I. Introduction

The Registration Data Access Protocol (RDAP) provides "RESTful" web services to retrieve registration data from Domain Name registrars/registries and Regional Internet Registries. The RDAP base protocol is defined by IETF STD 95. The global set of RDAP RFCs and Internet Drafts are referred to as the RDAP Specifications. See Appendix A for a listing.

The purpose of this document is to encapsulate the operational requirements for RDAP which, in conjunction with the RDAP Technical Implementation Guide, defines RDAP implementation in an ICANN operating environment. This document neither creates nor modifies existing policy,
rather it maps current policy requirements to the RDAP implementation with flexibility to incorporate future policy changes and the goal of minimal reengineering.

II. Policy Mapping

This document specifies the RDAP Policy requirements from the Registration Data Policy effective (date TBD) which builds upon existing RDDS policy and contractual requirements. For clarity, the Registration Data Policy has precedence in any conflict between this document and the Registration Data Policy. See Appendix C for a listing of source material used to create this RDAP Response Profile.

III. Response Requirements

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

1. General

1.1. These requirements represent the minimum baseline for RDAP query responses. RDAP server operators MAY output additional RDDS fields, RDAP events or RDAP roles without further approval by ICANN.

1.2. A server MUST indicate compliance with this specification by including the literal string “icann_rdap_response_profile_1” in the rdapConformance member for all responses provided by the server.

1.3. RDAP extensions

1.3.1. RDAP extensions to responses, 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.
1.3.2. RDAP extensions MUST NOT add browser executable code (e.g., Javascript) to the response.

1.4. In an entity object with an adr structure, the country name property MUST be empty and the cc property MUST be populated by a value from ISO 3166-1 alpha-2.

1.5. The topmost object in the RDAP response MUST contain an event of eventAction type last update of RDAP database with a value equal to the timestamp when the RDAP database was last updated.

1.6. Contact representation

1.6.1. An RDAP server MUST support the use of jCard [RFC7095] to represent contact information.

1.6.2. An RDAP server MAY use alternate formats to represent contact information if a mechanism that allows the RDAP client to request the alternate format is available. An RDAP server MUST only use the alternate format to represent contact information if the RDAP client requests the alternate format.

1.6.3. If the RDAP client does not request a supported contact representation, jCard MUST be the default representation for contact information.

2. Responses to Domain name RDAP queries

2.1. Domain Name - In response to a non-IDN domain query, the returned RDAP response MUST include a domain object and contain a ldhName member. In response to an IDN domain query, if the queried domain is an A-label, then the returned RDAP response MUST include a domain object and MUST contain an ldhName member and MAY contain a unicodeName member. In response to an IDN domain query, if the queried domain is an U-label, then the returned RDAP response MUST include a domain object and MUST contain an unicodeName member and MAY contain a ldhName member.

2.2. Registry Domain ID - 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.
2.3. Event and Event Actions - This section describes Events and Event Actions related to Domain name responses

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

- 2.3.1.1. Event of `eventAction` type `registration`
- 2.3.1.2. Event of `eventAction` type `expiration`

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

- 2.3.2.1. An event of `eventAction` type `registrar expiration`
- 2.3.2.2. Event of `eventAction` type `last changed` - The event of `eventAction` type `last changed` MUST be omitted from the response if the domain object has not been updated since it was created.
- 2.3.2.3. An event of `eventAction` type `transfer`, with the last date and time that the domain was transferred (for clarity: change of sponsoring registrar). The event of `eventAction` type `transfer` MUST be omitted from the response if the domain object has not been transferred since it was created.

2.4. Registrar (Registrar Entity)

2.4.1. Registrar - The `domain` object in the RDAP response MUST contain an `entity` with the `registrar` role (called registrar entity in this section) and a valid `fn` member MUST be present.

2.4.2. Registrar IANA ID - The `handle` of the `entity` MUST be equal to the IANA Registrar ID.

2.4.3. Registrar IANA ID - The `entity` with the `registrar` role in the RDAP response MUST contain a `publicIDs` member [RFC9083] 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.

