Registration Protocols Extensions (regext)
Internet-Draft
Intended status: Standards Track
Expires: 21 August 2026

G. Brown
A. Newton
ICANN

17 February 2026

Explicit RDAP Redirects
draft-ietf-regext-rdap-referrals-03

Abstract

This document describes an RDAP extension that allows RDAP clients to request to be redirected to a related RDAP record for a resource.

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 21 August 2026.

Table of Contents

1. Introduction

Many Registration Data Access Protocol (RDAP, described in [RFC7480], [RFC7481], [RFC9082], [RFC9083] and others) resources contain links to related RDAP resources.

For example, in the domain space, an RDAP record for a domain name received from the registry operator may include a link for the RDAP record for the same object provided by the sponsoring registrar (for example, see [gtld-rdap-profile]), while in the IP address space, an RDAP record for an address allocation may include links to enclosing or sibling prefixes.

In both cases, RDAP service users are often equally if not more interested in these related RDAP resources than the resource provided by the TLD registry or RIR.

While RDAP supports redirection of RDAP requests using HTTP redirections (which use a 3xx HTTP status and the "Location" header field, see Section 15.4 of [RFC9110]), it is not possible for RDAP

servers to know _a priori_ whether a client requesting an RDAP record is doing so because it wants to retrieve a related RDAP record, or its own, so it can only respond by providing the full RDAP response. The client must then parse that response in order to extract the relevant URL from the "links" property of the object.

This results in the wasteful expenditure of time, compute resources and bandwidth on the part of both the client and server.

This document describes an extension to RDAP that allows clients to request that an RDAP server redirect them to the URL of a related resource.

2. RDAP Redirect Request

To request a redirect to a related resource, the client sends an HTTP GET request with a URL of the form:

<base URL>redirects0_ref/<relation>/<lookup path>

The client replaces <base URL> with the applicable base URL (which, as per [RFC9224], has a trailing / character), <relation> with the desired relationship type, and <lookup path> with the lookup path of the object being sought (which, as per [RFC9082], does not have a leading / character).

For example, the URL of a redirect query for the domain example.com, where the base URL for the "com" TLD is https://rdap.example.com/rdap/, would be:

https://rdap.example.com/rdap/redirects0_ref/related/domain/example.com

The redirect query for the parent network of 192.0.2.42 with the base URL of https://rdap.example.net/ would be:

https://rdap.exampple.net/redirects0_ref/rdap-up/ip/192.0.2.42

Lookup paths for domain names, IP networks, autonomous system numbers, nameservers, and entities are described in [RFC9082]. Lookups defined by RDAP extensions may also use this extension.

Redirect requests for searches, where more than one object is returned, and help queries, as described by [RFC9083], are not supported. Servers MUST return an HTTP 400 for these requests.

3. RDAP Redirect Response

If the object specified in the request exists, a single appropriate link exists, and the client is authorised to perform the request, the server response MUST:

1. have an HTTP status code of 301 (Moved Permanently), 302 (Found), 303 (See Other), 307 (Temporary Redirect) or 308 (Permanent Redirect, see Section 15.4.9 of [RFC9110]); and

2. include an HTTP Location header field, whose value contains the URL of the linked resource.

If the server cannot find an appropriate link, the response MUST have an HTTP status of 404.

If an RDAP server holds in its datastore more than one relationship type for an object, a scenario that is possible but not common, only one of the URLs, as determined by server policy, can be returned.

The following examples use the HTTP/1.1 message exchange syntax as seen in [RFC9110].

An example of a redirect request from a domain registry to a domain registrar:

Client Request:

```
GET /redirects0_ref/related/domain/example.com HTTP/1.1
Accept: application/rdap+json
```

Server Response:

```
HTTP/1.1 307 Temporary Redirect
Location: https://registrar.example/domain/example.com
```

An example of a redirect request for a parent IPv4 network:

Client Request:

```
GET /redirects0_ref/rdap-up/ip/192.0.2.42 HTTP/1.1
Accept: application/rdap+json
```

Server Response:

```
HTTP/1.1 307 Temporary Redirect
Location: https://rir.example/ip/192.0.2.0/24
```

An example of a redirect request for a parent IPv6 network:

Client Request:

```
GET /redirects0_ref/rdap-up/ip/2001%3adb8%3a%3a1 HTTP/1.1
Accept: application/rdap+json"
```

Server Response:

```
HTTP/1.1 307 Temporary Redirect
Location: https://rir.example/ip/2001%3adb8%3a%3a/32
```

3.1. Selecting The Appropriate Link

When the server receives a redirect request, it must select which of an object’s links it should use to construct the response.

The rel property of the selected link MUST match <relation> path segment of the request. The type and hreflang properties of the link, if present, MUST match the Accept and (if specified) Accept-Language header fields of the request.

3.2. Caching by Intermediaries

To facilitate caching of RDAP resources by intermediary proxies, servers which provide a redirect based on the value of the Accept header field in the request MUST include a Vary header field (See Section 12.5.5 of [RFC9110]) in the response. This field MUST include accept, and MAY include other header field names.

Example:

```
Vary: accept, accept-language
```

3.3. Client Processing of Redirect Responses

Note that as per Section 10.2.2 of [!@RFC9110], the URI-reference in location header fields MAY be relative. For relative references, RDAP clients MUST compute the full URI using the request URI.

