Network Working Group J. Klensin
Internet-Draft November 28, 2008
Obsoletes: 3490, 3491 (if approved)
Updates: 3492 (if approved)
Intended status: Standards Track
Expires: June 1, 2009

Internationalized Domain Names in Applications (IDNA): Protocol
draft-ietf-idnabis-protocol-07.txt

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of 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 June 1, 2009.

Abstract

This document supplies the protocol definition for a revised and updated specification for internationalized domain names (IDNs). The rationale for these changes, the relationship to the older specification, and important terminology are provided in other documents. This document specifies the protocol mechanism, called Internationalizing Domain Names in Applications (IDNA), for registering and looking up IDNs in a way that does not require changes to the DNS itself. IDNA is only meant for processing domain names, not free text.

Table of Contents

1. Introduction

This document supplies the protocol definition for a revised and updated specification for internationalized domain names. Essential definitions and terminology for understanding this document and a road map of the collection of documents that make up IDNA2008 appear in [IDNA2008-Defs]. Appendix A discusses the relationship between this specification and the earlier version of IDNA (referred to here as "IDNA2003") and the rationale for these changes, along with considerable explanatory material and advice to zone administrators who support IDNs is provided in another documents, notably [IDNA2008-Rationale].

IDNA works by allowing applications to use certain ASCII string labels (beginning with a special prefix) to represent non-ASCII name labels. Lower-layer protocols need not be aware of this; therefore IDNA does not depend on changes to any infrastructure. In particular, IDNA does not depend on any changes to DNS servers, resolvers, or protocol elements, because the ASCII name service provided by the existing DNS is entirely sufficient for IDNA.

IDNA is applied only to DNS labels. Standards for combining labels into fully-qualified domain names and parsing labels out of those names are covered in the base DNS standards [RFC1034] [RFC1035] and their various updates. An application may, of course, apply locally-appropriate conventions to the presentation forms of domain names as discussed in [IDNA2008-Rationale].

While they share terminology, reference data, and some operations, this document describes two separate protocols, one for IDN registration (Section 4) and one for IDN lookup (Section 5).

1.1. Discussion Forum

[[anchor3: RFC Editor: please remove this section.]]

This work is being discussed in the IETF IDNABIS WG and on the mailing list idna-update@alvestrand.no

2. Terminology

General terminology applicable to IDNA, but with meanings familiar to those who have worked with Unicode or other character set standards and the DNS, appears in [IDNA2008-Defs]. Terminology that is an integral, normative, part of the IDNA definition, including the definitions of "ACE", appears in that document as well. Familiarity with the terminology materials in that document is assumed for

reading this one. The reader of this document is assumed to be familiar with DNS-specific terminology as defined in RFC 1034 [RFC1034].

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 BCP 14, RFC 2119 [RFC2119].

3. Requirements and Applicability

3.1. Requirements

IDNA conformance means adherence to the following requirements:

1. Whenever a domain name is put into an IDN-unaware domain name slot (see Section 2 and [IDNA2008-Defs]), it MUST contain only ASCII characters (i.e., must be either an A-label or an LDH-label), or must be a label associated with a DNS application that is not subject to either IDNA or the historical recommendations for "hostname"-style names [RFC1034].

2. Comparison of labels MUST be done on equivalent forms: either both A-Label forms or both U-Label forms. Because A-labels and U-labels can be transformed into each other without loss of information, these comparisons are equivalent. However, when the A-label form is compared, it MUST use an ASCII case-insensitive comparison as with all comparisons of DNS labels. Comparison is only valid if the putative labels have been verified to be either A-Labels or U-Labels.

3. Labels being registered MUST conform to the requirements of Section 4. Labels being looked up and the lookup process MUST conform to the requirements of Section 5.

3.2. Applicability

IDNA is applicable to all domain names in all domain name slots except where it is explicitly excluded. It is not applicable to domain name slots which do not use the LDH syntax rules.

