Internet Engineering Task Force
Internet-Draft
Intended status: Experimental
Expires: April 20, 2016

S. Morris
M. Sivaraman
Internet Systems Consortium

October 18, 2015

DNS catalog zones
draft-muks-dnsop-dns-catalog-zones-00

Abstract

This document describes a method for automatic zone catalog provisioning and synchronization among DNS primary and secondary nameservers by storing and transferring the catalogs as regular DNS zones.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

DNS nameservers implement AXFR and IXFR for zone data synchronization among a zone's primary its and secondary nameservers, but the list of zones served by the primary (called a catalog in [RFC1035]) is not automatically synchronized. The administrator of a DNS nameserver farm has to manually, or via an external application layer, synchronize such zone catalogs among primaries and their secondary nameservers. This can be inconvenient, error-prone and dependent on the nameserver implementation.

A method for automatic zone catalog provisioning and synchronization is useful, so that the zone catalog can be maintained in a reference location by an administrator, similar to zone data.

This document describes one such method, in which the catalog is represented as a regular DNS zone called a "catalog zone", and transferred using DNS zone transfers. The representation of catalogs within DNS zones is specified and nameserver requirements are listed so that DNS implementations can support catalog zones.

The contents and representation of catalog zones are described in Section 2. Nameserver behavior is described in Section 3. A glossary of some terms used in this memo is provided in Appendix A.

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 [RFC2119].

2. Catalog zones

2.1. Description

A catalog zone is a specially crafted DNS zone that contains, as DNS zone data, a list of DNS zones called member zones and associated template zone configuration common to all its member zones. An implementation of catalog zones MAY allow catalog zones to include

other catalog zones, but template zone configuration present in a catalog zone only applies to its immediate member zones. A catalog zone is meant to be used to provision DNS catalogs to secondary nameservers via zone transfers, for the purpose of setting up member zones to be served from these secondary nameservers.

A catalog zone uses some RR TYPEs such as PTR differently to achieve its purpose. Though this may be controversial, the situation is not different from other similar zone-based representations such as response-policy zones [RPZ]. However, none of the RR TYPEs used by catalog zones can incur any additional section processing during DNS QUERY. Other than being transmitted via zone transfers, catalog zones do not participate in the DNS and are not intended to be served via DNS QUERY [RFC1035].

Member zones' configuration is specified as a map of zone properties, represented as a subtree of a node [RFC1034] in the domain name space inside a catalog zone. This is described in Section 2.5. Each zone property has a name and an associated value of a specific data type. Zone property value data types are described in Section 2.6. A list of permitted zone property names and their data types is given in Section 2.9.

TBD: Transitive catalogs

2.2. Resource record fields

A catalog zone contains various resource records (RRs). They have NAME, TYPE, CLASS, TTL, RDLENGTH and RDATA as fields [RFC1035].

The NAME field contains the owner name of the respective RR. As with all DNS zones, the owner name must be a child of the catalog zone name.

The TYPE field depends on the type of catalog zone property value being represented. Section 2.6 describes how various zone property value types are represented.

The CLASS field of the RR MUST be set to the value 1 (IN) [RFC1035]. This is because some RR TYPEs such as APL used by catalog zones are defined only for the IN CLASS.

The TTL field's value is not specially defined by this memo. Catalog zones do not participate in the DNS and are not intended to be served via DNS QUERY [RFC1035], but this memo does not restrict their use.

The RDLENGTH field contains the length of the RDATA field.

The content of the RDATA field depends on the type of catalog zone property value being represented. Section 2.6 describes how various zone property value types are represented.

2.3. SOA and NS records at apex

Similar to any other DNS zone, a catalog zone would be expected to have a syntactically correct SOA record and one or more NS records at its apex.

The SOA record's SERIAL, REFRESH, RETRY and EXPIRE fields [RFC1035] are used during zone transfer. A catalog zone's SOA SERIAL field MUST increase when an update is made to the catalog zone's contents. Otherwise, secondary nameservers may not notice updates to the catalog zone's contents.