2.4.4. Other members MAY be present in the `entity` (as specified in RFC6350, the vCard Format Specification and its corresponding JSON mapping RFC7095).
2.4.5. Abuse Contact (email, phone) - An RDAP server MUST include an *entity* with the *abuse* role within the registrar *entity* which MUST include *tel* and *email* members, and MAY include other members.

2.5. Reseller - The returned *domain* object in the RDAP response MAY contain an *entity* with the *reseller* role, if the domain name was registered through a reseller.

2.6. Domain Status

2.6.1. The top-level domain object in the RDAP response MUST contain at least one *status* member [RFC9083] contained in the IANA RDAP JSON Values registry (https://www.iana.org/assignments/rdap-json-values/rdap-json-values.xhtml) of *status* type.

2.6.2. The *status* member value MUST conform to the Extensible Provisioning Protocol (EPP) and Registration Data Access Protocol (RDAP) Status Mapping [RFC8056].

2.6.3. A domain name RDAP response MUST contain a *notices* member with a *title* “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.

2.7. Contacts

2.7.1. The *domain* object in the RDAP response MAY contain entity instances, each of which includes a role signifying the relationship to the *domain* object.

2.7.2. The RDAP response by a registrar server to a domain query MUST contain an *entity* with the *registrant* role.

2.7.3. Except for the *entity* that includes the *registrar* role, the handle of an *entity* MUST be the ROID of the contact object, <contact:roid>, as defined in RFC5733.

2.7.4. The entity that includes the registrant role MAY include valid members *fn*, *org*, *adr*, *tel*, and *email*. Subject to the redaction requirements below, the following RDDS elements MUST be included in the *fn*, *org*, *adr*, *tel*, and *email* members based on the mappings defined in Appendix D when section 9.1 of the Registration Data Policy calls for the Publication of the RDDS element:
2.7.4.1. Registrant Name
2.7.4.2. Registrant Organization
2.7.4.3. Registrant Street
2.7.4.4. Registrant City
2.7.4.5. Registrant State/Province
2.7.4.6. Registrant Postal Code
2.7.4.7. Registrant Country
2.7.4.8. Registrant Phone
2.7.4.9. Registrant Email

2.7.5. Subject to the redaction requirements below, the entity that includes the registrant role MAY include the following RDDS data elements:
  2.7.5.1. Registrant Phone Ext
  2.7.5.2. Registrant Fax
  2.7.5.3. Registrant Fax Ext

2.7.6. The entity that includes the technical role MAY include valid members fn, tel, and email. Subject to the redaction requirements below, the following RDDS elements MUST be included in the fn, tel, and email members based on the mappings defined in Appendix D when section 9.1 of the Registration Data Policy calls for the Publication of the RDDS element:
  2.7.6.1. Tech Name
  2.7.6.2. Tech Phone
  2.7.6.3. Tech Email

2.7.7. When applying the redaction requirements in the Registration Data Policy, the redacted RDDS elements MUST be indicated using the Redacted Fields in the Registration Data Access Protocol (RDAP) Response (draft-ietf-regext-rdap-redacted, as defined by the Technical Implementation Guide 1.1.3) and using the redacted “name” member values included in Appendix E.

2.7.8. When applying the redaction requirements for an email address in the Registration Data Policy, the Registrar (1) MUST use the Redaction by Replacement Value Method in the Registration Data Access Protocol (RDAP) Response (draft-ietf-regext-rdap-redacted, as defined by the Technical Implementation Guide 1.1.3) for the email property, and (2) MUST comply with one of the following:

  2.7.8.1. The email property MUST contain a syntactically valid email address as defined in RFC5322, and) the entity object MUST NOT contain a contact-uri member.
2.7.8.2. The value of the contact-uri member in the entity object MUST be a syntactically valid HTTP URL as defined in RFC9110, and) the entity object MUST NOT contain an email member.

2.7.9. Notwithstanding the redaction requirements in section 2.7.7 and 2.7.8 where the Registered Name Holder provides its consent to publish data that would otherwise be redacted, Registrar MUST NOT redact such RDDS data elements.

