

RDAP Operational Profile for gTLD Registries and Registrars

3 December 2015
Version: 1.0
Status: Draft

DRAFT

1. Contents

1. CONTENTS	2
2. INTRODUCTION	3
3. RDAP OPERATIONAL PROFILE	4
APPENDIX A: OPEN ISSUES	16
1. STATUS CODES FOR DOMAINS.....	16
2. LAST UPDATE OF RDAP DATABASE	16
3. MULTIPLE HOST OBJECTS FOR THE SAME NAME SERVER NAME.....	16
4. REGISTRAR EXPIRATION DATE	17
APPENDIX B: DATA ELEMENTS MAPPINGS	18
DOMAIN NAME QUERY ELEMENTS (THIS SECTION APPLIES TO BOTH REGISTRIES AND REGISTRARS) ..	18
NAME SERVER ELEMENTS (THIS SECTION ONLY APPLIES TO REGISTRIES)	20
REGISTRAR QUERIES ELEMENTS (THIS SECTION ONLY APPLIES TO REGISTRIES)	20
APPENDIX C: EPP STATUS CODE MAPPING	22
APPENDIX D: RDAP IETF STANDARDS	23
APPENDIX E: OTHER REFERENCES.....	24

2. Introduction

Lookup and search services for name resources have traditionally been offered through WHOIS (see [RFC3912](#)), with additional services such as web pages and bulk download. Despite its wide deployment and usage, shortcomings of the WHOIS protocol became apparent with the years. These include the lack of a standard data model, the lack of support for internationalization and the inability to authenticate users and provide differentiated services to classes of users.

In 2012, The Internet Engineering Task Force (IETF) chartered the [WEIRDS](#) (Web Extensible Internet Registration Data Services) working group to determine the needs of the community. This working group concluded in early 2015 with the publication of several specifications ([RFC7480](#), [RFC7481](#), [RFC7482](#), [RFC7483](#), [RFC7484](#) and [RFC7485](#)) defining the behavior of the Registry Data Access Protocol (RDAP), a standardized replacement for WHOIS.

The goal of this document is to define an RDAP profile for gTLD Registries and Registrars. This document covers the features within the RDAP protocol that are mandatory, the basic parameters, the mandatory set of objects to be implemented, and other allowed optional objects.

3. RDAP Operational Profile

The purpose of this profile is to specify the RDAP requirements that are in line with the current Whois service requirements. The profile is built from the related IETF standards, requirements from the gTLD Registry Agreement (RA), Registrar Accreditation Agreement 2013 (RAA), Whois-related advisories and consensus policies published by ICANN.

This document should be read together with the following:

- gTLD Registry Agreement (RA), <https://newgtlds.icann.org/sites/default/files/agreements/agreement-approved-09jan14-en.htm>
- Registrar Accreditation Agreement 2013 (RAA), <https://www.icann.org/resources/pages/approved-with-specs-2013-09-17-en>
- Additional Whois Information Policy (AWIP), <https://www.icann.org/resources/pages/policy-awip-2014-07-02-en>
- RDDS Clarification Advisory, <https://www.icann.org/resources/pages/registry-agreement-raa-rdds-2015-04-27-en>

Contracted parties operating according to an agreement, which includes a clause to implement a successor protocol to WHOIS, are required to deploy RDAP. As of the time this document is published, the set of contracted parties subject to this are: RAA 2013 Registrars, gTLDs of the 2012 round (a.k.a. new gTLDs) and other gTLDs (e.g. .biz, .com, .info, .jobs, .name, .org, .xxx).

Section 1 contains requirements applicable to both gTLD Registries and Registrars. Section 2 only applies to gTLD Registries while Section 3 only applies to gTLD Registrars.

The term "RDDS fields" in this document refers to the key/value pairs listed in the RDDS Specification of the Registry Agreement (RA), the Registration Data Directory Service (WHOIS) Specification of the Registrar Accreditation Agreement (RAA) 2013 or the RDDS Clarification Advisory.

The term "MUST", "MUST NOT", "REQUIRED", "SHOULD" and "MAY" in this document are to be interpreted as described in [RFC2119](#).

1. The following requirements apply to both gTLD Registries and Registrars (i.e. contracted parties)

- 1.1. Within the present profile and through the RAA and RA, all references to Registration Data Directory Services (RDDS) apply to the following services: WHOIS (port 43), web-based WHOIS and RDAP.

1.2. RDDS fields defined as Optional in this document are REQUIRED to be included in a response, using the appropriate mapping as defined in Appendix B, when germane to the query and data exists in the Registry or Registrar database, as the case may be.

1.3. RDAP protocol:

1.3.1. The RDAP service MUST implement the following RFCs:

- [RFC7480](#) - HTTP Usage in the Registration Data Access Protocol (RDAP)
- [RFC7481](#) - Security Services for the Registration Data Access Protocol (RDAP)
- [RFC7482](#) - Registration Data Access Protocol (RDAP) Query Format
- [RFC7483](#) - JSON Responses for the Registration Data Access Protocol (RDAP)
- [RFC7484](#) - Finding the Authoritative Registration Data (RDAP) Service

1.3.2. The RDAP service MUST be provided over HTTPS. The RDAP service MUST use the best practices for secure use of TLS as described in [RFC7525](#) or its successors.

