Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: July 7, 2013

A. Newton
ARIN
S. Hollenbeck
Verisign Labs

January 3, 2013

JSON Responses for the Registration Data Access Protocol (RDAP)
draft-ietf-weirds-json-response-02

Abstract

This document describes JSON data structures representing registration information maintained by Regional Internet Registries (RIRs) and Domain Name Registries (DNRs). These data structures are used to form Registration Data Access Protocol (RDAP) query responses.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on July 7, 2013.

Table of Contents

1. Introduction 4
2. Terminology and Definitions 5
3. Common Data Types 6
4. Use of JSON 8
4.1. Signaling 8
4.2. Naming 8
5. Common Data Structures 10
5.1. RDAP Conformance 10
5.2. Notices 10
5.3. Language Identifier 11
5.4. An Example 12
6. Object Class Links 13
7. The Entity Object Class 14
7.1. The RIR Entity Object Class 15
7.2. The DNR Entity Object Class 16
8. The Nameserver Object Class 19
9. The Domain Object Class 21
9.1. The RIR Domain Object Class 21
9.2. The DNR Domain Object Class 24
10. The IP Network Object Class 29
11. Autonomous System Number Entity Object Class 32
12. Error Response Body 35
13. IANA Considerations 36
14. Internationalization Considerations 37
14.1. Character Encoding 37
14.2. URIs and IRIs 37
14.3. Language Tags 37
14.4. Internationalized Domain Names 37
15. Contributing Authors and Acknowledgements 38
16. References 39
16.1. Normative References 39
16.2. Informative References 40
Appendix A. Suggested Values 41
A.1. Status 41
A.2. Roles 41
A.3. Variant Relations 42
Appendix B. Suggested Data Modeling with the Entity Object Class 43
B.1. Registrants and Contacts 43
B.2. Registrars 44
Appendix C. IDN Query and Response Model 46
Appendix D. Postal Addresses vs Location 47
Appendix E. Motivations for Using JSON 48

Appendix F. Changelog 49
Authors' Addresses 51

1. Introduction

This document describes responses in the JSON [RFC4627] format for the RESTful web queries as defined by UNIFIED-RDAP-QUERY [I-D.ietf-weirds-rdap-query].

The data model for the responses consists of two major categories: responses returned by Regional Internet Registries (RIRs) for registrations data related to IP addresses, reverse DNS names, and Autonomous System numbers; and responses returned by Domain Name Registries (DNRs) for registration data related to forward DNS names. The RIR object classes are a proper subset of the DNR object classes. The current division between RIR and DNR object classes is given to illustrate an expectation of what data may be expected from an RIR vs a DNR. However, implementers should be aware that RIRs are not limited to the data in the RIR object classes (as an example, some RIRs have a notion of "status" for entities as defined in the DNR entity object class and may at some point start publishing that data).

Object classes defined in the document represent a minimal set of what a compliant client/server MUST understand to function correctly, however some deployments may want to include additional object classes to suit individual needs. Anticipating this need for extension, Section 4.2 of this document defines a mechanism for extending the JSON (objects) that are described in this document.

2. Terminology and Definitions

The following list describes terminology and definitions used throughout this document:

DNR: "Domain Name Registry".

member: data found with in an object as defined by JSON [RFC4627].

object: a data structure as defined by JSON [RFC4627].

object class: the definition of members that may be found in JSON objects described in this document.

object instance: an instantiation or specific instance of an object class.

RDAP: "Registration Data Access Protocol".

RIR: "Regional Internet Registry".

3. Common Data Types

JSON [RFC4627] defines the data types of a number, character string, boolean, array, object and null. This section describes the semantics and/or syntax reference for data types used in this document derived from the JSON character string.

'handle': DNRs and RIRs have registry-unique identifiers that may be used to specifically reference an object instance. The semantics of this data type as found in this document is to be a registry-unique reference to the closest enclosing object where the value is found. The data type names 'registryId', 'roid', 'nic-handle', 'registrationNo', etc... are terms often synonymous with this data type. In this document, the term 'handle' is used. The term exposed to users by clients is a presentation issue beyond the scope of this document.

IPv4 addresses: The representation of IPv4 addresses in this document uses the dotted-decimal notation described in [RFC1166]. An example of this textual representation is '192.0.2.0'.

IPv6 addresses: The representation of IPv6 addresses in this document follow the forms outlined in [RFC5952]. An example of this textual representation is '2001:db8::1:0:0:1'.

country codes: Where the identity of a geopolitical nation or country is needed, these identities are represented with the alpha-2 or 2 character country code designation as defined in [ISO.3166.1988]. The alpha-2 representation is used because it is freely available whereas the alpha-3 and numeric-3 standards are not.

domain names: Textual representations of DNS names follow the rules set forth in [RFC4343], specifically the case insensitivity and character escaping rules. Trailing periods are optional for both input and output.

email addresses: Textual representations of email addresses follow the syntax defined in [RFC5322].

dates and times: The syntax for values denoting dates and times is defined in [RFC3339].

URIs: The syntax for values denoting a Uniform Resource Identifier (URI) is defined by [RFC3986].

Many of the object classes defined in this document contain values representing telephone numbers. Servers are encouraged to provide

those telephone numbers in [E164] format, however clients MUST be prepared for telephone numbers that do not adhere to the [E164] standard.

Postal addresses also appear in some of the object classes. This document specifies no standard for postal addresses as many registries would have to undergo severe data cleanup efforts to meet such standards.

4. Use of JSON

4.1. Signaling

Clients may signal their desire for JSON using the "application/json" media type or the more specific media type "application/rdap" as specified in Section 13.

4.2. Naming

Clients processing JSON [RFC4627] responses SHOULD ignore values associated with unrecognized names. Servers MAY insert values signified by names into the JSON responses which are not specified in this document. Insertion of unspecified values into JSON responses SHOULD have names prefixed with a short identifier followed by an underscore followed by a meaningful name. The full JSON name, the prefix plus the underscore plus the meaningful name, SHOULD adhere to the character and name limitations of the prefix registry described in [I-D.ietf-weirds-using-http].

