Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: 6 December 2025

T. Harrison
APNIC
J. Singh
ARIN

4 June 2025

Registration Data Access Protocol (RDAP) Regional Internet Registry (RIR) Search
draft-ietf-regext-rdap-rir-search-19

Abstract

The Registration Data Access Protocol (RDAP) is used by Regional Internet Registries (RIRs) and Domain Name Registries (DNRs) to provide access to their resource registration information. The core specifications for RDAP define basic search functionality, but there are various search options related to IP addresses, IP prefixes, and ASNs, which are provided by RIRs via their Whois services, but for which there is no corresponding RDAP functionality. This document extends RDAP to support those search options.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 6 December 2025.

Copyright Notice

Table of Contents

1. Introduction

The Registration Data Access Protocol (RDAP) [RFC7480] is used by Regional Internet Registries (RIRs) and Domain Name Registries (DNRs) to provide access to their resource registration information. The core specifications for RDAP define basic search functionality, but this is limited to domains, nameservers, and entities. No searches were defined for IP networks or autonomous system numbers. In an effort to have RDAP reach feature parity with the existing RIR Whois [RFC3912] services in this respect, this document defines additional search options for IP networks and autonomous system numbers.

While this document is in terms of RIRs and DNRs for the sake of consistency with earlier RDAP documents such as [RFC9082] and [RFC9083], the functionality described here may be used by any RDAP server operator that hosts Internet Number Resource (INR) objects.

1.1. Conventions Used in This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

Indentation and whitespace in examples are provided only to illustrate element relationships, and are not a required feature of this protocol.

"..." in examples is used as shorthand for elements defined outside of this document, as well as to abbreviate elements that are too long.

2. Basic Searches

2.1. Path Segments

The new resource type path segments for basic search (similar to the searches defined in [RFC9082] and [RFC9083]) are:

'ips': Used to identify an IP network search using a pattern to match one of a set of IP network attributes.

'autnums': Used to identify an autonomous system number search using a pattern to match one of a set of autonomous system number attributes.

A search pattern matches a value where it equals the string representation of the value, or where it is a match for the value in accordance with the use of the asterisk ('*', US-ASCII value 0x2A) character for partial string matching as defined in Section 4.1 of [RFC9082]. For most searches, '*' may be used to match trailing characters only, and may appear in a search only once: see the previously-mentioned section for a complete definition of the relevant behaviour.

Section 4.1 of [RFC9082] describes the use of a trailing domain label suffix in a partial string search. It is not necessary that servers support this type of search pattern for the basic searches defined in this document, since those searches do not relate to domain name members.

2.2. IP Network Search

Syntax: ips?handle=<handle search pattern>

Syntax: ips?name=<name search pattern>

Searches for IP network (see Section 5.4 of [RFC9083]) information by handle are specified using the form:

ips?handle=XXXX

XXXX is a search pattern representing an IP network identifier whose syntax is specific to the registration provider. The following URL would be used to find information for IP networks with handles matching the "NET-199*" pattern:

https://example.com/rdap/ips?handle=NET-199*

Searches for IP network (see Section 5.4 of [RFC9083]) information by name are specified using the form:

ips?name=XXXX

XXXX is a search pattern representing an IP network identifier that is assigned to the network registration by the registration holder. The following URL would be used to find information for IP networks with names matching the "NET-EXAMPLE-*" pattern:

https://example.com/rdap/ips?name=NET-EXAMPLE-*

2.3. Autonomous System Number Search

Syntax: autnums?handle=<handle search pattern>

Syntax: autnums?name=<name search pattern>

Searches for autonomous system number (see Section 5.5 of [RFC9083]) information by handle are specified using the form:

autnums?handle=XXXX

XXXX is a search pattern representing an autonomous system number identifier whose syntax is specific to the registration provider. The following URL would be used to find information for autonomous system numbers with handles matching the "AS1*" pattern:

https://example.com/rdap/autnums?handle=AS1*

Searches for autonomous system number (see Section 5.5 of [RFC9083]) information by name are specified using the form:

autnums?name=XXXX

XXXX is a search pattern representing an autonomous system number identifier that is assigned to the autonomous system number registration by the registration holder. The following URL would be used to find information for autonomous system numbers with names matching the "ASN-EXAMPLE-*" pattern:

https://example.com/rdap/autnums?name=ASN-EXAMPLE-*

3. Relation Searches

This section defines searches and link relations for finding objects and sets of objects with respect to their position within a hierarchy.

3.1. Path Segments

The variables used in the path segments in this section include:

<relation>: A relation type, as defined in Section 3.2.2 of this document.

<IP address>: An IP address, as defined in Section 3.1.1 of [RFC9082].

<CIDR prefix>: The first address of a CIDR block, as defined in Section 3.1.1 of [RFC9082].

<CIDR length>: The prefix length for a CIDR block, as defined in Section 3.1.1 of [RFC9082].