1.3.3. A client MUST be able to successfully validate the TLS certificate used for the RDAP service with a *TLSA* record from the DNS (RFC 6698 and RFC 7671). The *Certificate Usage* field of the *TLSA* record MUST have a value of 1 or 3.

1.3.4. The TLS certificate used for the RDAP service MUST be issued by a Certificate Authority (CA) trusted by the major browsers and mobile operating systems such as the ones listed in the Mozilla Included CA Certificate List (<https://wiki.mozilla.org/CA:IncludedCAs>). The TLS certificate used for the RDAP service MUST be issued by a CA that follows the latest CAB Forum Baseline Requirements (<https://cabforum.org/baseline-requirements-documents>).

1.3.5. The RDAP service MUST support both GET and HEAD types of HTTP methods. HEAD requests are used to verify the existence of an object in the database, as specified in [RFC7480](#).

1.3.6. RDAP extensions, if used, MUST be registered in the IANA's RDAP Extensions registry (<https://www.iana.org/assignments/rdap-extensions/rdap-extensions.xhtml>), as defined in [RFC7480](#). Deployment of RDAP extensions in gTLD Registries operated under agreement with ICANN, are subject to approval by ICANN via the RSEP process.

- 1.3.7. An *rdapConformance* object [[RFC7483](#)] MUST be present in the topmost object of every response, and it MUST contain the conformance level of the RDAP protocol and of any extension, as specified in [RFC7483](#).
- 1.3.8. RDAP services MUST be available over both IPv4 and IPv6 transport. The resource records related to the RDAP service MUST be signed with DNSSEC, and the DNSSEC chain of trust from the root trust anchor to the name of the RDAP server MUST be valid at all times. The DNSSEC security algorithm used for zone signing at each level MUST be listed as standardized for Zone Signing in the IANA's Domain Name System Security (DNSSEC) Algorithm Numbers registry.
- 1.3.9. RDAP servers MUST NOT insert JSON members or objects that are not part of an ICANN-approved (e.g., per exhibit A of the RA) registered extension.
- 1.3.10. Contracted parties MUST use fully qualified domain names in RDAP responses.
- 1.4. Responses to RDAP queries:
 - 1.4.1. The RDAP server MUST support Internationalized Domain Name (IDN) RDAP lookup queries using A-label or U-label format [[RFC5890](#)] for domain name and name server objects. The RDAP server MUST accept a mixture of the two (i.e. A-label and U-label format) in the same RDAP lookup query.
 - 1.4.2. The source data used to generate the RDAP responses MUST be the same across all RDDS services (i.e. port-43 WHOIS, web-based WHOIS and RDAP).
 - 1.4.3. The case (i.e. uppercase and lowercase) of the data returned in RDAP responses MUST preserve the case received via EPP.
 - 1.4.4. The terms of service of the RDAP service MUST be specified in the *notices* object in the topmost JSON object of the response. The *notices* object MUST contain a *links* object [[RFC7483](#)]. The *links* object MUST contain an URL of the contracted party providing the RDAP service.
 - 1.4.5. In contact *entities* [[RFC7483](#)], phone numbers MUST be inserted as *tel* properties with a *voice* type parameter, as specified in [RFC6350](#), the vCard Format Specification and its corresponding JSON mapping [RFC7095](#).
 - 1.4.6. In contact *entities*, fax numbers are Optional RDDS fields, and MUST be inserted as *tel* properties with a *fax* type parameter, as specified in

[RFC6350](#), the vCard Format Specification and its corresponding JSON mapping [RFC7095](#).

- 1.4.7. RDAP Help queries [[RFC7482](#)] MUST be answered and include a *links* member with a URL to a document that provides usage information, policy and other explanatory material.
- 1.4.8. Truncated RDAP responses MUST contain a *notices* member describing the reason of the truncation. The *notices* object type MUST be of the form "Response truncated due to {authorization|load|unexplainable reason}".
- 1.4.9. Truncated RDAP objects MUST contain a *remarks* member describing the reason of the truncation. The *remarks* object type MUST be of the form "Result set truncated due to {authorization|load|unexplainable reason}".
- 1.4.10. An RDAP response MUST contain a *remarks* member with a *description* containing the string "This response conforms to the RDAP Operational Profile for gTLD Registries and Registrars version 1.0".
- 1.4.11. If permitted or required by an ICANN agreement provision, waiver, or Consensus Policy, an RDAP response may contain redacted registrant, administrative, technical and/or other contact information. If any information is redacted, the response MUST include a *remarks* member with *title* "Data Policy", *type* "object truncated due to authorization", a *description* containing the string "Some of the data in this object has been removed" and a *links* member with the elements *rel:alternate* and *href* indicating where the data policy can be found. An entity with redacted information MUST include the "removed" value in the *status* element.
- 1.4.12. An *entity* object within an RDAP response MUST contain an *events* member with the following events:
 - An event of *eventAction* type *registration*.
 - An event of *eventAction* type *last changed*. The event of *eventAction* type *last changed* MUST be omitted if the *Contact Object* (as defined in [RFC5733](#)) has not been updated since it was created.
 - An event of *eventAction* type *last update of RDAP database*. [[The *eventAction* type *last update of RDAP database* is pending registration in the IANA RDAP JSON Values registry]].
- 1.4.13. In the case of a TLD in which name servers are specified as *Host Objects* (as defined in [RFC5732](#)), a *nameserver* object within an RDAP response MUST contain an *events* member with the following events:

- An event of *eventAction* type *registration*.
- An event of *eventAction* type *last changed*. The event of *eventAction* type *last changed* MUST be omitted if the *Host Object* has not been updated since it was created.
- An event of *eventAction* type *last update of RDAP database*. [[The *eventAction* type *last update of RDAP database* is pending registration in the IANA RDAP JSON Values registry]].

1.4.14. The RDAP database (i.e. database used to generate the RDAP responses) MUST only include registration data from the Registry or Registrar database, as the case may be. The RDAP database must be updated within the allowed Service Level Requirement (SLR) (e.g. RDDS update time, ≤60 minutes). In a case where the contracted party is querying its database directly, and therefore, using real-time data, the *eventAction* type *last update of RDAP database* MUST show the timestamp of the response to the query. [[The *eventAction* type *last update of RDAP database* is pending registration in the IANA RDAP JSON Values registry]].

1.5. Responses to domain name RDAP queries:

- 1.5.1. The top-level domain object [RFC7483] in the RDAP response MUST contain the A-label format of the domain in the *ldhName* member [RFC7483].
- 1.5.2. The top-level domain object in the RDAP response MUST contain the U-label format of the domain in the *unicodeName* member [RFC7483], only if the domain name is an IDN.
- 1.5.3. The top-level domain object in the RDAP response MUST contain a *status* member [RFC7483] with the domain statuses in the SRS. The status MUST be a valid status type per the IANA's RDAP JSON Values registry.
- 1.5.4. The *status* member value of the RDAP domain, *nameserver* [RFC7483] and *entity* objects MUST conform to the values defined in IANA's RDAP JSON Values (<https://www.iana.org/assignments/rdap-json-values/rdap-json-values.xhtml>) of status type.
- 1.5.5. The *status* member of a domain object in the RDAP response MUST match the EPP Status codes in the SRS within the allowed timeframe (e.g. ≤60 minutes) to update the RDAP database from the Registry or Registrar database, as the case may be.
- 1.5.6. The *status* member of a domain object in the RDAP response MUST be a valid status type per the mapping table in Appendix C.

- 1.5.7. The *domain* object in the RDAP response MUST contain the name servers of the domain in the *nameservers* member. Each *nameserver* object MUST contain the following member: *ldhName*. The following members are Optional: *ipAddresses* [RFC7483], *unicodeName*, *handle* [RFC7483] (ROID of the host object, *<host:roid>* as defined in RFC5732), and *status*. In the case of a TLD in which name servers are specified as domain attributes, the *nameserver* object MUST NOT contain the following members: *handle* and *status*.
- 1.5.8. The *domain* object in the RDAP response MUST contain *entities* with the following roles. Exactly one *entity* per role MUST be present in the response, each of them with a *handle* (ROID of the contact object, *<contact:roid>*, as defined in RFC5733) and valid members *fn*, *adr*, *tel*, *email* (as specified in RFC6350, the vCard Format Specification and its corresponding JSON mapping RFC7095):
- *registrant*
 - *administrative*
 - *technical*
- 1.5.9. The *domain* object in the RDAP response MAY contain an *entity* of the *billing* role with a *handle* (ROID of the contact object, *<contact:roid>*, as defined in RFC5733) and valid members *fn*, *adr*, *tel*, *email*.
- 1.5.10. The following RDDS fields used to generate the *adr* member of the entities with the *registrant*, *administrative* and *technical* roles are REQUIRED to be included in the RDAP response:
- Registrant/Admin/Tech Street
 - Registrant/Admin/Tech City
 - Registrant/Admin/Tech Country
- 1.5.11. The following RDDS fields are Optional:
- Registrant/Admin/Tech Organization
 - Registrant/Admin/Tech State/Province
 - Registrant/Admin/Tech Postal Code
 - Registrant/Admin/Tech Phone Ext
 - Registrant/Admin/Tech Fax
 - Registrant/Admin/Tech Fax Ext
- 1.5.12. The *domain* object in the RDAP response MUST contain an *entity* with the *registrar* role (called registrar entity in this section). The *handle* of the *entity* MUST be equal to the IANA Registrar ID. A valid *fn* member MUST be present in the *registrar* entity. Other members MAY be present in the *entity* (as specified in RFC6350, the vCard Format Specification and its corresponding JSON mapping RFC7095). Contracted parties MUST include an *entity* with the *abuse* role (called

Abuse Entity in this section) within the registrar *entity*. The Abuse Entity MUST include *tel* and *email* members, and MAY include other members.