2.8. Name Server(s) - The domain object in the RDAP response MUST contain the name servers of the domain in the nameservers member.

2.8.1. Each nameserver object MUST contain the following member: ldhName.

2.8.2. The following members are Optional: ipAddresses [RFC9083], unicodeName, handle [RFC9083] (ROID of the host object, <host:roid> as defined in RFC5732), and status.

2.8.3. This section 2.8.3 only applies to Registries in which name servers are specified as domain attributes. The nameserver object MUST NOT contain the following members: handle and status, but the nameserver object MUST contain an ipAddresses member listing all IPv4 and IPv6 glue records for the in-domain name server (see RFC 8499).

2.9. DNSSEC - The domain object in the RDAP response MUST contain a secureDNS member [RFC9083] including at least a delegationSigned element. Other elements (e.g. dsData) 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.

2.10. RDDS Inaccuracy - A domain name RDAP response MUST contain a notices member with a title "RDDS Inaccuracy Complaint Form", a description containing the string "URL of the ICANN RDDS Inaccuracy Complaint Form: https://icann.org/wicf" and a links member with the https://icann.org/wicf URL.

2.11. Registrar-only requirements - the following requirements apply to registrars only.

2.11.1. A Registrar RDAP service MUST return an HTTP 404 response to a domain name request when the Registrar is not the Sponsoring Registrar for the domain name.

2.11.2. The domain object handle member 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.
2.11.3. The entity handle member in the RDAP response MUST contain the Repository Object Identifier (ROID of the contact object, &lt;contact:roid&gt;, 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 contact object’s unique identifier within the Registrar.

2.11.4. The eventAction type last changed MUST reflect 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.

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

3. Responses to Registrar Entity RDAP queries

This section only applies to Registries

3.1. Registrar (name, address, phone number, email) - In response to a query for a registrar entity, the returned RDAP response MUST be an entity with registrar role, with a handle and valid elements fn, adr, tel, email.

3.1.1. Registrar (Street, City, Country) - The adr member in the RDAP response for a Registrar query MUST at least contain the following RDDS fields: Street, City, Country.

3.1.2. Registrar (State/Province, Postal Code, Fax Number) - the following fields are optional in the adr member of the RDAP response: State/Province, Postal Code, Fax Number.

3.2. Contacts (Admin, Technical) - The RDAP response SHOULD contain at least two entity objects, with the administrative and technical roles respectively within the entity with the registrar role. The entities with the administrative and technical roles MUST contain valid fn, tel, email members, and MAY contain a handle and a valid adr element.

3.3. Appendix F contains non-normative information to assist in the implementation of this section.
4. Responses to Nameserver RDAP queries

This section only applies to Registries that support the host object model as described in RFC 5731.

4.1. Name Server (Name) - In response to a non-IDN Nameserver query the returned RDAP response MUST include a nameserver object and contain a ldhName member. In response to an IDN Nameserver query, if the queried Nameserver is an A-label, then the returned RDAP response MUST include a nameserver object and MUST contain an ldhName member and MAY contain a unicodeName member. In response to an IDN Nameserver query, if the queried Nameserver is an U-label, then the returned RDAP response MUST include a nameserver object and MUST contain an unicodeName member and MAY contain a ldhName member.

4.2. IP Address(es) - If the name server record includes IP addresses then the nameserver object MUST contain an ipAddresses member listing all IPv4 and IPv6 glue records for the in-domain Nameserver.

4.3. Registrar (Name, IANA ID) - The Registrar RDDS field is Optional; if present in the response, it MUST be represented as an entity with the registrar role. The handle of the entity with the registrar role MUST be equal to the IANA Registrar ID. If the Registrar does not have an IANA ID then the handle of the entity with the registrar role MUST equal "not applicable". If the Registrar has an IANA ID, then the entity with the registrar role in the RDAP response MUST contain a publicIDs member with a type value equal to the IANA Registrar ID. If the Registrar does not have an IANA ID then the RDAP response MUST NOT contain a publicIDs member.
Appendix A: RDAP IETF Standards

STD 95 - RDAP
https://www.rfc-editor.org/refs/ref-std95.txt
https://www.rfc-editor.org/info/std95