<domain name>: A fully-qualified domain name, as defined in Section 3.1.3 of [RFC9082].

<autonomous system number or range>: An autonomous system number, as defined in Section 3.1.2 of [RFC9082], or two such numbers separated by a single hyphen ('-', US-ASCII value 0x2D), where the second number is greater than the first.

<resource type search path segment>: A search path segment corresponding to an Internet Number Resource (INR) object class (i.e. an IP network address or range, autonomous system number or number range, or reverse domain name).

<object value>: a value used to identify an object for the purposes of a relation search relative to that object. One of <IP address>, <CIDR prefix> and <CIDR length> pair, <domain name>, or <autonomous system number or range>, depending on the type of search that is being performed.

<status>: an object status value, as defined in Section 4.6 of [RFC9083].

The new resource type path segments for relation search (similar to the searches defined in [RFC9082] and [RFC9083]) are:

'ips/rirSearch1/<relation>/<IP address>': Used to identify an IP network search using a relation and an IP address to match a set of IP networks.

'ips/rirSearch1/<relation>/<CIDR prefix>/<CIDR length>': Used to identify an IP network search using a relation and an IP address range to match a set of IP networks.

'autnums/rirSearch1/<relation>/<autonomous system number or range>': Used to identify an autonomous system number search using a relation and a single ASN or an ASN range to match a set of ASN objects.

'domains/rirSearch1/<relation>/<domain name>': Used to identify a reverse domain search using a relation and a reverse domain name to match a set of reverse domains.

3.2. Relation Search

Syntax: <resource type search path segment>/rirSearch1/<relation>/<object value>[?status=<status>]

The relation searches defined in this document rely on the syntax described above. Each search works in the same way for each object class.

The rirSearch1 path segment is used in the relation search URLs in order to provide a single namespace for those searches, and so that other searches can be defined underneath the top-level resource type search path segments.

3.2.1. Definitions

An INR object value may have a "parent" object and one or more "child" objects. The "parent" object is the next-least-specific object that exists in the relevant registry, while the "child" objects are the next-most-specific objects that exist in the relevant registry. For example, for a registry with the following IP network objects:

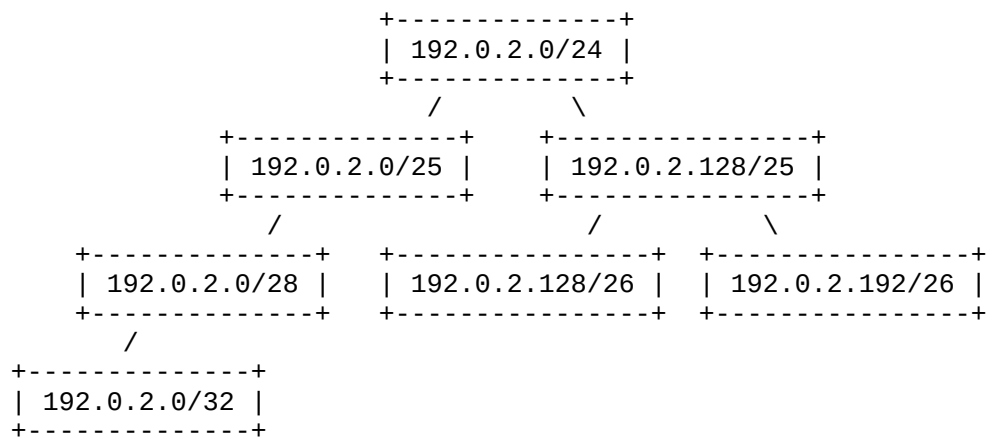

Figure 1: Example registry objects

the INR object value to parent/child object relationships are:

INR object value	Parent object
192.0.2.0/32	192.0.2.0/28
192.0.2.0/28	192.0.2.0/25
192.0.2.64/26	192.0.2.0/25
192.0.2.128/26	192.0.2.128/25
192.0.2.192/26	192.0.2.128/25
192.0.2.0/25	192.0.2.0/24
192.0.2.128/25	192.0.2.0/24
192.0.2.0/24	N/A

Table 1: Parent objects

INR object value	Child objects
192.0.2.0/24	192.0.2.0/25, 192.0.2.128/25
192.0.2.0/25	192.0.2.0/28
192.0.2.128/25	192.0.2.128/26, 192.0.2.192/26
192.0.2.64/26	N/A
192.0.2.128/26	N/A
192.0.2.192/26	N/A
192.0.2.0/28	192.0.2.0/32
192.0.2.0/32	N/A

Table 2: Child objects

(INR object values do not necessarily correspond to registry objects, because users can provide arbitrary object values as input to the searches defined in this document.)

