INTERNET-DRAFT   Eric A. Hall
Document: draft-ietf-crisp-firs-dns-02.txt   July 2003
Expires: February, 2004
Category: Standards-Track

Defining and Locating DNS Domains in the Federated Internet Registry Service

Status of this Memo

Copyright Notice

Abstract

This document defines LDAP schema and searching rules for DNS domain names, in support of the Federated Internet Registry Service (FIRS) described in [FIRS-ARCH] and [FIRS-CORE].

Table of Contents

1. Introduction

This specification defines the naming syntax, object classes, attributes, matching filters, and query processing rules for storing and locating DNS domain names in the FIRS service. Refer to [FIRS-ARCH] for information on the FIRS architecture and [FIRS-CORE] for the schema definitions and rules which govern the FIRS service as a whole.

Note that these rules and definitions only apply to domain name resources, and do not apply to domainComponent entries or any other domain name elements, unless explicitly defined. Also note that this specification governs reverse-lookup DNS domains for IPv4 and IPv6 address blocks, but that these entries are entirely different from the entries which govern the actual IPv4 and IPv6 address blocks themselves.

The definitions in this specification are intended to be used with FIRS. Their usage outside of FIRS is not prohibited, but any such usage is beyond this specification's scope of authority.

2. Prerequisites and Terminology

The complete set of specifications in the FIRS collection cumulative define a structured and distributed information service using LDAPv3 for the data-formatting and transport functions. This specification should be read in the context of that set, which currently includes [FIRS-ARCH], [FIRS-CORE], [FIRS-DNSRR], [FIRS-CONTCT], [FIRS-ASN], [FIRS-IPV4] and [FIRS-IPV6].

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.

3. Naming Syntax

The naming syntax for DNS domains in FIRS MUST follow the form of "cn=<inetDnsDomainSyntax>,cn=inetResources,<partition>", where <inetDnsDomainSyntax> is the DNS domain name resource, and where <partition> is a sequence of domainComponent relative distinguished names which identifies the scope of authority for the selected directory partition.

The inetDnsDomainSyntax is relatively unstructured, in that it uses standardized procedures to produce heavily-normalized DNS domain names rather than using structured syntax rules. This is partly necessary due to conflicting syntax rules in different specifications, but is also necessary to support existing LDAP systems which do not know the syntax rules.

The normalization procedure produces UTF-8 [RFC2279] domain names as output, with the resulting sequences being suitable for direct comparisons, substring searches, and a broad range of other matching operations.

This normalization procedure is as follows:

a. Any valid domain name MUST be accepted by FIRS-aware applications. This specifically includes ASCII characters outside of the traditional "hostname" subset, and also includes non-printable eight-bit code-point values such as Space, any of which are allowed by the domain name rules specified in STD 13 [STD13] and RFC 2181 [RFC2181].

   These code-point values MUST be escaped into an ASCII-safe form before they are stored and before they are used to seed assertion values. [STD13] and [RFC2253] both use a Reverse Solidus (Backslash) character followed by a three-digit decimal number to represent the code-point value, and this specification also requires FIRS implementations to use this process for all code-point values which need to be escaped. For example, "weird name.example.com" (where "weird name" is a valid domain name label with an embedded Space) MUST be stored as "weird\032name.example.com" in the directory, and query input MUST use this sequence as the basis of any resulting assertion value.

b. Domain names which explicitly specify the root domain MUST use a single Full-Stop (".") character. Other domain names MUST NOT have a trailing Full-Stop character, and any such character MUST be stripped.

c. In order to ensure that internationalized domain names are properly normalized and validated, all domain names MUST also undergo a round-trip conversion process using the mechanisms and rules specified in RFC 3490 [RFC3490].

   1. The first step in this process is to perform the "ToASCII" conversion operation specified in [RFC3490], with the "UseSTD3ASCIIRules" flag disabled. This step will reduce the input domain name to its canonical ASCII-compatible form, thus ensuring that the input data can be properly normalized.

   2. The second step in this process is to perform the "ToUnicode" conversion operation specified in [RFC3490], with the "UseSTD3ASCIIRules" flag disabled. This step will convert the ASCII-compatible sequence into a sequence of Unicode code-point values.

   3. The Unicode code-point values returned in step 3.c.2 MUST be converted to UTF-8 before the domain name is stored or transferred.

Once all of these steps have successfully completed, the domain name can be stored in the directory or used as an assertion value. Any fatal error conditions encountered during these conversions MUST result in a local failure; FIRS-aware applications MUST NOT store or transmit non-normalized domain names for any purposes.

