Finding the Authoritative Registration Data (RDAP) Service

Abstract

   This document specifies a method to find which Registration Data
   Access Protocol (RDAP) server is authoritative to answer queries for
   a requested scope, such as domain names, IP addresses, or Autonomous
   System numbers.

Status of This Memo

   This is an Internet Standards Track document.

   This document is a product of the Internet Engineering Task Force
   (IETF). It represents the consensus of the IETF community. It has
   received public review and has been approved for publication by the
   Internet Engineering Steering Group (IESG). Further information on
   Internet Standards is available in Section 2 of RFC 5741.

   Information about the current status of this document, any errata,
   and how to provide feedback on it may be obtained at
   http://www.rfc-editor.org/info/rfc7484.

Table of Contents

1. Introduction

   Querying and retrieving registration data from registries are defined
   in Registration Data Access Protocol (RDAP) [RFC7480] [RFC7482]
   [RFC7483]. These documents do not specify where to send the queries.
   This document specifies a method to find which server is
   authoritative to answer queries for the requested scope.

   Top-Level Domains (TLDs), Autonomous System (AS) numbers, and network
   blocks are delegated by IANA to Internet registries such as TLD
   registries and Regional Internet Registries (RIRs) that then issue
   further delegations and maintain information about them. Thus, the
   bootstrap information needed by RDAP clients is best generated from
   data and processes already maintained by IANA; the relevant
   registries already exist at [ipv4reg], [ipv6reg], [asreg], and
   [domainreg].

Per this document, IANA has created new registries based on a JSON
format specified in this document, herein named RDAP Bootstrap
Service Registries. These new registries are based on the existing
entries of the above mentioned registries. An RDAP client fetches
the RDAP Bootstrap Service Registries, extracts the data, and then
performs a match with the query data to find the authoritative
registration data server and appropriate query base URL.

2. Conventions Used in This Document

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].

3. Structure of the RDAP Bootstrap Service Registries

The RDAP Bootstrap Service Registries, as specified in Section 12
below, have been made available as JSON [RFC7159] objects, which can
be retrieved via HTTP from locations specified by IANA. The JSON
object for each registry contains a series of members containing
metadata about the registry such as a version identifier, a timestamp
of the publication date of the registry, and a description.
Additionally, a "services" member contains the registry items
themselves, as an array. Each item of the array contains a second-
level array, with two elements, each of them being a third-level
array.

Each element of the Services Array is a second-level array with two
elements: in order, an Entry Array and a Service URL Array.

The Entry Array contains all entries that have the same set of base
RDAP URLs. The Service URL Array contains the list of base RDAP URLs
usable for the entries found in the Entry Array. Elements within
these two arrays are not sorted in any way.

An example structure of the JSON output of a RDAP Bootstrap Service
Registry is illustrated:

```
{
    "version": "1.0",
    "publication": "YYYY-MM-DDTHH:MM:SSZ",
    "description": "Some text",
    "services": [
      [
        ["entry1", "entry2", "entry3"],
        [
          "https://registry.example.com/myrdap/",
          "http://registry.example.com/myrdap/"
        ]
      ],
      [
        ["entry4"],
        [
          "http://example.org/"
        ]
      ]
    ]
}
```

The formal syntax is described in Section 10.

The "version" corresponds to the format version of the registry.
This specification defines version "1.0".

The syntax of the "publication" value conforms to the Internet date/
time format [RFC3339]. The value is the latest update date of the
registry by IANA.

The optional "description" string can contain a comment regarding the
content of the bootstrap object.

Per [RFC7258], in each array of base RDAP URLs, the secure versions
of the transport protocol SHOULD be preferred and tried first. For
example, if the base RDAP URLs array contains both HTTPS and HTTP
URLs, the bootstrap client SHOULD try the HTTPS version first.

Base RDAP URLs MUST have a trailing "/" character because they are
concatenated to the various segments defined in [RFC7482].

JSON names MUST follow the format recommendations of [RFC7480]. Any
unrecognized JSON object properties or values MUST be ignored by
implementations.

Internationalized Domain Name labels used as entries or base RDAP
URLs in the registries defined in this document MUST be only
represented using their A-label form as defined in [RFC5890].

All Domain Name labels used as entries or base RDAP URLs in the
registries defined in this document MUST be only represented in
lowercase.