Similarly to the parent/child object relationships, each INR object value may have a "top" object, being the least-specific covering object that exists in the registry, and one or more "bottom" objects, being the most-specific objects that entirely cover the INR object value when taken together. Given the registry defined in the previous paragraph, the top and bottom object relationships are:

INR object value	Top object
192.0.2.0/32	192.0.2.0/24
192.0.2.0/28	192.0.2.0/24
192.0.2.64/26	192.0.2.0/24
192.0.2.128/26	192.0.2.0/24
192.0.2.192/26	192.0.2.0/24
192.0.2.0/25	192.0.2.0/24
192.0.2.128/25	192.0.2.0/24
192.0.2.0/24	N/A

Table 3: Top objects

INR object value	Bottom objects
192.0.2.0/24	192.0.2.0/25, 192.0.2.0/28, 192.0.2.0/32, 192.0.2.128/26, 192.0.2.192/26
192.0.2.0/25	192.0.2.0/25, 192.0.2.0/28, 192.0.2.0/32
192.0.2.128/25	192.0.2.128/26, 192.0.2.192/26
192.0.2.64/26	N/A
192.0.2.128/26	N/A
192.0.2.192/26	N/A
192.0.2.0/28	192.0.2.0/28, 192.0.2.0/32
192.0.2.0/31	192.0.2.0/28, 192.0.2.0/32
192.0.2.0/32	N/A

Table 4: Bottom objects

If there are no more-specific objects for a given INR object value, then the set of bottom objects for that INR object value will be empty. 192.0.2.0/32 is an example of such an INR object value.

It is not necessarily the case that the bottom objects for a given INR object value will be disjoint. For example, 192.0.2.0/28's bottom objects are 192.0.2.0/28 and 192.0.2.0/32. 192.0.2.0/32 is included because it is one of the most-specific objects (i.e. an object at the bottom of the object hierarchy) for 192.0.2.0/28, while 192.0.2.0/28 itself is included because it is the most-specific object for the other addresses within the range (i.e. those aside from 192.0.2.0/32).

The bottom objects for a given INR object value may include an object that is less-specific than that INR object value. For example, 192.0.2.0/31 is an INR object value that has a more-specific object, being 192.0.2.0/32, so the set of bottom objects must include at least that object. The most-specific object that covers the residual (i.e. 192.0.2.1/32) is 192.0.2.0/28, so it is included in the results as well.

3.2.2. Relations

3.2.2.1. Single-Result Searches

3.2.2.1.1. "rdap-up"

If the server receives a search containing the relation value "rdap-up", it will return the parent object for the specified INR object value as though that object had been requested directly. If no such object exists, it will respond with a HTTP 404 (Not Found) [RFC9110] search response.

3.2.2.1.2. "rdap-top"

If the server receives a search containing the relation value "rdap-top", it will return the top object for the specified INR object value as though that object had been requested directly. If no such object exists, it will respond with an HTTP 404 (Not Found) [RFC9110] search response.

3.2.2.2. Multiple-Result Searches

3.2.2.2.1. "rdap-down"

If the server receives a search containing the relation value "rdap-down", it will return the child objects for the specified INR object value. If no such objects exist, it will return an empty search response. Per the definitions section, this includes only immediate child objects.

3.2.2.2.2. "rdap-bottom"

If the server receives a search containing the relation value "rdap-bottom", it will return the bottom objects for the specified INR object value. If no such objects exist, it will return an empty search response.

3.3. Status

If the "status" argument is provided, then response processing will proceed as though all objects without the specified status had first been removed from the database. For example, if the registry objects from section 3.2.1 had the following statuses:

Object	Status
192.0.2.0/25	active
192.0.2.128/25	inactive
192.0.2.128/26	active
192.0.2.192/26	active

Table 5: Statuses

then a server receiving a "rdap-down" search request with the INR object value 192.0.2.0/24 and a "status" argument of "active" would return the objects 192.0.2.0/25, 192.0.2.128/26, and 192.0.2.192/26.

Status filtering is useful, for example, where the client is trying to find the delegation from an RIR to an RIR account holder: by using the "rdap-top" relation with a "status" of "active", the delegation from IANA to the RIR will be ignored, and the client will receive the delegation from the RIR to the account holder in the response instead.

By default, any valid status value may be used for status filtering. Server operators MAY opt not to support "status" filtering for the "rdap-down" and "rdap-bottom" link relations, in which case the server responds with an HTTP 501 (Not Implemented) [RFC9110] response code if it receives such a request. Server operators MAY also opt not to support "status" filtering for values other than "active" for the "rdap-up" and "rdap-top" link relations, in which case the server responds with an HTTP 501 (Not Implemented) [RFC9110] response code if it receives such a request.

While any valid status value may be used for status filtering, a given RDAP server may make use of only a small number of those status values for INR objects. For example, a status value like "client hold" would typically only be used by a DNR for a forward domain name object.