The inetDnsDomainSyntax syntax is as follows:

```
inetDnsDomainSyntax
( 1.3.6.1.4.1.7161.1.3.0 NAME 'inetDnsDomainSyntax' DESC 'A
  DNS domain name.' )
```

Note that the entry name of "cn=." encompasses the entire DNS domain namespace.

Note that any Reverse Solidus characters in the domain name will be further escaped when these sequences are transferred in LDAP messages. For example, "weird\032name.example.com" will be further escaped as "weird\\032name.example.com" when it is passed in an LDAP message (this secondary escape will be stripped upon receipt, leaving the escaped domain name in its original form). The use of Reverse Solidus characters is also frequently illegal as data in URLs, and these characters will probably be escaped before they are stored in a URL as data.

Also note that UTF-8 characters use character codes which are frequently illegal as data in URLs, and many of those octet values will probably be escaped before they are stored in a URL as data.

4. Object Classes and Attributes

DNS domain name entries in FIRS MUST use the inetDnsDomain object class, in addition to the mandatory object classes defined in [FIRS-CORE]. DNS domain name entries MUST be treated as containers capable of holding subordinate entries. If an entry exists as a referral source, the entry MUST also be defined with the referral object class, in addition to the above requirements.

The inetDnsDomain object class is a structural object class which is subordinate to the inetResources object class. The inetDnsDomain object class has no mandatory attributes, although it does have several optional attributes. The inetDnsDomain object class also inherits the attributes defined in the inetResources object class, including the "cn" naming attribute.

The schema definition for the inetDnsDomain object class is as follows:

```
inetDnsDomain
( 1.3.6.1.4.1.7161.1.3.1
  NAME 'inetDnsDomain'
  DESC 'DNS domain attributes.'
  SUP inetResources
  STRUCTURAL
  MAY ( inetDnsDelegationStatus $ inetDnsDelegationDate $
   inetDnsRegistrar $ inetDnsRegistry $ inetDnsContacts $
   inetDnsAuthServers ) )
```

The attributes from the inetDnsDomain object class are described below:

```
inetDnsAuthServers
( 1.3.6.1.4.1.7161.1.3.2
  NAME 'inetDnsAuthServers'
  DESC 'Authoritative DNS servers for this domain.'
  EQUALITY caseExactMatch
  SYNTAX 1.3.6.1.4.1.7161.1.3.1 )
```

The inetDnsAuthServers attribute provides a listing of the authoritative DNS servers associated with the domain name. The attribute is defined as multi-valued, with each attribute identifying the domain name of an authoritative nameserver.

```
inetDnsContacts
( 1.3.6.1.4.1.7161.1.3.3
  NAME 'inetDnsContacts'
  DESC 'Contacts for general administrative issues concerning
  this domain name.'
  EQUALITY caseIgnoreMatch
  SYNTAX 1.3.6.1.4.1.7161.1.7.1 )

inetDnsDelegationDate
( 1.3.6.1.4.1.7161.1.3.4
  NAME 'inetDnsDelegationDate'
  DESC 'Date this DNS domain name was delegated.'
  EQUALITY generalizedTimeMatch
  ORDERING generalizedTimeOrderingMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.24
  SINGLE-VALUE )

inetDnsDelegationStatus
( 1.3.6.1.4.1.7161.1.3.5
  NAME 'inetDnsDelegationStatus'
  DESC 'Delegation status of this domain name.'
  EQUALITY numericStringMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.36{2}
  SINGLE-VALUE )
```

NOTE: In an effort to facilitate internationalization and programmatic processing, the current status of a delegation is identified by a 16-bit integer. The values and status mapping is as follows:

0	Reserved delegation (permanently inactive)
1	Assigned and active (normal state)
2	Assigned but not yet active (new delegation)
3	Assigned but on hold (disputed)
4	Assignment revoked (database purge pending)
5	Variant registration (alias for canonical domain)

Additional values are reserved for future use, and are to be administered by IANA.

Note that there is no status code for "unassigned"; unassigned entries SHOULD NOT exist, and SHOULD NOT be returned as answers.

```
inetDnsRegistrar
( 1.3.6.1.4.1.7161.1.3.6
  NAME 'inetDnsRegistrar'
  DESC 'Registrar who delegated this domain name.'
  EQUALITY caseExactMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
```

NOTE: The inetDnsRegistrar attribute uses a URL to indicate the registrar who delegated the domain name. The attribute structure is identical to the labeledURI attribute, as defined in [RFC2798], including the URL and textual comments. The data can refer to any valid URL.

```
inetDnsRegistry
( 1.3.6.1.4.1.7161.1.3.7
  NAME 'inetDnsRegistry'
  DESC 'Registry where this domain name is managed.'
  EQUALITY caseExactMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
```

NOTE: The inetDnsRegistry attribute uses a URL to indicate the registry who is ultimately responsible for the domain name. The attribute structure is identical to the labeledURI attribute, as defined in [RFC2798], including the URL and textual comments. The data can refer to any valid URL.