1.5.13. The *entity* with the *registrar* role in the RDAP response MUST contain a *publicIDs* member [RFC7483] to identify the IANA Registrar ID from the IANA's Registrar ID registry (<https://www.iana.org/assignments/registrar-ids/registrar-ids.xhtml>). The type value of the *publicID* object MUST be equal to IANA Registrar ID.

1.5.14. The *domain* object in the RDAP response MUST contain the following events:

- An event of *eventAction* type *registration*.
- An event of *eventAction* type *expiration*.
- An event of *eventAction* type *last changed*. The event of *eventAction* type *last changed* MUST be omitted if the domain name has not been updated since it was created.
- An event of *eventAction* type *registrar expiration*. [[The *eventAction* type *registrar expiration* is pending registration in the IANA RDAP JSON Values registry]].
- An event of *eventAction* type *last update of RDAP database*. [[The *eventAction* type *last update of RDAP database* is pending registration in the IANA RDAP JSON Values registry]].

1.5.15. The *domain* object in the RDAP response MAY contain the following events:

- An event of *eventAction* type *last transferred*, with the last date and time that the domain was transferred. The event of *eventAction* type *last transferred* MUST be omitted if the domain name has not been transferred since it was created.

1.5.16. *Entities* MUST use jCard [RFC7095] structured addresses.

1.5.17. If the queried domain name is allocated, the following applies: If allocated variant domain names exist for the queried domain name, or if the domain name is an allocated variant domain name, the domain object in the RDAP response MUST contain a *variants* member [RFC7483]. The *variants relation* member MUST contain valid variant *relation* types as defined in the IANA's RDAP JSON Values registry. If the queried domain name is an allocated variant name, the original name MUST be included in the *variants* member.

1.5.18. A domain name RDAP response MUST contain a *remarks* member with a *title* "EPP Status Codes", a *description* containing the string "For more information on domain status codes, please visit

<https://icann.org/epp>" and a *links* member with the <https://icann.org/epp> URL.

- 1.5.19. The *domain* object in the RDAP response MUST contain a *secureDNS* member [RFC7483] including at least a *delegationSigned* element. Other elements (e.g. *dsData*, *maxSigLife*) of the *secureDNS* member MUST be included, if the domain name is signed and the elements are stored in the Registry or Registrar database, as the case may be.
- 1.5.20. A domain name RDAP response MUST contain a *remarks* member with a *title* "Whois Inaccuracy Complaint Form", a *description* containing the string "URL of the ICANN Whois Inaccuracy Complaint Form: <https://www.icann.org/wicf>" and a *links* member with the <https://www.icann.org/wicf> URL.
- 1.5.21. The returned *domain* object in the RDAP response MAY contain exactly one entity with the *reseller* role, if the domain name was registered through a reseller.
- 1.5.22. The *domain* object *handle* in the RDAP response MUST contain the Repository Object Identifier (ROID of the domain object, *<domain:roid>* as defined in [RFC5731] for the domain name object.

Note: a mapping of RDAP elements/objects is available in Appendix B.

2. The following requirements apply to Registries only:

- 2.1. Registries MUST support RDAP search requests for name servers by IP address as defined in [RFC7482] section 3.2.2.
- 2.2. If a Registry supports multiple host objects with the same name, the Registry MUST support the capability to respond with a set of host objects in response to a name server lookup, when an RFC defining this capability has been published.
- 2.3. The RDAP *domain* response MUST contain a *links* object, as defined in [RFC7483] section 4.2, in the topmost JSON object of the response, if the registration data of the registrant, administrative, or a technical contact is not available in the registry (i.e. a "thin" registration). The *links* object MUST contain the elements *rel:related* and *href* pointing to the Registrar's RDAP URL of the queried domain name object.
- 2.4. Registries offering Whois contact lookup (e.g., per exhibit A of their RA) MUST support RDAP lookup request for *entities* with any role within other objects using the *handle* (as described in 3.1.5 of [RFC7482]).
- 2.5. Sections 1.5.8, 1.5.9, 1.5.10 and 1.5.11 do not apply for "thin" registrations.

2.6. Reporting requirements:

- 2.6.1. Specification 3 of the RA specifies the format and content of the monthly reporting for Registry operators. The following rows are added to the Registry Functions Activity Report under section 2:

Field #	Field Name	Description
40	rdap-queries	Number of RDAP queries during the period.
41	rdap-rate-limit	Number of RDAP queries refused due to rate limiting for the period.
42	rdap-redirects	Number of HTTP redirects for the period.
42	rdap-authenticated	Number of authenticated RDAP queries for the period.
43	rdap-search-domain	Number of RDAP domain search queries for the period.
44	rdap-search-entity	Number of RDAP entity search queries for the period.
45	rdap-truncated-authorization	Number of RDAP responses truncated due to authorization. Includes both results and object truncation events.
46	rdap-truncated-load	Number of RDAP responses truncated due to server load. Includes both results and object truncation events.
47	rdap-truncated-unexplainable	Number of RDAP responses truncated due to unexplainable reasons. Includes both results and object truncation events.