3.4. Link Relations

Each of the relations defined in section 3.2.2 has a corresponding link relation that can be used for a link object contained within another RDAP object. When constructing these link objects, the server MUST use the corresponding search URL for the link target, or a URL that yields the same response as for the corresponding search as at the time of the request. The following is an elided example of an IPv4 response that makes use of those link relations:

```
{
  "startAddress": "192.0.2.0",
  "endAddress": "192.0.2.127",
  ...
  "links": [
    ...,
    {
      "value": "https://example.com/rdap/ip/192.0.2.0/25",
      "rel": "rdap-up",
      "href": ".../rdap/ips/rirSearch1/rdap-up/192.0.2.0/25",
      "type": "application/rdap+json"
    },
    {
      "value": "https://example.com/rdap/ip/192.0.2.0/25",
      "rel": "rdap-down",
      "href": ".../rdap/ips/rirSearch1/rdap-down/192.0.2.0/25",
      "type": "application/rdap+json"
    },
    {
      "value": "https://example.com/rdap/ip/192.0.2.0/25",
      "rel": "rdap-top",
      "href": ".../rdap/ips/rirSearch1/rdap-top/192.0.2.0/25",
      "type": "application/rdap+json"
    },
    {
      "value": "https://example.com/rdap/ip/192.0.2.0/25",
      "rel": "rdap-bottom",
      "href": ".../rdap/ips/rirSearch1/rdap-bottom/192.0.2.0/25",
      "type": "application/rdap+json"
    }
  ]
}
```

Figure 2: Example links in an IPv4 response

The following is an elided example of an IPv6 response that makes use of the link relations:

```
{
  "startAddress": "2001:db8:a::",
  "endAddress": "2001:db8:a:ffff:ffff:ffff:ffff:ffff",
  ...
  "links": [
    ...,
    {
      "value": "https://example.com/rdap/ip/2001:db8:a::/48",
      "rel": "rdap-up",
      "href": ".../rdap/ips/rirSearch1/rdap-up/2001:db8:a::/48",
      "type": "application/rdap+json"
    },
    {
      "value": "https://example.com/rdap/ip/2001:db8:a::/48",
      "rel": "rdap-down",
      "href": ".../rdap/ips/rirSearch1/rdap-down/2001:db8:a::/48",
      "type": "application/rdap+json"
    },
    {
      "value": "https://example.com/rdap/ip/2001:db8:a::/48",
      "rel": "rdap-top",
      "href": ".../rdap/ips/rirSearch1/rdap-top/2001:db8:a::/48",
      "type": "application/rdap+json"
    },
    {
      "value": "https://example.com/rdap/ip/2001:db8:a::/48",
      "rel": "rdap-bottom",
      "href": ".../rdap/ips/rirSearch1/rdap-bottom/2001:db8:a::/48",
      "type": "application/rdap+json"
    }
  ]
}
```

Figure 3: Example links in an IPv6 response

One additional link relation, "rdap-active", is defined for denoting a search with a "status" of "active". No other status link relations are defined, because the only known use cases for status filtering involve the "rdap-up" and "rdap-top" relations and the "active" status. The following is an elided example of an IPv4 response that makes use of those link relations:

```
{
  "startAddress": "192.0.2.0",
  "endAddress": "192.0.2.127",
  ...
  "links": [
    ...,
    {
      "value": "https://example.com/rdap/ip/192.0.2.0/25",
      "rel": "rdap-up rdap-active",
      "href":
        ".../rdap/ips/rirSearch1/rdap-up/192.0.2.0/25?status=active",
      "type": "application/rdap+json"
    },
    {
      "value": "https://example.com/rdap/ip/192.0.2.0/25",
      "rel": "rdap-top rdap-active",
      "href":
        ".../rdap/ips/rirSearch1/rdap-top/192.0.2.0/25?status=active",
      "type": "application/rdap+json"
    }
  ]
}
```

Figure 4: Example status links in an IPv4 response

The following is an elided example of an IPv6 response that makes use of the link relations:

```
{
  "startAddress": "2001:db8:a::",
  "endAddress": "2001:db8:a:ffff:ffff:ffff:ffff:ffff",
  ...
  "links": [
    ...,
    {
      "value": "https://example.com/rdap/ip/2001:db8:a::/48",
      "rel": "rdap-up rdap-active",
      "href":
      ".../rdap/ips/rirSearch1/rdap-up/2001:db8:a::/48?status=active",
      "type": "application/rdap+json"
    },
    {
      "value": "https://example.com/rdap/ip/2001:db8:a::/48",
      "rel": "rdap-top rdap-active",
      "href":
      ".../rdap/ips/rirSearch1/rdap-top/2001:db8:a::/48?status=active",
      "type": "application/rdap+json"
    }
  ]
}
```

Figure 5: Example status links in an IPv6 response

"rdap-active" is used only as a link relation in a link object. It cannot be used as a value for <relation> in the relation search URL defined in Section 3.2. Section 3.3 details status filtering for relation search URLs.