This implies that IDNA is applicable to many protocols that predate IDNA. Note that IDNs occupying domain name slots in those older protocols MUST be in A-label form until and unless those protocols and implementations of them are upgraded to be IDN-aware and that IDNs actually appearing in DNS queries or responses MUST be in A-label form.

3.2.1. DNS Resource Records

IDNA applies only to domain names in the NAME and RDATA fields of DNS resource records whose CLASS is IN.

There are currently no other exclusions on the applicability of IDNA to DNS resource records. Applicability depends entirely on the CLASS, and not on the TYPE except as noted below. This will remain true, even as new types are defined, unless there is a compelling reason for a new type that requires type-specific rules. The special naming conventions applicable to SRV records are examples of type-specific rules that are incompatible with IDNA coding. Hence the first two labels (the ones required to start in "_") on a record with TYPE SRV MUST NOT be A-labels or U-labels (while it would be possible to write a non-ASCII string with a leading underscore, conversion to an A-label would be impossible without loss of information because the underscore is not a letter, digit, or hyphen and is consequently DISALLOWED in IDNs). Of course, those labels may be part of a domain that uses IDN labels at higher levels in the tree.

3.2.2. Non-domain-name Data Types Stored in the DNS

Although IDNA enables the representation of non-ASCII characters in domain names, that does not imply that IDNA enables the representation of non-ASCII characters in other data types that are stored in domain names, specifically in the RDATA field for types that have structured RDATA format. For example, an email address local part is stored in a domain name in the RNAME field as part of the RDATA of an SOA record (hostmaster@example.com would be represented as hostmaster.example.com). IDNA specifically does not update the existing email standards, which allow only ASCII characters in local parts. Even though work is in progress to define internationalization for email addresses [RFC4952], changes to the email address part of the SOA RDATA would require action in, or updates to, other standards, specifically those that specify the format of the SOA RR.

4. Registration Protocol

This section defines the procedure for registering an IDN. The procedure is implementation independent; any sequence of steps that produces exactly the same result for all labels is considered a valid implementation.

Note that, while the registration and lookup protocols (Section 5) are very similar in most respects, they are different and implementers should carefully follow the steps they are implementing.

4.1. Proposed label

The registrant submits a request for an IDN. The user typically produces the request string by the keyboard entry of a character sequence in the local native character set (which might, of course, be Unicode).

4.2. Conversion to Unicode and Normalization

Some system routine, or a localized front-end to the IDNA process, ensures that the proposed label is a Unicode string or converts it to one as appropriate. Independent of its source form, the string MUST be in Unicode Normalization Form C (NFC [Unicode-UAX15]) before further processing in this protocol.

As a local implementation choice, the implementation MAY choose to map some forbidden characters to permitted characters (for instance mapping uppercase characters to lowercase ones), displaying the result to the user, and allowing processing to continue. This should be done very conservatively to prevent interoperability problems with lookup applications that do not follow exactly the same rules. In particular, it is strongly recommended that, to avoid any possible ambiguity, entities responsible for zone files ("registries") accept registrations only for A-labels (to be converted to U-labels by the registry as discussed above) or U-labels actually produced from A-labels, not forms expected to be converted by some other process.

4.3. Permitted Character and Label Validation

4.3.1. Input Format

[[anchor8: Note in -07 -- this section was formerly the second paragraph of Section 4.1. It may need additional work; suggestions welcome.]]

The registry MAY permit submission of labels in A-label form. If it does so, it MUST perform a conversion to a U-label, perform the steps and tests described below, and verify that the A-label produced by the step in Section 4.5 matches the one provided as input. If, for some reason, it does not, the registration MUST be rejected. If the conversion to a U-label is not performed, the registry MUST verify that the A-label is superficially valid, i.e., that it does not violate any of the rules of Punycode [RFC3492] encoding such as the prohibition on trailing hyphen-minus, appearance of non-basic characters before the delimiter, and so on. Invalid strings that appear to be A-labels MUST NOT be placed in DNS zones.

