

IETF
Internet-Draft
Intended status: Informational
Expires: August 17, 2016

A. Sullivan
Dyn, Inc.
February 14, 2016

The DNS Class Field is Not Useful
draft-sullivan-dns-class-useless-00

Abstract

Domain Name System Resource Records are identified in part by their class. The class field is not effective, and it is not used the way it appears to have been intended. This memo makes no recommendation about the DNS parameters registry, but urges those defining new RRTYPES to define them for all classes.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on August 17, 2016.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Internet-Draft

DNS CLASS Not Useful

February 2016

Table of Contents

| | | |
|-----------------------------|---|-------------------|
| 1. | Classes in the Domain Name System | 2 |
| 2. | Why classes are not as useful as they should be | 3 |
| 2.1. | Class data is in the wrong part of a resource record | 3 |
| 2.2. | Not all RRTYPEs are careful about class | 4 |
| 2.3. | A new class requires new delegations | 5 |
| 3. | DNS classes are effectively vestigial | 5 |
| 4. | Plea to authors: define new RRTYPEs for all classes | 5 |
| 5. | References | 5 |
| 5.1. | Informative References | 5 |
| 5.2. | URIs | 6 |
| Appendix A. | Discussion Venue | 6 |
| Appendix B. | Change History | 6 |
| | Author's Address | 6 |

[1.](#) Classes in the Domain Name System

The Domain Name System (DNS) [[RFC1034](#)] [[RFC1035](#)] includes two types of division: one by class, and one by "cuts". As [[RFC1034](#)] says,

The database for any class is organized, delegated, and maintained separately from all other classes. Since, by convention, the name spaces are the same for all classes, the separate classes can be thought of as an array of parallel namespace trees. Note that the data attached to nodes will be different for these different parallel classes. The most common reasons for creating a new class are the necessity for a new data format for existing types or a desire for a separately managed version of the existing name space.

As of this writing, there are only three "ordinary" classes assigned. Class 1 is the Internet or IN class. Class 3 is the Chaos or CH class. Class 4 is the Hesiod or HS class. Class 2 is noted in [[RFC1035](#)] as the CSNET or CS class, but the current registry (at [[1](#)]) no longer includes the assignment.

There are two other assigned classes; these have special purposes. Class 255, ANY or *, matches any class [[RFC1035](#)]. Class 254, NONE, is used in [[RFC2136](#)] to specify certain kinds of operations on RRsets.

Of the ordinary classes, only IN is found in common use on the Internet. Class CH is now sometimes used to carry certain kinds of name server metadata. It seems new classes might have been useful for a number of features that have been delivered in some other way.

Yet classes have not been used, which might suggest that classes are less useful than they otherwise appear to be.

Nevertheless, from time to time someone comes up with a suggestion for why a new class would solve some problem or other. The purpose of this memo is to offer some considerations for those contemplating such innovation.

[2.](#) Why classes are not as useful as they should be

There are three problems with classes that mean they cannot work as hoped. First, the class field is in the wrong part of the DNS message to be useful as a selector. Second, specification of resource record types has not always attended to the handling of types across classes, which means that new classes are likely to be, at best, of uncertain utility.

Apart from those technical problems, there is an administrative problem. The Internet name space starts from a common root, which means that any resolver needs to start from the same bootstrap mechanism no matter what classes it uses. But in order for that to work, the new class would need to be delegated from the root name servers.

[2.1.](#) Class data is in the wrong part of a resource record

As noted in [Section 1](#), classes are intended to divide the DNS into separate trees. Unfortunately, however, the class field does not appear in a convenient location in a DNS resource record (RR) to be effective for that purpose. DNS resource records provide the owner name before the class. As a practical matter, then, the namespace is primary.

The issues is made plain by considering the matching algorithm for name servers, described in [\[RFC1034\] section 4.3.2](#). Suppose there are two (imaginary) classes, EG and IE. Imagine, further, that the

same name has different RDATA in each class:

example.com EG CNAME example.net

example.com IE CNAME example.org

In principle, the above should work because, as noted in [Section 1](#), each class is supposed to be "delegated separately". Therefore, when the name server for example.com in class EG receives the query for example.com, it already knows which class database to search; similarly for example.com in class IE.

Yet while the class delegations are defined to be separate, there is of course no way to ensure that the NS record RDATA for example.com in class IE, and the NS record RDATA for example.com in class EG, are always different. Indeed, if the different classes of name space are truly managed separately, but the name space is by convention parallel, it would not be surprising that some name server ended up authoritative for the same name in different classes. (In [\[RFC1034\]](#), [section 3.6.1](#), there is an ambiguous example that appears to contain two classes from the same master file and for the same name.)

Therefore, the name server for example.com in every class needs to be able to identify the class for the QNAME before beginning to follow the resolution algorithm. But given the DNS message format, the name server cannot find the class until it knows the name. In effect, a multi-class name server needs to look at the QCLASS in the query it gets, and then select the class in question, before it starts trying to match name by name.