Since the "rdap-top" and "rdap-up" link relations resolve either to a single object or to an HTTP 404 (Not Found) [RFC9110] response, it is possible for a server to use a lookup URL (see Section 3.1 of [RFC9082]) in the "href" attribute in the link object. The following is an elided example of an IPv4 response that uses this approach:

```
{
  "startAddress": "192.0.2.0",
  "endAddress": "192.0.2.127",
  ...
  "links": [
    ...,
    {
      "value": "https://example.com/rdap/ip/192.0.2.0/25",
      "rel": "rdap-up",
      "href": "https://example.com/rdap/ip/192.0.2.0/24",
      "type": "application/rdap+json"
    }
  ]
}
```

Figure 6: Example single-result links in an IPv4 response

The following is an elided example of an IPv6 response that makes use of the approach:

```
{
  "startAddress": "2001:db8:a::",
  "endAddress": "2001:db8:a:ffff:ffff:ffff:ffff:ffff",
  ...
  "links": [
    ...,
    {
      "value": "https://example.com/rdap/ip/2001:db8:a::/48",
      "rel": "rdap-up",
      "href": "https://example.com/rdap/ip/2001:db8::/32",
      "type": "application/rdap+json"
    }
  ]
}
```

Figure 7: Example single-result links in an IPv6 response

Use of these link relations in responses is OPTIONAL. The absence in a response of a link for a specific relation does not necessarily mean that the corresponding search will return no results.

4. Responding To Searches

4.1. Single-Result Searches

The "rdap-up" and "rdap-top" relations are single-result searches. When processing these searches, if there is a result for the search, the server returns that object as though it were requested directly via a lookup URL (see Section 3.1 of [RFC9082]). If there is no result for the search, the server returns an HTTP 404 (Not Found) [RFC9110] response code.

4.2. Multiple-Result Searches

The "rdap-down" and "rdap-bottom" relations are multiple-result searches. As with [RFC9083], responses for these searches take the form of an array of object instances, where each instance is an appropriate object class for the search (i.e., a search beginning with /ips yields an array of IP network object instances, and a search beginning with /autnums yields an array of autonomous system number object instances). The IP network object class is defined in Section 5.4 of [RFC9083], and the autonomous system number object class is defined in Section 5.5 of [RFC9083]. The object instance arrays are contained within the response object.

The names of the arrays are as follows:

for /ips searches, the array is "ipSearchResults"; and

for /autnums searches, the array is "autnumSearchResults".

The following is an elided example of a response for an IPv4 network search:

```
{
  "rdapConformance": [ "rdap_level_0", "rirSearch1",
                       "ips", "ipSearchResults", ... ],
  ...
  "ipSearchResults": [
    {
      "objectClassName": "ip network",
      "handle": "XXXX-RIR",
      "startAddress": "192.0.2.0",
      "endAddress": "192.0.2.127",
      ...
    },
    {
      "objectClassName": "ip network",
      "handle": "YYYY-RIR",
      "startAddress": "192.0.2.0",
      "endAddress": "192.0.2.255",
      ...
    }
  ]
}
```

Figure 8: IPv4 network search response

The following is an elided example of a response for an IPv6 network search:

```
{
  "rdapConformance": [ "rdap_level_0", "rirSearch1",
                       "ips", "ipSearchResults", ... ],
  ...
  "ipSearchResults": [
    {
      "objectClassName": "ip network",
      "handle": "XXXX-RIR",
      "startAddress": "2001:db8:a::",
      "endAddress": "2001:db8:a:ffff:ffff:ffff:ffff:ffff",
      ...
    },
    {
      "objectClassName": "ip network",
      "handle": "YYYY-RIR",
      "startAddress": "2001:db8::",
      "endAddress": "2001:db8:ffff:ffff:ffff:ffff:ffff:ffff",
      ...
    }
  ]
}
```

Figure 9: IPv6 network search response

The following is an elided example of a response to an autonomous system number search:

```
{
  "rdapConformance": [ "rdap_level_0", "rirSearch1",
                       "autnums", "autnumSearchResults", ... ],
  ...
  "autnumSearchResults": [
    {
      "objectClassName": "autnum",
      "handle": "XXXX-RIR",
      "startAutnum": 64496,
      "endAutnum": 64496,
      ...
    },
    {
      "objectClassName": "autnum",
      "handle": "YYYY-RIR",
      "startAutnum": "64497",
      "endAutnum": "64497",
      ...
    }
  ]
}
```

Figure 10: ASN search response

Responses for relation searches for reverse domain objects have the same form as for a standard domain search response, per [RFC9083].

If the search can be processed by the server, but there are no results for the search, then the server returns an HTTP 404 (Not Found) [RFC9110] response code, with the body of the response containing an empty results array.

