Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 10, 2015

A. Newton
ARIN
S. Hollenbeck
Verisign Labs

October 7, 2014

Registration Data Access Protocol Query Format
draft-ietf-weirds-rdap-query-15

Abstract

This document describes uniform patterns to construct HTTP URLs that may be used to retrieve registration information from registries (including both Regional Internet Registries (RIRs) and Domain Name Registries (DNRs)) using "RESTful" web access patterns. These uniform patterns define the query syntax for the Registration Data Access Protocol (RDAP).

Status of This Memo

Copyright Notice

Table of Contents

1. 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 RFC 2119 [RFC2119].

1.1. Acronyms and Abbreviations

IDN: Internationalized Domain Name
IDNA: Internationalized Domain Names in Applications
DNR: Domain Name Registry
NFC: Unicode Normalization Form C
NFKC: Unicode Normalization Form KC

RDAP: Registration Data Access Protocol
REST: Representational State Transfer State Transfer. The term was first described in a doctoral dissertation [REST].
RESTful: An adjective that describes a service using HTTP and the principles of REST.
RIR: Regional Internet Registry

2. Introduction

This document describes a specification for querying registration data using a RESTful web service and uniform query patterns. The service is implemented using the Hypertext Transfer Protocol (HTTP) [RFC7230] and the conventions described in [I-D.ietf-weirds-using-http]. These uniform patterns define the query syntax for the Registration Data Access Protocol (RDAP).

The protocol described in this specification is intended to address deficiencies with the WHOIS protocol [RFC3912] that have been identified over time, including:

- Lack of standardized command structures,
- lack of standardized output and error structures,
- lack of support for internationalization and localization, and
- lack of support for user identification, authentication, and access control.

The patterns described in this document purposefully do not encompass all of the methods employed in the WHOIS and other RESTful web services of all of the RIRs and DNRs. The intent of the patterns described here are to enable queries of:

- networks by IP address,
- autonomous system numbers by number,
- reverse DNS meta-data by domain,
- name servers by name,
- registrars by name, and
- entities (such as contacts) by identifier.

Server implementations are free to support only a subset of these features depending on local requirements. If a server receives a query that it cannot process because it is not implemented it SHOULD return an HTTP 501 [RFC7231] error. It is also envisioned that each registry will continue to maintain WHOIS and/or other RESTful web services specific to their needs and those of their constituencies, and the information retrieved through the patterns described here may reference such services.

Likewise, future IETF standards may add additional patterns for additional query types. A simple pattern namespacing scheme is described in Section 5 to accommodate custom extensions that will not interfere with the patterns defined in this document or patterns defined in future IETF standards.

WHOIS services, in general, are read-only services. Therefore URL [RFC3986] patterns specified in this document are only applicable to the HTTP [RFC7231] GET and HEAD methods.

This document does not describe the results or entities returned from issuing the described URLs with an HTTP GET. The specification of these entities is described in [I-D.ietf-weirds-json-response].

Additionally, resource management, provisioning and update functions are out of scope for this document. Registries have various and divergent methods covering these functions, and it is unlikely a uniform approach for these functions will ever be possible.

HTTP contains mechanisms for servers to authenticate clients and for clients to authenticate servers (from which authorization schemes may be built) so such mechanisms are not described in this document. Policy, provisioning, and processing of authentication and authorization are out-of-scope for this document as deployments will have to make choices based on local criteria. Supported authentication mechanisms are described in [I-D.ietf-weirds-rdap-sec].

3. Path Segment Specification

The base URLs used to construct RDAP queries are maintained in an IANA registry described in [I-D.ietf-weirds-bootstrap]. Queries are formed by retrieving the appropriate base URL from the registry and appending a path segment specified in either Section 3.1 or Section 3.2. Generally, a registry or other service provider will provide a base URL that identifies the protocol, host and port, and this will be used as a base URL that the complete URL is resolved against, as per Section 5 of RFC 3986 [RFC3986]. For example, if the base URL is "http://example.com/rdap/", all RDAP query URLs will begin with "http://example.com/rdap/".

The bootstrap registry does not contain information for query objects that are not part of a global namespace, including entities and help. A base URL for an associated object is required to construct a complete query.