4.3.2. Rejection of Characters that are not Permitted

The candidate Unicode string is checked to verify that characters that IDNA does not permit do not appear in it. Those characters are identified in the "DISALLOWED" and "UNASSIGNED" lists that are specified in [IDNA2008-Tables] and described informally in [IDNA2008-Rationale]. Characters that are either DISALLOWED or UNASSIGNED MUST NOT be part of labels to be processed for registration in the DNS.

4.3.3. Label Validation

The proposed label (in the form of a Unicode string, i.e., a putative U-label) is then examined, performing tests that require examination of more than one character.

4.3.3.1. Rejection of Hyphen Sequences in U-labels

The Unicode string MUST NOT contain "--" (two consecutive hyphens) in the third and fourth character positions when the label is considered in "on the wire" order.

4.3.3.2. Leading Combining Marks

The first character of the string (when the label is considered in "on the wire" order) is examined to verify that it is not a combining mark. If it is a combining mark, the string MUST NOT be registered.

4.3.3.3. Contextual Rules

Each code point is checked for its identification as a character requiring contextual processing for registration (the list of characters appears as the combination of CONTEXTJ and CONTEXTO in [IDNA2008-Tables] as do the contextual rules themselves). If that indication appears, the table of contextual rules is checked for a rule for that character. If no rule is found, the proposed label is rejected and MUST NOT be installed in a zone file. If one is found, it is applied (typically as a test on the entire label or on adjacent characters within the label). If the application of the rule does not conclude that the character is valid in context, the proposed label MUST BE rejected. (See the IANA Considerations: IDNA Context Registry section of [IDNA2008-Tables].)

These contextual rules are required to support the use of characters that could be used, under other conditions, to produce misleading labels or to cause unacceptable ambiguity in label matching and interpretation. For example, labels containing invisible ("zero-

width") characters may be permitted in context with characters whose

presentation forms are significantly changed by the presence or absence of the zero-width characters, while other labels in which zero-width characters appear may be rejected.

4.3.3.4. Labels Containing Characters Written Right to Left

Additional special tests for right-to-left strings are applied. Strings that contain right to left characters that do not conform to the rule(s) identified in [IDNA2008-BIDI] MUST NOT be inserted as labels in zone files.

4.3.4. Registration Validation Summary

Strings that contain at least one non-ASCII character, have been produced by the steps above, whose contents pass the above tests, and are 63 or fewer characters long in ACE form (see Section 4.5), are U-labels.

To summarize, tests are made in Section 4.3 for invalid characters, invalid combinations of characters, and for labels that are invalid even if the characters they contain are valid individually.

4.4. Registry Restrictions

Registries at all levels of the DNS, not just the top level, are expected to establish policies about the labels that may be registered, and for the processes associated with that action. While exact policies are not specified as part of IDNA2008 and it is expected that different registries may specify different policies, there SHOULD be policies. Even a trivial policy (e.g., "anything can be registered in this zone that can be represented as an A-label - U-label pair") has value because it provides notice to users and applications implementers that the registry cannot be relied upon to provide even minimal user-protection restrictions. These per-registry policies and restrictions are an essential element of the IDNA registration protocol even for registries (and corresponding zone files) deep in the DNS hierarchy. As discussed in [IDNA2008-Rationale], such restrictions have always existed in the DNS. That document also contains a discussion and recommendations about possible types of rules.

The string produced by the above steps is checked and processed as appropriate to local registry restrictions. Application of those registry restrictions may result in the rejection of some labels or the application of special restrictions to others.

4.5. Punycode Conversion

The resulting U-label is converted to an A-label. The A-label, more precisely defined elsewhere, is the encoding of the U-label according to the Punycode algorithm [RFC3492] with the ACE prefix "xn--" added at the beginning of the string. This document updates RFC 3492 only to the extent of replacing the reference to the discussion of the ACE prefix. The ACE prefix is now specified in this document rather than as part of RFC 3490 or Nameprep [RFC3491] but is the same in both sets of documents.

The failure conditions identified in the Punycode encoding procedure cannot occur if the input is a U-label as determined by the steps above.

4.6. Insertion in the Zone

The A-label is registered in the DNS by insertion into a zone.