5. Reverse Search

RDAP reverse search is defined by [RFC9536]. That document limits reverse search to domains, nameservers, and entities. This document extends reverse search to cover IP networks and autonomous system numbers as well.

If a server receives a reverse search query with a searchable resource type (per the definition of that term in [RFC9536]) of "ips", then the reverse search will be performed on the IP network objects from its data store. Similarly, if a server receives a

reverse search query with a searchable resource type of "autnums", then the reverse search will be performed on the autonomous system number objects from its data store.

Additionally, Section 10 includes requests to register new entries for IP network and autonomous system number searches in the RDAP Reverse Search and RDAP Reverse Search Mapping IANA registries.

6. RDAP Conformance

A server that supports the functionality specified in this document MUST include additional string literals in the rdapConformance array of its responses, in accordance with the following:

* any response that includes an IP network basic search link, an IP network relation search link, or an IP network reverse search link includes the "rirSearch1" and "ips" literals;

* any response for an IP network basic search request, an IP network relation search request, or an IP network reverse search request includes the "rirSearch1", "ips", and "ipSearchResults" literals;

* any response that includes an ASN basic search link, an ASN relation search link, or an ASN reverse search link includes the "rirSearch1" and "autnums" literals;

* any response for an ASN basic search request, an ASN relation search request, or an ASN reverse search request includes the "rirSearch1", "autnums", and "autnumSearchResults" literals;

* any response that includes a domain relation search link includes the "rirSearch1" literal;

* any response for a domain relation search request includes the "rirSearch1" literal; and

* a response to a "/help" request includes the "rirSearch1", "ips", "ipSearchResults", "autnums", and "autnumSearchResults" literals.

Although responses will generally not include all of the rdapConformance string literals defined in this document, that is not meant to imply that a server can support only a portion of the functionality defined in this document.

The "ips", "autnums", "ipSearchResults", and "autnumSearchResults" extension identifiers are included here due to the requirements and recommendations set out in [RFC7480], [RFC9082], and [RFC9083]. These requirements and recommendations are such that an RDAP

extension identifier be used as a prefix in new path segments and JSON members introduced by the extension, and those strings are used as such as part of the basic searches defined in this document. Furthermore, using these strings as path segments helps to increase consistency among the basic searches for the core RDAP object classes.

As with the other core object classes (entity, domain, and nameserver), other documents may define additional reverse searches with IP networks and ASNs as the searchable resource type by registering those in the IANA RDAP reverse search registries. Because a given extension identifier must correspond to a single extension, though, any document making use of IP networks or ASNs as the searchable resource type must also implement the functionality described in this document.

The "1" in "rirSearch1" denotes that this is version 1 of the RIR search extension. New versions of the RIR search extension will use different extension identifiers. A version suffix is not used for the remaining identifiers defined by this document, partly because such a suffix would reduce consistency with the corresponding functionality for the other core object classes, and partly because it is very unlikely that the functionality associated with those identifiers will change.

7. Operational Considerations

When using a link object for a single-result search, a server may replace a search URL with a lookup URL, because the behaviour of the lookup URL is the same as for the search URL as at the time when the response is generated. One difference between these approaches is that when using the lookup URL, the server is effectively performing the search on behalf of the client as at the time of response generation. If there is no change to the relevant database state between the time when the original response is generated and the time when the client resolves the link relation target, then the search URL and the lookup URL will lead to the same result. However, if there is a change to the relevant database state, then the lookup URL may lead to a different result from that of the search URL. Server operators should consider which type of URL will be more effective for their clients when implementing this specification.

Where this document includes HTTP reason phrases, that is purely for the benefit of the reader, and is not an indication that those phrases need to be used as-is in responses.

8. Privacy Considerations

The search functionality defined in this document may affect the privacy of entities in the registry (and elsewhere) in various ways: see [RFC6973] for a general treatment of privacy in protocol specifications, and [RFC7481] for specific discussion about privacy threats with respect to the registration data provided by RDAP. Server operators should be aware of the tradeoffs that result from implementation of this functionality.

Many jurisdictions have laws or regulations that restrict the use of "Personal Data", per the definition in [RFC6973]. Given that, server operators should ascertain whether the regulatory environment in which they operate permits implementation of the functionality defined in this document.

9. Security Considerations

[RFC7481] describes security requirements and considerations for RDAP generally. Additionally, guidance as to the use of TLS has changed since that document was published: see [RFC8446] and [BCP195] for further detail.

[RFC9082] includes security considerations relating to object retrieval in RDAP. Those considerations are relevant here as well.

10. IANA Considerations

10.1. RDAP Extensions Registry

IANA is requested to register the following values in the RDAP Extensions Registry (https://www.iana.org/assignments/rdap-extensions/rdap-extensions.xhtml).

10.1.1. rirSearch1

Extension identifier: rirSearch1
Registry operator: Any
Published specification: [this document]
Contact: IETF <iesg@ietf.org>
Intended usage: This extension identifier is used for INR-specific search operations.