For entities, a base URL is retrieved for the service (domain, address, etc.) associated with a given entity. The query URL is

constructed by concatenating the base URL to the entity path segment specified in either Section 3.1.5 or Section 3.2.3.

For help, a base URL is retrieved for any service (domain, address, etc.) for which additional information is required. The query URL is constructed by concatenating the base URL to the help path segment specified in either Section 3.1.6.

3.1. Lookup Path Segment Specification

The resource type path segments for exact match lookup are:

- 'ip': Used to identify IP networks and associated data referenced using either an IPv4 or IPv6 address.
- 'autnum': Used to identify autonomous system registrations and associated data referenced using an AS Plain autonomous system number.
- 'domain': Used to identify reverse DNS (RIR) or domain name (DNR) information and associated data referenced using a fully-qualified domain name.
- 'nameserver': Used to identify a name server information query using a host name.
- 'entity': Used to identify an entity information query using a string identifier.

3.1.1. IP Network Path Segment Specification

Syntax: ip/<IP address> or ip/<CIDR prefix>/<CIDR length>

Queries for information about IP networks are of the form /ip/XXX/... or /ip/XXX/YY/... where the path segment following 'ip' is either an IPv4 dotted-decimal or IPv6 [RFC5952] address (i.e. XXX) or an IPv4 or IPv6 CIDR [RFC4632] notation address block (i.e. XXX/YY). Semantically, the simpler form using the address can be thought of as a CIDR block with a bitmask length of 32 for IPv4 and a bitmask length of 128 for IPv6. A given specific address or CIDR may fall within multiple IP networks in a hierarchy of networks, therefore this query targets the "most-specific" or smallest IP network which completely encompasses it in a hierarchy of IP networks.

The IPv4 and IPv6 address formats supported in this query are described in section 3.2.2 of [RFC3986], as IPv4address and IPv6address ABNF definitions. Any valid IPv6 text address format [RFC4291] can be used, compressed or not compressed. The restricted rules to write a text representation of an IPv6 address [RFC5952] are not mandatory. However, the zone id [RFC4007] is not appropriate in this context and therefore prohibited.

For example, the following URL would be used to find information for the most specific network containing 192.0.2.0:

```
http://example.com/rdap/ip/192.0.2.0
```

The following URL would be used to find information for the most specific network containing 192.0.2.0/24:

```
http://example.com/rdap/ip/192.0.2.0/24
```

The following URL would be used to find information for the most specific network containing 2001:db8::0:

```
http://example.com/rdap/ip/2001:db8::0
```

3.1.2. Autonomous System Path Segment Specification

Syntax: autnum/<autonomous system number>

Queries for information regarding autonomous system number registrations are of the form /autnum/XXX/... where XXX is an AS Plain autonomous system number [RFC5396]. In some registries, registration of autonomous system numbers is done on an individual number basis, while other registries may register blocks of autonomous system numbers. The semantics of this query are such that if a number falls within a range of registered blocks, the target of the query is the block registration, and that individual number registrations are considered a block of numbers with a size of 1.

For example, the following URL would be used to find information describing autonomous system number 12 (a number within a range of registered blocks):

```
http://example.com/rdap/autnum/12
```

The following URL would be used to find information describing 4-byte autonomous system number 65538:

```
http://example.com/rdap/autnum/65538
```

3.1.3. Domain Path Segment Specification

Syntax: domain/<domain name>

Queries for domain information are of the form /domain/XXXX/..., where XXXX is a fully-qualified (relative to the root) domain name [RFC1594] in either the in-addr.arpa or ip6.arpa zones (for RIRs) or a fully-qualified domain name in a zone administered by the server

operator (for DNRs). Internationalized domain names represented in either A-label or U-label format [RFC5890] are also valid domain names. See Section 6.1 for information on character encoding for the U-label format.

IDNs SHOULD NOT be represented as a mixture of A-labels and U-labels; that is, all internationalized labels in an IDN SHOULD be either A-labels or U-labels. It is possible for an RDAP client to assemble a query string from multiple independent data sources. Such a client might not be able to perform conversions between A-labels and U-labels. An RDAP server that receives a query string with a mixture of A-labels and U-labels MAY convert all the U-labels to A-labels, perform IDNA processing, and proceed with exact-match lookup. In such cases, the response to be returned to the query source may not match the input from the query source. Alternatively, the server MAY refuse to process the query.

