The Kitchen Sink Resource Record

Donald E. Eastlake 3rd

Status of This Document

Abstract

Periodically people are seized with a desire to put proprietary, complex, and/or obscure data into the Domain Name System (DNS). This draft defines a kitchen sink Resource Record that will satisfy this desire for the storage of miscellaneous structured information.

Acknowledgements

The suggestions of the following persons have improved this document and they are gratefully acknowledged:

Rob Austein
Johnny Eriksson
Michael A. Patton

Table of Contents

1. Introduction

The Domain Name System (DNS) provides a replicated distributed secure hierarchical database which stores "resource records" (RRs) under hierarchical domain names. This data is structured into zones which are independently maintained. [RFC 1034, 1035]

Numerous types of RRs have been defined. These support such critical functions as host name to address translation (A, AAAA, etc. RRs), automatic mail routing (MX etc. RRs), and other functions. In addition, there are RRs defined related to the zone structure and administration of the DNS (SOA, NS, and RP RRs), security (SIG, KEY, and NXT RRs), etc. There is a TXT RR for the inclusion of general human readable text.

New RRs that are reasonably spar tan and designed with some care are periodically added via the IETF standards process as new needs become apparent. But there are periodically people who want to put some proprietary, complex and/or large structured data in the DNS. They frequently come up with some way of reinterpreting the TXT RR, since that is one of the least constrained RR. This is likely a bad idea since all previous ways to reinterpreting the TXT RR have sunk without a trace. (Well, if they actually got an RFC out, it's still there, but, practically speaking, nobody actually uses it.)

If a new type of data is strongly needed for common interoperable use in the DNS, the best course is to design a new RR that efficiently meets the need through the IETF standards process. This draft defines an extremely general and flexible RR which can be used for other data, such as proprietary data where global interoperability is not a consideration. It includes representations of OSI ASN.1, MIME, XML, and, recursively, DNS RRs.

2. Kitchen Sink Resource Record

The symbol for the kitchen sink resource record is SINK. Its type number is <TBA>.

The RDATA portion of the SINK RR is structured as follows:


```
                                       1  1  1  1  1  1
         0  1  2  3  4  5  6  7  8  9  0  1  2  3  4  5
        +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
        |        meaning        |        coding         |
        +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
        |       subcoding       |                       /
        +--+--+--+--+--+--+--+--+                       /
        /                             data              /
        /                                               /
        +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
```

The "meaning", "coding", and "subcoding" octets are always present. The "data" portion is variable length and could be null in some cases. The size of the "data" portion can always be determined by subtracting 2 from the SINK resource record RDLENGTH. The coding octet gives the general structure of the data. The subcoding octet provides additional information depending on the value of the coding nibble.

[The primary objection to the previous version of this draft (draft-eastlake-kitchen-sink-05.txt) which I do not feel is answered by revision is as follows: If several different uses of SINK become popular, then DNS retrievals, which are based on RR type only, will get them all possibly resulting in wasted transfer bandwidth, unnecessary TCP (as opposed to UDP) transfers, etc.

I do not think this will be a serious problem and if it becomes one, future changes can be made such as a special DNS "extended query" that allows finer specification.

The only alternative I have thought of is to allocate a block of RR types to SINK. Then determine the actual RR type to use based on a hash of the meaning, coding, and subcoding octets and of any prefixes in the "data" fields based on those octets. But this would not guarantee that two or more popular SINK RRs wouldn't collide.]

2.1 The Meaning Octet

The meaning octet indicates whether any semantic labeling appears at the beginning of the data field and the format of such semantic labeling. This contrasts with the coding and subcoding octets which merely indicate format.

The types of labels available are chosen to be globally unique and under the control of some "owner". The owner designates the meaning associated with the labels they control. Where the label is a URI, it is recommended that a retrieval from the URI fetch material that would be helpful in determining this meaning. No a priori method is

defined for determining the meaning of other labels other than an out of band to the owner.

INITIAL ASSIGNED MEANING VALUES

0 - reserved.

1 - none.

2 - OID.
3 - domain name.
4 - URI.

5-254 - available for assignment, see section 6.

255 - reserved.

A meaning octet value of 1 indicates that there is no semantic labeling at the beginning of the data area. The information, whatever it is, coded according to the coding and subcoding octets, starts at the beginning of the data field.