The SOA record's MINIMUM field's value is not specially defined by this memo. Catalog zones do not participate in the DNS and are not intended to be served via DNS QUERY [RFC1035], but this memo does not restrict their use.

As catalog zones do not participate in the DNS, NS records at the apex are not used but they are still required so that catalog zones are syntactically correct DNS zones. No parent delegation for the catalog zone is required. Any valid DNS name can be used in the NSDNAME field of such NS records [RFC1035] and they MUST be ignored. A single NS RR with an NSDNAME field containing the absolute name "invalid." is recommended [RFC2606].

2.4. RR TYPEs to represent data

This section introduces new RR TYPEs to represent some types of data for which no convenient representation is currently available. These are used in Section 2.6. These are general-purpose RR TYPEs and their use is not limited to catalog zones.

2.4.1. BOOL RR TYPE (boolean condition)

The BOOL RR TYPE represents a single boolean condition in its RDATA.

2.4.1.1. BOOL RDATA wire format

The BOOL RDATA consists of a single octet (RDLENGTH is 1). A zero valued octet represents the false condition and a non-zero valued octet represents the true condition. Implementations SHOULD generate an octet value of 1 to represent true conditions.

2.4.1.2. BOOL RDATA presentation format

The BOOL RDATA's presentation is represented with the mnemonic "FALSE" for the false condition, and "TRUE" for the true condition. The presentation is case-insensitive.

2.4.1.3. BOOL RR example

The following example shows BOOL RRs:

```
                is-ready.example.org. 3600 IN BOOL TRUE
                allow-query.example.org. 3600 IN BOOL False
```

2.4.2. FLOAT RR TYPE (floating-point value)

The FLOAT RR TYPE represents a value in IEEE 754 double-precision floating-point representation in its RDATA.

2.4.2.1. FLOAT RDATA wire format

The FLOAT RDATA consists of a value in IEEE 754 double-precision floating-point representation [IEEE.754.1985] in network byte order, which occupies 8 octets (RDLENGTH is 8).

2.4.2.2. FLOAT RDATA presentation format

The FLOAT RDATA's presentation uses the representation of an unsuffixed "floating constant" as defined in the C programming language standard [ISO.9899.1990].

2.4.2.3. FLOAT RR example

The following example shows FLOAT RRs:

```
        x.sample-color.example.org. 3600 IN FLOAT 0.8
        y.sample-color.example.org. 3600 IN FLOAT 0.2
        z.sample-color.example.org. 3600 IN FLOAT 0.5
```

2.4.3. INT RR TYPE (64-bit signed integer value)

The INT RR TYPE represents a 64-bit signed integer in its RDATA.

2.4.3.1. INT RDATA wire format

The INT RDATA consists of a 64-bit signed integer in two's complement representation, in network byte order, which occupies 8 octets (RDLENGTH is 8).

2.4.3.2. INT RDATA presentation format

The INT RDATA's presentation uses the representation of an unsuffixed "integer constant" as defined in the C programming language standard [ISO.9899.1990] (of the type matching a 64-bit signed integer on that platform), with an optional minus prefix.

Scanners must read any of the various formats possible. Printers must output the RDATA in base-10 decimal format.

2.4.3.3. INT RR example

The following example shows a INT RR:

```
        counter-increment.example.org. 3600 IN INT -0x1
```

2.4.4. UINT RR TYPE (64-bit unsigned integer value)

The UINT RR TYPE represents a 64-bit unsigned integer in its RDATA.

2.4.4.1. UINT RDATA wire format

The UINT RDATA consists of a 64-bit unsigned integer, in network byte order, which occupies 8 octets (RDLENGTH is 8).

2.4.4.2. UINT RDATA presentation format

The UINT RDATA's presentation uses the representation of an unsuffixed "integer constant" as defined in the C programming language standard [ISO.9899.1990] (of the type matching a 64-bit signed integer on that platform).

Scanners must read any of the various formats possible. Printers must output the RDATA in base-10 decimal format.