The server MAY perform the match using either the A-label or U-label form. Using one consistent form for matching every label is likely to be more reliable.

The following URL would be used to find information describing the zone serving the network 192.0.2/24:

```
http://example.com/rdap/domain/2.0.192.in-addr.arpa
```

The following URL would be used to find information describing the zone serving the network 2001:db8:1::/48:

```
http://example.com/rdap/domain/1.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa
```

The following URL would be used to find information for the blah.example.com domain name:

```
http://example.com/rdap/domain/blah.example.com
```

The following URL would be used to find information for the xn--fo-5ja.example IDN:

```
http://example.com/rdap/domain/xn--fo-5ja.example
```

3.1.4. Name Server Path Segment Specification

Syntax: nameserver/<name server name>

The <name server name> parameter represents a fully qualified name as specified in RFC 952 [RFC0952] and RFC 1123 [RFC1123]. Internationalized names represented in either A-label or U-label

format [RFC5890] are also valid name server names. IDN processing for name server names uses the domain name processing instructions specified in Section 3.1.3. See Section 6.1 for information on character encoding for the U-label format.

The following URL would be used to find information for the ns1.example.com name server:

```
http://example.com/rdap/nameserver/ns1.example.com
```

The following URL would be used to find information for the ns1.xn--fo-5ja.example name server:

```
http://example.com/rdap/nameserver/ns1.xn--fo-5ja.example
```

3.1.5. Entity Path Segment Specification

Syntax: entity/<handle>

The <handle> parameter represents an entity (such as a contact, registrant, or registrar) identifier whose syntax is specific to the registration provider. For example, for some DNRs contact identifiers are specified in RFC 5730 [RFC5730] and RFC 5733 [RFC5733].

The following URL would be used to find information for the entity associated with handle XXXX:

```
http://example.com/rdap/entity/XXXX
```

3.1.6. Help Path Segment Specification

Syntax: help

The help path segment can be used to request helpful information (command syntax, terms of service, privacy policy, rate limiting policy, supported authentication methods, supported extensions, technical support contact, etc.) from an RDAP server. The response to "help" should provide basic information that a client needs to successfully use the service. The following URL would be used to return "help" information:

```
http://example.com/rdap/help
```

3.2. Search Path Segment Specification

A simple search to determine if an object exists (or not) without returning RDAP-encoded results can be performed using the HTTP HEAD method as described in Section 4.1 of [I-D.ietf-weirds-using-http].

The resource type path segments for search are:

- 'domains': Used to identify a domain name information search using a pattern to match a fully-qualified domain name.
- 'nameservers': Used to identify a name server information search using a pattern to match a host name.
- 'entities': Used to identify an entity information search using a pattern to match a string identifier.

RDAP search path segments are formed using a concatenation of the plural form of the object being searched for and an HTTP query string. The HTTP query string is formed using a concatenation of the question mark character ('?', ASCII value 0x003F), the JSON object value associated with the object being searched for, the equal sign character ('=', ASCII value 0x003D), and the search pattern. Search pattern query processing is described more fully in Section 4. For the domain and entity objects described in this document the plural object forms are "domains" and "entities".

Detailed results can be retrieved using the HTTP GET method and the path segments specified here.

3.2.1. Domain Search

Syntax: domains?name=<domain search pattern>

Syntax: domains?nsLdhName=<domain search pattern>

Syntax: domains?nsIp=<domain search pattern>

Searches for domain information by name are specified using this form:

```
/domains?name=XXXX
```

XXXX is a search pattern representing a domain name in "letters, digits, hyphen" format [RFC5890] in a zone administered by the server operator of a DNR. The following URL would be used to find DNR information for domain names matching the "example*.com" pattern:

```
http://example.com/rdap/domains?name=example*.com
```

Internationalized Domain Names (IDNs) in U-label format [RFC5890] can also be used as search patterns (see Section 4). Searches for these names are of the form /domains?name=XXXX, where XXXX is a search pattern representing a domain name in U-label format [RFC5890]. See Section 6.1 for information on character encoding for the U-label format.