An example of the inetDnsDomain object class in use is shown in Figure 1 below. The example includes attributes from the inetDnsDomain, inetResources, and inetAssociatedResources object classes.

```
cn=example.com,cn=inetResources,dc=netsol,dc=com
[top object class]
[inetResources object class]
[inetDnsDomain object class]
[inetAssociatedResources object class]
|
+-attribute: description
| value: "The example.com DNS domain"
|
+-attribute: inetDnsContacts
| value: "hostmaster@example.com"
|
+-attribute: inetAuthServers
| value: "ns1.example.net"
| value: "ns2.example.net"
|
+-attribute: inetAssociatedIpv4Network
  value: "192.0.2.0/24"
```

Figure 1: The entry for the example.com DNS domain name in the dc=netsol,dc=com partition.

5. Query Processing Rules

Queries for DNS domain names have several special requirements, as discussed in the following sections.

Refer to [FIRS-CORE] for general information about FIRS queries.

5.1. Query Pre-Processing

Clients MUST ensure that the query input is normalized according to the rules specified in section 3 before the input is used as the assertion value to the resulting LDAP query.

The authoritative partition for a DNS domain name is determined by mapping the normalized domain name to a sequence of domainComponent labels.

Since the domainComponent attribute is restricted to seven-bit characters, the normalized DNS domain name MUST be converted to its IDNA form using the "ToASCII" conversion operation specified in [RFC3490], with the "UseSTD3ASCIIRules" flag disabled (FIRS applications MAY reuse the output from the conversion performed in step 3.c.1 if the entire conversion process is known to have completed successfully). The resulting sequence of ASCII labels are used to form the domainComponent sequence which represents the authoritative partition for the DNS domain name.

As a simple example, "www.example.com" would be mapped to the "dc=www,dc=example,dc=com" authoritative partition, with this partition being used to seed the query process. As a slightly more complex example, the domain name of "weird name.example.com" would be mapped to "dc=weird\032name,dc=example,dc=com".

5.2. Query Bootstrapping

FIRS clients MUST use the top-down bootstrap model by default for DNS domain name queries. As such, the search base for default queries would be set to the right-most domainComponent relative distinguished name of the authoritative partition, rather than being set to the fully-qualified distinguished name of the authoritative partition.

FIRS clients MAY use the targeted or bottom-up bootstrap models for queries if necessary or desirable. However, it is not likely that entries will be found for all DNS domain name resources using these models. As such, the top-down bootstrap model will be the most useful in most cases, and MUST be used by default.

5.3. LDAP Matching

If the server advertises the inetDnsDomain object class in the firsVersion server control, FIRS clients MUST use the inetDnsDomainMatch extensible matching filter in LDAP searches for DNS domain name entries.

The inetDnsDomainMatch filter provides an identifier and search string format which collectively inform a queried server that a specific DNS domain name should be searched for, and that any inetDnsDomain object class entries which either match or are delegation parents to the assertion value should be returned.

The inetDnsDomainMatch extensibleMatch filter is defined as follows:

```
     inetDnsDomainMatch
     ( 1.3.6.1.4.1.7161.1.0.3 NAME 'inetDnsDomainMatch' SYNTAX
       inetDnsDomainSyntax )
```

The assertion value MUST be a normalized DNS domain name, using the inetDnsDomainSyntax syntax rules defined in section 3.

A FIRS server MUST compare the assertion value against the RDN of all entries in the inetResources container of the partition specified in the search base which have an object class of inetDnsDomain. Any entry with an object class of inetDnsDomain and with a relative distinguished name which is either equal to or is a delegation parent of the domain name provided in the assertion value MUST be returned to the client. Entries which are child delegations of the queried domain name MUST NOT be returned. Entries in other delegation hierarchies MUST NOT be returned. Entries which do not have an object class of inetDnsDomain MUST NOT be returned.

In order to ensure that all of the relevant entries are found (including any referrals), the search filters for these resources MUST specify the inetDnsDomain object class along with the search criteria. For example, "(&(objectclass=inetDnsDomain) (1.3.6.1.4.1.7161.1.0.3:=example.com))" with a search base of "cn=inetResources,dc=netsol,dc=com" would find all of the inetDnsDomain object class entries in the delegation path to the "example.com" domain in the "dc=netsol,dc=com" partition.