10.1.2. ips

Extension identifier: ips
Registry operator: Any
Published specification: [this document]

Contact: IETF <iesg@ietf.org>
Intended usage: This extension identifier is used for INR-specific search operations.

10.1.3. autnums

Extension identifier: autnums
Registry operator: Any
Published specification: [this document]
Contact: IETF <iesg@ietf.org>
Intended usage: This extension identifier is used for INR-specific search operations.

10.1.4. ipSearchResults

Extension identifier: ipSearchResults
Registry operator: Any
Published specification: [this document]
Contact: IETF <iesg@ietf.org>
Intended usage: This extension identifier is used for INR-specific search operations.

10.1.5. autnumSearchResults

Extension identifier: autnumSearchResults
Registry operator: Any
Published specification: [this document]
Contact: IETF <iesg@ietf.org>
Intended usage: This extension identifier is used for INR-specific search operations.

10.2. Link Relations Registry

IANA is requested to register the following values in the Link Relations Registry (https://www.iana.org/assignments/link-relations/link-relations.xhtml).

10.2.1. rdap-up

Relation Name: rdap-up
Description: Refers to an RDAP parent object in a hierarchy of objects.
Reference: [this document]

10.2.2. rdap-down

Relation Name: rdap-down
Description: Refers to a set of RDAP child objects in a hierarchy of

objects.
Reference: [this document]

10.2.3. rdap-top

Relation Name: rdap-top
Description: Refers to the topmost RDAP parent object in a hierarchy of objects.
Reference: [this document]

10.2.4. rdap-bottom

Relation Name: rdap-bottom
Description: Refers to the set of child RDAP objects that do not themselves have child objects, in a hierarchy of objects.
Reference: [this document]

10.2.5. rdap-active

Relation Name: rdap-active
Description: The target is for an RDAP RIR search that filters for the status "active".
Reference: [this document]

10.3. RDAP Reverse Search Registry

IANA is requested to register the following entries in the RDAP Reverse Search (https://www.iana.org/assignments/rdap-reverse-search/rdap-reverse-search.xhtml) registry.

10.3.1. fn

Property: fn
Description: The server supports the IP/autnum search based on the full name (a.k.a formatted name) of an associated entity.
Searchable Resource Type: ips, autnums
Related Resource Type: entity
Registrant: IETF
Contact Information: iesg@ietf.org
Reference: [this document]

10.3.2. handle

Property: handle
Description: The server supports the IP/autnum search based on the handle of an associated entity.
Searchable Resource Type: ips, autnums
Related Resource Type: entity

Registrant: IETF
Contact Information: iesg@ietf.org
Reference: [this document]

10.3.3. email

Property: email
Description: The server supports the IP/autnum search based on the email address of an associated entity.
Searchable Resource Type: ips, autnums
Related Resource Type: entity
Registrant: IETF
Contact Information: iesg@ietf.org
Reference: [this document]

10.3.4. role

Property: role
Description: The server supports the IP/autnum search based on the role of an associated entity.
Searchable Resource Type: ips, autnums
Related Resource Type: entity
Registrant: IETF
Contact Information: iesg@ietf.org
Reference: [this document]

10.4. RDAP Reverse Search Mapping Registry

IANA is requested to register the following entries in the RDAP Reverse Search Mapping (https://www.iana.org/assignments/rdap-reverse-search-mapping/rdap-reverse-search-mapping.xhtml) registry.

10.4.1. fn

Property: fn
Property Path: $.entities[*].vcardArray[1][?(@[0]==’fn’)][3]
Searchable Resource Type: ips, autnums
Related Resource Type: entity
Registrant: IETF
Contact Information: iesg@ietf.org
Reference: [this document]

10.4.2. handle

Property: handle
Property Path: $.entities[*].handle
Searchable Resource Type: ips, autnums
Related Resource Type: entity

Registrant: IETF
Contact Information: iesg@ietf.org
Reference: [this document]

10.4.3. email

Property: email
Property Path: $.entities[*].vcardArray[1][?(@[0]=='email')][3]
Searchable Resource Type: ips, autnums
Related Resource Type: entity
Registrant: IETF
Contact Information: iesg@ietf.org
Reference: [this document]

10.4.4. role

Property: role
Property Path: $.entities[*].roles
Searchable Resource Type: ips, autnums
Related Resource Type: entity
Registrant: IETF
Contact Information: iesg@ietf.org
Reference: [this document]

11. Implementation Status

> | Note to the RFC Editor - remove this section before
> | publication, as well as the reference to RFC 7942.

This section records the status of known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft, and is based on a proposal described in [RFC7942]. The description of implementations in this section is intended to assist the IETF in its decision processes in progressing drafts to RFCs. Please note that the listing of any individual implementation here does not imply endorsement by the IETF. Furthermore, no effort has been spent to verify the information presented here that was supplied by IETF contributors. This is not intended as, and must not be construed to be, a catalogue of available implementations or their features. Readers are advised to note that other implementations may exist.