Searches for domain information by name server name are specified using this form:

```
/domains?nsLdhName=YYYY
```

YYYY is a search pattern representing a host name in "letters, digits, hyphen" format [RFC5890] in a zone administered by the server operator of a DNR. The following URL would be used to search for domains delegated to name servers matching the "ns1.example*.com" pattern:

```
http://example.com/rdap/domains?nsLdhName=ns1.example*.com
```

Searches for domain information by name server IP address are specified using this form:

```
/domains?nsIp=ZZZZ
```

ZZZZ is a search pattern representing an IPv4 [RFC1166] or IPv6 [RFC5952] address. The following URL would be used to search for domains that have been delegated to name servers that resolve to the "192.0.2.0" address:

```
http://example.com/rdap/domains?nsIp=192.0.2.0
```

3.2.2. Name Server Search

Syntax: nameservers?name=<name server search pattern>

Syntax: nameservers?ip=<name server search pattern>

Searches for name server information by name server name are specified using this form:

```
/nameservers?name=XXXX
```

XXXX is a search pattern representing a host name in "letters, digits, hyphen" format [RFC5890] in a zone administered by the server operator of a DNR. The following URL would be used to find DNR information for name server names matching the "ns1.example*.com" pattern:

```
http://example.com/rdap/nameservers?name=ns1.example*.com
```

Internationalized name server names in U-label format [RFC5890] can also be used as search patterns (see Section 4). Searches for these names are of the form /nameservers?name=XXXX, where XXXX is a search pattern representing a name server name in U-label format [RFC5890]. See Section 6.1 for information on character encoding for the U-label format.

Searches for name server information by name server IP address are specified using this form:

```
/nameservers?ip=YYYY
```

YYYY is a search pattern representing an IPv4 [RFC1166] or IPv6 [RFC5952] address. The following URL would be used to search for name server names that resolve to the "192.0.2.0" address:

```
http://example.com/rdap/nameservers?ip=192.0.2.0
```

3.2.3. Entity Search

Syntax: entities?fn=<entity name search pattern>

Syntax: entities?handle=<entity handle search pattern>

Searches for entity information by name are specified using this form:

```
/entities?fn=XXXX
```

where XXXX is a search pattern representing an entity name as specified in Section 6.1 of [I-D.ietf-weirds-json-response]. The following URL would be used to find information for entity names matching the "Bobby Joe*" pattern.

```
http://example.com/rdap/entities?fn=Bobby%20Joe*
```

Searches for entity information by handle are specified using this form:

```
/entities?handle=XXXX
```

where XXXX is a search pattern representing an entity handle as specified in Section 6.1 of [I-D.ietf-weirds-json-response]. The following URL would be used to find information for entity names matching the "CID-40*" pattern.

```
http://example.com/rdap/entities?handle=CID-40*
```

URLs MUST be properly encoded according to the rules of [RFC3986]. In the example above, "Bobby Joe*" is encoded to "Bobby%20Joe*".

4. Query Processing

Servers indicate the success or failure of query processing by returning an appropriate HTTP response code to the client. Response codes not specifically identified in this document are described in [I-D.ietf-weirds-using-http].

4.1. Partial String Searching

Partial string searching uses the asterisk ('*', ASCII value 0x002A) character to match zero or more trailing characters. A character string representing multiple domain name labels MAY be concatenated to the end of the search pattern to limit the scope of the search. For example, the search pattern "exam*" will match "example.com" and "example.net". The search pattern "exam*.com" will match "example.com". Note that these search patterns include implied beginning and end of string regular expression markers, and the "example*.com" search would be translated into a POSIX regular expression as "^example.*\.com$". Additional pattern matching processing is beyond the scope of this specification.

If a server receives a search request but cannot process the request because it does not support a particular style of partial match searching, it SHOULD return an HTTP 422 [RFC4918] error. When returning a 422 error, the server MAY also return an error response body as specified in Section 7 of [I-D.ietf-weirds-json-response] if the requested media type is one that is specified in [I-D.ietf-weirds-using-http].

Partial matching is not feasible across combinations of Unicode characters because Unicode characters can be combined with another Unicode character or characters. Servers SHOULD NOT partially match combinations of Unicode characters where a Unicode character may be legally combined with another Unicode character or characters. It should be noted, though, that it may not always be possible to detect possible cases where a character could have been combined with another character, but was not, because of the way combining characters can be combined with many other characters.

