

IETF
Internet-Draft
Updates: [6895](#) (if approved)
Intended status: Best Current Practice
Expires: January 9, 2017

A. Sullivan
Dyn, Inc.
July 8, 2016

**The DNS Is Not Classy: DNS Classes Considered Useless
draft-sullivan-dns-class-useless-03**

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 suspends additions to the DNS class registry pending greater clarity on how classes might be used, and until such clarification requires 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 January 9, 2017.

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.

Table of Contents

- [1. Classes in the Domain Name System](#) [2](#)
- [2. Why classes are not as useful as they might be](#) [3](#)
 - [2.1. Matching rules are class-independent](#) [3](#)
 - [2.1.1. Really parallel trees](#) [4](#)
 - [2.2. Not all RRTYPEs are careful about class](#) [5](#)
 - [2.3. The DNS RRTYPE registry and meaning](#) [6](#)
 - [2.4. A new class requires new delegations](#) [6](#)
- [3. DNS classes are effectively vestigial](#) [7](#)
- [4. New RRTYPEs are for all classes](#) [7](#)
- [5. Security considerations](#) [7](#)
- [6. IANA Considerations](#) [7](#)
- [7. Acknowledgements](#) [8](#)
- [8. References](#) [8](#)
 - [8.1. Normative References](#) [8](#)
 - [8.2. Informative References](#) [8](#)
- [Appendix A. Discussion Venue](#) [9](#)
- [Appendix B. Change History](#) [9](#)
- [Author's Address](#) [10](#)

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 <http://www.iana.org/assignments/dns-parameters/dns-parameters.xml#dns-parameters-2>) 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 today. 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. (It is also worth observing that classes divide the database, and not the namespace itself. In other words, classes are part of the implementation of the DNS and not a natural feature of domain names as such.) In some ways, the motivations for classes are made clearer by reading [[RFC0882](#)] along with the other DNS specifications.

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 might be

There are four (or, depending on how one counts, five) problems that make classes less useful than they might be. First, the rules for name matching are independent of the class. Second, specification of resource record types has not always attended to the handling of types across classes; this makes new classes of (at best) uncertain utility. Third, the DNS RRTYPE registry does not have a way of indicating significant differences in meaning for the same type among different classes.

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, any new class would need to be delegated from the existing root name servers, or else a new set of policies about how to select the alternative roots would be required. A wrinkle (or possibly a separate problem) is that some possible uses of classes appear not really to require a shared global root. (For instance, it is not clear that Hesiod names really needed to be part of the global namespace.)

2.1. Matching rules are class-independent

As noted in [Section 1](#), classes are intended to divide the DNS into separate trees. The class field does not, however, affect the

matching rules for names, so as a practical matter the namespace is primary.

The issue 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 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 example that appears to contain two classes from the same master file and for the same name. This illustrates the principle that the same name server could be authoritative for the same name in different classes. Note that the example might be a mistake, since according to [\[RFC1035\], section 5.2](#), all entries in a master file for a zone should have the same class. In any case, it is plain that the name is primary, and having matched the name one can then select data according to the class. But this means that the matching rules for names cannot differ across classes, and that makes classes less useful for extending DNS capabilities than they might at first seem.

Once one describes the resolution pattern this way, and given that the IN class is so widely used and other classes so rarely, it is not too surprising that some naive implementations simply assume IN for every resource record. That assumption, of course, makes the class division in the DNS again less useful.

[2.1.1](#). Really parallel trees

It appears that the notion of "parallel namespace trees" is stronger than one might have hoped if one wanted to use classes to do something new in the DNS. When one considers how classes are treated in [\[RFC1034\]](#) and [\[RFC1035\]](#) and their predecessors, that parallelism

becomes less surprising. The examples are all of alternative networking systems to the Internet. Moreover, [\[RFC1034\]](#) says, "Because we want the name space to be useful in dissimilar networks and applications, we provide the ability to use the same name space with different protocol families or management."

If one thinks of classes as simply the way to resolve the same name depending on which sort of networking technology is being used, a strong expectation of completely parallel trees is not surprising. Indeed, in an environment of many different networking and internetworking technologies, it would have been surprising if, when one changed network technologies, a name referred to a completely different target system. At the same time, parallel namespace trees are not formally required.

Since the time when the DNS was defined, Internet technology has largely won out over other network technologies. In addition, the last time a fundamentally new networking technology was introduced to the Internet (with IPv6 in [\[RFC1883\]](#)), the designers treated it as just another part of the IN class (and introduced the AAAA record, in [\[RFC1886\]](#)). So, the reason for the class field in the first place has withered away; and, when the opportunity came to use classes in their originally intended way, the designers of the technology decided not to use them.

2.2. Not all RRTYPES are careful about class

RRTYPES can either be class-independent, or else they can return very different data depending on the class in question. Not every RRTYPE specification is clear about its definition under various classes. For instance, the original 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 many RRs in every class except IN will be treated as unknown. For the purposes of resolution, that might not matter. But administrators and users will be reluctant to embrace a class that does not have good input and validation tools -- a problem that already vexes adoption of new RRTYPES in class IN.