Meaning octet values of 2, 3, or 4, indicate, on the other hand, that a semantic label is present. A value of two indicates that a BER [BER] encoded OID appears prefixed by an OID length count as a single unsigned octet. A value of three indicates that a DNS domain name appears in wire format with name compression prohibited. And a value of four indicates that a null octet terminated URI appears.

2.2 The Coding and Subcoding Octets

The coding octet gives the major method by which the data in the data field is encoded. It should always have a meaningful value. The subcoding octet is intended to give additional coding details. Although the subcoding octet is always present, it must be interpreted in the context of the coding octet. For any coding octet value which does not specify subcoding octet value meanings, the subcoding octet MUST be ignored and SHOULD be zero.

While not explicitly mentioned below, the data field will actually start with a semantic label is indicated by the meaning octet. If such a semantic label is present, any data prefix required by the coding or subcoding octet.

CODING OCTET VALUES

0 - reserved.

1 - ASN.1. See section 2.2.1.

2 - DNS RRs. The data portion consists of DNS resource records as they would be transmitted in a DNS response section. The subcoding octet is the number of RRs in the data area as an unsigned integer. Domain names may be compressed via pointers as in DNS replies. The origin for the pointers is the beginning of the RDATA section of the SINK RR. Thus the SINK RR is safe to cache since only code that knows how to parse the data portion of a SINK RR need know of and can expand these compressions.

3 - MIME structured data [RFC 2045, 2046]. The data portion is a MIME structured message. The "MIME-Version:" header line may be omitted unless the version is other than "1.0". The top level Content-Transfer-Encoding may be encoded into the subcoding octet (see section 2.2.2). Note that, to some extent, the size limitations of DNS RRs may be overcome in the MIME case by using the "Content-Type: message/external-body" mechanism.

4 - Text tagged data. The data potion consists of text formated as specified in the TXT RR except that the first and every subsequent odd numbered text item is considered to be a tag labeling the immediately following text item. If there are an odd number of text items overall, then the last is considered to label a null text item. Syntax of the tags is as specified in RFC 2396 for the "Authority Component" without the two leading slashes ("//") or trailing slash using the DNS for authority. Thus any organization with a domain name can assign tags without fear of conflict. The subcodings octet specifies the encoding of the labeled text items as specified in section 2.2.3.

5 - HTML. The subcoding octet indicates the version of HTML.

6 - XML. The subcoding octet is not used.

7-251 - Available for assignment, see section 6.

252 - Private coding format indicated by an OID. The format of the data portion is indicated by an initial BER encoded OID which is prefixed by an OID octet count give as an unsigned octet. The subcoding octet is available for whatever use the private formating wishes to make of it.

253 - Private coding format indicated by a domain name. The format of the data portion is indicated by an initial wire

format domain name with compression prohibited. The subcoding octet is available for whatever use the private formating wishes to make of it.

254 - Private coding format indicated by a URI. The format of the data portion is indicated by an initial URI [RFC 2396] which is terminated by a zero valued octet followed by the data with that format. The subcoding octet is available for whatever use the private formating wishes to make of it. The manner in which the URL specifies the format is not defined but presumably the retriever will recognize the URI.

255 - reserved.

NOTE: the existence of a DNS RR coding and the infinite possibilities of ASN.*s, XML, and MIME permit one to SINK to even greater depths by nesting SINKs.

2.2.1 ASN.* Subcodings

If the coding octet indicates the data is ASN.1 derived, then the data is prefixed by an OID designating the version of ASN.1 used. [Can anyone provide the values for the common varieties of ASN.1?]

For ASN.* data, a specific concrete encoding must be chosen as indicated by the subcoding octet.

ASN.* SUBCODINGS

0 - reserved.
1 - BER (Basic Encoding Rules [BER]).
2 - DER (Distinguished Encoding Rules [DER]).
3 - PER (Packed Encoding Rules) Aligned.
4 - PER Unaligned.
5 - CER (Canonical Encoding Rules).
6-253 - available for assignment, see section 6.
254 - private. This subcoding will never be assigned to a standard set of encoding rules. An OID preceded by a one octet unsigned length appears at the beginning of the data area after the ASN coding OID.
255 - reserved.