According to [RFC7942], "this will allow reviewers and working groups to assign due consideration to documents that have the benefit of running code, which may serve as evidence of valuable experimentation and feedback that have made the implemented protocols more mature. It is up to the individual working groups to use this information as they see fit".

11.1. APNIC RDAP Implementation

* Responsible Organization: Asia-Pacific Network Information Centre (APNIC)
* Location: https://github.com/APNIC-net/rdap-rmp-demo/tree/rir-search
* Description: This implementation includes support for the various searches and responses described in this document.
* Level of Maturity: This is a proof-of-concept implementation.
* Coverage: This implementation includes all of the features described in this specification.
* Contact Information: Tom Harrison, tomh@apnic.net

11.2. RIPE NCC RDAP Implementation

* Responsible Organization: RIPE NCC
* Location: https://github.com/RIPE-NCC/whois
* Description: This implementation includes support for several of the searches and responses as at version 14 of this document.
* Level of Maturity: This is a production implementation.
* Coverage: This implementation includes IP and domain relation searches, as well as the links that correspond to those searches.
* Contact Information: Ed Shryane, eshryane@ripe.net

12. Acknowledgements

The authors wish to thank Mario Loffredo, Andy Newton, Antoin Verschuren, James Gould, Scott Hollenbeck, Orie Steele, Russ Housley, John Levine, Stewart Bryant, Mark Nottingham, Mohamed Boucadair, Deb Cooley, テ詠ic Vyncke, and Roman Danyliw for document review and associated comments.

13. References

13.1. Normative References

[BCP195] Best Current Practice 195, <https://www.rfc-editor.org/info/bcp195>. At the time of writing, this BCP comprises the following:

Moriarty, K. and S. Farrell, "Deprecating TLS 1.0 and TLS 1.1", BCP 195, RFC 8996, DOI 10.17487/RFC8996, March 2021, <https://www.rfc-editor.org/info/rfc8996>.

Sheffer, Y., Saint-Andre, P., and T. Fossati, "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", BCP 195, RFC 9325, DOI 10.17487/RFC9325, November 2022, <https://www.rfc-editor.org/info/rfc9325>.

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

[RFC7481] Hollenbeck, S. and N. Kong, "Security Services for the Registration Data Access Protocol (RDAP)", STD 95, RFC 7481, DOI 10.17487/RFC7481, March 2015, <https://www.rfc-editor.org/info/rfc7481>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

[RFC9082] Hollenbeck, S. and A. Newton, "Registration Data Access Protocol (RDAP) Query Format", STD 95, RFC 9082, DOI 10.17487/RFC9082, June 2021, <https://www.rfc-editor.org/info/rfc9082>.

[RFC9083] Hollenbeck, S. and A. Newton, "JSON Responses for the Registration Data Access Protocol (RDAP)", STD 95, RFC 9083, DOI 10.17487/RFC9083, June 2021, <https://www.rfc-editor.org/info/rfc9083>.

[RFC9110] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "HTTP Semantics", STD 97, RFC 9110, DOI 10.17487/RFC9110, June 2022, <https://www.rfc-editor.org/info/rfc9110>.

[RFC9536] Loffredo, M. and M. Martinelli, "Registration Data Access Protocol (RDAP) Reverse Search", RFC 9536, DOI 10.17487/RFC9536, April 2024, <https://www.rfc-editor.org/info/rfc9536>.

13.2. Informative References

[RFC3912] Daigle, L., "WHOIS Protocol Specification", RFC 3912, DOI 10.17487/RFC3912, September 2004, <https://www.rfc-editor.org/info/rfc3912>.

[RFC6973] Cooper, A., Tschofenig, H., Aboba, B., Peterson, J., Morris, J., Hansen, M., and R. Smith, "Privacy Considerations for Internet Protocols", RFC 6973, DOI 10.17487/RFC6973, July 2013, <https://www.rfc-editor.org/info/rfc6973>.

[RFC7480] Newton, A., Ellacott, B., and N. Kong, "HTTP Usage in the Registration Data Access Protocol (RDAP)", STD 95, RFC 7480, DOI 10.17487/RFC7480, March 2015, <https://www.rfc-editor.org/info/rfc7480>.

[RFC7942] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", BCP 205, RFC 7942, DOI 10.17487/RFC7942, July 2016, <https://www.rfc-editor.org/info/rfc7942>.

Authors' Addresses

Tom Harrison
Asia Pacific Network Information Centre
6 Cordelia St
South Brisbane QLD 4101
Australia
Email: tomh@apnic.net

Jasdip Singh
American Registry for Internet Numbers
PO Box 232290
Centreville, VA 20120
United States of America
Email: jasdips@arin.net