RFC8056 – Extensible Provisioning Protocol (EPP) and Registration Data Access Protocol (RDAP) Status Mapping
https://www.ietf.org/info/rfc8056
Describes the mapping of the Extensible Provisioning Protocol (EPP) statuses with the statuses registered for us in the Registration Data Access Protocol (RDAP).

jCard: The JSON Format for vCard
https://www.rfc-editor.org/info/rfc7095

vCard Format Specification
https://www.rfc-editor.org/info/rfc6350
EPP Status Code (ICANN)
https://www.icann.org/epp

Appendix B: Other Technical References

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 RFC9083.

IANA Bootstrap Service Registry for Domain Name Space
https://www.iana.org/assignments/rdap-dns/rdap-dns.xhtml
Draft Final Report from the Expert Working Group on Internationalized Registration Data

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

Mozilla Included CA Certificate List
https://wiki.mozilla.org/CA:IncludedCAs

Redacted Fields in the Registration Data Access Protocol (RDAP) Response
Describes an RDAP extension for explicitly identifying redacted RDAP response fields, using JSONPath as the default expression language.

Appendix C: Policy References

gTLD Base 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

Registry Registration Data Directory Services Consistent Labeling and Display Policy (CL&D),

Temporary Specification for gTLD Registration Data –

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 (RDDS clarification Advisory)

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

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

Final Report on the Thick Whois Policy Development Process

ICANN Whois Marketing Restriction Policy
https://www.icann.org/resources/pages/registrars/consensus-policies/wmrp-en

Registration Data Policy (EPDP Phase 1)
[insert link once finalized]
Appendix D: Data Element Mappings

The tables below show the mapping between the data elements in the registration data policy and the RDAP response element. Including a data element in this appendix does not imply its inclusion in any RDAP response.

## Domain Name Responses:

<table>
<thead>
<tr>
<th>Data Element</th>
<th>RDAP Response Element</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain Name</td>
<td>ldhName</td>
</tr>
<tr>
<td>Registry Domain ID</td>
<td>handle</td>
</tr>
<tr>
<td>Updated Date</td>
<td>events.eventAction “last changed”</td>
</tr>
<tr>
<td>Creation Date</td>
<td>events.eventAction “registration”</td>
</tr>
<tr>
<td>Registry Expiry Date</td>
<td>events.eventAction “expiration”</td>
</tr>
<tr>
<td>Domain Status</td>
<td>status object</td>
</tr>
<tr>
<td>Name Server</td>
<td>nameservers.ldhname</td>
</tr>
<tr>
<td>DNSSEC Elements</td>
<td>secureDNS object</td>
</tr>
<tr>
<td>Last update of RDDS</td>
<td>Events.eventAction “last update of RDAP database”</td>
</tr>
<tr>
<td>Registrar</td>
<td>Entities.role registrar</td>
</tr>
<tr>
<td>Registrar</td>
<td>Entities.roles.registrar</td>
</tr>
<tr>
<td>Registrar IANA ID</td>
<td>publicIDs.identifier</td>
</tr>
<tr>
<td>Registrar Abuse Contact</td>
<td>Entities.role abuse email</td>
</tr>
<tr>
<td>Email</td>
<td></td>
</tr>
<tr>
<td>Table</td>
<td>Value</td>
</tr>
<tr>
<td>-------</td>
<td>-------</td>
</tr>
<tr>
<td>Registrar Abuse Contact Phone</td>
<td>Entities.role abuse phone</td>
</tr>
<tr>
<td>Registrar Registration Expiration Date</td>
<td>events.eventAction “registrar expiration”</td>
</tr>
<tr>
<td>Reseller</td>
<td>Entities.roles reseller</td>
</tr>
<tr>
<td>Registrant</td>
<td>Entities.role registrant</td>
</tr>
<tr>
<td>Registry Registrant ID</td>
<td>Entity.handle</td>
</tr>
<tr>
<td>Registrant Name</td>
<td>jCard “fn”</td>
</tr>
<tr>
<td>Registrant Organization</td>
<td>Org</td>
</tr>
<tr>
<td>Registrant Street</td>
<td>Grouped into adr member while complying with section 1.4 above</td>
</tr>
<tr>
<td>Registrant City</td>
<td></td>
</tr>
<tr>
<td>Registrant State/Province</td>
<td></td>
</tr>
<tr>
<td>Registrant Postal Code</td>
<td></td>
</tr>
<tr>
<td>Registrant Country</td>
<td></td>
</tr>
<tr>
<td>Registrant Phone</td>
<td>Tel type parameter voice</td>
</tr>
<tr>
<td>Registrant Phone Ext</td>
<td>Ext</td>
</tr>
<tr>
<td>Registrant Fax</td>
<td>Tel type parameter Fax</td>
</tr>
<tr>
<td>Registrant Fax Ext</td>
<td>Ext</td>
</tr>
<tr>
<td>Registrant Email</td>
<td>Email</td>
</tr>
<tr>
<td>Technical Contact</td>
<td>Entities.role Technical</td>
</tr>
<tr>
<td>Registry Tech ID</td>
<td>Entity.handle</td>
</tr>
<tr>
<td>Tech Name</td>
<td>jCard “fn”</td>
</tr>
<tr>
<td>Data Element</td>
<td>RDAP Response Element</td>
</tr>
<tr>
<td>-----------------------</td>
<td>----------------------------------------------------</td>
</tr>
<tr>
<td>Name Server(s)</td>
<td>nameserver.ldhName</td>
</tr>
<tr>
<td>Name Server IP Address(es)</td>
<td>nameserver.ipAddresses</td>
</tr>
<tr>
<td>Registrar</td>
<td>Entities.roles registrar</td>
</tr>
<tr>
<td>Last Update of RDDS</td>
<td>events.eventAction “last update of RDAP database”</td>
</tr>
</tbody>
</table>
Appendix E: Redacted Fields in the Registration Data Access Protocol (RDAP) Response “redacted name” JSON Values