2.7. RDAP Bootstrapping requirements:

- 2.7.1. The base URL of RDAP services MUST be registered in the IANA's Bootstrap Service registry for Domain Name Space (<https://www.iana.org/assignments/rdap-dns/rdap-dns.xhtml>), as described in [RFC7484](#). A separate entry is required for each TLD.
- 2.7.2. When the RDAP service base URL needs to be changed, the previous URL and the new one MUST remain in operation until: 1) the IANA's Bootstrap Service registry for Domain Name Space is updated, and 2) the date and time in the Expires HTTP header of a HTTP/GET request performed on the IANA's Bootstrap registry for Domain Name Space (after the new URL has been published) has elapsed.
- 2.7.3. An IANA's Bootstrap registry for Domain Name Space entry MUST be populated with an HTTPS URL only.

2.8. Response to registrar queries:

- 2.8.1. In response to registrar queries, the returned RDAP response MUST be an *entity* with *registrar* role, with a *handle* and valid elements *fn*, *adr*, *tel*, *email*.

- 2.8.2. Registrar object lookup using an entity search on the *fn* element MUST be supported.
 - 2.8.3. Registries MUST support lookup for *entities* with the *registrar* role within other objects using the *handle* (as described in 3.1.5 of [RFC7482](#)). The *handle* of the *entity* with the *registrar* role MUST be equal to IANA Registrar ID. The *entity* with the *registrar* role in the RDAP response MUST contain a *publicIDs* member to identify the IANA Registrar ID from the IANA's Registrar ID registry. The type value of the *publicID* object MUST be equal to IANA Registrar ID.
 - 2.8.4. The *adr* member in the RDAP response for a Registrar query MUST at least contain the following RDDS fields:
 - Street
 - City
 - Country
 - 2.8.5. The following RDDS fields in the RDAP response for a Registrar query are Optional:
 - State/Province
 - Postal Code
 - Fax Number
 - 2.8.6. The RDAP response SHOULD contain at least two *entities*, with the *administrative* and *technical* roles respectively within the *entity* with the *registrar* role. The *entities* with the *administrative* and *technical* roles MUST contain a *handle* and valid *fn*, *tel*, *email* members, and MAY contain a valid *adr* element.
- 2.9. Responses to nameserver RDAP queries:
- 2.9.1. Registries MUST support *nameserver* lookup queries based on the name server's name as specified in 3.1.4 of [RFC7482](#).
 - 2.9.2. The name server's name MUST be specified in the *ldhName* in A-label format.
 - 2.9.3. All known glue record IPv4 and IPv6 addresses for the name server MUST be listed in the *ipAddresses* member.
 - 2.9.4. The *unicodeName* member MUST be present in the response to a *nameserver* lookup, if the name server has an IDN label.
 - 2.9.5. The Registrar RDDS field is Optional; if present, it MUST be represented as an *entity* with the *registrar* role. The *handle* of the *entity* with the *registrar* role MUST be equal to IANA Registrar ID. The entity with the *registrar* role in the RDAP response MUST contain a *publicIDs* member

to identify the IANA Registrar ID from the IANA's Registrar ID registry. The type value of the *publicID* object MUST be equal to IANA Registrar ID.

- 2.9.6. In the case of a Registry in which name servers are specified as domain attributes, the existence of a name server used as an attribute for an allocated domain name MUST be treated as equivalent to the existence of a host object.
- 2.9.7. The *nameserver* object MUST contain the following member: *ldhName*. The following members are Optional: *ipAddresses* [RFC7483], *unicodeName*, *handle* [RFC7483] (ROID of the host object, *<host:roid>* as defined in RFC5732) and *status*. In the case of a Registry in which name servers are specified as domain attributes, the *nameserver* object MUST NOT contain the following members: *handle* and *status*.

3. The following requirements apply to Registrars only:

3.1. Responses to domain name RDAP queries:

- 3.1.1. A Registrar is REQUIRED to respond with information regarding domain names for which the Registrar is the Sponsoring Registrar.
- 3.1.2. A Registrar MUST return an HTTP 404 response when the Registrar is not the Sponsoring Registrar for the domain name.
- 3.1.3. The *domain* object *handle* in the RDAP response MUST contain the Repository Object Identifier (ROID of the domain object, *<domain:roid>* as defined in RFC5731) for the Domain Name object. For example, a Registrar could obtain the ROID from the Registry via EPP and cache the information locally after creating or gaining a domain name via a transfer.
- 3.1.4. The *entity handle* in the RDAP response MUST contain the Repository Object Identifier (ROID of the contact object, *<contact:roid>*, as defined in RFC5733) for the Contact object. For example, a Registrar could obtain the ROID from the Registry via EPP and cache the information locally. The RAA 2013 defines that this information MUST be shown if available from the Registry. If this information is not available from the Registry (e.g., a "thin" Registry), the *handle* MUST contain the unique identifier within the Registrar.
- 3.1.5. The *eventAction* type *last changed* reflects the date and time of the latest successful update known to the Registrar. Registrars are not required to constantly refresh this date from the Registry.

