Added a temporary note about page breaks and the figures.

    o Modified terminology slightly to mention "underscore labels" and
       to revise the statements about equivalence.

A.9. Version -08

    o Corrected several typos, at least one of them confusing (NR-LDH-
       Label instead or R-LDH-Label).

    o Added new text to the discussion of U-labels and A-labels (end of
       Section 2.3.2.1) to support the text in Protocol about not-yet-
       validated strings. See the note there.

A.10. Version -09

   o Added a pointer to the Mapping document.

   o Updated references to other documents.

A.11. Version -10

   o Updated references to other documents.

   o Corrected several issues pointed out by Andrew Sullivan, including
      editorial problems.

   o Former Section 2.3.3 removed as redundant with Section 2.3.2.1,
      which has been rewritten somewhat for clarity.

   o Remove placeholders/ requests for further clarification because
      there have been no suggestions.

   o Adjusted definition of "IDN".

   o Note: No further changes made or required for mapping other than
      an update to the references and removal of a placeholder that
      turned out to be unnecessary.

   o Note: Corrections and improvements to figure formatting and
      insertion of section numbers have been deferred to the post-WG-LC
      version. I only want to do that once.

A.12. Version -11

   o Added new material to Section 4.4 on visually similar characters
      and contexts.

   o Added definition for "IDNA2003".

   o Reformatted Figure 1 again.

   o Adjusted Acknowledgments to remove Mark Davis's name, per his
      request and advice from IETF Trust Counsel.

   o Adjusted description of 63 character limit on labels and added
      Security Consideration material on that subject.

   o Incorporated other changes from WG Last Call, including a note
      about upper case A-labels.

   o Several small editorial changes.

   o Inserted section numbers in references to other IDNA2008
      documents.

A.13. Version -12

   This is the version of the document produced to reflect comments on
   IETF Last Call. For the convenience of those who made comments and
   of the IESG in evaluating them, this section therefore identifies
   non-editorial changes made in response to Last Call comments in
   somewhat more detail than may be usual.

   o DNS length limit clarified to be in octets, not "characters"
      (whatever those are) in several places. (SM review, 20091011)

   o Clarify relationship between label equivalence and mapping. (SM
      review, 20091011)

   o Corrected accidently-dropped text in A-label definition. (Dan
      Winship, note of 20091013)

   o Made another round of patches to the case-sensitivity of A-labels.
      (James Mitchell, 20091014)

   o Removed the "selected protocols" handwaving from Section 2.3.2.1.
      (Peter Saint-Andre, 20091019)

   o Some minor corrections requested by Suresh Krishnan in the Gen-ART
      review of 20091016.

A.14. Version -13

   Version incorporating IESG post-Last-Call review and evaluation
   comments.

   o Editorial and reference corrections, including bringing Unicode
      reference to 5.2.

Author's Address

    John C Klensin
    1770 Massachusetts Ave, Ste 322
    Cambridge, MA 02140
    USA

    Phone: +1 617 245 1457
    Email: john+ietf@jck.com