5. Domain Name Lookup Protocol

Lookup is conceptually different from registration and different tests are applied on the client. Although some validity checks are necessary to avoid serious problems with the protocol (see Section 5.5ff.), the lookup-side tests are more permissive and rely on the assumption that names that are present in the DNS are valid. That assumption is, however, a weak one because the presence of wild cards in the DNS might cause a string that is not actually registered in the DNS to be successfully looked up.

5.1. Label String Input

The user supplies a string in the local character set, typically by typing it or clicking on, or copying and pasting, a resource identifier, e.g., a URI [RFC3986] or IRI [RFC3987] from which the domain name is extracted. Alternately, some process not directly involving the user may read the string from a file or obtain it in some other way. Processing in this step and the next two are local matters, to be accomplished prior to actual invocation of IDNA, but at least the two steps in Section 5.2 and Section 5.3 must be accomplished in some way.

5.2. Conversion to Unicode

The string is converted from the local character set into Unicode, if it is not already Unicode. The exact nature of this conversion is

beyond the scope of this document, but may involve normalization as

described in Section 4.2. The result MUST be a Unicode string in NFC
form.

5.3. Character Changes in Preprocessing or the User Interface

The Unicode string MAY then be processed to prevent confounding of user expectations. For instance, it might be reasonable, at this step, to convert all upper case characters to lower case, if this makes sense in the user's environment, but even this should be approached with caution due to some edge cases: in the long term, it is probably better for users to understand IDNs strictly in lower-case, U-label, form. More generally, preprocessing may be useful to smooth the transition from IDNA2003, especially for direct user input, but with similar cautions. In general, IDNs appearing in files and those transmitted across the network as part of protocols are expected to be in either ASCII form (including A-labels) or to contain U-labels, rather than being in forms requiring mapping or other conversions.

Other examples of processing for localization might be applied, especially to direct user input, at this point. They include interpreting various characters as separating domain name components from each other (label separators) because they either look like periods or are used to separate sentences, mapping halfwidth or fullwidth East Asian characters to the common form permitted in labels, or giving special treatment to characters whose presentation forms are dependent only on placement in the label. Such localization changes are also outside the scope of this specification.

Recommendations for preprocessing for global contexts (i.e., when local considerations do not apply or cannot be used) and for maximum interoperability with labels that might have been specified under liberal readings of IDNA2003 are given in [IDNA2008-Rationale]. It is important to note that the intent of these specifications is that labels in application protocols, files, or links are intended to be in U-label or A-label form. Preprocessing MUST NOT map a character that is valid in a label as specified elsewhere in this document or in [IDNA2008-Tables] into another character. Excessively liberal use
of preprocessing, especially to strings stored in files, poses a threat to consistent and predictable behavior for the user even if not to actual interoperability.

Because these transformations are local, it is important that domain names that might be passed between systems (e.g., in IRIs) be U-labels or A-labels and not forms that might be accepted locally as a consequence of this step. This step is not standardized as part of
IDNA, and is not further specified here.

5.4. A-label Input

If the input to this procedure appears to be an A-label (i.e., it starts in "xn--"), the lookup application MAY attempt to convert it to a U-label and apply the tests of Section 5.5 and the conversion of Section 5.6 to that form. If the label is converted to Unicode (i.e., to U-label form) using the Punycode decoding algorithm, then the processing specified in those two sections MUST be performed, and the label MUST be rejected if the resulting label is not identical to the original. See also Section 6.1.

That conversion and testing SHOULD be performed if the domain name will later be presented to the user in native character form (this requires that the lookup application be IDNA-aware). If those steps are not performed, the lookup process SHOULD at least make tests to determine that the string is actually an A-label, examining it for the invalid formats specified in the Punycode decoding specification.
Applications that are not IDNA-aware will obviously omit that testing; others MAY treat the string as opaque to avoid the additional processing at the expense of providing less protection and information to users.