Consider the following JSON response with JSON names. "handle" and "remarks" are JSON names specified in this document.

```
{
    "handle" : "ABC123",
    "remarks" :
    [
      "she sells seas shells",
      "down by the seashore"
    ]
}
```

Figure 1

If The Registry of the Moon desires to express information not found in this specification, it might select "lunarNic" as its identifying prefix and insert, as an example, the name "lunarNic_beforeOneSmallStep" to signify registrations occuring before the first moon landing and the name "lunarNic_harshMistressNotes" containing other descriptive text.

Consider the following JSON response with JSON names, some of which should be ignored by clients without knowledge of their meaning.

```
{
  "handle" : "ABC123",
  "lunarNic_beforeOneSmallStep" : "TRUE THAT!",
  "remarks" :
  [
    "she sells seas shells",
    "down by the seashore"
  ],
  "lunarNic_harshMistressNotes" :
  [
    "In space,",
    "nobody can hear you scream."
  ]
}
```

Figure 2

Insertion of unrecognized names ignored by clients may also be used for future revisions to this specification.

Clients processing JSON responses MUST be prepared for values specified in this document to be absent from a response as no JSON value listed is required to appear in a response. In other words, servers MAY remove values as is needed by the policies of the server operator.

Finally, all JSON names specified in this document are case sensitive. Both servers and clients MUST transmit and process them according to the character casing specified.

5. Common Data Structures

This section defines three common data structures to be used in respones. Each of these datatypes MAY appear within any object class of a response, but the intended purpose is that they will be mostly used in the top-most object class of a response.

5.1. RDAP Conformance

The first data structure is named "rdapConformance" and is simply an array of strings, each providing a hint as to the specifications used in the construction of the response.

An example rdapConformance data structure.

```
"rdapConformance" :
[
  "rdap_level_0"
]
```

Figure 3

The string literal "rdap_level_0" signifies conformance with this specification. When custom JSON values are inserted into responses, conformance to those custom specifications should use a string prefixed with the appropriate identifier from the IANA prefix identifier registry specified in [I-D.ietf-weirds-using-http]. For example, if the fictional Registry of the Moon want to signify that their JSON responses are conformant with their registered extensions, the string used might be "lunarNIC_level_0".

Example rdapConformance structure with custom extensions noted.

```
"rdapConformance" :
[
  "rdap_level_0",
  "lunarNic_level_0"
]
```

Figure 4

5.2. Notices

The second data structure is named "notices" and is an array of objects. Each object contains a "title" string representing the title of the notice object, an array of strings named "description" for the purposes of conveying any descriptive text about the notice, and an optional "links" object as described in Section 6.

An exmaple of the notices data structure.

```
"notices" :
[
  {
    "title" : "Terms of Use",
    "description" :
    [
      "This service is subject to The Registry of the Moons",
      "terms of service."
    ],
    "links" :
    [
      {
        "value" : "http://example.net/entity/XXXX",
        "rel" : "alternate",
        "type" : "text/html",
        "href" : "http://www.example.com/terms_of_use.html"
      }
    ]
  }
]
```

Figure 5

5.3. Language Identifier

The third data structure is a simple JSON name/value of "lang" with a string containing a language identifier as described by [RFC5646].

```
"lang" : "mn-Cyrl-MN"
```

Figure 6

5.4. An Example

This is an example response with both rdapConformance and notices embedded.

```
{
  "rdapConformance" :
  [
    "rdap_level_0"
  ],
  "notices" :
  [
    {
      "title" : "Content Redacted",
      "description" :
      [
        "Without full authorization, content has been redacted.",
        "Sorry, dude!"
      ],
      "links" :
      [
        {
          "value" : "http://example.net/ip/192.0.2.0/24",
          "rel" : "alternate",
          "type" : "text/html",
          "href" : "http://www.example.com/redaction_policy.html"
        }
      ]
    }
  ],
  "lang" : "en",
  "startAddress" : "192.0.2.0",
  "endAddress" : "192.0.2.255",
  "handle" : "XXXX-RIR",
  "ipVersion" : 4,
  "name": "NET-RTR-1",
  "description" : [ "A network used for example documentation" ],
  "parentHandle" : "YYYY-RIR",
  "remarks" :
  [
    "she sells seas shells",
    "down by the seashore"
  ]
}
```

Figure 7

6. Object Class Links

Each object class defined in this document may have links to other resources on the Internet. The relationship of these links is defined by the IANA registry described by [RFC5988].

The following is an example of the link structure of object classes

```
{
  "value" : "http://example.com/context_uri",
  "rel" : "self",
  "href" : "http://example.com/target_uri",
  "hreflang" : [ "en", "ch" ],
  "title" : [ "title1", "title2" ],
  "media" : "screen",
  "type" : "application/json"
}
```

Figure 8

The JSON name/values of "rel", "href", "hreflang", "title", "media", and "type" correspond to values found in Section 5 of [RFC5988]. The "value" JSON value is the context URI as described by [RFC5988]. The "value", "rel", and "href" JSON values MUST be specified. All other JSON values are optional.

Within an object class, these structures are to be in an array named "links".

This is an example of the "links" array as it might be found in an object class.

```
"links" :
[
    {
      "value" : "http://example.com/ip/2001:db8::123",
      "rel" : "self",
      "href" : "http://example.com/ip/2001:db8::123"
    },
    {
      "value" : "http://example.com/ip/2001:db8::123",
      "rel" : "up",
      "href" : "http://example.com/ip/2001:db8::/48"
    }

]
```

7. The Entity Object Class

The entity object class appears throughout this document and is an appropriate response for the /entity/XXXX query defined in UNIFIED-RDAP-QUERY [I-D.ietf-weirds-rdap-query]. This object class represents the information of organizations, corporations, governments, non-profits, clubs, individual persons, and informal groups of people. All of these representations are so similar that it is best to represent them in JSON [RFC4627] with one construct, the entity object class, to aid in the re-use of code by implementers.

