INTERNET-DRAFT
Document: draft-ietf-crisp-firs-ipv6-03.txt
Expires: March, 2004
Category: Experimental

Eric A. Hall

August 2003

Defining and Locating IPv6 Address Blocks in the Federated Internet Registry Service

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC 2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

Copyright Notice

Abstract

This document defines LDAP schema and searching rules for IPv6 address blocks, in support of the Federated Internet Registry Service (FIRS) described in [FIRS-ARCH] and [FIRS-CORE].

Table of Contents

1. Introduction.....2

2. Prerequisites and Terminology.....2
3. Naming Syntax.....3
4. Object Classes and Attributes.....4
5. Query Processing Rules.....8
5.1. Query Pre-Processing.....8
5.2. LDAP Matching.....9
5.3. Example Query.....11
6. Security Considerations.....12
7. IANA Considerations.....12
8. Normative References.....12
9. Changes from Previous Versions.....13
10. Author's Address.....14
11. Acknowledgments.....15
12. Full Copyright Statement.....15

1. Introduction

This specification defines the naming syntax, object classes, attributes, matching filters, and query processing rules for storing and locating IPv6 address blocks 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 reverse-lookup DNS domains for IPv6 address blocks are managed as DNS domain entries in [FIRS-DNS]. These are entirely different network resources, and should not be confused with IPv6 address block entries.

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-DNS], [FIRS-DNSRR], [FIRS-CONTCT], [FIRS-ASN] and [FIRS-IPV4].

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 IPv6 address blocks in FIRS MUST follow the form of "cn=<inetIpv6NetworkSyntax>,cn=inetResources,<partition>", where <inetIpv6NetworkSyntaxf> is the IPv6 address block resource, and where <partition> is a sequence of domainComponent relative distinguished names which identifies the scope of authority for the selected directory partition.

The inetIpv6NetworkSyntax rules use the uncompressed, 32-nibble notation, terminated with a subnet "prefix". The network address consists of eight sub-components, each of which are separated by a colon character, and which each consist of four hexadecimal values that represent one nibble. The entire sequence is followed by a "/" character and a three-digit decimal "prefix" value.

Entries which use the inetIpv6NetworkSyntax MUST use the starting address from a range of inclusive addresses, and MUST use CIDR prefix notation. In this manner, it is possible to create an inetIpv6Network entry for a range of addresses of any size (including a single host).

Each of the 16-bit colon-separated values MUST be written in the uncompressed form. Nibbles with a value of zero MUST be represented by the hexadecimal sequence of "0000".

If an input string does not match this syntax, a FIRS-aware application MAY attempt to manipulate the input string to form a valid value. For example, if a user enters a zero-compressed IPv6 address such as "3ffe:ffff::", the application MAY convert the input value to "3ffe:ffff:0000:0000:0000:0000:0000:0000/32" in order to form a valid inetIpv6NetworkSyntax form.

An augmented BNF for this syntax is as follows:

```
inetIpv6NetworkSyntax = inetIpv6Octet ":" inetIpv6Octet ":"
  inetIpv6Octet ":" inetIpv6Octet ":" inetIpv6Octet ":"
  inetIpv6Octet ":" inetIpv6Octet ":" inetIpv6Octet "/"
  inetIpv6Prefix

inetIpv6Octet = 4*4nibblePart

nibblePart = hexadecimal digit between "0" and "F" inclusive

inetIpv6Prefix = decimal value between "1" and "128"
  inclusive, with the non-affective leading zeroes removed
```

The inetIpv6NetworkSyntax syntax is as follows:

```
inetIpv6NetworkSyntax
( 1.3.6.1.4.1.7161.1.6.0
  NAME 'inetIpv6NetworkSyntax'
  DESC 'An IPv6 address and prefix.' )
```

For example, an IPv6 network with a range of addresses between "3ffe:ffff::" and "3ffe:ffff:ffff:ffff:ffff:ffff:ffff:ffff" would be written as "cn=3ffe:ffff:0000:0000:0000:0000:0000:0000/32". Similarly, a host address of "3ffe:ffff::1:2:3:4" would be written as "cn=3ffe:ffff:0000:0000:0001:0002:0003:0004/128".

Note that the entry name of "cn=0000:0000:0000:0000:0000:0000:0000:0000/0" encompasses the entire IPv6 address space.

Note that the use of "/" is illegal as data in URLs, and MUST be escaped before it is stored in a URL as data.