Domain names MUST be compared on label boundaries, and MUST NOT be compared through simple character matching. Given two entries of "cn=example.com" and "cn=an-example.com", only the first would match an assertion value of "example.com".

Note that the entry name of "cn=." encompasses the entire DNS domain namespace. When used in conjunction with referrals, this entry MAY be used to redirect all inetDnsDomainMatch queries to another partition for subsequent processing.

The matching filters defined in this specification MUST be supported by FIRS clients and servers. FIRS servers MAY support additional sub-string filters, soundex filters, or any other filters they wish (these may be required to support generic LDAP clients), although FIRS clients MUST NOT expect any additional filters to be available.

If the server does not advertise support for the inetDnsDomain object class in the firsVersion server control, the client MAY choose to emulate this matching process through the use of locally-constructed filters. Since the inetDnsDomainMatch filter simply locates all of the entries in the delegation path to the named domain, it is possible that a client could emulate this query by generating distinct queries for any entries associated with the parent domains.

For example, if the user asked for information about the "www.example.com" domain name resource but the server does not advertise support for the inetDnsDomain object class, the client could theoretically issue distinct queries for inetDnsDomain entries named "cn=com", "cn=example.com" and "cn=www.example.com".

As stated earlier, however, if the server advertises support for the inetDnsDomain object class in the firsVersion control, then the client MUST use the inetDnsDomainMatch filter defined above.

5.4.   Example Query

The following example assumes that the user has specified "www.example.com" as the query value:

a. Normalize the input, which is "www.example.com" in this case.

b. Determine the authoritative partition, which is "dc=www,dc=example,dc=com" in this case. By default, queries for DNS domain names use the top-down model, meaning that the right-most relative distinguished name of "dc=com" will be used.

c. Determine the search base for the query, which will be "cn=inetResources,dc=com" if the defaults are used.

d. Initiate a DNS lookup for the SRV resource records associated with "_ldap._tcp.com." For the purpose of this example, assume that this lookup succeeds, with the DNS response message indicating that "firs.iana.org" is the preferred LDAP server.

e. Submit an LDAPv3 query to the specified server, using "(&(objectclass=inetDnsDomain) (1.3.6.1.4.1.7161.1.3.8:=www.example.com))" as the matching filter, "cn=inetResources,dc=com" as the search base, and the global query defaults defined in [FIRS-CORE].

f. Assume that the queried server returns a continuation reference referral which points to "ldap://cn=inetResources,dc=netsol,dc=com". The distinguished name element of "cn=inetResources,dc=netsol,dc=com" will be used as the new search base, while "dc=netsol,dc=com" will be used as the new authoritative partition.

g. Initiate a DNS lookup for the SRV resource records associated with "_ldap._tcp.netsol.com." For the purpose of this example, assume that this lookup succeeds, with the DNS response message indicating that "firs.netsol.org" is the preferred LDAP server.

h. Submit an LDAPv3 query to the specified server, using "(&(objectclass=inetDnsDomain) (1.3.6.1.4.1.7161.1.3.8:=www.example.com))" as the matching filter, "cn=inetResources,dc=netsol,dc=com" as the search base, and the global query defaults defined in [FIRS-CORE].

i. Assume that no other referrals are received. Display the answer data which has been received and exit the query.

6. Variant Domain Names

Some domain operators have policies which require that variant forms of a domain name be assigned or reserved whenever the underlying domain name is registered. For example, a domain operator may choose to reserve look-alike forms of "foo" (including "f00" and "fo0" and so forth), thereby preventing other entities from registering the look-alike domain name.

This document reserves the inetDnsDelegationStatus attribute value of "5" specifically for use with the look-alike domains. In this model, the canonical domain name would have a typical entry, while all of the look-alike domains would have entries with the inetDnsDelegationStatus attribute value of "5", and would only exist as referrals to the canonical domain name's entry. Searches and lookups for the variant domain names would return referrals which point to the canonical domain name entry.

An entry for the canonical domain name MUST exist in the appropriate partition(s). These entries MAY include the variant domain names as values of the optional inetAssociatedDnsDomains attribute, if desired.

7. Security Considerations

Security considerations are discussed in [FIRS-ARCH].

8. IANA Considerations

This specification assumes the existence of partitions for each of the top-level domain names in the global DNS namespace, with the expectation that FIRS-capable LDAP servers will be established for each of these partitions, and with these partition containing domain delegation entries which will provide referrals to the appropriate registrar's partitions. It is expected that IANA will encourage top-level domain registry operators to oversee the creation and management of these resources.

