

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 12, 2019

M. Blanchet
Viagenie
March 11, 2019

Finding Additional Registration Data (RDAP) Service
draft-blanchet-regext-entityid2rdapserver-00

Abstract

This document specifies a method to find which Registration Data Access Protocol (RDAP) server is authoritative to answer additional information for a query already answered by another server. It is based on an entity id to RDAP server location mapping registry managed by IANA. One use case of this specification is the domain registry RDAP server providing a referral URL to the registrar RDAP server, based on the registrar entity id, for information that the registrar is authoritative for such as the contact or reseller information.

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 September 12, 2019.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	2
2.	Conventions Used in This Document	3
3.	High-Level Functional Description	3
4.	Registry of Entity to RDAP server location	3
5.	Identifying the Entity	4
6.	Structure of the Entity to RDAP Server Location Registry . .	4
7.	Formal Definition	6
7.1.	Imported JSON Terms	6
7.2.	Registry Syntax	6
8.	Recursive Referrals	7
9.	Merging the Data Received from Multiple RDAP Servers	7
10.	Security Considerations	7
11.	IANA Considerations	8
12.	Discussion of this draft	8
13.	References	9
13.1.	Normative References	9
13.2.	Informative References	10
13.3.	URIs	10
	Acknowledgements	11
	Author's Address	11

[1.](#) Introduction

Finding the authoritative Registration Data Access Protocol (RDAP) server is specified in [\[RFC7484\]](#). In some use cases, the authoritative server answering an RDAP query may not have all the information, but instead another server has the missing information. However, the first server may not know the location (URL) of that other server, but just an organization identifier, therefore it can not send a link or redirect, as described in [\[RFC7483\]](#). Operationally, the location of the other server will need to be known to many servers, where storing the mapping centrally enables the scalable management of the locations..

The typical use case is for domain registries where the RDAP server of the domain registry is not authoritative for or does not have some

information for the query, but the registrar, a separate entity from the domain registry, is authoritative and does have that additional information. The information may include contact, reseller, expiration date information. The registry RDAP server needs to provide a referral location (URL) to the client, or provide the

organization identifier for the client to map to a location (URL), to enable the client to retrieve the information from the registrar RDAP server.

This specification is generic to include other possible current or future cases, so it does not focus on the specific thin domain registry-registrar case while it uses that use case for examples.

[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.](#) High-Level Functional Description

The functional description of this proposal is as follows:

1. an RDAP client finds the authoritative RDAP server using [[RFC7484](#)],
2. the client sends its query to the authoritative RDAP server,
3. the authoritative RDAP server returns an answer as described in [[RFC7483](#)] with a reference to the identifier of an entity who has more data for this query,
4. The client finds the RDAP server of the entity by either:
 - A. Using a referral URL returned by the server based on the server using this specification,
 - B. Using this specification to find the RDAP server of the entity, based on the entity identifier,
5. the client sends the same query to the RDAP server of that

entity,

6. the server returns an answer as described in [[RFC7483](#)],
7. the client shows all the information received from both servers.

[4.](#) Registry of Entity to RDAP server location

While it is expected that the RDAP servers will be managed by organizations, this specification uses the term "entity" to support any generic case. This specification defines a registry managed by IANA which maps an entity Id to its RDAP server location (URL). The

Blanchet

Expires September 12, 2019

[Page 3]

Internet-Draft

Finding Additional RDAP Service

March 2019

RDAP server location information description is similar to [[RFC7484](#)] so that RDAP client can parse similarly for both registries.

[5.](#) Identifying the Entity

The organization identifier used in the RDAP answer is the key to the entry of the RDAP server registry specified in this document. This key should be unique in the registry. For the specific case of gTLD domain registries, ICANN through IANA has created a registry of gTLD accredited registrars [[1](#)]. In that registry, a registrar is identified by a number. For ccTLD domain registries, some registrars may not be in this registry, as they do not need to be accredited by ICANN. In a generic way not related to domain registries, there should be a registry of entities providing a unique number for these entities. IANA already have a registry of organizations identifiers, as numbers, the Private Enterprise Numbers [[2](#)] registry, with a policy of first come first serve without any limitation, with easy registration procedure [[3](#)], used in multiple contexts for IETF protocols. This document suggests to use this registry for the assignment of unique numbers to entities. Therefore, this document specifies two namespaces for the entity identification: one for accredited gTLD registrars and one from the IANA private enterprise numbers registry. In the case of domain registries where a registrar is not in the first list, that registrar can easily get a unique organization number from the IANA organizations registry in a timely manner. This specification defines a registry which maps an entity id to its RDAP server location (URL). Therefore, the entity id with its namespace creates a unique key to the registry.

6. Structure of the Entity to RDAP Server Location Registry

The Entity to RDAP Server Location registry, as specified in [Section 11](#) 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 JSON objects. Each object has a key which uniquely defines the entity and the value is an array of its RDAP server URLs.

An example structure of the JSON output of the registry is illustrated:

Blanchet

Expires September 12, 2019

[Page 4]

Internet-Draft

Finding Additional RDAP Service

March 2019

```
{
  "version": "1.0",
  "publication": "YYYY-MM-DDTHH:MM:SSZ",
  "description": "Some text",
  "services": {
    "entry1-accregids":
      [
        "https://registrar2.example.com/myrdap/",
        "http://registrar2.example.com/myrdap/"
      ],
    "entry2-pen":
      [
        "https://registrar4.example.com/rdap/"
      ],
    "entry3-accregids":
      [
        "https://myregistrar.example.com/rdap/"
      ]
  }
}
```

The formal syntax is described in [Section 7](#).

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 registry.

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 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.