Clients should avoid submitting a partial match search of Unicode characters where a Unicode character may be legally combined with another Unicode character or characters. Partial match searches with incomplete combinations of characters where a character must be

combined with another character or characters are invalid. Partial match searches with characters that may be combined with another character or characters are to be considered non-combined characters (that is, if character x may be combined with character y but character y is not submitted in the search string then character x is a complete character and no combinations of character x are to be searched).

4.2. Associated Records

Conceptually, a name-record in a database may include a link to an associated name-record, which may include a link to another such record, and so on. If an implementation is to return more than one name-record in response to a query, information from the records thereby identified is returned.

Note that this model includes arrangements for associated names, including those that are linked by policy mechanisms and names bound together for some other purposes. Note also that returning information that was not explicitly selected by an exact-match lookup, including additional names that match a relatively fuzzy search as well as lists of names that are linked together, may cause privacy issues.

5. Extensibility

This document describes path segment specifications for a limited number of objects commonly registered in both RIRs and DNRs. It does not attempt to describe path segments for all of the objects registered in all registries. Custom path segments can be created for objects not specified here using the process described in Section 6 of "HTTP usage in the Registration Data Access Protocol (RDAP)" [I-D.ietf-weirds-using-http].

Custom path segments can be created by prefixing the segment with a unique identifier followed by an underscore character (0x5F). For example, a custom entity path segment could be created by prefixing "entity" with "custom_", producing "custom_entity". Servers MUST return an appropriate failure status code for a request with an unrecognized path segment.

6. Internationalization Considerations

There is value in supporting the ability to submit either a U-label (Unicode form of an IDN label) or an A-label (ASCII form of an IDN label) as a query argument to an RDAP service. Clients capable of processing non-ASCII characters may prefer a U-label since this is more visually recognizable and familiar than A-label strings, but

clients using programmatic interfaces might find it easier to submit and display A-labels if they are unable to input U-labels with their keyboard configuration. Both query forms are acceptable.

Internationalized domain and name server names can contain character variants and variant labels as described in RFC 4290 [RFC4290]. Clients that support queries for internationalized domain and name server names MUST accept service provider responses that describe variants as specified in "JSON Responses for the Registration Data Access Protocol" [I-D.ietf-weirds-json-response].

6.1. Character Encoding Considerations

Servers can expect to receive search patterns from clients that contain character strings encoded in different forms supported by HTTP. It is entirely possible to apply filters and normalization rules to search patterns prior to making character comparisons, but this type of processing is more typically needed to determine the validity of registered strings than to match patterns.

An RDAP client submitting a query string containing non-US-ASCII characters converts such strings into Unicode in UTF-8 encoding. It then performs any local case mapping deemed necessary. Strings are normalized using Normalization Form C (NFC, [Unicode-UAX15]); note that clients might not be able to do this reliably. UTF-8 encoded strings are then appropriately percent-encoded [RFC3986] in the query URL.

After parsing any percent-encoding, an RDAP server treats each query string as Unicode in UTF-8 encoding. If a string is not valid UTF-8, the server can immediately stop processing the query and return an HTTP 400 error response code.

When processing queries, there is a difference in handling DNS names, including those including putative U-labels, and everything else. DNS names are treated according to the DNS matching rules as described in Section 3.1 of RFC 1035 [RFC1035] for NR-LDH labels and the matching rules described in Section 5.4 of RFC 5891 [RFC5891] for U-labels. Matching of DNS names proceeds one label at a time, because it is possible for a combination of U-labels and NR-LDH labels to be found in a single domain or host name. The determination of whether a label is a U-label or an NR-LDH label is based on whether the label contains any characters outside of the US-ASCII letters, digits, or hyphen (the so-called LDH rule).

For everything else, servers map fullwidth and halfwidth characters to their decomposition equivalents. Servers convert strings to the same coded character set of the target data that is to be looked up

or searched and each string is normalized using the same normalization that was used on the target data. In general, storage of strings as Unicode is RECOMMENDED. For the purposes of comparison, Normalization Form KC (NFKC, [Unicode-UAX15]) with case folding is used to maximize predictability and the number of matches. Note the use of case-folded NFKC as opposed to NFC in this case.