2.4.4.3. UINT RR example

The following example shows a UINT RR:

```
        max-query-rate.example.org. 3600 IN UINT 3600
```

2.5. Zone properties map and owner names

Member zones' configuration is specified as a map of zone properties, represented as a subtree of a node [RFC1034] in the domain name space inside a catalog zone. A subtree of child nodes is used for a nested map, occuping another label level. A map element's key (property name) is represented in the label at that level. For example, if a

catalog zone is named "catalog1.example.org." and contains a property with name "prop0", the corresponding owner name of the node representing that property is "prop0.catalog1.example.org."

Zone property names are case-insensitive. Each zone property may use only one data type for its values. A list of permitted zone property names and their data types is given in Section 2.9.

Many properties are single-valued, but some properties can be collections with thousands of values. An example is the list of member zones within a catalog zone, which can be larger than any single RDATA instance can allow. Multiple RRs are used to represent such properties.

TBD: Currently a hashing method in owner names is used to split the elements of such properties with multiple RRs into individual RRsets, one per RR. This needs to be revisited as IXFR and DNS UPDATE both allow individual RRs within an RRset to be modified. The hashing method used is described in the appropriate property value data types in Section 2.6.

2.6. Zone property value data types

2.6.1. Strings

A property with a string value is specified using a single TXT RR [RFC1035] with owner name set to the name of the property as a sub-domain of the catalog zone name, and RDATA set to the property value.

For example, if a catalog zone is named "catalog1.example.org." and contains a property "prop0" with string value "Example", the corresponding RR may look as follows:

```
        prop0.catalog1.example.org. 3600 IN TXT "Example"
```

Here, "prop0" can contain multiple TXT RRs at that node of the domain name space [RFC1034]. The single string property SHOULD be checked by the implementation.

2.6.2. Booleans

A property with a boolean value is specified using a single BOOL RR (see Section 2.4.1) with owner name set to the name of the property as a sub-domain of the catalog zone name, and RDATA set to the property value.

For example, if a catalog zone is named "catalog1.example.org." and contains a property "active" with boolean value false, the corresponding RR may look as follows:

```
        active.catalog1.example.org. 3600 IN BOOL false
```

Here, "active" can contain multiple BOOL RRs at that node of the domain name space [RFC1034]. The single boolean property SHOULD be checked by the implementation.

2.6.3. Integers

A property with an integer value is specified using a single INT RR (see Section 2.4.3) for signed integers, or UINT RR (see Section 2.4.4) for unsigned integers, with owner name set to the name of the property as a sub-domain of the catalog zone name, and RDATA set to the property value.

For example, if a catalog zone is named "catalog1.example.org." and contains a property "min-ttl" with unsigned integer value 300, the corresponding RR may look as follows:

```
        min-ttl.catalog1.example.org. 3600 IN UINT 300
```

Here, "min-ttl" can contain multiple UINT RRs at that node of the domain name space [RFC1034]. The single integer property SHOULD be checked by the implementation.

2.6.4. Floating-point values

A property with a floating-point value is specified using a single FLOAT RR (see Section 2.4.2) with owner name set to the name of the property as a sub-domain of the catalog zone name, and RDATA set to the property value.

For example, if a catalog zone is named "catalog1.example.org." and contains a property "decay-rate" with value 0.33333333, the corresponding RR may look as follows:

```
        decay-rate.catalog1.example.org. 3600 IN FLOAT 0.33333333
```

Here, "decay-rate" can contain multiple FLOAT RRs at that node of the domain name space [RFC1034]. The single floating-point property SHOULD be checked by the implementation.

2.6.5. Single names

A property with a single name as value is specified using a PTR RR [RFC1035] with owner name set to the name of the property as a sub-domain of the catalog zone name, and RDATA set to the property value.

For example, if a catalog zone is named "catalog1.example.org." and contains a property "prop1" with value "val1.example.com.", the corresponding RR may look as follows:

```
        prop1.catalog1.example.org. 3600 IN PTR val1.example.com.
```

Here, "prop1" can contain multiple PTR RRs at that node of the domain name space [RFC1034]. The single name property SHOULD be checked by the implementation.