Many of the members of the entity object class are repeated in other object classes described later in this document.

7.1. The RIR Entity Object Class

The following is an example of an RIR entity:

```
    {
      "handle" : "XXXX",
      "entityNames": [ "Joe Bob, Inc.", "Bobby Joe Shopping" ],
      "roles" : [ "registrant" ],
      "postalAddress" :
      [
        "123 Maple Ave",
        "Suite 90001",
        "Vancouver",
        "BC",
        "12393"
      ],
      "emails" : [ "joe@bob.com", "bob@joe.com" ],
      "phones" :
      {
        "office" : [ "1-958-555-4321", "1-958-555-4322" ],
        "fax" :    [ "1-958-555-4323" ],
        "mobile" : [ "1-958-555-4324" ]
      },
      "remarks" :
      [
        "she sells seas shells",
        "down by the seashore"
      ],
      "links" :
      [
        {
          "value" : "http://example.com/entity/XXXX",
          "rel" : "self",
          "href" : "http://example.com/entity/XXXX"
        }
      ],
      "registrationDate" : "1990-12-31T23:59:60Z",
      "lastChangedDate" : "1990-12-31T23:59:60Z",
      "lastChangedBy" : "joe@bob.com"
    }
```

This object as the following members.

- handle -- a string representing an registry unique identifier of the entity

- entityNames -- an array of strings, each signifying the name of the entity
- roles -- an array of strings, each signifying the relationship an object would have with its closest containing object.
- postalAddress -- an array of string, each representing a line in a postal address.
- emails -- an array of strings, each containing an email address for the entity
- phones -- an object containing telephone information associated with the entity, with the following members:
  - office -- an array of strings, each being a telephone number
  - fax -- an array of strings, each being a telephone number
  - mobile -- an array of strings, each being a telephone number
- remarks -- an array of strings, each containing comments about the entity
- links -- see Section 6
- registrationDate -- a string containing the date the entity was registered
- lastChangedDate -- a string containing the date of last change made to the entity
- lastChangedBy -- a string containing an identifier of the party responsible for the last change made to the entity registration

7.2. The DNR Entity Object Class

The DNR entity object class is a superset of the RIR entity object class (Section 7.1). It has the following additional members:

- registrationBy -- a string containing an identifier of the party responsible for the registration of the entity
- sponsoredBy -- a string containing an identifier of the party through which the registration was made, such as an IANA approved registrar

- resoldBy -- a string containing an identifier of the party originating the registration of the entity.

- status -- an array of strings indicating the state of the entity

- port43 -- a string containing the fully-qualified host name of the WHOIS [RFC3912] server where the object instance may be found.

The following is an example of a DNR entity:

```
{
  "handle" : "XXXX",
  "entityNames": [ "Joe Bob, Inc.", "Bobby Joe Shopping" ],
  "status" : [ "validated", "locked" ],
  "postalAddress" :
  [
    "123 Maple Ave",
    "Suite 90001",
    "Vancouver",
    "BC",
    "12393"
  ],
  "emails" : [ "joe@bob.com", "bob@joe.com" ],
  "phones" :
  {
    "office" : [ "1-958-555-4321", "1-958-555-4322" ],
    "fax" :    [ "1-958-555-4323" ],
    "mobile" : [ "1-958-555-4324" ]
  },
  "remarks" :
  [
    "she sells seas shells",
    "down by the seashore"
  ],
  "links" :
  [
    {
      "value" : "http://example.com/entity/XXXX",
      "rel" : "self",
      "href" : "http://example.com/entity/XXXX"
    }
  ],
  "port43" : "whois.example.net",
  "registrationDate" : "1990-12-31T23:59:60Z",
  "registrationBy" : "ABC123",
  "lastChangedDate" : "1990-12-31T23:59:60Z",
  "lastChangedBy" : "ABC123",
  "sponsoredBy" : "SponsorXYZ",
  "resoldBy" : "ResellerPDQ"
}
```

8. The Nameserver Object Class

The nameserver object class is used by both RIRs and DNRs. Unlike other object classes used by both registries where the RIR object class is a subset of the DNR object class, a clear delineation is not made with the nameserver object class because some DNRs have the same or a similar registration model as the RIRs. RIRs and some DNRs register or expose nameserver information as an attribute of a domain name, while other DNRs model nameservers as "first class objects".

The nameserver object class accommodates both models and degrees of variation in between.

The following is an example of a nameserver object.

```
{
  "handle" : "XXXX",
  "name" : "ns1.example.com",
  "status" : [ "active" ],
  "ipAddresses" : [ "192.0.2.1", "192.0.2.2" ],
  "remarks" :
  [
    "she sells seas shells",
    "down by the seashore"
  ],
  "links" :
  [
    {
      "value" : "http://example.net/nameserver/xxxx",
      "rel" : "self",
      "href" : "http://example.net/nameserver/xxxx"
    }
  ],
  "port43" : "whois.example.net",
  "registrationDate" : "1990-12-31T23:59:60Z",
  "registrationBy" : "ABC123",
  "lastChangedDate" : "1990-12-31T23:59:60Z",
  "lastChangedBy" : "ABC123",
  "sponsoredBy" : "SponsorXYZ",
  "resoldBy" : "ResellerPDQ"
}
```

Figure 9

Figure 9 is an example of a nameserver object with all values given. Registries using a first-class nameserver data model would embed this

in domain objects as well as allowing references to it with the /nameserver query type (all depending on the registry operators policy). Other registries may pare back the information as needed. Figure 10 is an example of a nameserver object as would be found in RIRs and some DNRs, while Figure 11 is an example of a nameserver object as would be found in other DNRs.