7. IANA Considerations

This document does not specify any IANA actions.

8. Security Considerations

Security services for the operations specified in this document are described in "Security Services for the Registration Data Access Protocol" [I-D.ietf-weirds-rdap-sec].

Search functionality typically requires more server resources (such as memory, CPU cycles, and network bandwidth) when compared to basic lookup functionality. This increases the risk of server resource exhaustion and subsequent denial of service due to abuse. This risk can be mitigated by developing and implementing controls to restrict search functionality to identified and authorized clients. If those clients behave badly, their search privileges can be suspended or revoked. Rate limiting as described in Section 5.5 of "HTTP usage in the Registration Data Access Protocol (RDAP)" [I-D.ietf-weirds-using-http] can also be used to control the rate of received search requests. Server operators can also reduce their risk by restricting the amount of information returned in response to a search request.

Search functionality also increases the privacy risk of disclosing object relationships that might not otherwise be obvious. For example, a search that returns IDN variants [RFC6927] that do not explicitly match a client-provided search pattern can disclose information about registered domain names that might not be otherwise available. Implementers need to consider the policy and privacy implications of returning information that was not explicitly requested.

9. Acknowledgements

This document is derived from original work on RIR query formats developed by Byron J. Ellacott of APNIC, Arturo L. Servin of LACNIC, Kaveh Ranjbar of the RIPE NCC, and Andrew L. Newton of ARIN. Additionally, this document incorporates DNR query formats originally described by Francisco Arias and Steve Sheng of ICANN and Scott Hollenbeck of Verisign Labs.

The authors would like to acknowledge the following individuals for their contributions to this document: Francisco Arias, Marc Blanchet, Ernie Dainow, Jean-Philippe Dionne, Behnam Esfahbod, John Klensin, Edward Lewis, John Levine, Mark Nottingham, and Andrew Sullivan.

10. References

10.1. Normative References

[I-D.ietf-weirds-bootstrap]
Blanchet, M. and G. Leclanche, "Finding the Authoritative Registration Data (RDAP) Service", draft-ietf-weirds-bootstrap-07 (work in progress), September 2014.

[I-D.ietf-weirds-json-response]
Newton, A. and S. Hollenbeck, "JSON Responses for the Registration Data Access Protocol (RDAP)", draft-ietf-weirds-json-response-09 (work in progress), September 2014.

[I-D.ietf-weirds-rdap-sec]
Hollenbeck, S. and N. Kong, "Security Services for the Registration Data Access Protocol", draft-ietf-weirds-rdap-sec-09 (work in progress), September 2014.

[I-D.ietf-weirds-using-http]
Newton, A., Ellacott, B., and N. Kong, "HTTP usage in the Registration Data Access Protocol (RDAP)", draft-ietf-weirds-using-http-12 (work in progress), September 2014.

[RFC0952] Harrenstien, K., Stahl, M., and E. Feinler, "DoD Internet host table specification", RFC 952, October 1985.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

[RFC1123] Braden, R., "Requirements for Internet Hosts - Application and Support", STD 3, RFC 1123, October 1989.

[RFC1166] Kirkpatrick, S., Stahl, M., and M. Recker, "Internet numbers", RFC 1166, July 1990.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.

[RFC4632] Fuller, V. and T. Li, "Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan", BCP 122, RFC 4632, August 2006.

[RFC4918] Dusseault, L., "HTTP Extensions for Web Distributed Authoring and Versioning (WebDAV)", RFC 4918, June 2007.

[RFC5396] Huston, G. and G. Michaelson, "Textual Representation of Autonomous System (AS) Numbers", RFC 5396, December 2008.

[RFC5730] Hollenbeck, S., "Extensible Provisioning Protocol (EPP)", STD 69, RFC 5730, August 2009.

[RFC5733] Hollenbeck, S., "Extensible Provisioning Protocol (EPP) Contact Mapping", STD 69, RFC 5733, August 2009.

[RFC5890] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework", RFC 5890, August 2010.

[RFC5891] Klensin, J., "Internationalized Domain Names in Applications (IDNA): Protocol", RFC 5891, August 2010.

[RFC5952] Kawamura, S. and M. Kawashima, "A Recommendation for IPv6 Address Text Representation", RFC 5952, August 2010.