5.5. Validation and Character List Testing

As with the registration procedure, the Unicode string is checked to verify that all characters that appear in it are valid as input to IDNA lookup processing. As discussed above and in [IDNA2008-Rationale], the lookup check is more liberal than the registration one. Putative labels with any of the following characteristics MUST BE rejected prior to DNS lookup:

- Labels containing code points that are unassigned in the version of Unicode being used by the application, i.e., in the "Unassigned" Unicode category or the UNASSIGNED category of [IDNA2008-Tables].

- Labels that are not in NFC form.

- Labels containing prohibited code points, i.e., those that are assigned to the "DISALLOWED" category in the permitted character table [IDNA2008-Tables].

- Labels containing code points that are shown in the permitted character table as requiring a contextual rule and that are flagged as requiring exceptional special processing on lookup ("CONTEXTJ" in the Tables) but do not conform to that rule.

- Labels containing other code points that are shown in the permitted character table as requiring a contextual rule ("CONTEXTO" in the tables), but for which no such rule appears in the table of rules. Applications resolving DNS names or carrying out equivalent operations are not required to test contextual rules for "CONTEXTO" characters, only to verify that a rule exists (although they MAY make such tests to give better information to the user).

- Labels whose first character is a combining mark.

In addition, the application SHOULD apply the following test. The test may be omitted in special circumstances, such as when the lookup application knows that the conditions are enforced elsewhere, because an attempt to look up and resolve such strings will almost certainly lead to a DNS lookup failure except when wildcards are present in the zone. However, applying the test is likely to give much better information about the reason for a lookup failure -- information that may be usefully passed to the user when that is feasible -- than DNS resolution failure information alone. In any event, lookup applications should avoid attempting to resolve labels that are invalid under that test.

- Verification that the string is compliant with the requirements for right to left characters, specified in [IDNA2008-BIDI].

For all other strings, the lookup application MUST rely on the presence or absence of labels in the DNS to determine the validity of those labels and the validity of the characters they contain. If they are registered, they are presumed to be valid; if they are not, their possible validity is not relevant. A lookup application that declines to process a string that conforms to the rules above and does not look it up in the DNS is not in conformance with this protocol.

5.6. Punycode Conversion

The validated string, a U-label, is converted to an A-label using the Punycode algorithm with the ACE prefix added.

5.7. DNS Name Resolution

The A-label is looked up in the DNS, using normal DNS resolver procedures.

6. Name Server Considerations

[[anchor16: Note in draft: If we really want this document to contain only information that is necessary to proper implementation of IDNA by implementers who are familiar with the DNS, the material in this section is either tutorial, explanatory, or totally unnecessary. Should some or all of it be moved back to Rationale?]]

6.1. Processing Non-ASCII Strings

Existing DNS servers do not know the IDNA rules for handling non-ASCII forms of IDNs, and therefore need to be shielded from them. All existing channels through which names can enter a DNS server database (for example, master files (as described in RFC 1034) and DNS update messages [RFC2136]) are IDN-unaware because they predate IDNA. Other sections of this document provide the needed shielding by ensuring that internationalized domain names entering DNS server databases through such channels have already been converted to their equivalent ASCII A-label forms.

Because of the design of the algorithms in Section 4 and Section 5 (a domain name containing only ASCII codepoints can not be converted to an A-label), there can not be more than one A-label form for any given U-label.

As specified in RFC 2181 [RFC2181], the DNS protocol explicitly allows domain labels to contain octets beyond the ASCII range (0000..007F), and this document does not change that. Note, however, that there is no defined interpretation of octets 0080..00FF as characters. If labels containing these octets are returned to applications, unpredictable behavior could result. The A-label form, which cannot contain those characters, is the only standard representation for internationalized labels in the DNS protocol.

6.2. DNSSEC Authentication of IDN Domain Names

DNS Security [RFC2535] is a method for supplying cryptographic verification information along with DNS messages. Public Key Cryptography is used in conjunction with digital signatures to provide a means for a requester of domain information to authenticate the source of the data. This ensures that it can be traced back to a trusted source, either directly or via a chain of trust linking the source of the information to the top of the DNS hierarchy.

