INTERNET-DRAFT
Updates RFC 1034, 1035
Expires January 2006

Donald E. Eastlake 3rd
Motorola Laboratories
July 2005

Domain Name System (DNS) Case Insensitivity Clarification

<draft-ietf-dnsext-insensitive-06.txt>

Donald E. Eastlake 3rd

Status of This Document

Copyright Notice

Abstract

Domain Name System (DNS) names are "case insensitive". This document explains exactly what that means and provides a clear specification of the rules. This clarification updates RFCs 1034 and 1035.

Acknowledgements

The contributions to this document of Rob Austein, Olafur Gudmundsson, Daniel J. Anderson, Alan Barrett, Marc Blanchet, Dana, Andreas Gustafsson, Andrew Main, Thomas Narten, and Scott Seligman are gratefully acknowledged.

Table of Contents

1. Introduction

The Domain Name System (DNS) is the global hierarchical replicated distributed database system for Internet addressing, mail proxy, and other information. Each node in the DNS tree has a name consisting of zero or more labels [STD 13][RFC 1591, 2606] that are treated in a case insensitive fashion. This document clarifies the meaning of "case insensitive" for the DNS. This clarification updates RFCs 1034 and 1035 [STD 13].

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].

2. Case Insensitivity of DNS Labels

DNS was specified in the era of [ASCII]. DNS names were expected to look like most host names or Internet email address right halves (the part after the at-sign, "@") or be numeric as in the in-addr.arpa part of the DNS name space. For example,

```
    foo.example.net.
    aol.com.
    www.gnu.ai.mit.edu.
or  69.2.0.192.in-addr.arpa.
```

Case varied alternatives to the above would be DNS names like

```
    Foo.ExamplE.net.
    AOL.COM.
    WWW.gnu.AI.mit.EDU.
or  69.2.0.192.in-ADDR.ARPA.
```

However, the individual octets of which DNS names consist are not limited to valid ASCII character codes. They are 8-bit bytes and all values are allowed. Many applications, however, interpret them as ASCII characters.

2.1 Escaping Unusual DNS Label Octets

In Master Files [STD 13] and other human readable and writable ASCII contexts, an escape is needed for the byte value for period (0x2E, ".") and all octet values outside of the inclusive range of 0x21 ("!") to 0x7E ("~"). That is to say, 0x2E and all octet values in the two inclusive ranges 0x00 to 0x20 and 0x7F to 0xFF.

One typographic convention for octets that do not correspond to an ASCII printing graphic is to use a back-slash followed by the value of the octet as an unsigned integer represented by exactly three decimal digits.

The same convention can be used for printing ASCII characters so that they will be treated as a normal label character. This includes the back-slash character used in this convention itself which can be expressed as \092 or \\ and the special label separator period (".") which can be expressed as and \046 or \. respectively. It is advisable to avoid using a backslash to quote an immediately following non-printing ASCII character code to avoid implementation difficulties.

A back-slash followed by only one or two decimal digits is undefined. A back-slash followed by four decimal digits produces two octets, the first octet having the value of the first three digits considered as a decimal number and the second octet being the character code for the fourth decimal digit.

2.2 Example Labels with Escapes

The first example below shows embedded spaces and a period (".") within a label. The second one show a 5-octet label where the second octet has all bits zero, the third is a backslash, and the fourth octet has all bits one.

```
      Donald\032E\.\032Eastlake\0323rd.example.
and   a\000\\\255z.example.
```

3. Name Lookup, Label Types, and CLASS

The original DNS design decision was made that comparisons on name lookup for DNS queries should be case insensitive [STD 13]. That is to say, a lookup string octet with a value in the inclusive range of 0x41 to 0x5A, the upper case ASCII letters, MUST match the identical value and also match the corresponding value in the inclusive range 0x61 to 0x7A, the lower case ASCII letters. And a lookup string octet with a lower case ASCII letter value MUST similarly match the identical value and also match the corresponding value in the upper case ASCII letter range.

(Historical Note: the terms "upper case" and "lower case" were invented after movable type. The terms originally referred to the two font trays for storing, in partitioned areas, the different

physical type elements. Before movable type, the nearest equivalent

terms were "majuscule" and "minuscule".)

One way to implement this rule would be, when comparing octets, to subtract 0x20 from all octets in the inclusive range 0x61 to 0x7A before the comparison. Such an operation is commonly known as "case folding" but implementation via case folding is not required. Note that the DNS case insensitivity does NOT correspond to the case folding specified in [iso-8859-1] or [iso-8859-2]. For example, the octets 0xDD (\221) and 0xFD (\253) do NOT match although in other contexts, where they are interpreted as the upper and lower case version of "Y" with an acute accent, they might.