- 3.1.6. The *status* element MUST reflect the latest known set of EPP statuses in the Registry. Registrars are not required to constantly refresh the EPP statuses from the Registry.

DRAFT

Appendix A: Open Issues

The following issues have been identified in the RDAP base specification required to mirror the current RDDS requirements. The following section describes the issues found, and the possible solutions. Implementers are advised that the RFC(s) published in the future to handle this missing functionality may be different from the proposed solution in this section.

1. Status Codes for Domains

The current Whois requirements (see, Additional Whois Information Policy, <https://www.icann.org/resources/pages/policy-awip-2014-07-02-en>) require the use of the EPP domain statuses codes in responses.

This issue is discussed in the IETF document “Extensible Provisioning Protocol (EPP) and Registration Data Access Protocol (RDAP) Status Mapping” (<https://tools.ietf.org/html/draft-gould-epp-rdap-status-mapping>). This Internet Draft suggests new RDAP status codes to cover the EPP domain statuses that cannot be mapped to the currently defined RDAP statuses.

Possible Solutions

Add RDAP statuses to the IANA's RDAP JSON Values registry mapping to the missing EPP domain name statuses.

2. Last update of RDAP database

The base RDAP specification does not define an element to specify the last date and time that the RDDS database used by the RDAP service was updated from the SRS. This element is required to map the "Last update of WHOIS database" RDDS field.

Possible Solutions

A potential solution would be to add a new event action (e.g. "last update of RDAP database") to specify the last date and time that the database used by the RDAP service was updated from the Registry or Registrar database, as the case may be.

3. Multiple host objects for the same name server name

Items 29, 30, 32 of the RDDS Clarification Advisory (<https://www.icann.org/resources/pages/registry-agreement-raa-rdds-2015-04-27-en>) cover the case of the existence of multiple host objects for the same name server name. This requirement is not supported by the RDAP specification.

Possible Solutions

The RDAP name server lookup specification could be extended to provide the information of multiple name servers.

4. Registrar expiration date

RDAP does not include an event to specify the registrar registration expiration date as described in the RAA 2013.

Possible Solutions

A potential solution would be to add a new event action (e.g. "registrar expiration") to specify the registrar registration expiration date.

Appendix B: Data Elements Mappings

Domain Name Query Elements (this section applies to both registries and registrars)

The table below provides data elements mappings of RDDS fields to RDAP for Domain Name queries.

RDDS field		Location in RDAP Response
Registry	Registrar	
General		
WHOIS Server / Referral URL		links object with rel:related
	WHOIS Server / Referral URL	[Not applicable in RDAP]
Last update of WHOIS database	Last update of WHOIS database	[Pending solution, see Section 4]
Domains		
Domain Name	Domain Name	ldhName
Domain ID	Registry Domain ID	handle
Updated Date	Updated Date	events.eventAction “last changed”
Creation Date	Creation Date	events.eventAction “registration”
Registry Expiry Date		events.eventAction “expiration”
Registrar Registration Expiration Date	Registrar Registration Expiration Date	[Pending solution, see Section 4]
Domain Status	Domain Status	status object
Name Server	Name Server	nameservers.ldhname
DNSSEC	DNSSEC	secureDNS object
Internationalized Domain Name	Internationalized Domain Name	unicodeName
Registrar		
Sponsoring Registrar	Registrar	entities.roles registrar
Sponsoring Registrar IANA ID	Registrar IANA ID	publicIDs.identifier
Registrar Abuse Contact Email	Registrar Abuse Contact Email	entities.role abuse
Registrar Abuse Contact Phone	Registrar Abuse Contact Phone	entities role abuse
Reseller		
Reseller	Reseller	entities.roles reseller
Registrant Contact		entities role registrant
		entity.handle
Registrant ID	Registry Registrant ID	
Registrant Name	Registrant Name	jCard “fn”
Registrant Organization	Registrant Organization	org
Registrant Street	Registrant Street	Grouped into the adr member.
Registrant City	Registrant City	
Registrant State/Province	Registrant State/Province	
Registrant Postal Code	Registrant Postal Code	
Registrant Country	Registrant Country	
Registrant Phone Number	Registrant Phone Number	tel with a type parameter voice
Registrant Phone Number Ext	Registrant Phone Number Ext	ext