IDNA specifies that all internationalized domain names served by DNS servers that cannot be represented directly in ASCII MUST use the

A-label form. Conversion to A-labels MUST be performed prior to a zone being signed by the private key for that zone. Because of this

ordering, it is important to recognize that DNSSEC authenticates a domain name containing A-labels or conventional LDH-labels, not U-labels. In the presence of DNSSEC, no form of a zone file or query response that contains a U-label may be signed or the signature validated.

One consequence of this for sites deploying IDNA in the presence of DNSSEC is that any special purpose proxies or forwarders used to transform user input into IDNs must be earlier in the lookup flow than DNSSEC authenticating nameservers for DNSSEC to work.

6.3. Root and other DNS Server Considerations

IDNs in A-label form will generally be somewhat longer than current domain names, so the bandwidth needed by the root servers is likely to go up by a small amount. Also, queries and responses for IDNs will probably be somewhat longer than typical queries historically, so EDNS0 [RFC2671] support may be more important (otherwise, queries and responses may be forced to go to TCP instead of UDP).

7. Security Considerations

The general security principles and issues for IDNA appear in [IDNA2008-Defs] with additional explanation in [IDNA2008-Rationale]. The comments below are specific to the registration and loopup protocols specified in this document, but should be read in the context of the material in the first of those documents and the definitions and specifications, identified there, on which this one depends.

This memo 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 Requirement 2 of Section 3.1. 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.

8. IANA Considerations

IANA actions for this version of IDNA are specified in [IDNA2008-Tables] and discussed informally in [IDNA2008-Rationale]. The components of IDNA described in this document do not require any IANA actions.

9. Contributors

While the listed editor held the pen, the original versions of this document represent the joint work and conclusions of an ad hoc design team consisting of the editor and, in alphabetic order, Harald Alvestrand, Tina Dam, Patrik Faltstrom, and Cary Karp. This document draws significantly on the original version of IDNA [RFC3490] both conceptually and for specific text. This second-generation version would not have been possible without the work that went into that first version and its authors, Patrik Faltstrom, Paul Hoffman, and Adam Costello. While Faltstrom was actively involved in the creation of this version, Hoffman and Costello were not and should not be held responsible for any errors or omissions.

10. Acknowledgements

This revision to IDNA would have been impossible without the accumulated experience since RFC 3490 was published and resulting comments and complaints of many people in the IETF, ICANN, and other communities, too many people to list here. Nor would it have been possible without RFC 3490 itself and the efforts of the Working Group that defined it. Those people whose contributions are acknowledged in RFC 3490, [RFC4690], and [IDNA2008-Rationale] were particularly important.

Specific textual changes were incorporated into this document after suggestions from the other contributors, Stephane Bortzmeyer, Vint

Cerf, Mark Davis, Paul Hoffman, Kent Karlsson, Erik van der Poel, Marcos Sanz, Andrew Sullivan, Ken Whistler, and other WG participants. Special thanks are due to Paul Hoffman for permission to extract material from his Internet-Draft to form the basis for Appendix A

11. References

11.1. Normative 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-Defs] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework", November 2008, <https://datatracker.ietf.org/drafts/draft-ietf-idnabis-defs/>.

[IDNA2008-Tables] Faltstrom, P., "The Unicode Codepoints 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

[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.

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

[Unicode-PropertyValueAliases]

The Unicode Consortium, "Unicode Character Database: PropertyValueAliases", March 2008, <http://www.unicode.org/Public/UNIDATA/PropertyValueAliases.txt>.

[Unicode-RegEx]
The Unicode Consortium, "Unicode Technical Standard #18: Unicode Regular Expressions", May 2005, <http://www.unicode.org/reports/tr18/>.

[Unicode-Scripts]
The Unicode Consortium, "Unicode Standard Annex #24: Unicode Script Property", February 2008, <http://www.unicode.org/reports/tr24/>.

[Unicode-UAX15]
The Unicode Consortium, "Unicode Standard Annex #15: Unicode Normalization Forms", 2006, <http://www.unicode.org/reports/tr15/>.