Registry Registrations

After the Redacted Fields in the Registration Data Access Protocol (RDAP) Response draft (draft-ietf-regext-rdap-redacted) becomes an RFC, the following is the list of “redacted name” registrations for use in redacting the fields defined in the gTLD RDAP Profile.

Value: Registry Domain ID
Type: redacted name
Description: Redacted domain object class “handle” member. The “removal” redacted “path” member JSONPath for Figure “Unredacted RDAP Lookup Response” of draft-ietf-regext-rdap-redacted is “$.handle”.
Registrant Name: ICANN
Registrant Contact Information: globalsupport@icann.org

Value: Registry Registrant ID
Type: redacted name
Description: Redacted entity object class, with “registrant” role, “handle” member. The “removal” redacted “path” member JSONPath for Figure “Unredacted RDAP Lookup Response” of draft-ietf-regext-rdap-redacted is “$.entities[?(@.roles[0]=='registrant')].handle”.
Registrant Name: ICANN
Registrant Contact Information: globalsupport@icann.org

Value: Registrant Name
Type: redacted name
Description: Redacted entity object class, with “registrant” role, name property. When using jCard, redacting the “vcard” “fn” property. The “emptyValue” redacted “path” member JSONPath for Figure “Unredacted RDAP Lookup Response” of draft-ietf-regext-rdap-redacted is “$.entities[?(@.roles[0]=='registrant')].vcardArray[1][?(@[0]=='fn')][3]”.
Registrant Name: ICANN
Registrant Contact Information: globalsupport@icann.org

Value: Registrant Organization
Type: redacted name
Description: Redacted entity object class, with “registrant” role, organization property. When using jCard, redacting the “vcard” “org” property. The “removal” redacted “path” member JSONPath for Figure “Unredacted RDAP Lookup Response” of draft-ietf-regext-rdap-redacted is “$.entities[?(@.roles[0]=='registrant')].vcardArray[1][?(@[0]=='org')]”.