3.1 Original DNS Label Types

DNS labels in wire-encoded names have a type associated with them. The original DNS standard [RFC 1035] had only two types. ASCII labels, with a length of from zero to 63 octets, and indirect (or compression) labels which consist of an offset pointer to a name location elsewhere in the wire encoding on a DNS message. (The ASCII label of length zero is reserved for use as the name of the root node of the name tree.) ASCII labels follow the ASCII case conventions described herein and, as stated above, can actually contain arbitrary byte values. Indirect labels are, in effect, replaced by the name to which they point which is then treated with the case insensitivity rules in this document.

3.2 Extended Label Type Case Insensitivity Considerations

DNS was extended by [RFC 2671] to have additional label type numbers available. (The only such type defined so far is the BINARY type [RFC 2673] which is now Experimental [RFC 3363].)

The ASCII case insensitivity conventions only apply to ASCII labels, that is to say, label type 0x0, whether appearing directly or invoked by indirect labels.

3.3 CLASS Case Insensitivity Considerations

As described in [STD 13] and [RFC 2929], DNS has an additional axis for data location called CLASS. The only CLASS in global use at this time is the "IN" or Internet CLASS.

The handling of DNS label case is not CLASS dependent. With the

original design of DNS, it was intended that a recursive DNS resolver

be able to handle new CLASSes that were unknown at the time of its implementation. This requires uniform handling of label case insensitivity. Should it become desireable, for example, to allocate a CLASS with "case sensitive ASCII labels" for example, it would be necessary to allocate a new label type for these labels.

4. Case on Input and Output

While ASCII label comparisons are case insensitive, [STD 13] says case MUST be preserved on output, and preserved when convenient on input. However, this means less than it would appear since the preservation of case on output is NOT required when output is optimized by the use of indirect labels, as explained below.

4.1 DNS Output Case Preservation

[STD 13] views the DNS namespace as a node tree. ASCII output is as if a name was marshaled by taking the label on the node whose name is to be output, converting it to a typographically encoded ASCII string, walking up the tree outputting each label encountered, and preceding all labels but the first with a period ("."). Wire output follows the same sequence but each label is wire encoded and no periods inserted. No "case conversion" or "case folding" is done during such output operations, thus "preserving" case. However, to optimize output, indirect labels may be used to point to names elsewhere in the DNS answer. In determining whether the name to be pointed to, for example the QNAME, is the "same" as the remainder of the name being optimized, the case insensitive comparison specified above is done. Thus such optimization may easily destroy the output preservation of case. This type of optimization is commonly called "name compression".

4.2 DNS Input Case Preservation

Originally, DNS data came from an ASCII Master File as defined in [STD 13] or a zone transfer. DNS Dynamic update and incremental zone transfers [RFC 1995] have been added as a source of DNS data [RFC 2136, 3007]. When a node in the DNS name tree is created by any of such inputs, no case conversion is done. Thus the case of ASCII labels is preserved if they are for nodes being created. However, when a name label is input for a node that already exist in DNS data being held, the situation is more complex. Implementations are free to retain the case first loaded for such a label or allow new input

to override the old case or even maintain separate copies preserving

the input case.

For example, if data with owner name "foo.bar.example" is loaded and then later data with owner name "xyz.BAR.example" is input, the name of the label on the "bar.example" node, i.e. "bar", might or might not be changed to "BAR" in the DNS stored data or the actual input case could be preserved. Thus later retrieval of data stored under "xyz.bar.example" in this case can return all data with "xyz.BAR.example" or all data with "xyz.bar.example" or even, when more than one RR is being returned, a mixture of these two cases. This last case is unlikely because optimization of answer length through indirect labels tends to cause only copy of the name tail ("bar.example" or "BAR.example") to be used for all returned RRs. Note that none of this has any effect on the number of completeness of the RR set returned, only on the case of the names in the RR set returned.

The same considerations apply when inputting multiple data records with owner names differing only in case. For example, if an "A" record is the first resourced record stored under owner name "xyz.BAR.example" and then a second "A" record is stored under "XYZ.BAR.example", the second MAY be stored with the first (lower case initial label) name or the second MAY override the first so that only an upper case initial label is retained or both capitalizations MAY be kept in the DNS stored data. In any case, a retrieval with either capitalization will retrieve all RRs with either capitalization.

Note that the order of insertion into a server database of the DNS name tree nodes that appear in a Master File is not defined so that the results of inconsistent capitalization in a Master File are unpredictable output capitalization.

5. Internationalized Domain Names

A scheme has been adopted for "internationalized domain names" and "internationalized labels" as described in [RFC 3490, 3454, 3491, and 3492]. It makes most of [UNICODE] available through a separate application level transformation from internationalized domain name to DNS domain name and from DNS domain name to internationalized domain name. Any case insensitivity that internationalized domain names and labels have varies depending on the script and is handled entirely as part of the transformation described in [RFC 3454] and [RFC 3491] which should be seen for further details. This is not a part of the DNS as standardized in STD 13.

6. Security Considerations

The equivalence of certain DNS label types with case differences, as clarified in this document, can lead to security problems. For example, a user could be confused by believing two domain names differing only in case were actually different names.