4. Object Classes and Attributes

IPv6 address block entries in FIRS MUST use the inetIpv6Network object class, in addition to the mandatory object classes defined in [FIRS-CORE]. IPv6 address block entries MUST be treated as containers capable of holding subordinate entries.

If an entry exists as a referral source, the entry MUST be defined with the referral object class, in addition to the other object classes defined above. Referral sources MUST NOT contain

subordinate entries. Refer to section 3.5 of [FIRS-CORE] for more information on referral entries in FIRS.

The inetIpv6Network object class is a structural object class which is subordinate to the inetResources object class. The inetIpv6Network object class has no mandatory attributes, although it does have several optional attributes. The inetIpv6Network object class also inherits the attributes defined in the inetResources object class, including the "cn" naming attribute.

The schema definition for the inetIpv6Network object class is as follows:

```
inetIpv6Network
( 1.3.6.1.4.1.7161.1.6.1
  NAME 'inetIpv6Network'
  DESC 'IPv6 network attributes.'
  SUP inetResources
  STRUCTURAL
  MAY ( inetIpv6DelegationStatus $ inetIpv6DelegationDate $
   inetIpv6Registrar $ inetIpv6Registry $ inetIpv6Contacts $
   inetIpv6RoutingContacts $ inetIpv6ParentNetwork $
   inetIpv6SiblingNetworks $ inetIpv6ChildNetworks ) )
```

The attributes from the inetIpv6Network object class are described below:

```
inetIpv6Contacts
( 1.3.6.1.4.1.7161.1.6.2
  NAME 'inetIpv6Contacts'
  DESC 'Contacts for general administrative issues concerning
  this address block.'
  EQUALITY caseIgnoreMatch
  SYNTAX 1.3.6.1.4.1.7161.1.4.0 )

inetIpv6DelegationDate
( 1.3.6.1.4.1.7161.1.6.3
  NAME 'inetIpv6DelegationDate'
  DESC 'Date this address block was delegated.'
  EQUALITY generalizedTimeMatch
  ORDERING generalizedTimeOrderingMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.24
  SINGLE-VALUE )
```

```
inetIpv6DelegationStatus
( 1.3.6.1.4.1.7161.1.6.4
  NAME 'inetIpv6DelegationStatus'
  DESC 'Delegation status of this address block.'
  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)

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.

```
inetIpv6Registrar
( 1.3.6.1.4.1.7161.1.6.5
  NAME 'inetIpv6Registrar'
  DESC 'Registrar or sub-registry who delegated this address
  block.'
  EQUALITY caseExactMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
```

NOTE: The inetIpv6Registrar attribute uses a URL to indicate the registrar who delegated the address block. 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.

```
        inetIpv6Registry
        ( 1.3.6.1.4.1.7161.1.6.6
          NAME 'inetIpv6Registry'
          DESC 'Regional registry where this address block is
          managed.'
          EQUALITY caseExactMatch
          SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )

          NOTE: The inetIpv6Registry attribute uses a URL to indicate
          the registry who is ultimately responsible for the address
          block. 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.

        inetIpv6ParentNetworks
        ( 1.3.6.1.4.1.7161.1.6.7
          NAME 'inetIpv6ParentNetworks'
          DESC 'IPv6 parent networks directly associated with this
          address block.'
          EQUALITY caseIgnoreMatch
          SYNTAX 1.3.6.1.4.1.7161.1.6.0 )

        inetIpv6SiblingNetworks
        ( 1.3.6.1.4.1.7161.1.6.8
          NAME 'inetIpv6SiblingNetworks'
          DESC 'IPv6 sibling networks directly associated with this
          address block.'
          EQUALITY caseIgnoreMatch
          SYNTAX 1.3.6.1.4.1.7161.1.6.0 )

        inetIpv6ChildNetworks
        ( 1.3.6.1.4.1.7161.1.6.9
          NAME 'inetIpv6ChildNetworks'
          DESC 'IPv6 child networks directly associated with this
          address block.'
          EQUALITY caseIgnoreMatch
          SYNTAX 1.3.6.1.4.1.7161.1.6.0 )
```

```
inetIpv6RoutingContacts
( 1.3.6.1.4.1.7161.1.6.10
  NAME 'inetIpv6RoutingContacts'
  DESC 'Contacts for routing-related problems with this
  address block.'
  EQUALITY caseIgnoreMatch
  SYNTAX 1.3.6.1.4.1.7161.1.4.0 )
```

An example of the inetIpv6Network object class is shown in Figure 1 below. The example includes attributes from the inetIpv6Network, inetResources, and inetAssociatedResources object classes.

