Network Working Group J. Klensin
Internet-Draft December 11, 2008
Obsoletes: 3490 (if approved)
Intended status: Standards Track
Expires: June 14, 2009

Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework

draft-ietf-idnabis-defs-05.txt

Status of this Memo

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.

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]. It 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 what names are permitted in DNS zone files

- About policies related to names and naming, and

- About 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.

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:

- 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".

- 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.

- 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".

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

- 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 2.3.1.1 below), based on Unicode 5.1 [Unicode51] 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 specifications 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".

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 [Unicode51] 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 [Unicode51]).

2.2. DNS-related Terminology

When discussing the DNS, this document generally assumes the terminology used in the DNS specifications [RFC1034] [RFC1035]. 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", 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]). 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".

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.

2.3.1. Terms for IDN Label Codings

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

To improve clarity, this subsection of the document introduces three new terms. In the next subsection, it defines a historical term to be slightly more precise for IDNA contexts. The relationship among these terms and some others is illustrated in Figure 1.

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, trivially, be ASCII strings that conform to the traditional "hostname" (or "LDH") rule and that do not contain "--" as the third and fourth character. 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 2.3.1.5) 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 2.3.1.5), followed by a string that is a valid output of the Punycode algorithm 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. A string meeting that above requirements is still not an A-label unless it can be decoded into a U-label.

- 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 -- in an Internet transmission context this will normally be UTF-8 -- and subject to the constraints about permitted characters that are specified in the Protocol and Tables documents as well as the symmetry constraint described. 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 must be capable of being produced by conversion from a U-label and a U-label must be capable of being produced by conversion from an A-label. 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, such as rules about lengths of strings, apply to whichever of the U-label or A-label would be more restrictive. For the U-label, constraints imposed by existing protocols and their presentation forms make the length restriction apply to the length in octets of the UTF-8 form of those labels (which will always be greater than or equal to the length in code points). The exception to this, of course, is that the restriction to ASCII characters does not apply to the U-label.

A different way to look at these terms, which may be more clear to some readers, is that U-labels, A-labels, and LDH-labels (see the next subsection) are disjoint categories that, together, make up the forms of legitimate strings for use in domain names that describe hosts. Of the three, only A-labels and LDH-labels can actually appear in DNS zone files or queries; U-labels can appear, along with the other two, in presentation and user interface forms and in selected protocols other than those of the DNS itself. Strings that do not conform to the rules for one of these three categories and, in particular, strings that contain "--" in the third and fourth character position but are:

- not A-labels or

- cannot be processed as U-labels or A-labels as described in these specifications,

are invalid in IDNA-conformant applications as labels in domain names that identify Internet hosts or similar resources.

2.3.1.2. LDH-label and Internationalized Label

These specifications use the term "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 not an IDN. It should be stressed that an A-label obeys the "hostname" rules and is sometimes described as "LDH-conformant", or in similar language, but it is not an LDH-label as that term is defined in these specifications.

2.3.1.3. Internationalized Domain Name

An "internationalized domain name" (IDN) is a domain name that may contain any mixture of LDH-labels, A-labels, or U-labels. This implies that every conventional domain name is an IDN (which implies that it is possible for a domain name to be an IDN without it containing any non-ASCII characters). 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. 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 LDH-label, A-label, or U-label. There are some standardized DNS label formats, such as those for service location (SRV) records [RFC2782], that do not fall into any of the three categories and hence are not internationalized labels.

2.3.1.4. 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. Equivalent labels in IDNA are treated as alternate forms of the same label, just as "foo" and "Foo" are treated as alternate forms of the same label.

2.3.1.5. 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".

2.3.1.6. 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.