The syntax of the keys is as follows:

- o entity: a unsigned integer encoded in ASCII

- o separator: the 0x2d ASCII hyphen
- o namespace: either "accregids" for an entity id from the IANA accredited registrar Ids registry or "pen" for an entity id from the IANA Private Enterprise Numbers registry

[7.](#) 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.

[7.1.](#) 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 \[RFC7159\]](#)

[7.2.](#) Registry Syntax

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

- o rdap-entity2server-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
- o TDB
- o service-uri-list: an ARRAY, where each ARRAY-VALUE is a service-

uri

o service-uri: a STRING

8. Recursive Referrals

This specification does not restrict the use of recursive links. For example, the answer from the additional RDAP server may itself contain reference to other servers, hence the possibility of recursion. However, without limits, this may end up with infinite recursion. Based on its use case, the RDAP client should set a limit on the number of referrals it will follow. In the specific case of thin domain registries with registrars RDAP servers, there should be a limit of 2 levels: the domain registry RDAP server and the registrar RDAP server.

9. Merging the Data Received from Multiple RDAP Servers

The answer from the additional RDAP server may contain data that overlaps with the answer from the initial authoritative RDAP server. This document does not specify which data should be chosen in case of overlaps or conflicts, since it depends on the use case. In the specific case of thin domain registries with registrars RDAP servers, the data received by all RDAP servers should be additive and shown appropriately to the user. For example, if the domain registry RDAP server answer contains an expiration date for the domain queried, and if the registrar RDAP server answer also contains an expiration date, then the two expiration dates are shown to the user of the RDAP client.

10. Security Considerations

By providing a method to find an entity 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](#)].

11. IANA Considerations

IANA has created the RDAP Entity to RDAP Server Location Registry, listed below, and made them available as JSON objects. The contents of these registries are described in [Section 6](#) with the formal syntax specified in [Section 7](#).

Because this registry will be accessed by software, the download demand 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.

Software accessing 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 [\[RFC7159\]](#).

The registry entries may not be sorted.

[12.](#) Discussion of this draft

this is a todo list for the author on topics to be done/resolved, or comments received. This section will disappear when draft is finished.

- o should this document merge with [RFC7484](#) and become "[RFC7484](#)-bis"
- o identify the exact field the first server refers to the entity
- o Additional namespaces may be added with updates of this specification. we don't want to setup a registry of namespace, do we?
- o The process for adding or updating entries in these registries should be defined here

alternate structure proposed by James Gould:

```
{
  "description": "RDAP bootstrap file for registry to registrar referrals",
  "publication": "2019-02-14T02:00:02Z",
  "repositories": [
    {
      "id": "ICANN",
      "description": "ICANN registrar repository for ICANN accredited registrar",
      "registrars": [
        {
          "id": "292",
          "description": "MarkMonitor",
          "url": "https://rdap.markmonitor.com/rdap/"
        }
      ]
    },
    {
      "id": "US",
      "description": "US registry repository for US registrars",
      "registrars": [
        {
          "id": "9999",
          "description": "Example non-ICANN accredited registrar for .US ccTLD",
          "url": "https://rdap.registrar.example/rdap/"
        }
      ]
    }
  ]
}
```

[13.](#) References

[13.1.](#) Normative References

- [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>>.
- [RFC3339] Klyne, G. and C. Newman, "Date and Time on the Internet: Timestamps", [RFC 3339](#), DOI 10.17487/RFC3339, July 2002, <<https://www.rfc-editor.org/info/rfc3339>>.
- [RFC7159] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", [RFC 7159](#), DOI 10.17487/RFC7159, March 2014, <<https://www.rfc-editor.org/info/rfc7159>>.

13.2. Informative References

- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", [RFC 5246](#), DOI 10.17487/RFC5246, August 2008, <<https://www.rfc-editor.org/info/rfc5246>>.
- [RFC7258] Farrell, S. and H. Tschofenig, "Pervasive Monitoring Is an Attack", [BCP 188](#), [RFC 7258](#), DOI 10.17487/RFC7258, May 2014, <<https://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](#), DOI 10.17487/RFC7480, March 2015, <<https://www.rfc-editor.org/info/rfc7480>>.
- [RFC7481] Hollenbeck, S. and N. Kong, "Security Services for the Registration Data Access Protocol (RDAP)", [RFC 7481](#), DOI 10.17487/RFC7481, March 2015, <<https://www.rfc-editor.org/info/rfc7481>>.
- [RFC7482] Newton, A. and S. Hollenbeck, "Registration Data Access Protocol (RDAP) Query Format", [RFC 7482](#), DOI 10.17487/RFC7482, March 2015, <<https://www.rfc-editor.org/info/rfc7482>>.
- [RFC7483] Newton, A. and S. Hollenbeck, "JSON Responses for the Registration Data Access Protocol (RDAP)", [RFC 7483](#), DOI 10.17487/RFC7483, March 2015, <<https://www.rfc-editor.org/info/rfc7483>>.
- [RFC7484] Blanchet, M., "Finding the Authoritative Registration Data (RDAP) Service", [RFC 7484](#), DOI 10.17487/RFC7484, March 2015, <<https://www.rfc-editor.org/info/rfc7484>>.

13.3. URIs

- [1] <https://www.iana.org/assignments/registrar-ids/registrar-ids.xhtml>

- [2] <https://www.iana.org/assignments/enterprise-numbers/enterprise-numbers>
- [3] <https://pen.iana.org/pen/PenApplication.page>

Blanchet Expires September 12, 2019 [Page 10]

Internet-Draft Finding Additional RDAP Service March 2019

Acknowledgements

The following people have provided comments and reviews improving the document significantly (in no particular order): Audric Schiltnknecht, Julien Bernard, James Gould.

Author's Address

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

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