The following is an example of the simplest nameserver object.

```
{
  "name" : "ns1.example.com"
}
```

Figure 10

The following is an example of a simple nameserver object that might be commonly used by DNRs.

```
{
  "name" : "ns1.example.com",
  "ipAddresses" : [ "2001:db8::123", "2001:db8::124" ]
}
```

Figure 11

The nameserver object class has the following members:

- handle -- a string representing an registry unique identifier of the nameserver
- name -- a string containing the DNS name of the nameserver
- ipAddresses -- an array of strings containing IPv4 and/or IPv6 addresses of the nameserver

The members "status", "remarks", "links", "port43", "sponsoredBy", "resoldBy", "registrationBy", "registrationDate", "lastChangedDate", and "lastChangedBy" take the same form of the members of the same name of the entity object (Section 7).

9. The Domain Object Class

The domain object class represents a DNS name and point of delegation. For RIRs these delegation points are in the reverse DNS tree, whereas for DNRs these delegation points are in the forward DNS tree. The RIR domain object class is a subset of the DNR object class.

In both cases, the high level structure of the domain object class consists of information about the domain registration, nameserver information related to the domain name, and entities related to the domain name (e.g. registrant information, contacts, etc...).

The following is an elided example of the domain object showing the high level structure.

```
{
  "handle" : "XXX",
  "name" : "blah.example.com",
  ...
  "nameServers" :
  [
    ...
  ],
  ...
  "entities" :
  [
    ...
  ]
}
```

9.1. The RIR Domain Object Class

The following is an example of a JSON object representing a reverse DNS delegation point or the RIR domain object class.

```
{
  "handle" : "XXXX",
  "name" : "192.in-addr.arpa",
  "nameServers" :
  [
    { "name" : "ns1.rir.net" },
    { "name" : "ns2.rir.net" }
  ],
  "delegationKeys" :
```

```
    [
      {
        "algorithm": 7,
        "digest" : "E68C017BD813B9AE2F4DD28E61AD014F859ED44C",
        "digestType" : 1,
        "keyTag" : 53814
      }
    ],
    "remarks" :
    [
      "she sells seas shells",
      "down by the seashore"
    ],
    "links" :
    [
      {
        "value": "http://example.net/domain/XXXX",
        "rel" : "self",
        "href" : "http://example.net/domain/XXXXX"
      }
    ],
    "registrationDate" : "1990-12-31T23:59:60Z",
    "lastChangedDate" : "1990-12-31T23:59:60Z",
    "lastChangedBy" : "joe@bob.com",
    "entities" :
    [
      {
        "handle" : "XXXX",
        "entityNames": [ "Joe Bob, Inc.", "Bobby Joe Shopping" ],
        "roles" : [ "registrant" ],
        "postalAddress" :
        [
          "123 Maple Ave",
          "Suite 90001",
          "Vancouver",
          "BC",
          "12393"
          ],
        "emails" : [ "joe@bob.com", "bob@joe.com" ],
        "phones" :
        {
          "office" : [ "1-958-555-4321", "1-958-555-4322" ],
          "fax" :    [ "1-958-555-4323" ],
          "mobile" : [ "1-958-555-4324" ]
        },
        "remarks" :
        [
          "she sells seas shells",
```

```
          "down by the seashore"
        ],
        "links" :
        [
          {
            "value": "http://example.net/entity/xxxx",
            "rel" : "self",
            "href" : "http://example.net/entity/xxxx"
          }
        ],
        "registrationDate" : "1990-12-31T23:59:60Z",
        "lastChangedDate" : "1990-12-31T23:59:60Z",
        "lastChangedBy" : "joe@bob.com"
      }
    ]
}
```

The following is a description of the members of this object:

- handle -- a string representing a registry unique identifier of the domain object instance

- name -- a string denoting the DNS zone name, which is a domain name

- nameservers -- an array of nameserver objects as defined by Section 8

- delegationKeys -- an array of objects, each with the following members:

  * algorithm -- an integer as specified by the algorithm field of a DNS DS record as specified by RFC 4034 [RFC4034] in presentation format

  * digest -- an string as specified by the digest field of a DNS DS record as specified by RFC 4034 in presentation format

  * digestType -- an integer as specified by the digest type field of a DNS DS record as specified by RFC 4034 in presentation format

  * keyTag -- an integer as specified by the key tag field of a DNS DS record as specified by RFC 4034 in presentation format

- entities -- an array of entity objects as defined by Section 7.1.

The members "remarks", "links", "registrationDate", "lastChangedDate", and "lastChangedBy" take the same form of the members of the same name of the entity object (Section 7).

9.2. The DNR Domain Object Class

The DNR domain object class is a superset of the RIR domain object class (Section 9.1) and has the following additional members.

- variants -- an array of objects, each containing the following values:
  - relation -- an array of strings, with each string denoting the relationship between the variants and the containing domain object.
  - variantNames -- an array of strings, each being a variant domain of the containing domain object.
- expirationDate -- a string containing the date and time this domain name registration will expire
- registrationBy -- a string containing an identifier of the party responsible for the registration of the domain name
- sponsoredBy -- a string containing an identifier of the party through which the registration was made, such as an IANA approved registrar
- resoldBy -- a string containing an identifier of the party originating the registration of the domain name
- status -- an array of strings indicating the state of the domain name
- transferDate -- a string containing the date and time this domain name was transferred
- port43 -- a string containing the fully-qualified host name of the WHOIS [RFC3912] server where the object instance may be found.