It is further expected that IANA will oversee the creation and management of the root domain's LDAP SRV resource records, the "dc=." LDAP partition, and the necessary LDAP servers.

The inetDnsDelegationStatus attribute uses numeric code values. It is expected that IANA will manage the assignment of these values.

Additional IANA considerations are discussed in [FIRS-ARCH].

9. Normative References

[FIRS-ARCH] Hall, E. "The Federated Internet Registry Service: Architecture and Implementation Guide", draft-ietf-crisp-firs-arch-02, July 2003.

[FIRS-ASN] Hall, E. "Defining and Locating Autonomous System Numbers in the Federated Internet Registry Service", draft-ietf-crisp-firs-asn-02, July 2003.

[FIRS-CONTCT] Hall, E. "Defining and Locating Contact Persons in the Federated Internet Registry Service", draft-ietf-crisp-firs-contact-02, July 2003.

[FIRS-CORE]   Hall, E. "The Federated Internet Registry Service: Core Elements", draft-ietf-crisp-firs-core-02, July 2003.

[FIRS-DNS]   Hall, E. "Defining and Locating DNS Domains in the Federated Internet Registry Service", draft-ietf-crisp-firs-dns-02, July 2003.

[FIRS-DNSRR] Hall, E. "Defining and Locating DNS Resource Records in the Federated Internet Registry Service", draft-ietf-crisp-firs-dnsrr-02, July 2003.

[FIRS-IPV4]   Hall, E. "Defining and Locating IPv4 Address Blocks in the Federated Internet Registry Service", draft-ietf-crisp-firs-ipv4-02, July 2003.

[FIRS-IPV6]   Hall, E. "Defining and Locating IPv6 Address Blocks in the Federated Internet Registry Service", draft-ietf-crisp-firs-ipv6-02, July 2003.

[RFC2181]   Elz, R., and Bush, R. "Clarifications to the DNS Specification", RFC 2181, July 1997.

[RFC2247]   Kille, S., Wahl, M., Grimstad, A., Huber, R., and Sataluri, S. "Using Domains in LDAP/X.500 DNs", RFC 2247, January 1998.

[RFC2251]   Wahl, M., Howes, T., and Kille, S. "Lightweight Directory Access Protocol (v3)", RFC 2251, December 1997.

[RFC2252]   Wahl, M., Coulbeck, A., Howes, T., and Kille, S. "Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions", RFC 2252, December 1997.

[RFC2254]   Howes, T. "The String Representation of LDAP Search Filters", RFC 2254, December 1997.

[RFC2279]   Yergeau, F. "UTF-8, a transformation format of ISO 10646", RFC 2279, January 1998.

[RFC3490]   Faltstrom, P., Hoffman, P., and Costello, A. "Internationalizing Domain Names in Applications (IDNA)", RFC 3490, March 2003.

[STD13] Mockapetris, P. "Domain names - concepts and facilities", STD 13, RFC 1034 and "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

[US-ASCII] Cerf, V. "ASCII format for Network Interchange", RFC 20, October 1969.

10. Changes from Previous Versions

draft-ietf-crisp-firs-dns-02:

- Several clarifications and corrections have been made.
- Several attributes had their OIDs changed. NOTE THAT THIS IS AN INTERNET DRAFT, AND THAT THE OIDS ARE SUBJECT TO ADDITIONAL CHANGES AS THIS DOCUMENT IS EDITED.

draft-ietf-crisp-firs-dns-01:

- Several clarifications and corrections have been made.

draft-ietf-crisp-firs-dns-00:

- Restructured the document set.
- "Attribute references" have been eliminated from the specification. All referential attributes now provide actual data instead of URL pointers to data. Clients that wish to retrieve these values will need to start new queries using the data values instead of URLs.
- The various modified* operational attributes have been eliminated as unnecessary.
- Several attributes had their OIDs changed. NOTE THAT THIS IS AN INTERNET DRAFT, AND THAT THE OIDS ARE SUBJECT TO ADDITIONAL CHANGES AS THIS DOCUMENT IS EDITED.

draft-ietf-crisp-lw-dns-01:

- Added discussion for internationalized domain names.
- Moved attribute-specific security requirements to the Security section.

11. Author's Address

Eric A. Hall
ehall@ehsco.com

12. Acknowledgments

Funding for the RFC editor function is currently provided by the Internet Society.

Portions of this document were funded by Verisign Labs.

The first version of this specification was co-authored by Andrew Newton of Verisign Labs, and subsequent versions continue to be developed with his active participation.

13. Full Copyright Statement