The figure on this page illustrates the relationships among some of the terms defined above. The parenthesized numbers refer to the notes below the figure.

```
              _______________________          _______________________
              | ASCII Labels        |          |  Non-ASCII           |
              |                     |          |                      |
              |  ___________________|          |    __________________|
              |  |LDH-conforming (1)|          |    | U-label (2)     |
              |  |                  |          |    |_________________|
              |  |  ________________|          |    |                 |
              |  |  | LDH-label     |          |    |  Binary Label   |
              |  |  |_______________|          |    | (including      |
              |  |  | A-label       |          |    |  high bit on)   |
              |  |  |_______________|          |    |_________________|
              |  |  |               |          |    |                 |
              |  |  | Broken IDN    |          |    | Bit String      |
              |  |  |   e.g., xn--?,|          |    |   Label         |
              |  |  |      abc--def |          |    |_________________|
              |  |  |_______________|          |______________________|
              |  |__________________|
              |  ___________________|
              |  |Not-LDH-Conforming|
              |  |                  |
              |  |  ________________|
              |  |  |SRV & SRV-like |
              |  |  | e.g., _tcp    |
              |  |  |_______________|
              |  |  | Leading or    |
              |  |  |  trailing     |
              |  |  |  hyphens      |
              |  |  |_______________|
              |  |  | Other non-LDH |
              |  |  |  ASCII chars  |
              |  |  | e.g., #$%&_   |
              |  |  |_______________|
              |  |__________________|
              |_____________________|

        (1) These subtypes are indistinguishable to IDNA-unaware
                applications.
        (2) To IDNA-unaware applications, U-labels are
                indistinguishable from Binary ones.
```

Figure 1: IDNA and Related DNS Terminology Space

2.3.2. Strings Proposed to be Used or Looked Up as Labels

Strings are encountered at many places in these specifications that are expected to be processed as labels of particular types but that are not yet fully validated to conform to the requirements for the particular type of label in question. If XYZ is a type of label (e.g., "A" for A-label or "U" for a U-label), then the term "putative XYZ-label" is used to refer to such a string before it is fully validated or tested.

Similarly, terms similar to "a string in the form of an XYZ-label" are used to refer to a string that appears to obey the syntax for an XYZ-label on superficial examination. Specifically, a string that would comply with the LDH syntax except that some characters are non-ASCII is considered to be in the form of a U-label and one that starts in "xn--" and is otherwise all-ASCII is considered to be in the form of an A-label.

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 righmost 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 servers. 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, 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. 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.

4.3. Visually Similar Characters

To help prevent confusion between characters that are visually similar, it is suggested that implementations provide visual indications where a domain name contains multiple scripts (or what are considered multiple scripts in a local environment in which some mixed-script use is normal). 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. 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].

4.4. 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 (i.e., are LDH-labels or A-labels), the proper comparison reduces to the same case-insensitive ASCII comparison that has always been used for ASCII labels.

The introduction of IDNA means 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.

4.5. 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 (something that is now recognized as a considerable source of risk). 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.

4.6. 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 and Bill McQuillan.

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/>.

[Unicode51] The Unicode Consortium, "The Unicode Standard, Version 5.1.0", 2008.

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 (http://www.unicode.org/versions/Unicode5.1.0/).

6.2. Informative References

[IDNA2008-Bidi] Alvestrand, H. and C. Karp, "An updated IDNA criterion for right to left scripts", July 2008, <https://datatracker.ietf.org/drafts/draft-ietf-idnabis-bidi/>.

[IDNA2008-Protocol] Klensin, J., "Internationalized Domain Names in Applications (IDNA): Protocol", November 2008, <https://datatracker.ietf.org/drafts/draft-ietf-idnabis-protocol/>.

[IDNA2008-Rationale]
Klensin, J., "Internationalized Domain Names for Applications (IDNA): Background, Explanation, and Rationale", November 2008, <https://datatracker.ietf.org/drafts/draft-ietf-idnabis-rationale/>.

[IDNA2008-Tables]
Faltstrom, P., "The Unicode Code Points and IDNA", July 2008, <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.

[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.

[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.

[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

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

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

A.3. Version -02

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

- Some definitions clarified. Added one about string order.

- Usual small editorial tuning.

A.4. Version -03

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

A.5. Version -04

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

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

- Definitions of categories of labels supplemented with a picture.

- Explicit text added (Section 2.3.2) to define strings that look like A-labels or U-labels but are not.

A.6. Version -05

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

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

Full Copyright Statement

Intellectual Property