The following is an example of a JSON object representing a forward DNS delegation point or the DNR domain object class.

```
{
```

```
     "handle" : "XXXX",
     "name" : "blah.example.com",
     "variants" :
     [
       {
         "relation" : [ "registered", "conjoined" ],
         "variantNames" : [ "blah2.example.com", "blah3.example.com" ]
       },
       {
         "relation" : [ "unregistered", "restrictedRegistration" ],
         "variantNames" : [ "blah3.example.com", "blah4.example.com" ]
       }
     ],
     "status" : [ "locked", "transferProhibited" ],
     "nameServers" :
     [
       {
         "handle" : "XXXX",
         "name" : "ns1.example.com",
         "status" : [ "active" ],
         "ipAddresses" :
         [
           "2001:db8::123", "2001:db8::124",
           "192.0.2.1", "192.0.2.2"
         ],
         "remarks" :
         [
           "she sells seas shells",
           "down by the seashore"
         ],
         "links" :
         [
           {
             "value" : "http://example.net/nameserver/XXXX".
             "rel" : "self",
             "href" : "http://example.net/nameserver/XXXX"
           }
         ],
         "registrationDate" : "1990-12-31T23:59:60Z",
         "registrationBy" : "ABC123",
         "lastChangedDate" : "1990-12-31T23:59:60Z",
         "lastChangedBy" : "ABC123",
         "sponsoredBy" : "SponsorXYZ",
         "resoldBy" : "ResellerPDQ"
       },
       {
         "handle" : "XXXX",
         "name" : "ns2.example.com",
```

```
        "status" : [ "active" ],
        "ipAddresses" :
        [
          "2001:db8::125", "2001:db8::126",
          "192.0.2.3", "192.0.2.4"
        ],
        "remarks" :
        [
          "she sells seas shells",
          "down by the seashore"
        ],
        "links" :
        [
          {
            "value" : "http://example.net/nameserver/XXXX",
            "rel" : "self",
            "href" : "http://example.net/nameserver/XXXX"
          }
        ],
        "registrationDate" : "1990-12-31T23:59:60Z",
        "registrationBy" : "ABC123",
        "lastChangedDate" : "1990-12-31T23:59:60Z",
        "lastChangedBy" : "ABC123",
        "sponsoredBy" : "SponsorXYZ",
        "resoldBy" : "ResellerPDQ"
      }
    ],
    "delegationKeys" :
    [
      {
        "algorithm": 7,
        "digest" : "E68C017BD813B9AE2F4DD28E61AD014F859ED44C",
        "digestType" : 1,
        "keyTag" : 53814
      }
    ],
    "remarks" :
    [
      "she sells seas shells",
      "down by the seashore"
    ],
    "links" :
    [
      {
        "value": "http://example.net/domain/XXXX",
        "rel" : "self",
        "href" : "http://example.net/domain/XXXX"
      }
```

```
    ],
    "port43" : "whois.example.net",
    "registrationDate" : "1990-12-31T23:59:60Z",
    "registrationBy" : "ABC123",
    "lastChangedDate" : "1990-12-31T23:59:60Z",
    "lastChangedBy" : "ABC123",
    "sponsoredBy" : "SponsorXYZ",
    "resoldBy" : "ResellerPDQ",
    "expirationDate" : "2016-12-31T23:59:60Z",
    "transferDate" : "1990-12-31T23:59:60Z",
    "entities" :
    [
      {
        "handle" : "XXXX",
        "entityNames": [ "Joe Bob, Inc.", "Bobby Joe Shopping" ],
        "status" : [ "validated", "locked" ],
        "postalAddress" :
        [
          "123 Maple Ave",
          "Suite 90001",
          "Vancouver",
          "BC",
          "12393"
        ],
        "emails" : [ "joe@bob.com", "bob@joe.com" ],
        "phones" :
        {
          "office" : [ "1-958-555-4321", "1-958-555-4322" ],
          "fax" :    [ "1-958-555-4323" ],
          "mobile" : [ "1-958-555-4324" ]
        },
        "remarks" :
        [
          "she sells seas shells",
          "down by the seashore"
        ],
        "links" :
        [
          {
            "value" : "http://example.net/entity/xxxx",
            "rel" : "self",
            "href" : "http://example.net/entity/xxxx"
          }
        ],
        "registrationDate" : "1990-12-31T23:59:60Z",
        "registrationBy" : "ABC123",
        "lastChangedDate" : "1990-12-31T23:59:60Z",
        "lastChangedBy" : "ABC123",
```

```
        "sponsoredBy" : "SponsorXYZ",
        "resoldBy" : "ResellerPDQ"
      }
    ]
  }
```

10. The IP Network Object Class

The IP Network object class models IP network registrations found in RIRs and is the expected response for the /ip query as defined by [I-D.ietf-weirds-rdap-query]. There is no equivalent object class for DNRs. The high level structure of the IP network object class consists of information about the network registration and entities related to the IP network (e.g. registrant information, contacts, etc...).

The following is an elided example of the IP network object type showing the high level structure.

```
{
  "handle" : "XXX",
  ...
  "entities" :
  [
    ...
  ]
}
```

The following is an example of the JSON object for the network registration information

```
{
  "handle" : "XXXX-RIR",
  "startAddress" : "2001:db8::0",
  "endAddress" : "2001:db8::0:FFFF:FFFF:FFFF:FFFF:FFFF",
  "ipVersion" : 6,
  "name": "NET-RTR-1",
  "description" : [ "A network used for routing" ],
  "type" : "DIRECT ALLOCATION",
  "country" : "AU",
  "parentHandle" : "YYYY-RIR",
  "status" : [ "allocated" ],
  "remarks" :
  [
    "she sells seas shells",
    "down by the seashore"
  ],
  "links" :
  [
    {
      "value" : "http://example.ent/ip/2001:db8::/48",
```

```
        "rel" : "self",
        "href" : "http://example.net/ip/2001:db8::/48"
      },
      {
        "value" : "http://example.net/ip/2001:db8::/48",
        "rel" : "up",
        "href" : "http://example.net/ip/2001:C00::/23"
      }
    ],
    "registrationDate" : "20110509",
    "lastChangedDate" : "20110509",
    "lastChangedBy" : "joe@bob.com",
    "entities" :
    [
      {
        "handle" : "XXXX",
        "entityNames": [ "Joe Bob, Inc.", "Bobby Joe Shopping" ],
        "roles" : [ "registrant" ],
        "postalAddress" :
        [
          "123 Maple Ave",
          "Suite 90001",
          "Vancouver",
          "BC",
          "12393"
        ],
        "emails" : [ "joe@bob.com", "bob@joe.com" ],
        "phones" :
        {
          "office" : [ "1-958-555-4321", "1-958-555-4322" ],
          "fax" :    [ "1-958-555-4323" ],
          "mobile" : [ "1-958-555-4324" ]
        },
        "remarks" :
        [
          "she sells seas shells",
          "down by the seashore"
        ],
        "links" :
        [
          {
            "value" : "http://example.net/entity/xxxx",
            "rel" : "self",
            "href" : "http://example.net/entity/xxxx"
          }
        ],
        "registrationDate" : "1990-12-31T23:59:60Z",
        "lastChangedDate" : "1990-12-31T23:59:60Z",
```

```
        "lastChangedBy" : "joe@bob.com"
      }
    ]
  }
```