Registrant Name: ICANN
Registrant Contact Information: globalsupport@icann.org

Value: Registrant Street
Type: redacted name
Description: Redacted entity object class, with “registrant” role, street properties. When using jCard, the “vcard” “adr” street properties (vCard “ADR-component-pobox”, “ADR-component-ext”, and “ADR-component-street”). The “emptyValue” redacted “path” member JSONPath for Figure “Unredacted RDAP Lookup Response” of draft-ietf-regext-rdap-redacted is “$.entities[?(@.roles[0]=='registrant')].vcardArray[1][?(@[0]=='adr')][3]:3].

Registrant Name: ICANN
Registrant Contact Information: globalsupport@icann.org

Value: Registrant City
Type: redacted name
Description: Redacted entity object class, with “registrant” role, city property. When using jCard, the “vcard” “adr” locality property (vCard “ADR-component-locality”). The “emptyValue” redacted “path” member JSONPath for Figure “Unredacted RDAP Lookup Response” of draft-ietf-regext-rdap-redacted is “$.entities[?(@.roles[0]=='registrant')].vcardArray[1][?(@[0]=='adr')][3][3].

Registrant Name: ICANN
Registrant Contact Information: globalsupport@icann.org

Value: Registrant Postal Code
Type: redacted name
Description: Redacted entity object class, with “registrant” role, postal code property. When using jCard, the “vcard” “adr” code property (vCard “ADR-component-code”). The “emptyValue” redacted “path” member JSONPath for Figure “Unredacted RDAP Lookup Response” of draft-ietf-regext-rdap-redacted is “$.entities[?(@.roles[0]=='registrant')].vcardArray[1][?(@[0]=='adr')][3][5].

Registrant Name: ICANN
Registrant Contact Information: globalsupport@icann.org

Value: Registrant Phone
Type: redacted name
Description: Redacted entity object class, with “registrant” role, voice phone property. When using jCard, the “vcard” “tel” property with type “voice”. The “removal” redacted “path” member
Registrant Name: ICANN
Registrant Contact Information: globalsupport@icann.org

Value: Registrant Phone Ext
Type: redacted name
Description: Redacted entity object class, with “registrant” role, voice phone extension property. When using jCard, the “vcard” “tel” property extension value with type “voice”. The “removal” redacted “path” member JSONPath for Figure “Unredacted RDAP Lookup Response” of draft-ietf-regext-rdap-redacted is “$.entities[?(@.roles[0]==’registrant’)].vcardArray[1][?(@[1].type==’voice’)]”.

Registrant Name: ICANN
Registrant Contact Information: globalsupport@icann.org

Value: Registrant Fax
Type: redacted name
Description: Redacted entity object class, with “registrant” role, fax phone property. When using jCard, the “vcard” “tel” property with type “fax”. The “removal” redacted “path” member JSONPath for Figure “Unredacted RDAP Lookup Response” of draft-ietf-regext-rdap-redacted is “$.entities[?(@.roles[0]==’registrant’)].vcardArray[1][?(@[1].type==’fax’)]”.

Registrant Name: ICANN
Registrant Contact Information: globalsupport@icann.org

Value: Registrant Fax Ext
Type: redacted name
Description: Redacted entity object class, with “registrant” role, fax phone extension property. When using jCard, the “vcard” “tel” property with type “fax”. The “removal” redacted “path” member JSONPath for Figure “Unredacted RDAP Lookup Response” of draft-ietf-regext-rdap-redacted is “$.entities[?(@.roles[0]==’registrant’)].vcardArray[1][?(@[1].type==’fax’)]”.

Registrant Name: ICANN
Registrant Contact Information: globalsupport@icann.org

Value: Registrant Email
Type: redacted name
Description: Redacted entity object class, with “registrant” role, email property. When using jCard, the “vcard” “email” property. The “removal” redacted “path” member JSONPath for Figure “Unredacted RDAP Lookup Response” of draft-ietf-regext-rdap-redacted is “$.entities[?(@.roles[0]==’registrant’)].vcardArray[1][?(@[0]==’email’)]”. The “replacementValue” redacted “path” member JSONPath for Figure “Unredacted RDAP Lookup Response” of draft-ietf-regext-rdap-redacted is
 Registrant Name:  ICANN
Registrant Contact Information: globalsupport@icann.org