4. RDAP Conformance

Servers which implement this specification MUST include the string "redirects0" in the "rdapConformance" array in responses to RDAP "help" queries.

5. Bootstrap Use Case

The primary use case of this extension is a one-hop redirect, where the client is not interested in the use of this extension beyond the first redirect. Another use case is querying a bootstrap redirect server for the authoritative source of information according to the IANA RDAP bootstrap information.

Client Request:

```
GET /redirects0_ref/rdap-bootstrap/ip/2001%3adb8%3a%3a1 HTTP/1.1
Accept: application/rdap+json"
```

Server Response:

```
HTTP/1.1 307 Temporary Redirect
Location: https://rir1.example/ip/2001%3adb8%3a%3a/32
```

Other uses cases may exist, but for this specific use case, this document registers the "rdap-bootstrap" link relationship type.

6. Multi-Hop Redirect Limitations

In some scenarios, a target server might have a policy to issue another redirect using this extension. For example:

Client Request to rir1.example:

```
GET /redirects0_ref/rdap-top/ip/2001%3adb8%3a%3a1 HTTP/1.1
Accept: application/rdap+json"
```

Server Response:

```
HTTP/1.1 307 Temporary Redirect
Location: https://rir2.example/redirects0_ref/rdap-top/ip/2001%3adb8%3a%3a/32
```

In this scenario rir1.example is redirecting to rir2.example with a "/redirects0_ref" path. However, not all servers may support this extension. Therefore, the "/redirects0_ref" path defined in this specification MUST only be used in an HTTP redirect if the server issuing the redirect is assured that the target server of the redirect supports this extension.

Furthermore, servers SHOULD only use the "/redirects0_ref" path in an HTTP redirect when the link relationship type is one for a terminal relationship such as "rdap-top" and "rdap-bottom" (i.e., "rdap-up" and "rdap-down" do not explicitly express a relationship that is the end of a series of redirects).

7. IANA Considerations

7.1. RDAP Extension Identifier

IANA is requested to register the following value in the [rdap-extensions] Registry:

Extension identifier: redirects0

Registry operator: any.

Published specification: this document.

Contact: the authors of this document.

Intended usage: this extension allows clients to request to be redirected to a related resource for an RDAP resource.

7.2. Link Relations

IANA is requested to register the following value into the [link-relations] registry:

Relation Name: rdap-bootstrap

Description: Refers to an RDAP object for which a reference can be derived from RFC 9224.

Reference: This document once published as an RFC.

8. Security Considerations

A malicious HTTP redirect has the potential to create an infinite loop, which can exhaust resources on both client and server side.

To prevent such loops, RDAP servers which receive redirect requests for the self relation MUST respond with a 400 HTTP status.

As described in Section 15.4 of [!@RFC9110], when processing server responses, RDAP clients SHOULD detect and intervene in cyclical redirections.

9. Change Log

This section is to be removed before publishing as an RFC.

9.1. Changes from 02 to 03

* Consistely refer to "redirect" instead of "referral". This includes changing the extension identifier to redirects0 and the document title.

* Added Section 5 and Section 7.2.

* Correct specification of the redirect query path.

* Updated Section 4 to limit the use of the extension identifier to help responses.

* Include 308 in the list of redirection HTTP status codes.

Thanks to Jasdip Singh for identifying the last three of these issues.

9.2. Changes from 01 to 02

* Add reference to [gtld-rdap-profile] which describes how gTLD RDAP servers link to registrar RDAP resoures.

* Include <base path> in the path specification, and remove the / between <relation> and <lookup path> so that naive URL construction works.

* Reuse the language from RFC 7480 on HTTP status codes used for redirection.

* Fix HTTP status code in the examples.

* Described the risk of redirection loops and things clients and servers have to do.

9.3. Changes from 00 to 01

* Switch to using a path segment and a 30x redirect.

* Describe how the server behaves when multiple links exist.

9.4. Changes from draft-brown-rdap-referrals-02 to draft-ietf-regext-rdap-referrals-00

* Nothing apart from the name.

9.5. Changes from 01 to 02

* add this change log.

9.6. Changes from 00 to 01

* change extension identifer from registrar_link_header to redirects0.

10. References

10.1. Normative References

[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>.

[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>.

[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>.

10.2. Informative References

[RFC9224] Blanchet, M., "Finding the Authoritative Registration Data Access Protocol (RDAP) Service", STD 95, RFC 9224, DOI 10.17487/RFC9224, March 2022, <https://www.rfc-editor.org/info/rfc9224>.

[gtld-rdap-profile] ICANN, "gTLD RDAP Profile", 2024, <https://www.icann.org/gtld-rdap-profile>.

[link-relations]
IANA, "Link Relations", <https://www.iana.org/assignments/link-relations/link-relations.xhtml>.

[rdap-extensions]
IANA, "RDAP Extensions", <https://www.iana.org/assignments/rdap-extensions/rdap-extensions.xhtml>.

Authors' Addresses

Gavin Brown
ICANN
12025 Waterfront Drive, Suite 300
Los Angeles, CA 90292
United States of America
Email: gavin.brown@icann.org
URI: https://icann.org

Andy Newton
ICANN
12025 Waterfront Drive, Suite 300
Los Angeles, CA 90292
United States of America
Email: andy.newton@icann.org
URI: https://icann.org