[RFC7230] Fielding, R. and J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", RFC 7230, June 2014.

[RFC7231] Fielding, R. and J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content", RFC 7231, June 2014.

[Unicode-UAX15]
The Unicode Consortium, "Unicode Standard Annex #15: Unicode Normalization Forms", September 2013, <http://www.unicode.org/reports/tr15/>.

10.2. Informative References

[REST] Fielding, R. and R. Taylor, "Principled Design of the Modern Web Architecture", ACM Transactions on Internet Technology Vol. 2, No. 2, May 2002.

[RFC1594] Marine, A., Reynolds, J., and G. Malkin, "FYI on Questions and Answers - Answers to Commonly asked "New Internet User" Questions", RFC 1594, March 1994.

[RFC3912] Daigle, L., "WHOIS Protocol Specification", RFC 3912, September 2004.

[RFC4007] Deering, S., Haberman, B., Jinmei, T., Nordmark, E., and B. Zill, "IPv6 Scoped Address Architecture", RFC 4007, March 2005.

[RFC4290] Klensin, J., "Suggested Practices for Registration of Internationalized Domain Names (IDN)", RFC 4290, December 2005.

[RFC6927] Levine, J. and P. Hoffman, "Variants in Second-Level Names Registered in Top-Level Domains", RFC 6927, May 2013.

[RFC7159] Bray, T., "The JavaScript Object Notation (JSON) Data Interchange Format", RFC 7159, March 2014.

Appendix A. Change Log

Initial -00: Adopted as working group document.

-01: Added "Conventions Used in This Document" section. Added normative reference to draft-ietf-weirds-rdap-sec and some wrapping text in the Security Considerations section.

-02: Removed "unified" from the title. Rewrote the last paragraph of section 2. Edited the first paragraph of section 3 to more clearly note that only one path segment is provided. Added "bitmask" to "length" in section 3.1. Changed "lowest IP network" to "smallest IP network" in section 3.1. Added "asplain" to the description of autonomous system numbers in section 3.2. Minor change from "semantics is" to "semantics are" in section 3.2. Changed the last sentence in section 4 to more clearly specify error response behavior. Added acknowledgements. Added a paragraph in the introduction regarding future IETF standards and extensibility.

-03: Changed 'query' to 'lookup' in document title to better describe the 'exact match lookup' purpose of this document. Included a multitude of minor additions and clarifications provided by Marc Blanchet and Jean-Philippe Dionne. Modified the domain and name server sections to include support for IDN U-labels.

-04: Updated the domain and name server sections to use .example IDN U-labels. Added text to note that mixed IDN labels SHOULD NOT be used. Fixed broken sentences in Section 6.

-05: Added "help" path segment.

- -06: Added search text and removed or edited old search text.
- -07: Fixed query parameter typo by replacing "/?" with "?". Changed "asplain" to "AS Plain". Added entity search by handle. Corrected section references. Updated IDN search text.
- -08: Revised URI formats and added IANA instructions to create a registry entry for the "rdap" well-known prefix. Revised search processing text and added search privacy consideration. Synchronized examples with response draft.
- -09: More search processing and URI prefix updates. Updated fully-qualified domain name reference.
- -10: Added name server search by IP address.
- -11: Replaced reference to RFC 4627 with reference to RFC 7159. Replaced .well-known with bootstrap-defined prefix. Replaced references to RFC 2616 with references to RFC 7231 and draft-ietf-httpbis-http2, adding a note to make it clear that 2616 is an acceptable reference if http2 isn't ready when needed.
- -12: IDN label processing clarification. Added domain search by name server name and name server IP address. Minor text editing for consistency in the search sections. Replaced reference to draft-ietf-httpbis-http2 with a reference to RFC 7230 and removed reference note.
- -13: Added HTTP HEAD reference in Section 3.2.
- -14: Address WG last call comments.
- -15: Address AD review comments.

Authors' Addresses

Andrew Lee Newton
American Registry for Internet Numbers
3635 Concorde Parkway
Chantilly, VA 20151
US

Email: andy@arin.net
URI: http://www.arin.net

Scott Hollenbeck
Verisign Labs
12061 Bluemont Way
Reston, VA 20190
US

Email: shollenbeck@verisign.com
URI: http://www.verisignlabs.com/