```
cn=3ffe:ffff:0000:0000:0000:0000:0000:0000/32,
   cn=inetResources,dc=arin,dc=net
[top object class]
[inetResources object class]
[inetIpv6Network object class]
[inetAssociatedResources object class]
|
+-attribute: description
| value: "The example.net top-level network"
|
+-attribute: inetIpv6Contacts
| value: "hostmaster@example.com"
|
+-attribute: inetAssociatedAsNumbers
| value: "65535"
|
+-attribute: inetIpv6Registrar
  value: "http://www.arin.net/ (ARIN)"
```

Figure 1: The 3ffe:ffff:0000:0000:0000:0000:0000:0000/32 address block in the dc=arin,dc=net directory partition.

5. Query Processing Rules

Queries for IPv6 address blocks 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

FIRS clients MUST use the targeted bootstrap model by default for IPv6 address block queries, using the "ip6.arpa" zone as the seed domain for the initial query.

FIRS clients MAY use the top-down or bottom-up bootstrap models for queries if necessary or desirable. However, it is not likely that entries will be found for all IPv6 address block resources using these models. As such, the targeted bootstrap model will be the most useful in most cases, and MUST be used by default.

When the bottom-up bootstrap model is used, the authoritative partition for an IPv6 address block is determined by mapping the normalized input to an associated reverse-lookup DNS domain name (using the process as defined in RFC 1886 [RFC1886], as amended by RFC 3152 [RFC3152]), and then mapping the resulting DNS domain name to a sequence of domainComponent labels. The subnet prefix sequence MUST be stripped from the input address block as part of this mapping process (note that these rules only apply to the mapping process by which an authoritative partition is constructed, and does not apply to the process by which the entry-specific relative distinguished name is constructed). Due to the 128-bit addresses and the rules defined in [RFC1886], a fully-formed IPv6 reverse-lookup domain name will have 34 labels, which result in very large distinguished names.

For example, an IPv6 address of "3ffe:ffff:0000:0000:0000:0000:0000:0000/32" would be mapped to the reverse-lookup DNS domain name of "0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.f.f.f.f.e.f.f.3.ip6.arpa." which would in turn be mapped to "dc=0,dc=0,dc=0,dc=0,dc=0,dc=0,dc=0,dc=0,dc=0,dc=0,dc=0,dc=0,dc=0,dc=0,dc=0,dc=0,dc=0,dc=0,dc=0,dc=0,dc=0,dc=0,dc=0,dc=0,dc=0,dc=0,dc=f,dc=f,dc=f,dc=f,dc=e,dc=f,dc=f,dc=3,dc=ip6,dc=arpa", which would then be used as the authoritative partition for the bottom-up bootstrap process.

5.2. LDAP Matching

If the server advertises the inetIpv6Network object class and inetIpv6NetworkMatch matching filter in the inetResourcesControl server control, FIRS clients MUST use the inetIpv6NetworkMatch matching filter in LDAP searches for IPv6 network entries.

The inetIpv6NetworkMatch filter provides an identifier and search string format which collectively inform a queried server that a specific IPv6 address should be searched for, and that any matching inetIpv6network object class entries should be returned.

The inetIpv6NetworkMatch matching filter is defined as follows:

```
inetIpv6NetworkMatch
( 1.3.6.1.4.1.7161.1.6.0.1
  NAME 'inetIpv6NetworkMatch'
  SYNTAX 1.3.6.1.4.1.7161.1.6.0 )
```

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.

A FIRS server MUST compare the assertion value against the distinguished name of all entries within and beneath the container of the partition specified in the search base. Any entry in that hierarchy with an object class of inetIpv4Network and a distinguished name that is clearly superior to the IPv6 address provided in the assertion value MUST be returned. Entries which do not have an object class of inetIpv6Network MUST NOT be returned. Entries which are not clearly superior to the queried address MUST NOT be returned.

Note that "superiority" means that the address ranges specified in the entry names clearly encompass the address range specified in the assertion value. This can be reverse-computed by repeatedly shrinking the prefix size of the address in the assertion value, and using the resulting network/prefix pair as a matching value.

An example of this matching logic for IPv4 addresses is shown in section 5.2 of [FIRS-IPV4].

Note that the entry name of "cn=0000:0000:0000:0000:0000:0000:0000:0000/0" encompasses the entire IPv6 address space. When used in conjunction with referrals, this entry MAY be used to redirect all inetIpv6NetworkMatch 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 matching filters, although FIRS clients MUST NOT expect any additional filters to be available.

If the server does not advertise support for the inetIpv6NetworkMatch matching filter in the inetResourcesControl server control, the client MAY choose to emulate this matching

filter through the use of locally-constructed equalityMatch filters. However, this process can result in incomplete answers in some cases, so if the server advertises support for the inetIpv6NetworkMatch matching filter in the inetResourcesControl control, the client MUST use it.

5.3. Example Query

The following example assumes that the user has specified "3ffe:ffff:0000:0000:0000:0000:0000:0000/32" as the query value:

a. Normalize the input, which is "3ffe:ffff:0000:0000:0000:0000:0000:0000/32" in this case.

b. Determine the canonical authoritative partition.

   1. Map the input sequence to the reverse-lookup domain name, which is "0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.f.f.f.f.e.f.f.3.ip6.arpa." in this case.

   2. Determine the domain name which is appropriate for the bootstrap model in use. In the default case of a targeted query, use the label sequence of "ip6.arpa". In the case of a bottom-up query, use the label sequence determined in step 5.3.b.1. In the case of a top-down query, set the domain name to "arpa".

   3. Map the domain name to an authoritative partition, which would be "dc=ip6,dc=arpa" if the default bootstrap model were in use.

c. Determine the search base for the query, which will be "cn=inetResources,dc=ip6,dc=arpa" if the defaults are used.

d. Initiate a DNS lookup for the SRV resource records associated with "_ldap._tcp.ip6.arpa." 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 "(1.3.6.1.4.1.7161.1.6.0.1:= 3ffe:ffff:0000:0000:0000:0000:0000:0000/32)" as the matching filter, "cn=inetResources,dc=ip6,dc=arpa" as the search base, and the global query defaults defined in [FIRS-CORE].

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

6. Security Considerations

Security considerations are discussed in [FIRS-ARCH].

7. IANA Considerations

This specification uses the "dc=ip6,dc=arpa" directory partition by default. It is expected that authoritative LDAP partitions will be mapped to that zone, and that FIRS-capable LDAP servers will be established to service this partition, with this partition containing IPv6-specific entries which will provide referrals to the appropriate RIR partitions. It is further expected that IANA will oversee the creation and management of the ip6.arpa domain's LDAP SRV resource records, the "dc=ip6,dc=arpa" LDAP partition, and the necessary LDAP servers.

The inetIpv6DelegationStatus 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].

8. Normative References

[RFC1886] Thomson, S., and Huitema, C. "DNS Extensions to support IP version 6", RFC 1886, December 1995.

[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.

[RFC3152] Bush, R. "Delegation of IP6.ARPA", RFC 3152, August 2001.

[FIRS-ARCH] Hall, E. "The Federated Internet Registry Service: Architecture and Implementation Guide", draft-ietf-crisp-firs-arch-03, August 2003.

[FIRS-ASN] Hall, E. "Defining and Locating Autonomous System Numbers in the Federated Internet Registry Service", draft-ietf-crisp-firs-asn-03, August 2003.

[FIRS-CONTCT] Hall, E. "Defining and Locating Contact Persons in the Federated Internet Registry Service", draft-ietf-crisp-firs-contact-03, August 2003.

[FIRS-CORE] Hall, E. "The Federated Internet Registry Service: Core Elements", draft-ietf-crisp-firs-core-03, August 2003.

[FIRS-DNS] Hall, E. "Defining and Locating DNS Domains in the Federated Internet Registry Service", draft-ietf-crisp-firs-dns-03, August 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-ipv4-03, August 2003.

9. Changes from Previous Versions

draft-ietf-crisp-firs-ipv6-03:

* Several clarifications and corrections have been made.

* Added the inetIpv6ParentNetworks, inetIpv6SiblingNetworks, and inetIpv6ChildNetworks attributes.

* 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-ipv6-02:

* Several clarifications and corrections have been made.

* Changed the default bootstrap model to use targeted queries, with "ip6.arpa" as the default zone and "dc=ip6,dc=arpa" as the default partition.

* 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-ipv6-01:

* Several clarifications and corrections have been made.

draft-ietf-crisp-firs-ipv6-00:

* 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 attribute-specific operational attributes have been eliminated as unnecessary.

* The inetIpv6Registrar and inetIpv6Registry attributes were added.

* 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.

* Several typographical errors have been fixed.

* Some unnecessary text has been removed.

10. Author's Address

Eric A. Hall
ehall@ehsco.com

11. 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. Edward Lewis also contributed significant feedback to this specification in the later stages of its developments.

12. Full Copyright Statement