The following is a description of the members of this object:

- handle -- a string representing an RIR unique identifier of the network registration
- startAddress -- the starting IP address of the network, either IPv4 or IPv6
- endAddress -- the ending IP address of the network, either IPv4 or IPv6
- ipVersion -- an integer signifying the IP protocol version of the network: 4 signifying an IPv4 network, 6 signifying an IPv6 network
- name -- an identifier assigned to the network registration by the registration holder
- description -- an array of strings containing descriptive text about the network registration
- type -- a string containing an RIR specific classification of the network
- country -- a string containing the name of the 2 character country code of the network
- parentHandle -- a string containing an RIR unique identifier of the parent network of this network registration
- status -- an array of strings indicating the state of the IP network
- entities -- an array of entity objects as defined by Section 7.1.

The members "remarks", "links", "registrationDate", "lastChangedDate", and "lastChangedBy" take the same form of the members of the same name of the entity object (Section 7.1).

11. Autonomous System Number Entity Object Class

The Autonomous System Number (autnum) object class models Autonomous System Number registrations found in RIRs and represents the expected response to an /autnum query as defined by [I-D.ietf-weirds-rdap-query]. There is no equivalent object class for DNRs. The high level structure of the autnum object class consists of information about the network registration and entities related to the autnum registration (e.g. registrant information, contacts, etc...), and is similar to the IP Network entity object class.

The following is an example of a JSON object representing an autnum.

```
{
  "handle" : "XXXX-RIR",
  "startAutnum" : "10",
  "endAutnum" : "15",
  "name": "AS-RTR-1",
  "description" : [ "AS for Exchange" ],
  "type" : "DIRECT ALLOCATION",
  "country": "AU",
  "remarks" :
  [
    "she sells seas shells",
    "down by the seashore"
  ],
  "links" :
  [
    {
      "value" : "http://example.net/autnum/xxxx",
      "rel" : "self",
      "href" : "http://example.net/autnum/xxxx"
    }
  ],
  "registrationDate" : "20110509",
  "lastChangedDate" : "20110509",
  "lastChangedBy" : "joe@bob.com",
  "entities" :
  [
    {
      "handle" : "XXXX",
      "entityNames": [ "Joe Bob, Inc.", "Bobby Joe Shopping" ],
      "roles" : [ "registrant" ],
      "postalAddress" :
      [
        "123 Maple Ave",
```

```
          "Suite 90001",
          "Vancouver",
          "BC",
          "12393"
        ],
        "emails" : [ "joe@bob.com", "bob@joe.com" ],
        "phones" :
        {
          "office" : [ "1-958-555-4321", "1-958-555-4322" ],
          "fax" :    [ "1-958-555-4323" ],
          "mobile" : [ "1-958-555-4324" ]
        },
        "remarks" :
        [
          "she sells seas shells",
          "down by the seashore"
        ],
        "links" :
        [
          {
            "value" : "http://example.net/entity/XXXX",
            "rel" : "self",
            "href" : "http://example.net/entity/XXXX"
          }
        ],
        "registrationDate" : "1990-12-31T23:59:60Z",
        "lastChangedDate" : "1990-12-31T23:59:60Z",
        "lastChangedBy" : "joe@bob.com"
      }
    ]
  }
```

The following is a description of the members of this object:

- handle -- a string representing an RIR unique identifier of the autnum registration

- startAutnum -- the starting number [RFC5396] in the block of autonomous system numbers

- endAutnum -- the ending number [RFC5396] in the block of autonomous system numbers

- name -- an identifier assigned to the autnum registration by the registration holder

- description -- an array of strings containing descriptive text about the autnum registration

- type -- a string containing an RIR specific classification of the autnum

- country -- a string containing the name of the 2 character country code of the autnum

The members "remarks", "links", "registrationDate", "lastChangedDate", and "lastChangedBy" take the same form of the members of the same name of the entity object (Section 7.1).

12. Error Response Body

Some non-answer responses may return entity bodies with information that could be more descriptive.

The basic structure of that response is an object class containing an error code number (corresponding to the HTTP response code) followed by a string named "title" followed by an array of strings named "description".

This is an example of the JSON version of the common response body.

```
{
  "errorCode": 418,
  "title": "Your beverage choice is not available",
  "description":
  [
    "I know coffee has more ummppphhh.",
    "But I cannot provide."
  ]
}
```

Figure 12

A client MAY simply use the HTTP response code as the server is not required to include error data in the response body. However, if a client wishes to parse the error data, it SHOULD first check that the Content-Type header contains the appropriate media type.

13. IANA Considerations

This specification registers the "application/rdap" media type.

Type name: application

Subtype name: rdap

Required parameters: n/a

Encoding considerations: n/a

Security considerations: n/a

Interoperability considerations: n/a

Published specification: [[this document]]

Applications that use this media type: RDAP

Additional information: n/a

Person & email address to contact for further information: Andy Newton &andy@hxr.us&

Intended usage: COMMON

Restrictions on usage: none

Author: Andy Newton

Change controller: IETF

14. Internationalization Considerations

14.1. Character Encoding

The default text encoding for JSON and XML responses in RDAP is UTF-8, and all servers and clients MUST support UTF-8. Servers and clients MAY optionally support other character encodings.