4. Bootstrap Service Registry for Domain Name Space

The JSON output of this registry contains domain label entries
attached to the root, grouped by base RDAP URLs, as shown in this
example.

```
{
    "version": "1.0",
    "publication": "YYYY-MM-DDTHH:MM:SSZ",
    "description": "Some text",
    "services": [
      [
        ["net", "com"],
        [
          "https://registry.example.com/myrdap/"
        ]
      ],
      [
        ["org", "mytld"],
        [
          "http://example.org/"
        ]
      ],
      [
        ["xn--zckzah"],
        [
          "https://example.net/rdapxn--zckzah/",
          "http://example.net/rdapxn--zckzah/"
        ]
      ]
    ]
}
```

The domain name's authoritative registration data service is found by
doing the label-wise longest match of the target domain name with the
domain values in the Entry Arrays in the IANA Bootstrap Service
Registry for Domain Name Space. The match is done per label, from
right to left. If the longest match results in multiple entries,
then those entries are considered equivalent. The values contained

in the Service URL Array of the matching second-level array are the valid base RDAP URLs as described in [RFC7482].

For example, a domain RDAP query for a.b.example.com matches the com entry in one of the arrays of the registry. The base RDAP URL for this query is then taken from the second element of the array, which is an array of base RDAP URLs valid for this entry. The client chooses one of the base URLs from this array; in this example, it chooses the only one available, "https://registry.example.com/myrdap/". The segment specified in [RFC7482] is then appended to the base URL to complete the query. The complete query is then "https://registry.example.com/myrdap/domain/a.b.example.com".

If a domain RDAP query for a.b.example.com matches both com and example.com entries in the registry, then the longest match applies and the example.com entry is used by the client.

If the registry contains entries such as com and goodexample.com, then a domain RDAP query for example.com only matches the com entry because matching is done on a per-label basis.

The entry for the root of the domain name space is specified as "".

5. Bootstrap Service Registries for Internet Numbers

   This section discusses IPv4 and IPv6 address space and Autonomous System numbers.

   For IP address space, the authoritative registration data service is found by doing a longest match of the target address with the values of the arrays in the corresponding RDAP Bootstrap Service Registry for Address Space. The longest match is done the same way as for routing: the addresses are converted in binary form and then the binary strings are compared to find the longest match up to the specified prefix length. The values contained in the second element of the array are the base RDAP URLs as described in [RFC7482]. The longest match method enables covering prefixes of a larger address space pointing to one base RDAP URL while more specific prefixes within the covering prefix are being served by another base RDAP URL.

5.1. Bootstrap Service Registry for IPv4 Address Space

   The JSON output of this registry contains IPv4 prefix entries,
   specified in Classless Inter-domain Routing (CIDR) format [RFC4632]
   and grouped by RDAP URLs, as shown in this example.

```
   {
       "version": "1.0",
       "publication": "2024-01-07T10:11:12Z",
       "description": "RDAP Bootstrap file for example registries.",
       "services": [
         [
           ["1.0.0.0/8", "192.0.0.0/8"],
           [
             "https://rir1.example.com/myrdap/"
           ]
         ],
         [
           ["28.2.0.0/16", "192.0.2.0/24"],
           [
             "http://example.org/"
           ]
         ],
         [
           ["28.3.0.0/16"],
           [
             "https://example.net/rdaprir2/",
             "http://example.net/rdaprir2/"
           ]
         ]
       ]
   }
```

   For example, a query for "192.0.2.1/25" matches the "192.0.0.0/8"
   entry and the "192.0.2.0/24" entry in the example registry above.
   The latter is chosen by the client given the longest match. The base
   RDAP URL for this query is then taken from the second element of the
   array, which is an array of base RDAP URLs valid for this entry. The
   client chooses one of the base URLs from this array; in this example,
   it chooses the only one available, "http://example.org/". The
   {resource} specified in [RFC7482] is then appended to the base URL to
   complete the query. The complete query is then "https://example.org/
   ip/192.0.2.1/25".

5.2. Bootstrap Service Registry for IPv6 Address Space

   The JSON output of this registry contains IPv6 prefix entries, using
   [RFC4291] text representation of the address prefixes format, grouped
   by base RDAP URLs, as shown in this example.

```
   {
       "version": "1.0",
       "publication": "2024-01-07T10:11:12Z",
       "description": "RDAP Bootstrap file for example registries.",
       "services": [
         [
           ["2001:0200::/23", "2001:db8::/32"],
           [
             "https://rir2.example.com/myrdap/"
           ]
         ],
         [
           ["2600::/16", "2100:ffff::/32"],
           [
             "http://example.org/"
           ]
         ],
         [
           ["2001:0200:1000::/36"],
           [
             "https://example.net/rdaprir2/",
             "http://example.net/rdaprir2/"
           ]
         ]
       ]
   }
```

   For example, a query for "2001:0200:1000::/48" matches the
   "2001:0200::/23" entry and the "2001:0200:1000::/36" entry in the
   example registry above. The latter is chosen by the client given the
   longest match. The base RDAP URL for this query is then taken from
   the second element of the array, which is an array of base RDAP URLs
   valid for this entry. The client chooses one of the base URLs from
   this array; in this example, it chooses "https://example.net/
   rdaprir2/" because it's the secure version of the protocol. The
   segment specified in [RFC7482] is then appended to the base URL to
   complete the query. The complete query is, therefore,
   "https://example.net/rdaprir2/ip/2001:0200:1000::/48". If the target
   RDAP server does not answer, the client can then use another URL
   prefix from the array.