2.6.6. Unordered list of names

Let N be an absolute name formed by concatenating the RDATA hash (see Appendix A), the name of the property, and the catalog zone name in that order, such that N is a unique owner name in the catalog zone.

Then, a property containing an unordered list of names as value is specified using multiple PTR RRs [RFC1035] with owner name set to N, and each RR's RDATA set to each name in the list of the property's value respectively.

For example, if a catalog zone is named "catalog1.example.org." and contains a property "prop2" with its value being an unordered list of two names "a.example.com." and "b.example.com.", the corresponding RRs may look as follows:

```
       <hash1>.prop2.catalog1.example.org. 3600 IN PTR a.example.com.
       <hash2>.prop2.catalog1.example.org. 3600 IN PTR b.example.com.
```

Here, "prop2"'s subtree child nodes (in the domain name space [RFC1034]) can contain multiple PTR RRs at each child. For example, <hash1>.prop2 may contain multiple PTR RRs at that node. The single name property SHOULD be checked by the implementation.

2.6.7. List of network addresses

A property with a list of network addresses as value is specified using a single APL RR [RFC3123] with owner name set to the name of the property as a sub-domain of the catalog zone name, and RDATA set to the property value. In its presentation format, the "!" character (corresponding to the negation flag) is used to negate a network element. The exact meaning of a negated network element is left to

be described by the property that APL is used for. Note that the AFL RR TYPE is defined only for the IN(1) RR CLASS.

For example, if a catalog zone is named "catalog1.example.org." and contains a property "allow" with value [192.0.2.0/24, 198.51.100.0/24] as the list of networks, the corresponding RR may look as follows:

```
        allow.catalog1.example.org. 3600 IN APL (1:192.0.2.0/24
                                                 1:198.51.100.0/24)
```

Here, "allow" can contain multiple APL RRs at that node of the domain name space [RFC1034]. The single APL RR property SHOULD be checked by the implementation.

2.6.8. Single host address

A single host address is represented using the list of network addresses data type (see Section 2.6.7) with a suitable network and prefix to result in a single network address.

2.6.9. Comments

Comments may be added anywhere in a catalog zone using a scheme such as NOTE RRs [I-D.hunt-note-rr]. This memo does not depend on NOTE RRs and it is only suggested here as an informative reference.

2.7. Catalog zone version

The catalog zone version is specified by an unsigned integer property with the property name "version". All catalog zones MUST have this property present. Primary and secondary nameservers MUST NOT use catalog zones with an unexpected value value in this property, but they may be transferred as ordinary zones. For this memo, the "version" property value MUST be set to 0.

For example, if a catalog zone is named "catalog1.example.org.", the corresponding RR MUST look as follows:

```
        version.catalog1.example.org. 3600 IN UINT 0
```

Here, "version" can contain multiple UINT RRs at that node of the domain name space [RFC1034]. The single UINT RR property SHOULD be checked by the implementation.

2.8. List of member zones

The list of member zones are specified as an unordered list (see Section 2.6.6) of names under the owner name "zones" where "zones" is a sub-domain of the catalog zone.

The names of member zones are represented on the RDATA side instead of as part of owner names so that the entire name of a zone (that is technically possible [RFC1035]) can be represented correctly.

For example, if a catalog zone is named "catalog1.example.org." and lists 3 zones "example.com.", "example.net." and "example.org.", the RRs may look as follows:

```
        <hash>.zones.catalog1.example.org. 3600 IN PTR example.com.
        <hash>.zones.catalog1.example.org. 3600 IN PTR example.net.
        <hash>.zones.catalog1.example.org. 3600 IN PTR example.org.
```

2.9. Zone configuration properties

TBD: Prepare a list of zone configuration properties that are common to DNS implementations. This is so that a company may manage a catalog zone using a Windows DNS server as the primary, and a secondary nameserver hosting service may pick up the common properties and may use a different nameserver implementation such as BIND or NSD on a POSIX operating system to serve it.