Value:  Registry Tech ID
Type:  redacted name
Description:  Redacted entity object class, with “technical” role, “handle” member. The "removal" redacted “path” member JSONPath for Figure “Unredacted RDAP Lookup Response” of draft-ietf-regext-rdap-redacted is 

Registrant Name:  ICANN
Registrant Contact Information: globalsupport@icann.org

Value:  Tech Name
Type:  redacted name
Description:  Redacted entity object class, with “technical” role, name property. When using jCard, redacting the “vcard” “fn” property. The “emptyValue” redacted “path” member JSONPath for Figure “Unredacted RDAP Lookup Response” of draft-ietf-regext-rdap-redacted is 

Registrant Name:  ICANN
Registrant Contact Information: globalsupport@icann.org

Value:  Tech Phone
Type:  redacted name
Description:  Redacted entity object class, with “technical” role, voice phone property. When using jCard, the “vcard” “tel” property with type “voice”. The “removal" redacted “path” member JSONPath for Figure “Unredacted RDAP Lookup Response” of draft-ietf-regext-rdap-redacted is  

Registrant Name:  ICANN
Registrant Contact Information: globalsupport@icann.org

Value:  Tech Phone Ext
Type:  redacted name
Description:  Redacted entity object class, with “technical” role, voice phone extension property. When using jCard, the “vcard” “tel” property extension value with type “voice”. The “removal” redacted “path” member JSONPath for Figure “Unredacted RDAP Lookup Response” of draft-ietf-regext-rdap-redacted is  

Registrant Name:  ICANN
Registrant Contact Information: globalsupport@icann.org

Value:  Tech Email
Type: redacted name
Description: Redacted entity object class, with “technical” role, email property. When using jCard, the “vcard” “email” property. The “removal” redacted “path” member JSONPath for Figure “Unredacted RDAP Lookup Response” of draft-ietf-regext-rdap-redacted is “$.entities[?(@.roles[0]=='technical')].vcardArray[1][?(@[0]=='email')]. The “replacementValue” redacted “path” member JSONPath for Figure “Unredacted RDAP Lookup Response” of draft-ietf-regext-rdap-redacted is “$.entities[?(@.roles[0]=='technical')].vcardArray[1][?(@[0]=='email')][3]” or using the “replacementPath” to the “vcard” “contact-uri” property.
Registrant Name: ICANN
Registrant Contact Information: globalsupport@icann.org]
Appendix F: Data Element Mappings

The tables below show the mapping between data elements and the RDAP response element. Including a data element in this appendix does not imply its inclusion in any RDAP response.

Registrar Responses:

<table>
<thead>
<tr>
<th>Data Element</th>
<th>RDAP Response Element</th>
</tr>
</thead>
<tbody>
<tr>
<td>Registrar</td>
<td>jCard fn</td>
</tr>
<tr>
<td>Registrar Street</td>
<td></td>
</tr>
<tr>
<td>Registrar City</td>
<td></td>
</tr>
<tr>
<td>Registrar State/Province</td>
<td>Grouped into the adr member, while complying with section 1.4 above</td>
</tr>
<tr>
<td>Registrar Postal Code</td>
<td></td>
</tr>
<tr>
<td>Registrar Country</td>
<td></td>
</tr>
<tr>
<td>Registrar Phone</td>
<td>Tel with a type parameter voice</td>
</tr>
<tr>
<td>Registrar Fax</td>
<td>Tel with a type parameter fax</td>
</tr>
<tr>
<td>Registrar Email</td>
<td>email</td>
</tr>
<tr>
<td>Registrar Admin/Tech Contact</td>
<td>Entity.role administrative or technical</td>
</tr>
<tr>
<td>Administrative/Technical Contact</td>
<td>jCard fn</td>
</tr>
<tr>
<td>Contact Phone</td>
<td>Tel with a type parameter voice</td>
</tr>
<tr>
<td>Contact Fax</td>
<td>Tel with a type parameter fax</td>
</tr>
<tr>
<td>Contact Email</td>
<td>email</td>
</tr>
</tbody>
</table>