5.3. Bootstrap Service Registry for AS Number Space

   The JSON output of this contains Autonomous Systems number ranges
   entries, grouped by base RDAP URLs, as shown in this example. The
   Entry Array is an array containing the list of AS number ranges
   served by the base RDAP URLs found in the second element. The array
   always contains two AS numbers represented in decimal format that
   represents the range of AS numbers between the two elements of the
   array. A single AS number is represented as a range of two identical
   AS numbers.

```
   {
       "version": "1.0",
       "publication": "2024-01-07T10:11:12Z",
       "description": "RDAP Bootstrap file for example registries.",
       "services": [
         [
           ["2045-2045"],
           [
             "https://rir3.example.com/myrdap/"
           ]
         ],
         [
           ["10000-12000", "300000-400000"],
           [
             "http://example.org/"
           ]
         ],
         [
           ["64512-65534"],
           [
             "http://example.net/rdaprir2/",
             "https://example.net/rdaprir2/"
           ]
         ]
       ]
   }
```

   For example, a query for AS 65411 matches the 64512-65534 entry in
   the example registry above. The base RDAP URL for this query is then
   taken from the second element of the array, which is an array of base
   RDAP URLs valid for this entry. The client chooses one of the base
   URLs from this array; in this example, it chooses
   "https://example.net/rdaprir2/". The segment specified in [RFC7482]
   is then appended to the base URL to complete the query. The complete
   query is, therefore, "https://example.net/rdaprir2/autnum/65411". If
   the server does not answer, the client can then use another URL
   prefix from the array.

6. Entity

   Entities (such as contacts, registrants, or registrars) can be
   queried by handle as described in [RFC7482]. Since there is no
   global namespace for entities, this document does not describe how to
   find the authoritative RDAP server for entities. However, it is
   possible that, if the entity identifier was received from a previous
   query, the same RDAP server could be queried for that entity, or the
   entity identifier itself is a fully referenced URL that can be
   queried.

7. Non-existent Entries or RDAP URL Values

   The registries may not contain the requested value. In these cases,
   there is no known RDAP server for that requested value, and the
   client SHOULD provide an appropriate error message to the user.

8. Deployment and Implementation Considerations

   This method relies on the fact that RDAP clients are fetching the
   IANA registries to then find the servers locally. Clients SHOULD NOT
   fetch the registry on every RDAP request. Clients SHOULD cache the
   registry, but use underlying protocol signaling, such as the HTTP
   Expires header field [RFC7234], to identify when it is time to
   refresh the cached registry.

   If the query data does not match any entry in the client cached
   registry, then the client may implement various methods, such as the
   following:

   o In the case of a domain object, the client may first query the DNS
      to see if the respective entry has been delegated or if it is
      mistyped information by the user. The DNS query could be to fetch
      the NS records for the TLD domain. If the DNS answer is negative,
      then there is no need to fetch the new version of the registry.
      However, if the DNS answer is positive, this may mean that the
      currently cached registry is no longer current. The client could
      then fetch the registry, parse, and then do the normal matching as
      specified above. This method may not work for all types of RDAP
      objects.

   o If the client knows the existence of an RDAP aggregator or
      redirector and its associated base URL, and trusts that service,
      then it could send the query to the redirector, which would
      redirect the client if it knows the authoritative server that
      client has not found.

Some authorities of registration data may work together on sharing
their information for a common service, including mutual redirection
[REDIRECT-RDAP].

When a new object is allocated, such as a new AS range, a new TLD, or
a new IP address range, there is no guarantee that this new object
will have an entry in the corresponding bootstrap RDAP registry,
since the setup of the RDAP server for this new entry may become live
and registered later. Therefore, the clients should expect that even
if an object, such as TLD, IP address range, or AS range is
allocated, the existence of the entry in the corresponding bootstrap
registry is not guaranteed.