Registrant Fax	Registrant Fax	tel with a type parameter fax
Registrant Fax Ext	Registrant Fax Ext	ext
Registrant Email	Registrant Email	email
Administrative Contact		entity role administrative
Admin ID	Registry Admin ID	entity.handle
Admin Name	Admin Name	jCard "fn"
Admin Organization	Admin Organization	org
Admin Street	Admin Street	Grouped into the adr member.
Admin City	Admin City	
Admin State/Province	Admin State/Province	
Admin Postal Code	Admin Postal Code	
Admin Country	Admin Country	
Admin Phone Number	Admin Phone Number	tel with a type parameter voice
Admin Phone Number Ext	Admin Phone Number Ext	ext
Admin Fax	Admin Fax	tel with a type parameter fax
Admin Fax Ext	Admin Fax Ext	ext
Admin Email	Admin Email	email
Technical Contact		entitites.role technical
Tech ID	Registry Tech ID	entity.handle
Tech Name	Tech Name	jCard "fn"
Tech Organization	Tech Organization	org
Tech Street	Tech Street	Grouped into the adr member.
Tech City	Tech City	
Tech State/Province	Tech State/Province	
Tech Postal Code	Tech Postal Code	
Tech Country	Tech Country	
Tech Phone Number	Tech Phone Number	tel with a type parameter voice
Tech Phone Number Ext	Tech Phone Number Ext	ext
Tech Fax	Tech Fax	tel with a type parameter fax
Tech Fax Ext	Tech Fax Ext	ext
Tech Email	Tech Email	email
Billing Contact		entities role billing
Billing ID	Registry Billing ID	entity.handle
Billing Name	Billing Name	jCard "fn"
Billing Organization	Billing Organization	org
Billing Street	Billing Street	Grouped into the adr member.
Billing City	Billing City	
Billing State/Province	Billing State/Province	
Billing Postal Code	Billing Postal Code	
Billing Country	Billing Country	
Billing Phone Number	Billing Phone Number	tel with a type parameter voice
Billing Phone Number Ext	Billing Phone Number Ext	ext
Billing Fax	Billing Fax	tel with a type parameter fax
Billing Fax Ext	Billing Fax Ext	ext
Billing Email	Billing Email	email

Name Server Elements (this section only applies to registries)

The table below provides data elements mappings of RDDS fields to RDAP for Name Server queries.

RDDS field	Location in RDAP Response
Registry	
Server Name	nameserver.ldhName
IP Address	nameserver.ipAddresses
Registrar	entities.roles registrar
WHOIS Server / Referral URL	[Not applicable in RDAP]
Last update of WHOIS database (RA and RAA)	[Pending solution, see Section 4]

Registrar Queries Elements (this section only applies to registries)

The table below provides data elements mappings of RDDS fields to RDAP for Registrar queries.

RDDS field	Location in RDAP Response
Registry	
General	
WHOIS Server / Referral URL	[Not applicable in RDAP]
Last update of WHOIS database (RA and RAA)	[Pending solution, see Section 4]
Registrar	entities.role registrar
Registrar Name	jCard fn
Registrar Street	Grouped into the adr member.
Registrar City	
Registrar State/Province	
Registrar Postal Code	
Registrar Country	
Registrar Phone Number	tel with a type parameter voice
Registrar Phone Number Ext	ext
Registrar Fax	tel with a type parameter fax
Registrar Fax Ext	ext
Registrar Email	email
Registrar Administrative Contact	entity.role administrative
Admin Contact	jCard fn
Phone Number	tel with a type parameter voice
Phone Number Ext	ext
Fax Number	tel with a type parameter fax
Fax Number Ext	ext
Email	email
Registrar Technical Contact	entitites.role technical
Technical Contact	jCard fn
Phone Number	tel with a type parameter voice

Phone Number Ext	ext
Fax Number	tel with a type parameter fax
Fax Number Ext	ext
Email	email

DRAFT

Appendix C: EPP status code mapping

EPP Status Code	RDAP Status Code (*: new RDAP Status Code proposed in https://draft-gould-epp-rdap-status-mapping)	Comments
addPeriod	<i>add period*</i>	
autoRenewPeriod	<i>auto renew period*</i>	
clientDeleteProhibited	<i>client delete prohibited*</i>	
clientHold	<i>client hold*</i>	
clientRenewProhibited	<i>client renew prohibited*</i>	
clientTransferProhibited	<i>client transfer prohibited*</i>	
clientUpdateProhibited	<i>client update prohibited*</i>	
clientDeleteProhibited	<i>client delete prohibited*</i>	
inactive	inactive	
linked (contact status from RFC5733)	associated	Applies to nameservers and entities, when they are associated with a network resource or domain.
ok	active	
pendingCreate	pending create	
pendingDelete	pending delete	
pendingRenew	pending renew	
pendingRestore	<i>pending restore*</i>	
pendingTransfer	pending transfer	
pendingUpdate	pending update	
redemptionPeriod	<i>redemption period*</i>	
renewPeriod	<i>renew period*</i>	
serverDeleteProhibited	<i>server delete prohibited*</i>	
serverHold	<i>server hold*</i>	
serverRenewProhibited	<i>server renew prohibited*</i>	
serverTransferProhibited	<i>server transfer prohibited*</i>	
serverUpdateProhibited	<i>server update prohibited*</i>	
transferPeriod	<i>transfer period*</i>	

Appendix D: RDAP IETF Standards

RDAP standards are a set of specifications, which together provide a complete RDAP service. Each specification is briefly described below.