Furthermore, a domain name may be used in contexts other than the DNS. It could be used as a case sensitive index into some data base or file system. Or it could be interpreted as binary data by some integrity or authentication code system. These problems can usually be handled by using a standardized or "canonical" form of the DNS ASCII type labels, that is, always mapping the ASCII letter value octets in ASCII labels to some specific pre-chosen case, either upper case or lower case. An example of a canonical form for domain names (and also a canonical ordering for them) appears in Section 6 of [RFC 4034]. See also [RFC 3597].

Finally, a non-DNS name may be stored into DNS with the false expectation that case will always be preserved. For example, although this would be quite rare, on a system with case sensitive email address local parts, an attempt to store two "RP" records that differed only in case would probably produce unexpected results that might have security implications. That is because the entire email address, including the possibly case sensitive local or left hand part, is encoded into a DNS name in a readable fashion where the case of some letters might be changed on output as described above.

Copyright and Disclaimer

Normative References

[ASCII] - ANSI, "USA Standard Code for Information Interchange", X3.4, American National Standards Institute: New York, 1968.

[RFC 1034, 1035] - See [STD 13].

[RFC 1995] - M. Ohta, "Incremental Zone Transfer in DNS", August 1996.

[RFC 2119] - S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", March 1997.

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

[RFC 3007] - B. Wellington, "Secure Domain Name System (DNS) Dynamic Update", November 2000.

[RFC 3597] - Andreas Gustafsson, "Handling of Unknown DNS RR Types", draft-ietf-dnsext-unknown-rrs-05.txt, March 2003.

[RFC 4034} - Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", RFC 4034, March 2005.

[STD 13]
- P. Mockapetris, "Domain names - concepts and facilities", RFC 1034, November 1987.
- P. Mockapetris, "Domain names - implementation and specification", RFC 1035, November 1987.

Informative References

[ISO 8859-1] - International Standards Organization, Standard for Character Encodings, Latin-1.

[ISO 8859-2] - International Standards Organization, Standard for Character Encodings, Latin-2.

[RFC 1591] - J. Postel, "Domain Name System Structure and Delegation", March 1994.

[RFC 2606] - D. Eastlake, A. Panitz, "Reserved Top Level DNS Names", June 1999.

[RFC 2929] - D. Eastlake, E. Brunner-Williams, B. Manning, "Domain Name System (DNS) IANA Considerations", September 2000.

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

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

[RFC 3092] - D. Eastlake 3rd, C. Manros, E. Raymond, "Etymology of Foo", 1 April 2001.

[RFC 3363] - Bush, R., Durand, A., Fink, B., Gudmundsson, O., and T. Hain, "Representing Internet Protocol version 6 (IPv6) Addresses in the Domain Name System (DNS)", RFC 3363, August 2002.

[RFC 3454] - P. Hoffman, M. Blanchet, "Preparation of Internationalized String ("stringprep")", December 2002.

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

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

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

[UNICODE] - The Unicode Consortium, "The Unicode Standard", <http://www.unicode.org/unicode/standard/standard.html>.

Changes Between Draft Version

RFC Editor: The following summaries of changes between draft versions are to be removed before publication.

-02 to -03 Changes

The following changes were made between draft version -02 and -03:

1. Add internationalized domain name section and references.

2. Change to indicate that later input of a label for an existing DNS name tree node may or may not be normalized to the earlier input or override it or both may be preserved.

3. Numerous minor wording changes.

-03 to -04 Changes

The following changes were made between draft versions -03 and -04:

1. Change to conform to the new IPR, Copyright, etc., notice requirements.

2. Change in some section headers for clarity.

3. Drop section on wildcards.

4. Add emphasis on loss of case preservation due to name compression.

5. Add references to RFCs 1995 and 3092.

-04 to -05 Changes

The following changes were made between draft versions -04 and -05:

1. More clearly state that this draft updates RFCs 1034, 1035 [STD 13].

2. Add informative references to ISO 8859-1 and ISO 8859-2.

3. Fix hyphenation and capitalization nits.

-05 to -06 Changes

The following changes were made between draft version -05 and -06.

1. Add notation to the RFC Editor that the draft version change summaries are to be removed before RFC publication.

2. Additional text explaining why labe case insensitivity is CLASS independent.

3. Changes and additional text clarifying that the fact that inconsistent case in data loaded into DNS may result in unpredicatable or inconsistent case in DNS storage but has no effect on the completeness of RR sets retrieved.

4. Add reference to [RFC 3363] and update reference to [RFC 2535] to be to [RFC 4034].

Author's Address

Donald E. Eastlake 3rd
Motorola Laboratories
155 Beaver Street
Milford, MA 01757 USA

Telephone: +1 508-786-7554 (w)

EMail: Donald.Eastlake@motorola.com

Expiration and File Name

This draft expires January 2006.

Its file name is draft-ietf-dnsext-insensitive-06.txt.