9. Limitations

   This method does not provide a direct way to find authoritative RDAP
   servers for any other objects than the ones described in this
   document. In particular, the following objects are not bootstrapped
   with the method described in this document:

   o entities

   o queries using search patterns that do not contain a terminating
      string that matches some entries in the registries

   o nameservers

   o help

10. Formal Definition

   This section is the formal definition of the registries. The
   structure of JSON objects and arrays using a set of primitive
   elements is defined in [RFC7159]. Those elements are used to
   describe the JSON structure of the registries.

10.1. Imported JSON Terms

   o OBJECT: a JSON object, defined in Section 4 of [RFC7159]

   o MEMBER: a member of a JSON object, defined in Section 4 of
      [RFC7159]

   o MEMBER-NAME: the name of a MEMBER, defined as a "string" in
      Section 4 of [RFC7159]

   o MEMBER-VALUE: the value of a MEMBER, defined as a "value" in
      Section 4 of [RFC7159]

   o ARRAY: an array, defined in Section 5 of [RFC7159]

   o ARRAY-VALUE: an element of an ARRAY, defined in Section 5 of
      [RFC7159]

   o STRING: a "string", as defined in Section 7 of [RFC7159]

10.2. Registry Syntax

   Using the above terms for the JSON structures, the syntax of a
   registry is defined as follows:

   o rdap-bootstrap-registry: an OBJECT containing a MEMBER version and
      a MEMBER publication, an optional MEMBER description, and a MEMBER
      services-list

   o version: a MEMBER with MEMBER-NAME "version" and MEMBER-VALUE a
      STRING

   o publication: a MEMBER with MEMBER-NAME "publication" and MEMBER-
      VALUE a STRING

   o description: a MEMBER with MEMBER-NAME "description" and MEMBER-
      VALUE a STRING

   o services-list: a MEMBER with MEMBER-NAME "services" and MEMBER-
      VALUE a services-array

   o services-array: an ARRAY, where each ARRAY-VALUE is a service

   o service: an ARRAY of 2 elements, where the first ARRAY-VALUE is a
      an entry-list and the second ARRAY-VALUE is a service-uri-list

   o entry-list: an ARRAY, where each ARRAY-VALUE is an entry

   o entry: a STRING

   o service-uri-list: an ARRAY, where each ARRAY-VALUE is a service-
      uri

   o service-uri: a STRING

11. Security Considerations

   By providing a bootstrap method to find RDAP servers, this document
   helps to ensure that the end users will get the RDAP data from an
   authoritative source, instead of from rogue sources. The method has
   the same security properties as the RDAP protocols themselves. The
   transport used to access the registries can be more secure by using
   TLS [RFC5246], which IANA supports.

   Additional considerations on using RDAP are described in [RFC7481].

12. IANA Considerations

   IANA has created the RDAP Bootstrap Services Registries, listed
   below, and made them available as JSON objects. The contents of
   these registries are described in Section 3, Section 4, and
   Section 5, with the formal syntax specified in Section 10.

   The process for adding or updating entries in these registries
   differs from the normal IANA registry processes: these registries are
   generated from the data, processes, and policies maintained by IANA
   in their allocation registries ([ipv4reg], [ipv6reg], [asreg], and
   [domainreg]), with the addition of new RDAP server information.

   IANA will create and update RDAP Bootstrap Services Registries
   entries from the allocation registries as those registries are
   updated.

   This document does not change any policies related to the allocation
   registries; IANA has provided a mechanism for collecting the RDAP
   server information. The RDAP Bootstrap Services Registries will
   start empty and will be gradually populated as registrants of domains
   and address spaces provide RDAP server information to IANA.

   IANA has created a new top-level category on the Protocol Registries
   page, <http://www.iana.org/protocols>. The group is called
   "Registration Data Access Protocol (RDAP)". Each of the RDAP
   Bootstrap Services Registries has been made available for general
   public on-demand download in the JSON format, and that registry's URI
   is listed directly on the Protocol Registries page.

   Other normal registries will be added to this group by other
   documents, but the reason the URIs for these registries are clearly
   listed on the main page is to make those URIs obvious to implementers
   -- these are registries that will be accessed by software, as well as
   by humans using them for reference information.

Because these registries will be accessed by software, the download
demand for the RDAP Bootstrap Services Registries may be unusually
high compared to normal IANA registries. The technical
infrastructure by which registries are published will need to be
reviewed and might need to be augmented.

As discussed in Section 8, software that accesses these registries
will depend on the HTTP Expires header field to limit their query
rate. It is, therefore, important for that header field to be
properly set to provide timely information as the registries change,
while maintaining a reasonable load on the IANA servers. The HTTP
Content-Type returned to clients accessing these JSON- formatted
registries MUST be "application/json", as defined in [RFC7159].