RFC7480 - HTTP Usage in the Registration Data Access Protocol (RDAP)

<https://www.rfc-editor.org/rfc/rfc7480.txt>

Describes usage of HTTP transport for RDAP, error messages, RDAP extensions, rate limiting and internationalization with URIs.

RFC7481 - Security Services for the Registration Data Access Protocol (RDAP)

<https://www.rfc-editor.org/rfc/rfc7481.txt>

Covers access control, authentication, authorization, privacy, data confidentiality and RDAP services availability considerations.

RFC7482 - Registration Data Access Protocol (RDAP) Query Format

<https://www.rfc-editor.org/rfc/rfc7482.txt>

Defines the URL patterns for networks, autonomous systems, reverse DNS, name servers, registrars and entities queries. Also covers help requests, search (wildcards) and internationalization in requests.

RFC7483 - JSON Responses for the Registration Data Access Protocol (RDAP)

<https://www.rfc-editor.org/rfc/rfc7483.txt>

Defines JSON object classes for domains, name servers, entities, IP networks and autonomous system numbers. Describe answers to help queries, searches, JSON-embedded error codes and truncated answers.

RFC7484 - Finding the Authoritative Registration Data (RDAP) Service

<https://www.rfc-editor.org/rfc/rfc7484.txt>

Describes a method to find the authoritative server for RDAP data.

IANA RDAP JSON Values Registry

<https://www.iana.org/assignments/rdap-json-values/rdap-json-values.xhtml>

This registry defines valid values for RDAP JSON status, role, notices and remarks, event action, and domain variant relation, as defined in RFC7483.

IANA Bootstrap Service Registry for Domain Name Space

<https://www.iana.org/assignments/rdap-dns/rdap-dns.xhtml>

Appendix E: Other References

RFC7485 - Inventory and Analysis of WHOIS Registration Objects

<https://www.rfc-editor.org/rfc/rfc7485.txt>

Extensible Provisioning Protocol (EPP) and Registration Data Access Protocol (RDAP) Status Mapping

<https://tools.ietf.org/html/draft-gould-epp-rdap-status-mapping>

jCard: The JSON Format for vCard

<https://tools.ietf.org/html/rfc7095>

vCard Format Specification

<https://tools.ietf.org/html/rfc6350>

gTLD Registry Agreement

<https://newgtlds.icann.org/sites/default/files/agreements/agreement-approved-09jan14-en.htm>

2013 Registrar Accreditation Agreement

<https://www.icann.org/resources/pages/approved-with-specs-2013-09-17-en>

ICANN Advisories

<https://www.icann.org/resources/pages/advisories-2012-02-25-en>

Advisory: Clarifications to the Registry Agreement, and the 2013 Registrar Accreditation Agreement (RAA) regarding applicable Registration Data Directory Service (Whois) Specifications (a.k.a. RDDS clarification Advisory)

<https://www.icann.org/resources/pages/registry-agreement-raa-rdds-2015-04-27-en>

Advisory: Registrar Implementation of the 2013 RAA's Whois Requirements

<https://www.icann.org/news/announcement-2013-07-31-en>

ICANN Consensus Policies

<https://www.icann.org/resources/pages/registrars/consensus-policies-en>

Additional Whois Information Policy

<https://www.icann.org/resources/pages/policy-awip-2014-07-02-en>

EPP Status Code (ICANN)

<https://www.icann.org/epp>

Draft Final Report from the Expert Working Group on Internationalized Registration Data

<https://gnso.icann.org/en/issues/ird/ird-draft-final-10mar15-en.pdf>

Study to Evaluate Available Solutions for the Submission and Display of Internationalized Contact Data

<https://www.icann.org/en/system/files/files/transform-dnrd-02jun14-en.pdf>

Final Report on the Thick Whois Policy Development Process

<https://gnso.icann.org/en/issues/whois/thick-final-21oct13-en.pdf>

ICANN Whois Marketing Restriction Policy

<https://www.icann.org/resources/pages/registrars/consensus-policies/wmrp-en>

gTLD Applicant Guidebook

<https://newgtlds.icann.org/en/applicants/agb/guidebook-full-04jun12-en.pdf>

RIPE RDAP Implementation (beta)

<https://github.com/RIPE-NCC/whois/wiki/RDAP>

CNNIC RDAP Project Git

<https://github.com/cnnic/rdap>

APNIC RDAP Conformance Git

<https://github.com/APNIC-net/rdap-conformance>

Viagenie RDAP Test Suite

<http://rdap.viagenie.ca>

Mozilla Included CA Certificate List

<https://wiki.mozilla.org/CA:IncludedCAs>

CAB Forum Baseline Requirements

<https://cabforum.org/baseline-requirements-documents>