TBD: We may specify that unrecognized zone property names must be ignored, or that nameserver specific properties must be specified using the "x-" prefix similar to MIME type naming.

2.9.1. zone-soa-default-serial

TBD.

2.9.2. zone-soa-default-refresh

TBD.

2.10. Zone properties specific to a member zone

Member zones in a catalog zone share template zone configuration that is common to all member zones in that catalog. This section describes the syntax that can be used to specify zone properties specific to single member zones.

Let N be an absolute name formed by concatenating the member zone name hash (see Appendix A) and the catalog zone name in that order, such that N is a unique owner name in the catalog zone.

Zone properties specific to a particular member zone are specified under the respective sub-domain N.

For example, if a catalog zone is named "catalog1.example.org." and a member zone "example.com." contains a property "prop0" with string (see Section 2.6.1) value "Example", the corresponding RR may look as follows:

```
        prop0.<m-hash>.catalog1.example.org. 3600 IN TXT "Example"
```

As another example, if a catalog zone is named "cat1.example.org." and a member zone "example.com." contains a property "prop2" with its value being an unordered list (see Section 2.6.6) of two names "a.example.com." and "b.example.com.", the corresponding RRs may look as follows:

```
     <hash>.prop2.<m-hash>.cat1.example.org. 3600 IN PTR a.example.com.
     <hash>.prop2.<m-hash>.cat1.example.org. 3600 IN PTR b.example.com.
```

2.11. Examples of catalog zones

TBD.

3. Nameserver behavior and requirements

3.1. General requirements

TBD: Explain nameserver behavior in a more detailed way here. It is under-specified.

As it is a regular DNS zone, a catalog zone can be transferred using DNS zone transfers among nameservers. Catalog zones do not participate in the DNS and are not intended to be served via DNS QUERY. It may be inconvenient to serve some contents of catalog zones via DNS queries anyway due to the nature of their representation. A separate method of querying entries inside the catalog zone may be made available by nameserver implementations (see Section 3.3).

Catalog updates should be automatic, i.e., when a nameserver that supports catalog zones completes a zone transfer for a catalog zone, it SHOULD apply changes to the catalog within the running nameserver automatically without any manual intervention.

As with regular zones, primary and secondary nameservers for a catalog zone may be operated by different administrators. The secondary nameservers may be configured to synchronize catalog zones from the primary, but the primary's administrators may not have any administrative access to the secondaries.

A catalog zone can be updated via DNS UPDATE on a reference primary nameserver, or via zone transfers. Nameservers MAY allow loading and transfer of broken zones with incorrect catalog zone syntax (as they are treated as regular zones), but nameservers MUST NOT process such broken zones as catalog zones. For the purpose of catalog processing, the broken catalogs MUST be ignored.

If there is a clash between an existing member zone's name and an incoming member zone's name (via transfer or update), the new instance of the zone MUST be ignored and an error SHOULD be logged.

When zones are introduced into a catalog zone, a primary MUST first make the new zones available for transfers before making the updated catalog zone available for transfer, or sending NOTIFY for the catalog zone to secondaries. Note that secondary nameservers may attempt to transfer the catalog zone upon refresh timeout, so care must be taken to make the member zones available before any update to the list of member zones is visible in the catalog zone.

When zones are deleted from a catalog zone, a primary MAY delete the member zone immediately after notifying secondaries. It is up to the secondary nameserver to handle this condition correctly.

TBD: Transitive primary-secondary relationships

3.2. Updating catalog zones

TBD: Explain updating catalog zones using DNS UPDATE.

3.3. Implementation notes

Catalog zones on secondary nameservers would have to be setup manually, perhaps as static configuration, similar to how ordinary DNS zones are configured. Members of such catalog zones will be automatically synchronized by the secondary after the catalog zone is configured.

