

INTERNET-DRAFT
Clarifies STD0013
Expires August 2003

Donald E. Eastlake 3rd
Motorola Laboratories
February 2003

Domain Name System (DNS) Case Insensitivity Clarification

<[draft-ietf-dnsext-insensitive-01.txt](#)>

Donald E. Eastlake 3rd

Status of This Document

Distribution of this document is unlimited. Comments should be sent to the DNSEXT working group at namedroppers@ops.ietf.org.

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of RFC 2026](#). Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/1id-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

Abstract

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 should not have any interoperability consequences.

Acknowledgements

The contributions to this document of Rob Austein, Olafur Gudmundsson, Daniel J. Anderson, Alan Barrett, and Andrew Main are gratefully acknowledged.

Table of Contents

Status of This Document.....	1
Abstract.....	1
Acknowledgements.....	2
Table of Contents.....	2
1 . Introduction.....	3
2 . Case Insensitivity of DNS Labels.....	3
2.1 Escaping Unusual DNS Label Octets.....	3
2.2 Example Labels with Escapes.....	4
2.3 Name Lookup Case Insensitivity.....	4
2.4 Original DNS Label Types.....	5
3 . Additional DNS Case Insensitivity Considerations.....	5
3.1 CLASS Case Insensitivity Considerations.....	5
3.2 Extended Label Type Case Insensitivity Considerations..	5
4 . Case on Input and Output.....	6
4.1 DNS Output Case Preservation.....	6
4.2 DNS Input Case Preservation.....	6
4.3 Wildcard Matching.....	7
5 . Security Considerations.....	7
Normative References.....	9
Informative References.....	9
Author's Address.....	10
Expiration and File Name.....	10

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] which have been specified as being treated in a case insensitive fashion. This document clarifies the meaning of "case insensitive" for this application.

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

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

2.1 Escaping Unusual DNS Label Octets

An escape is needed for all octet values outside of the inclusive range of 0x21 ("!") to 0x7E ("~"). That is to say, 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 show them as 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. This includes the back-slash character used in this convention itself and the special label separator period (".") which can be expressed as `\092` and `\046` respectively.

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.

Octets, other than those corresponding to the ASCII digits 0 through 9, can also be protected from recognition, so that they will be treated as a normal label character, by a second convention: preceding them with a back-slash. This is the most commonly used technique for protecting back slash ("`\`") and period ("`.`"). However, it is advisable to avoid using this on other than printing ASCII characters.

2.2 Example Labels with Escapes

The first example below shows embedded spaces and a period (".") within a label. The second one show a 4 octet label where the second octet has all bits zero and the third octet has all bits one.

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

2.3 Name Lookup Case Insensitivity

The 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 became wide spread for printing. 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.

2.4 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 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 above. 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. Additional DNS Case Insensitivity Considerations

This section clarifies the effect of DNS CLASS and extended Label Type on case insensitivity.

3.1 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.

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].)

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

4. Case on Input and Output

While ASCII label comparisons are case insensitive, case MUST be preserved on output, except when output is optimized by the use of indirect labels, and preserved when possible on input.

4.1 DNS Output Case Preservation

[STD 13] views the DNS namespace as a node tree. ASCII output is as if a name was marshalled 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. 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 is the "same" as the remainder of the name being optimized, the case insensitive comparison specified above is done. Thus such optimization MAY destroy the output preservation of case. This type of optimization is commonly called "name compression".

4.2 DNS Input Case Preservation

Originally, DNS input came from an ASCII Master File as defined in [STD 13]. DNS Dynamic update has been added as a source of DNS data [RFC 2136, 3007]. When a node in the DNS name tree is created by such input, no case conversion is done and the case of ASCII labels is preserved if they are for nodes being created. However, no change is made in the name label on nodes that already exist if the DNS data being augmented or updated. It is quite common for higher level nodes to already exist.

For example, if data with owner name "foo.bar.example" is input 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", is not changed to

"BAR". Thus later retrieval of data stored under "xyz.bar.example" in

this case can easily result is obtaining data with "xyz.BAR.example". The same considerations apply inputting multiple data records with owner names differing only in case. From the example above, if an "A" record is stored under owner name "xyz.BAR.example" and then a second "A" record under "XYZ.BAR.example", the second will be stored at the node with the first (lower case initial label) name.

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.

4.3 Wildcard Matching

There is one additional instance of note, which reflects the general rules that output case reflects input case unless there is conflicting capitalization in the DNS database or the output case is hidden by name compression. This is when a query matches a wild card in the DNS database at a server. In that case, the answer SHOULD reflect the input case of the label or labels that matched the wildcard unless they are replaced by an indirect label which MAY point to a name with different capitalization.

5. 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 an index into some case sensitive data base 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 map 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 8 of \[RFC 2535\]](#). See also [\[UNKRR\]](#).

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

D. Eastlake 3rd

[Page 7]

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.

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 2119] - "Key words for use in RFCs to Indicate Requirement Levels", S. Bradner, 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 2535] - D. Eastlake, "Domain Name System Security Extensions", March 1999.

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

[STD 13]

- P. Mockapetris, "Domain names - concepts and facilities", [RFC 1034](#), November 1987.

- P. Mockapetris, "Domain names - implementation and specification", [RFC 1035](#), November 1987.

[UNKRR] - Andreas Gustafsson, "Handling of Unknown DNS RR Types", [draft-ietf-dnsext-unknown-rrs-04.txt](#), September 2002.

Informative References

[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.

Author's Address

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

Telephone: +1 508-851-8280 (w)
 +1 508-634-2066 (h)
EMail: Donald.Eastlake@motorola.com

Expiration and File Name

This draft expires August 2003.

Its file name is [draft-ietf-dnsext-insensitive-01.txt](#).