2.2.2 MIME Subcodings

If the coding octet indicates the data is MIME structured, the precise encoding is given by the subcoding octets as listed below.

MIME SUBCODINGS

0 - reserved, see section 6.
1 - 7bit.
2 - 8bit.
3 - binary.
4 - quoted-printable.
5 - base64.
6 - 253 - available for assignment, see section 6.
254 - private. The data portion must start with an "x-" token denoting the private content-transfer-encoding immediately followed by one null (zero) octet followed by the remainder of the MIME object.
255 - reserved, see section 6.

2.2.3 Text Subcodings

If the coding octet indicates the data is text, the exact encoding of the text items is indicated by the subcoding octet as follows:

TEXT SUBCODINGS

0 - reserved, see section 6.
1 - ASCII.
2 - UTF-7 [RFC 1642].
3 - UTF-8 [RFC 2044].
4 - ASCII with MIME header escapes [RFC 2047].
5 - 253 - available for assignment, see section 6.
254 - private. Each text item must start with a domain name [RFC 1034] denoting the private text encoding immediately followed by one null (zero) octet followed by the remainder of the text item.
255 - reserved, see section 6.

3. Master File Representation

SINK resource records may appear as lines in zone master files. The meaning, coding, and subcoding appear as unsigned decimal integers. The data portion can be quite long. It is represented in base 64 [RFC 2045] and may be divided up into any number of white space separated substrings, down to single base 64 digits, which are concatenated to obtain the full data. These substrings can span lines using the standard parenthesis. (This type of base64 master file data is also required to support the DNS KEY and SIG security RRs [RFC 2535] in any case.)

4. Performance Considerations

Currently DNS is optimized for small data transfers, generally not exceeding 512 octets including overhead. Larger transfers are less efficient but do work correctly and efforts are underway to make them more efficient.

It is easy to create very large RRs or RR sets using SINK. DNS administrators should think carefully about this and may wish to discourage large RRs or RR sets. Consideration should also be given to putting zones from which large RRs or RR sets will be commonly retrieved on separate hosts which can be tuned for the load this will represent.

5. Security Considerations

Since the SINK resource record can be used to store arbitrary data in the DNS, this data could have security consequences, particularly if it is control, executable, macro, or interpretable information or very large and might cause buffer overflow. Due care should be taken. [RFC 2535] covers data original authentication of the data in the domain name system including SINK RRs.

6. IANA Considerations

Assignment of specific meaning to the values listed herein as "reserved" requires an IETF standards action.

All other assignments are by IETF consensus.

The many provisions for private indicita specified by separately allocated OIDs, domain names, or URIs should cover most requirements for private or proprietary values.

References

[ASN.1] Abstract Syntax Notation One, C.C.I.T.T. X.208.

[BER] Basic Encoding Rules for ASN.1, C.C.I.T.T. X.209.

[DER] Distinguished Encoding Rules for ASN.1, ISO/IEC 8825-1 | ITU-T Rec. X.690..

[RFC 1034] - P. Mockapetris, "Domain names - concepts and facilities", 11/01/1987.

[RFC 1035] - P. Mockapetris, "Domain names - implementation and specification", 11/01/1987.

[RFC 1642] - D. Goldsmith, M. Davis, "UTF-7 - A Mail-Safe Transformation Format of Unicode", 07/13/1994.

[RFC 2044] - F. Yergeau, "UTF-8, a transformation format of Unicode and ISO 10646", 10/30/1996.

[RFC 2045] - N. Freed, N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", 12/02/1996.

[RFC 2046] - N. Freed, N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", 12/02/1996.

[RFC 2047] - K. Moore, "MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text", 12/02/1996.

[RFC 2396] - T. Berners-Lee, R. Fielding, L. Masinter, "Uniform Resource Identifiers (URI): Generic Syntax", August 1998.

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

Author's Address

Donald E. Eastlake 3rd
IBM
65 Shindegan Hill Road
Carmel, 10512 USA

Telephone: +1 914-276-2668 (h)
             +1 914-784-7913 (w)
FAX: +1 914-784-3833 (w)
EMail: dee3@us.ibm.com

Expiration and File Name

This draft expires December 1999.

Its file name is draft-ietf-dnsind-kitchen-sink-00.txt.