Because of how information in the RDAP Bootstrap Services Registries
is grouped and formatted, the registry entries may not be sortable.
It is, therefore, not required or expected that the entries be sorted
in any way.

12.1. Bootstrap Service Registry for IPv4 Address Space

Entries in this registry contain at least the following:

o a CIDR [RFC4632] specification of the network block being
   registered.

o one or more URLs that provide the RDAP service regarding this
   registration.

12.2. Bootstrap Service Registry for IPv6 Address Space

Entries in this registry contain at least the following:

o an IPv6 prefix [RFC4291] specification of the network block being
   registered.

o one or more URLs that provide the RDAP service regarding this
   registration.

12.3. Bootstrap Service Registry for AS Number Space

Entries in this registry contain at least the following:

o a range of Autonomous System numbers being registered.

o one or more URLs that provide the RDAP service regarding this
   registration.

12.4. Bootstrap Service Registry for Domain Name Space

   Entries in this registry contain at least the following:

   o a domain name attached to the root being registered.

   o one or more URLs that provide the RDAP service regarding this
      registration.

13. References

13.1. Normative References

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

   [RFC3339] Klyne, G. and C. Newman, "Date and Time on the Internet:
              Timestamps", RFC 3339, July 2002,
              <http://www.rfc-editor.org/info/rfc3339>.

   [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing
              Architecture", RFC 4291, February 2006,
              <http://www.rfc-editor.org/info/rfc4291>.

   [RFC4632] Fuller, V. and T. Li, "Classless Inter-domain Routing
              (CIDR): The Internet Address Assignment and Aggregation
              Plan", BCP 122, RFC 4632, August 2006,
              <http://www.rfc-editor.org/info/rfc4632>.

   [RFC5890] Klensin, J., "Internationalized Domain Names for
              Applications (IDNA): Definitions and Document Framework",
              RFC 5890, August 2010,
              <http://www.rfc-editor.org/info/rfc5890>.

   [RFC7159] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data
              Interchange Format", RFC 7159, March 2014,
              <http://www.rfc-editor.org/info/rfc7159>.

13.2. Informative References

   [REDIRECT-RDAP]
              Martinez, C., Zhou, L., and G. Rada, "Redirection Service
              for Registration Data Access Protocol", Work in Progress,
              draft-ietf-weirds-redirects-04, July 2014.

   [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security
              (TLS) Protocol Version 1.2", RFC 5246, August 2008,
              <http://www.rfc-editor.org/info/rfc5246>.

   [RFC7071] Borenstein, N. and M. Kucherawy, "A Media Type for
              Reputation Interchange", RFC 7071, November 2013,
              <http://www.rfc-editor.org/info/rfc7071>.

   [RFC7234] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke,
              Ed., "Hypertext Transfer Protocol (HTTP/1.1): Caching",
              RFC 7234, June 2014,
              <http://www.rfc-editor.org/info/rfc7234>.

   [RFC7258] Farrell, S. and H. Tschofenig, "Pervasive Monitoring Is an
              Attack", BCP 188, RFC 7258, May 2014,
              <http://www.rfc-editor.org/info/rfc7258>.

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

   [RFC7481] Hollenbeck, S. and N. Kong, "Security Services for the
              Registration Data Access Protocol", RFC 7481, March 2015,
              <http://www.rfc-editor.org/info/rfc7481>.

   [RFC7482] Newton, A. and S. Hollenbeck, "Registration Data Access
              Protocol Query Format", RFC 7482, March 2015,
              <http://www.rfc-editor.org/info/rfc7482>.

   [RFC7483] Newton, A. and S. Hollenbeck, "JSON Responses for the
              Registration Data Access Protocol (RDAP)", RFC 7483, March
              2015, <http://www.rfc-editor.org/info/rfc7483>.

   [asreg]   IANA, "Autonomous System (AS) Numbers",
              <http://www.iana.org/assignments/as-numbers>.

   [domainreg]
              IANA, "Root Zone Database",
              <http://www.iana.org/domains/root/db>.

   [ipv4reg] IANA, "IPv4 Address Space Registry",
              <http://www.iana.org/assignments/ipv4-address-space>.

   [ipv6reg] IANA, "IPv6 Global Unicast Address Assignments",
              <http://www.iana.org/assignments/
              ipv6-unicast-address-assignments>.

Acknowledgements

Author's Address

   Marc Blanchet
   Viagenie
   246 Aberdeen
   Quebec, QC G1R 2E1
   Canada

   EMail: Marc.Blanchet@viagenie.ca
   URI: http://viagenie.ca