11.2. Informative 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.

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

[RFC2136] Vixie, P., Thomson, S., Rekhter, Y., and J. Bound, "Dynamic Updates in the Domain Name System (DNS UPDATE)", RFC 2136, April 1997.

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

[RFC2535] Eastlake, D., "Domain Name System Security Extensions", RFC 2535, March 1999.

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

[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.

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

[RFC3987] Duerst, M. and M. Suignard, "Internationalized Resource Identifiers (IRIs)", RFC 3987, January 2005.

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

[RFC4952] Klensin, J. and Y. Ko, "Overview and Framework for Internationalized Email", RFC 4952, July 2007.

[Unicode] The Unicode Consortium, "The Unicode Standard, Version 5.0", 2007.

Boston, MA, USA: Addison-Wesley. ISBN 0-321-48091-0

Appendix A. Summary of Major Changes from IDNA2003

1. Update base character set from Unicode 3.2 to Unicode version-agnostic.

2. Separate the definitions for the "registration" and "lookup" activities.

3. Disallow symbol and punctuation characters except where special exceptions are necessary.

4. Remove the mapping and normalization steps from the protocol and have them instead done by the applications themselves, possibly in a local fashion, before invoking the protocol.

5. Change the way that the protocol specifies which characters are allowed in labels from "humans decide what the table of codepoints contains" to "decision about codepoints are based on Unicode properties plus a small exclusion list created by humans".

6. Introduce the new concept of characters that can be used only in specific contexts.

7. Allow typical words and names in languages such as Dhivehi and Yiddish to be expressed.

8. Make bidirectional domain names (delimited strings of labels, not just labels standing on their own) display in a less surprising fashion whether they appear in obvious domain name contexts or as part of running text in paragraphs.

9. Remove the dot separator from the mandatory part of the protocol.

10. Make some currently-valid labels that are not actually IDNA labels invalid.

Appendix B. Change Log

[[anchor25: RFC Editor: Please remove this appendix.]]

B.1. Changes between Version -00 and -01 of draft-ietf-idnabis-protocol

- Corrected discussion of SRV records.
- Several small corrections for clarity.
- Inserted more "open issue" placeholders.

B.2. Version -02

- Rewrote the "conversion to Unicode" text in Section 5.2 as requested on-list.
- Added a comment (and reference) about EDNS0 to the "DNS Server Conventions" section, which was also retitled.
- Made several editorial corrections and improvements in response to various comments.
- Added several new discussion placeholder anchors and updated some older ones.

B.3. Version -03

- Trimmed change log, removing information about pre-WG drafts.
- Incorporated a number of changes suggested by Marcos Sanz in his note of 2008.07.17 and added several more placeholder anchors.
- Several minor editorial corrections and improvements.
- "Editor" designation temporarily removed because the automatic posting machinery does not accept it.

B.4. Version -04

- Removed Contextual Rule appendices for transfer to Tables.
- Several changes, including removal of discussion anchors, based on discussions at IETF 72 (Dublin)
- Rewrote the preprocessing material (Section 5.3) somewhat.

B.5. Version -05

- Updated part of the A-label input explanation (Section 5.4) per note from Erik van der Poel.

B.6. Version -06

- Corrected a few typographical errors.
- Incorporated the material (formerly in Rationale) on the relationship between IDNA2003 and IDNA2008 as an appendix and pointed to the new definitions document.
- Text modified in several places to recognize the dangers of interaction between DNS wildcards and IDNs.
- Text added to be explicit about the handling of edge and failure cases in Punycode encoding and decoding.
- Revised for consistency with the new Definitions document and to make the text read more smoothly.

B.7. Version -07

- Multiple small textual and editorial changes and clarifications.

- Requirement for normalization clarified to apply to all cases and conditions for preprocessing further clarified.

- Substantive change to Section 4.3.1, turning a SHOULD to a MUST (see note from Mark Davis, 19 November, 2008 18:14 -0800).

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

Copyright (C) The IETF Trust (2008).

This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.