14.2. URIs and IRIs

[I-D.ietf-weirds-using-http] defines the use of URIs and IRIs in RDAP.

14.3. Language Tags

Section 5.3 defines the use of language tags in the JSON responses defined in this document.

14.4. Internationalized Domain Names

Appendix C illustrates the model for query and response regarding internationalized domain names (IDNs).

15. Contributing Authors and Acknowledgements

This document is derived from original work on RIR response in JSON 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 word on DNR responses in JSON by Ning Kong, Linlin Zhou, Jiagui Xie, and Sean Shen of CNNIC.

The components of the DNR object classes are derived from a categorization of WHOIS response formats created by Ning Kong, Linlin Zhou, and Guangqing Deng of CNNIC, Steve Sheng and Francisco Arias of ICANN, Ray Bellis of Nominet, and Frederico Neves of NIC.BR.

Ed Lewis of Neustar contributed significant review comments and provided clarifying text.

16. References

16.1. Normative References

[RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.

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

[RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

[RFC3339] Klyne, G., Ed. and C. Newman, "Date and Time on the Internet: Timestamps", RFC 3339, July 2002.

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

[RFC4034] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", RFC 4034, March 2005.

[RFC4343] Eastlake, D., "Domain Name System (DNS) Case Insensitivity Clarification", RFC 4343, January 2006.

[RFC4627] Crockford, D., "The application/json Media Type for JavaScript Object Notation (JSON)", RFC 4627, July 2006.

[RFC5322] Resnick, P., Ed., "Internet Message Format", RFC 5322, October 2008.

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

[RFC5646] Phillips, A. and M. Davis, "Tags for Identifying Languages", BCP 47, RFC 5646, September 2009.

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

[RFC5988] Nottingham, M., "Web Linking", RFC 5988, October 2010.

[ISO.3166.1988] International Organization for Standardization, "Codes for the representation of names of countries, 3rd edition",

ISO Standard 3166, August 1988.

[I-D.ietf-weirds-rdap-query]
Newton, A. and S. Hollenbeck, "RDAP Query Format", draft-ietf-weirds-rdap-query-00 (work in progress), September 2011.

[I-D.ietf-weirds-using-http]
Newton, A., Ellacott, B., and N. Kong, "Using HTTP for RESTful Whois Services by Internet Registries", draft-ietf-weirds-using-http-01 (work in progress), May 2012.

[E164] ITU-T, "The International Public Telecommunication Number Plan", Recommendation E.164, May 1997.

16.2. Informative References

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

[RFC3730] Hollenbeck, S., "Extensible Provisioning Protocol (EPP)", RFC 3730, March 2004.

[JSON_acendancy]
MacVittie, "The Stealthy Ascendancy of JSON", 04 2011.

[JSON_performance_study]
Montana State University - Bozeman, Montana State University - Bozeman, Montana State University - Bozeman, and Montana State University - Bozeman, "Comparison of JSON and XML Data Interchange Formats: A Case Study", 2009.

Appendix A. Suggested Values

Due to the wide variation between the hundreds of registry operators and the on-going policy refinement by registry communities, values of some data cannot be formally standardized. This section lists suggested values for such data but is not nor will ever be a complete list of values and their meanings.

A.1. Status

Many of the object classes have a member named 'status'. This member is an array of strings, with each string denoting a status associated with the containing object. The following is a list of suggested values to use in the 'status' array:

- 'validated' -- Signifies that the data of the object instance has been found to be accurate. This type of status is usually found on entity object instances to note the validity of identifying contact information.

- 'update prohibited' -- Updates to the object instance are forbidden.

- 'transfer prohibited' -- Transfers of the registration from one registrar to another are forbidden. This type of status normally applies to DNR domain names.

- 'delete prohibited' -- Deletion of the registration of the object instance is forbidden. This type of status normally applies to DNR domain names.

A.2. Roles

Entity object classes have a member named 'roles'. This member is an array of strings, with each string indicating the role or relationship the entity object instance has with a containing object, such as a domain name or IP network. An entity object instance can have more than one type of relationship with a containing object. The following is a list of suggested values to use in the 'roles' array:

- 'registrant' -- The entity object instance is the registrant of the registration.

- 'tech' -- The entity object instance is a technical contact for the registration.

- 'admin' -- The entity object instance is an administrative contact for the registration.

- 'abuse' -- The entity object instance handles network abuse issues on behalf of the registrant of the registration.

- 'billing' -- The entity object instance handles payment and billing issues on behalf of the registrant of the registration.

- 'registrar' -- The entity object instance represents the authority responsible for the registration in the registry.

A.3. Variant Relations

Section 9.2 describes a structure for noting variants of domain names and the relationship those variants have with a registered domain name. The following is a list of suggested values to use as the variant relation values:

- 'registered' -- the variant names are registered in the registry.

- 'unregistered' -- the variant names are not found in the registry.

- 'restrictedRegistration' -- registration of the variant names is restricted to certain parties or within certain rules.

- 'openRegistration' -- registration of the variant names is available to generally qualified registrants.

- 'conjoined' -- registration of the variant names is conjoined with the registration of the containing domain registration.

Appendix B. Suggested Data Modeling with the Entity Object Class

B.1. Registrants and Contacts

This document does not provide specific object classes for registrants and contacts. Instead the entity object class may be used to represent a registrant or contact. When the entity object is embedded inside a containing object such as a domain name or IP network, the 'roles' string array can be used to signify the relationship. It is recommended that the values from Appendix A.2 be used.