An administrator would want to query data from a catalog zone. Typical queries may include dumping the list of member zones, dumping a member zone's effective configuration, querying a specific property value of a member zone, etc. Because of the syntax of catalog zones, it may not be possible to perform these queries efficiently (or in

some cases, at all) using DNS QUERY. The list of member zones may not fit in a single DNS message. The set of present properties for a zone cannot be queried using a single DNS QUERY.

Implementations are advised to provide a tool that uses either the output of AXFR or an out-of-band method to perform queries on catalog zones.

4. Security considerations

As catalog zones are transmitted using DNS zone transfers, it is absolutely essential for these transfers to be protected from unexpected modifications on the route. So, it is a requirement that catalog zone transfers MUST be authenticated using TSIG [RFC2845]. A primary nameserver MUST NOT serve a catalog zone for transfer without using TSIG and a secondary nameserver MUST abandon an update to a catalog zone that was received without using TSIG.

DNS UPDATE [RFC2136] to catalog zones similarly MUST be authenticated using TSIG.

Zone transfers of member zones MUST similarly be authenticated using TSIG [RFC2845]. The TSIG shared secrets used for member zones are identical to those used for the catalog zones. Details of the shared secrets MUST NOT be mentioned anywhere in the catalog zone data.

Catalog zones do not need to be signed using DNSSEC. Their zone transfers are authenticated by TSIG. Signed zones MUST be handled normally by nameservers, and their contents MUST NOT be DNSSEC-validated.

5. IANA considerations

This document defines new resource record types, titled BOOL, FLOAT, INT, and UINT (see Section 2.4), assigned values as follows from the Resource Record (RR) TYPEs space [to be removed upon publication: https://www.iana.org/assignments/dns-parameters/dns-parameters.xhtml#dns-parameters-11].

TYPE	Value	Meaning
BOOL	TBD	Boolean value
FLOAT	TBD	IEEE 754 double-precision number
INT	TBD	64-bit signed integer (two's complement)
UINT	TBD	64-bit unsigned integer

6. Acknowledgements

Catalog zones originated as the chosen method among various proposals that were evaluated at ISC for easy zone management. The chosen method of storing the catalog as a regular DNS zone was proposed by Stephen Morris.

We later discovered that Paul Vixie's earlier [Metazones] proposal implemented a similar approach and reviewed it. Catalog zones borrows some syntax ideas from Metazones, as both share this scheme of representing the catalog as a regular DNS zone.

Thanks to Ray Bellis, Brian Conry, Evan Hunt, Witold Krecicki, Victoria Risk for reviewing draft proposals and providing support, comments and suggestions.

Thanks to BIND users who reviewed draft proposals and offered comments and suggestions.

7. References

7.1. Normative references

[FIPS.180-1.1995]
National Institute of Standards and Technology, "Secure Hash Standard", FIPS PUB 180-1, April 1995, <http://www.itl.nist.gov/fipspubs/fip180-1.htm>.

[IEEE.754.1985]
Institute of Electrical and Electronics Engineers, "Standard for Binary Floating-Point Arithmetic", IEEE Standard 754, August 1985.

[ISO.9899.1990]
International Organization for Standardization, "Programming languages - C", ISO Standard 9899, 1990.

[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>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[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>.

[RFC2606] Eastlake 3rd, D. and A. Panitz, "Reserved Top Level DNS Names", BCP 32, RFC 2606, DOI 10.17487/RFC2606, June 1999, <http://www.rfc-editor.org/info/rfc2606>.

[RFC2845] Vixie, P., Gudmundsson, O., Eastlake 3rd, D., and B. Wellington, "Secret Key Transaction Authentication for DNS (TSIG)", RFC 2845, DOI 10.17487/RFC2845, May 2000, <http://www.rfc-editor.org/info/rfc2845>.

[RFC3123] Koch, P., "A DNS RR Type for Lists of Address Prefixes (APL RR)", RFC 3123, DOI 10.17487/RFC3123, June 2001, <http://www.rfc-editor.org/info/rfc3123>.

7.2. Informative references

[I-D.hunt-note-rr]
Hunt, E. and D. Mahoney, "A DNS Resource Record for Confidential Comments (NOTE RR)", draft-hunt-note-rr-01 (work in progress), May 2014.