Once one describes the resolution pattern this way, it seems obvious that a class needs to be selected in order to resolve a QNAME. But in that case, it is rather inconvenient that the class field does not come first in a resource record; and it is not too surprising that, given that the IN class is so widely used and others so rarely, some naive implementations simply assume IN for every resource record. That assumption, of course, makes the class division in the DNS again less useful.

[2.2](#). Not all RRTYPEs are careful about class

Some RRTYPEs are defined in a class-dependent way. For instance, the

A record (type 1) is defined in [\[RFC1035\]](#) to be for class IN only. Perhaps unfortunately, the IANA registry for RRTYPEs (at [\[2\]](#)) does not include an indicator for the class in which the RRTYPE is defined.

In addition, not every RRTYPE specification is clear about its definition under various classes. For instance, the specification of type 55, HIP, appears not to state the class(es) for which it is defined [\[RFC5205\]](#). The specification of LOC, type 29 (in [\[RFC1876\]](#)), says that the type is defined for classes other than IN, but also says, "The semantics of such use are not defined by this memo."

One might argue that this issue is resolved by [\[RFC3597\]](#), because it specifies that an unknown class and type combination is to be handled as unknown. Formally, of course, that means that every type can be handled regardless of class. But it would appear to reduce the utility of classes yet further, by increasing the probability that every RR in every class except IN will be treated as unknown.

[2.3.](#) A new class requires new delegations

The Internet's DNS system is part of the common name space of the Internet, and that common name space starts from a common root (see [\[RFC2826\]](#) for the arguments about why this must be true). In order to provide for the resolution of a new class, the root name servers would need to respond to resolution requests for that class and provide the delegation data. Current policies about domain name delegation in the root zone appear to apply to the IN class, and it is not clear where responsibility would lie for the policies about a new class. At the very least, a new policy of this sort would need to be worked out.

[3.](#) DNS classes are effectively vestigial

Given the considerations above, it is plain that DNS classes are unlikely to be useful in the future. Designers of new name systems should consider the design of classes in the DNS. If a similar feature is desirable, its design needs to be different in order to be useful. Given the the way the DNS has managed to thrive effectively without classes, however, it would be worth asking whether the feature is useful at all.

4. Plea to authors: define new RRTYPEs for all classes

When defining a new RRTYPE, it is best to ensure that there is a clear definition of the class for which the RRTYPE is defined. In the absence of strong reasons to do otherwise, new types should be defined for all classes.

5. References

5.1. Informative References

- [RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, [RFC 1034](#), DOI 10.17487/RFC1034, November 1987, <<http://www.rfc-editor.org/info/rfc1034>>.
- [RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, [RFC 1035](#), DOI 10.17487/RFC1035, November 1987, <<http://www.rfc-editor.org/info/rfc1035>>.
- [RFC1876] Davis, C., Vixie, P., Goodwin, T., and I. Dickinson, "A Means for Expressing Location Information in the Domain Name System", [RFC 1876](#), DOI 10.17487/RFC1876, January 1996, <<http://www.rfc-editor.org/info/rfc1876>>.

Sullivan

Expires August 17, 2016

[Page 5]

Internet-Draft

DNS CLASS Not Useful

February 2016

- [RFC2136] Vixie, P., Ed., Thomson, S., Rekhter, Y., and J. Bound, "Dynamic Updates in the Domain Name System (DNS UPDATE)", [RFC 2136](#), DOI 10.17487/RFC2136, April 1997, <<http://www.rfc-editor.org/info/rfc2136>>.
- [RFC2826] Internet Architecture Board, "IAB Technical Comment on the Unique DNS Root", [RFC 2826](#), DOI 10.17487/RFC2826, May 2000, <<http://www.rfc-editor.org/info/rfc2826>>.
- [RFC3597] Gustafsson, A., "Handling of Unknown DNS Resource Record (RR) Types", [RFC 3597](#), DOI 10.17487/RFC3597, September 2003, <<http://www.rfc-editor.org/info/rfc3597>>.
- [RFC5205] Nikander, P. and J. Laganier, "Host Identity Protocol (HIP) Domain Name System (DNS) Extensions", [RFC 5205](#), DOI

10.17487/RFC5205, April 2008,
<<http://www.rfc-editor.org/info/rfc5205>>.

[5.2.](#) URIs

- [1] <http://www.iana.org/assignments/dns-parameters/dns-parameters.xml#dns-parameters-2>
- [2] <http://www.iana.org/assignments/dns-parameters/dns-parameters.xml#dns-parameters-4>

[Appendix A.](#) Discussion Venue

This Internet-Draft is discussed on the IAB Internet Names and Identifiers Program public list: inip-discuss@iab.org.

[Appendix B.](#) Change History

00:

- * Initial version

Author's Address

Andrew Sullivan
Dyn, Inc.
150 Dow St
Manchester, NH 03101
U.S.A.

Email: asullivan@dyn.com