The following is an example of an elided containing object with an embedded entity that is both a registrant and admin contact:

```
{
  ...
  "entities" :
  [
    {
      "handle" : "XXXX",
      "entityNames": [ "Joe Bob, Inc.", "Bobby Joe Shopping" ],
      "roles" : [ "registrant", "admin" ],
      "postalAddress" :
      [
        "123 Maple Ave",
        "Suite 90001",
        "Vancouver",
        "BC",
        "12393"
      ],
      "emails" : [ "joe@bob.com", "bob@joe.com" ],
      "phones" :
      {
        "office" : [ "1-958-555-4321", "1-958-555-4322" ],
        "fax" :    [ "1-958-555-4323" ],
        "mobile" : [ "1-958-555-4324" ]
      },
      "remarks" :
      [
        "she sells seas shells",
        "down by the seashore"
      ],
      "registrationDate" : "1990-12-31T23:59:60Z",
      "lastChangedDate" : "1990-12-31T23:59:60Z",
      "lastChangedBy" : "joe@bob.com"
    }
```

```
    ]
  }
```

B.2. Registrars

This document does not provide a specific object class for registrars, but like registrants and contacts (see Appendix B.1) the 'roles' string array maybe used.

The following is an example of an elided containing object with an embedded entity that is a registrar:

```
{
  ...
  "entities" :
  [
    {
      "handle" : "XXXX",
      "names": [ "RegistrarsRUS" ],
      "roles" : [ "registrar" ],
      "postalAddress" :
      [
        "1212 Tulip Ave",
        "Suite 1",
        "Marina Del Rey",
        "CA",
        "12393-2193"
      ],
      "emails" : [ "joe@bob.com", "bob@joe.com" ],
      "phones" :
      {
        "office" : [ "1-958-555-4321", "1-958-555-4322" ],
        "fax" :    [ "1-958-555-4323" ],
        "mobile" : [ "1-958-555-4324" ]
      },
      "remarks" :
      [
        "we registrar for less!"
      ],
      "links" :
      [
        {
          "value" : "http://example.net/entity/XXXX",
          "rel" : "alternate",
          "type" : "text/html",
          "href" : "http://www.example.com"
        }
      ]
    }
  ]
}
```

Appendix C. IDN Query and Response Model

Internationalized Domain Names (IDNs) differ from other types of domain names because multiple domain names as would be represented by a name in Master File format (see [RFC4343]) may be registered by a single IDN. IDNs are based on Unicode, and Unicode can have multiple means for encoding the same word depending on the character set and language being used. And the rules for determining which IDN encoding maps to a "wire-format" domain name vary from DNR to DNR.

When an IDN maps to multiple domain names, the various mappings are called variants. The DNR Domain object class (Section 9.2) represents the variants using a string array.

The following is an example of an elided DNR domain object with variants.

```
{
  "handle" : "XXXX",
  "name" : "blah.example.com",
  "variants" : [ "blah2.example.com", "blah3.example.com" ],
  ...
}
```

Because IDNs can have multiple targets in a mapping and due to the variance in DNR mapping rules, it is up to the client to reduce an IDN to a domain name in Master File format so as to narrow the lookup of the domain name to the proper subset. A query of a DNR using the IDN itself might map across multiple registrations depending on the mapping rules of the DNR.

Appendix D. Postal Addresses vs Location

The postal address data listed in the entity object class (Section 7) does not necessarily represent location. The intent of this information is to provide a means to send postal mail to an entity. While in some cases it may also be the location of the entity, there is no guarantee that the two are the same.

Additionally, the postal address data represented in this document does not follow any specific standard for postal addresses because many registries do not keep postal address data in an internationalized standard form. Publication of such data in a format that suggests an internationalized standard form when such data is not known to be well-formed for that purpose would be misleading.

Appendix E. Motivations for Using JSON

This section addresses a common question regarding the use of JSON over other data formats, most notably XML.

It is often pointed out that many DNRs and one RIR support the EPP [RFC3730] standard, which is an XML serialized protocol. The logic is that since EPP is a common protocol in the industry it follows that XML would be a more natural choice. While EPP does enfluence this specification quite a bit, EPP serves a different purpose which is the provisioning of Internet resources between registries and accredited registrars and serves a much narrower audience than that envisioned for RDAP.

By contrast, RDAP has a broader audience and is designed for public consumption of data. Experience from RIRs with first generation RESTful web services for Whois indicate a large percentage of clients operate within browsers and other platforms where full-blown XML stacks are not readily available and where JSON is a better fit.

Additionally, while EPP is used in much of the DNR community it is not a unversial constant in that industry. And finally, EPP's use of XML predates the specification of JSON. If EPP had been defined today, it may very well have used JSON instead of XML.

Beyond the specific DNR and RIR communities, the trend in the broader Internet industry is also switching to JSON over XML, especially in the area of RESTful web services (see [JSON_acendancy]). Studies have also found that JSON is generally less bulky and consequently faster to parse (see [JSON_performance_study]).

Appendix F. Changelog

Initial -00 Adopted as working group document 2012-September-18.

-01

Minor spelling corrections. Changed "Registry Data" to "Registration Data" for the sake of consistency.

Transitioned to RFC 5988 links and relationshipt types from our own custom "uris" structure.

Some examples had 'status' as a string. Those have been corrected as 'status' is always an array of strings.

Domain variants can now have a multi-valued relationship with domain registrations.

"names" in the entity object class was changed to "entityNames".

Some IP address examples change to IPv6.

Change phone number examples and added reference to E.164.

Added section on motivations for using JSON.

Added error response body section.

Added JSON naming section.

Added common data structures section.

Added the IANA Considerations section and the media type registration.

Added 'lang' name/value.

Added internationalization considerations section.

-02

Removed level from media type registration.

Textual changes as given by Ed Lewis.

Fixed objectclass linking example noted by Francisco Obispo

Fixed a lot of other examples called out by Alex Sergeyev

Added a note that JSON names are case sensitive

Added 'status' to IP networks as suggested by Alex Sergeyev

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/