[I-D.ietf-dnsop-dns-terminology]
Hoffman, P., Sullivan, A., and K. Fujiwara, "DNS Terminology", draft-ietf-dnsop-dns-terminology-05 (work in progress), September 2015.

[Metazones]
Vixie, P., "Federated Domain Name Service Using DNS Metazones", 2005, <http://ss.vix.su/~vixie/mz.pdf>.

[RPZ] Vixie, P. and V. Schryver, "DNS Response Policy Zones (DNS RPZ)", 2010, <http://ftp.isc.org/isc/dnsrpz/isc-tn-2010-1.txt>.

Appendix A. Glossary

Catalog zone: A DNS zone containing a DNS catalog, that is, a list of DNS zones and associated template zone configuration common to all member zones.

Member zone: A DNS zone whose configuration is published inside a catalog zone.

Primary nameserver: An authoritative server configured to be the source of zone transfer to one or more [secondary] nameservers. Also see [I-D.ietf-dnsop-dns-terminology].

RDATA hash: The hexadecimal format 40-digit SHA-1 [FIPS.180-1.1995] digest, of the RDATA of the corresponding RR. For RDATA containing DNS names, no name compression must be in use, i.e., the name must be in its full expanded wire data format when it is hashed.

Member zone name hash: The hexadecimal format 40-digit SHA-1 [FIPS.180-1.1995] digest, of a zone name in uncompressed wire format.

Secondary nameserver: An authoritative server which uses zone transfer to retrieve the zone. Also see [I-D.ietf-dnsop-dns-terminology].

Zone property: A configuration parameter of a zone, sometimes also called a zone option.

Appendix B. Open issues and discussion (to be removed before final publication)

1. Config options

   We want catalog zones to be adopted by multiple DNS implementations. Towards this, we have to generalize zone config options and adopt a minimal set that we can expect most implementations to support.

2. Catalog zone and member zones on different primary nameservers

   Will it be possible to setup a catalog zone on one nameserver as primary, and allow its member zones to be served by different primary namesservers?

3. Transitive relationships

   For a catalog zone, a secondary nameserver may be a primary nameserver to a different set of nameservers in a nameserver farm. In these transitive relationships, zone configuration options (such as also-notify and allow-transfer) may differ based on the location of the primary in the hierarchy. It may not be possible to specify this within a catalog zone.

4. Templates

   Are support for config and zone data templates useful at this level? They would add complexity across implementations. If added, it would be better to restrict templates at the primary nameserver and let the secondary receive regular expanded zones.

5. Overriding controls

   A way to override zone config options (as prescribed by the catalog zones) on secondary nameservers was requested. As this would be configured outside catalog zones, it may be better to leave this to implementations.

6. Use of hashing

   Should use of hashing be completely removed, and replaced with the same common owner name for all property RRs in a collection? Both IXFR and DNS UPDATE allow changing individual RRs in a RRset.

7. Choice of hash function

   Should a different faster hash function be chosen to replace SHA-1 when computing catalog member zone name hashes?

8. Overriding existing RR types

   This memo currently overrides only the PTR RR TYPE's meaning as PTR is currently used for reverse lookups. But such overridden use seems like a non-issue as PTR is defined to be a pointer to any name in [RFC1035].

9. APL limits

   APL can only support as many networks as can fit in its RDATA. Though a very large number of networks can be listed in a single RDATA field, it is still limited in size. Will this limitation become a problem for any users?

Appendix C. Change History (to be removed before final publication)

o draft-muks-dnsop-dns-catalog-zones-00
  Initial public draft.

Authors' Addresses

Stephen Morris
Internet Systems Consortium
950 Charter Street
Redwood City, CA 94063
US

Email: stephen@isc.org
URI: http://www.isc.org/

Mukund Sivaraman
Internet Systems Consortium
950 Charter Street
Redwood City, CA 94063
US

Email: muks@mukund.org
URI: http://www.isc.org/