2.3. The DNS RRTYPE registry and meaning

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. In [RFC1034], section 3.6, the A record is also defined for class CH. Perhaps unfortunately, the IANA registry for RRTYPEs (at <http://www.iana.org/assignments/dns-parameters/dns-parameters.xml#dns-parameters-4>) does not include an indicator for the class(es) in which the RRTYPE is defined.

It appears, therefore, that the "meaning" field of an RRTYPE definition is required to be class-independent, even though the RDATA for a given type may vary dramatically. For instance, in the case of the A record the RDATA is either a 32-bit IPv4 address or else a domain name and a 16-bit octal address. Across classes, even the number of fields may differ for the same type.

This appears to be yet more evidence for the "strict parallelism" explanation in Section 2.1.1. At the same time, [RFC1034] is not perfectly clear that a data type must have the same meaning in every class, and [RFC6895] does not contain clear instructions on the topic. Moreover, given the vastly different RDATA allowable for the same type across classes it is hard to be certain what is entailed by says that they all "have the same meaning", unless there is a strict requirement that a class only ever differs based on the underlying network technology.

Therefore, if classes were to be used for purposes other than alternative low-level network technologies, the RRTYPE registry ought to be altered to indicate the meaning of a type in each class for which the type is defined. Such an alteration appears to be of questionable value given the overwhelming dominance of the IN class.

2.4. 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 for any use where a class had truly global scope.

Alternatively, it is possible to imagine resolvers using a different set of root servers for different classes of query. Such a solution

merely moves the policy problem around, for it would be necessary to develop policies about root server systems for new classes whenever names in some class need to be resolvable in a globally-unambiguous way.

3. DNS classes are effectively vestigial

Given the considerations above, it is far from obvious that DNS classes are likely to be useful in the future. At the very least, in order that they could become useful, a number of clarifications to the DNS protocol and operation specifications would be necessary.

In the interests of encouraging interoperation, therefore, additions to the DNS CLASS registry (at <<http://www.iana.org/assignments/dns-parameters/dns-parameters.xhtml#dns-parameters-2>>) are suspended until such clarifications are forthcoming. New class definitions henceforth will require standards action.

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 at least clearer and possibly different in order to be useful. Given the the way the DNS has managed to thrive without really using classes, however, it would be worth asking whether the feature is useful at all.

4. New RRTYPEs are for all classes

As long as the DNS CLASS registry suspension is in effect, new RRTYPE allocations are required to be defined in a class-independent way.

5. Security considerations

This memo creates no new security issues. It might be argued that it could in principle reduce security issues by eliminating a potential source for confusion on the Internet, but classes are so little used that there is probably no improvement in practice.

6. IANA Considerations

IANA is hereby requested to update the Domain Name System (DNS) Parameters registry as follows:

- o Update the DNS CLASSes sub-registry to add a reference to this document.
- o Update the DNS CLASSes sub-registry Registration Procedures field to "Standards Action" for decimal classes 1-65279 inclusive.

- o Add this document to the Resource Record (RR) TYPEs sub-registry references.

7. Acknowledgements

The author appreciates comments and observations from Mark Andrews, Rob Austein, Ray Bellis, Stephane Bortzmeyer, Avri Doria, John Klensin, Shane Kerr, Warren Kumari, Ed Lewis, Mukund Sivaraman, Paul Vixie, and Lixia Zhang.

8. References

8.1. Normative References

- [RFC6895] Eastlake 3rd, D., "Domain Name System (DNS) IANA Considerations", [BCP 42](#), [RFC 6895](#), DOI 10.17487/RFC6895, April 2013, <<http://www.rfc-editor.org/info/rfc6895>>.

8.2. Informative References

- [RFC0882] Mockapetris, P., "Domain names: Concepts and facilities", [RFC 882](#), DOI 10.17487/RFC0882, November 1983, <<http://www.rfc-editor.org/info/rfc882>>.
- [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>>.
- [RFC1883] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", [RFC 1883](#), DOI 10.17487/RFC1883, December 1995, <<http://www.rfc-editor.org/info/rfc1883>>.
- [RFC1886] Thomson, S. and C. Huitema, "DNS Extensions to support IP version 6", [RFC 1886](#), DOI 10.17487/RFC1886, December 1995, <<http://www.rfc-editor.org/info/rfc1886>>.

- [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>>.

[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

Note to RFC Editor: this section should be removed prior to publication as an RFC.

00:

- * Initial version

01:

- * Clarify the distinction between database and domain names as such
- * Address question of closing registry
- * Minor fixes of text

02:

- * Eliminate argument from class position in message
- * Sharpen argument from primacy of matching rules
- * Note network-technology history of class.

- * Change to status: update 6895 and close class registry

03:

- * Incorporate feedback from DNSOP meeting at IETF 95
- * Step back from closing registry: "suspend" registration until the specification is clearer about how classes work
- * Call for clarification of classes

Author's Address

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

Email: asullivan@dyn.com

