Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 13, 2013

F. Arias
G. Lozano
ICANN
S. Noguchi
JPRS
J. Gould
C. Thippeswamy
VeriSign

March 12, 2013

Domain Name Registration Data (DNRD) Objects Mapping
draft-arias-noguchi-dnrd-objects-mapping-02

Abstract

This document specifies the format, contents and semantics of Domain Name Registration Data (DNRD) Escrow deposits for a Domain Name Registry. It includes the following objects:

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 September 13, 2013.

Copyright Notice

include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction 4
2. Models 4
3. Terminology 5
4. General Conventions 6
4.1. Date and Time 6
4.2. Country names 6
4.3. Telephone numbers 6
4.4. IP addresses 6
5. Object Description 6
5.1. RDE Domain object 6
5.2. RDE Host object 10
5.3. RDE Contact object 12
5.4. RDE Registrar object 16
5.5. RDE IDN Practices 19
5.6. RDE NNDN 20
5.7. RDE EPP Parameters object 21
5.8. RDE Policy object 23
5.9. Header object 23
6. RDE IDN Variants handling 24
7. Profile 24
8. Appendix A. Example of a full deposit using the XML model only 25
9. Appendix B. Example of differential deposit using the XML model only 30
10. Appendix C. Data escrow agent extended verification process . 31
11. Appendix D. Data escrow notifications 32
11.1. Notifications from Registry Operators to Third Parties . . 32
11.2. Notifications from Data Escrow Agents to Third Parties . . 34
11.3. Formal Syntax 35
11.4. Policy Object 58
12. Internationalization Considerations 60
13. IANA Considerations 60
14. Security Considerations 63
15. Acknowledgments 63
16. Change History 64
16.1. Changes from draft-arias-noguchi-registry-data-escrow-02 to -dnrd-objects-mapping-00 64
16.2. Changes from version 00 to 01 64
16.3. Changes from version 01 to 02 65
17. References 65

17.1. Normative References 65
17.2. Informative References 66
Authors' Addresses 66

1. Introduction

This document defines the data escrow structure of the standard set of objects for a Domain Name Registry which include:

- Domain: Internet domain names that are typically provisioned in a Domain Name Registry using the EPP domain name mapping [RFC5731]. The attributes defined in the EPP domain name mapping [RFC5731] are fully supported by this document.

- Host: Internet host names that are typically provisioned in a Domain Name Registry using the EPP host mapping [RFC5732]. The attributes defined in the EPP host mapping [RFC5732] are fully supported by this document.

- Contact: Individual or organization social information provisioned in a Domain Name Registry using the EPP contact mapping [RFC5733]. The attributes defined in the EPP contact mapping [RFC5733] are fully supported by this document.

- Registrar: The organization that sponsors objects like domains, hosts, and contacts in a Domain Name Registry.

- NNDN: A lightweight domain object that is not linked to a Registrar.

This document defines the following pseudo-objects:

- IDN practices: Internationalized Domain Names (IDN) included in the Domain Object Data Escrow include references to the languages rules that define the set of character code points allowed for a specific language.

- EPP parameters: Definition of the specific EPP parameters supported by the Registry Operator.

- Header: Used to specify counters of objects in the SRS database at a certain point in time (watermark).

- Policy: Used to specify OPTIONAL elements from this specification that are REQUIRED based on the business model of the registry.

2. Models

This document defines two different models that can be used to deposit data escrow objects:

- XML: The XML model includes all of the deposit information (meta-data and data) in an XML document. The definition of the XML format is fully defined in the XML schemas.

- CSV: The CSV model uses XML to define the data escrow format of the data contained in referenced Comma-Separated Values (CSV) files.

The data escrow deposit MAY contain a mix of both models but an object MUST be escrowed only in one model.

3. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14, [RFC2119].

REGISTRY. In the context of this draft the definition will be overloaded (from the definition in the base protocol) to indicate an organization providing Registry Services for a REGISTRY-CLASS DOMAIN NAME.

REGISTRY-CLASS DOMAIN NAME (RCDN): Refers to a top-level domain (TLD) or any other domain name at any level in the DNS tree for which a Registry (either directly or through and affiliate company) provides Registry Services for other organizations or individuals. For example: .COM, .ORG, .BIZ, .CO.JP, .B.BR.

REGISTRY SERVICES. Services offered by the Registry critical to the following tasks: the provisioning of domain names on receipt of requests and data from registrars; responding to registrar queries for status information relating to the DNS servers for the RCDN; dissemination of RCDN zone files; operation of the Registry DNS servers; and responding to queries for contact and other information concerning DNS registrations in the RCDN. Any other products or services that only a Registry is capable of providing, by reason of its designation as the Registry. Typical examples of Registry Services are: DNS resolution for the RCDN, WHOIS and EPP.

ALLOCATED. A status of some label with respect to a zone, whereby the label is associated administratively to some entity that has requested the label. This term (and its cognates "allocation" and "to allocate") may represents the first step on the way to delegation in the DNS.

4. General Conventions

4.1. Date and Time

Numerous fields indicate "dates", such as the creation and expiry dates for domain names. These fields SHALL contain timestamps indicating the date and time in UTC as specified in [RFC3339], with no offset from the zero meridian.

4.2. Country names

Country identifiers SHALL be represented using two character identifiers as specified in [ISO-3166-1].

4.3. Telephone numbers

Telephone numbers (both voice and fascimile) SHALL be formatted based on structures defined in [ITU-E164]. Telephone numbers described in this specification are character strings that MUST begin with a plus sign ("+", ASCII value 0x002B), followed by a country code defined in [ITU-E164], followed by a dot (".", ASCII value 0x002E), followed by a sequence of digits representing the telephone number.

4.4. IP addresses

IP addresses syntax MUST conform either to, Internet Protocol [RFC0791], for IPv4 addresses, or IP Version 6 Addressing Architecture [RFC4291], for IPv6 addresses.

5. Object Description

This section describes the base objects supported by this specification:

5.1. RDE Domain object

The RDE domain object is based on the EPP domain name mapping specified in [RFC5731]. There are two elements used in this format related to domains: the domain object per se, used inside the <contents> element and the <rdeDomain:delete> object used inside the <deletes> element.

5.1.1. <domain> object

The domain element is based on the EPP domain <info> response for an authorized client (see Section 3.1.2. of [RFC5731]) with additional data from an EPP <transfer> Query Response, see Section 3.1.3. of

[RFC5731], RGP status from [RFC3915], and data from the EPP <secDns: create> command, see Section 5.2.1. of [RFC5910].

A <domain> element substitutes for the <abstractDomain> abstract element to define a concrete definition of a domain. The <abstractDomain> element can be replaced by other domain definitions using the XML schema substitution groups feature.

The <domain> element contains the following child elements:

- A <name> element that contains the fully qualified name of the domain name object.
- A <roid> element that contains the repository object identifier assigned to the domain name object when it was created.
- An OPTIONAL <uName> element that contains the name of the domain name in Unicode character set. It MUST be provided if available.
- An OPTIONAL <idnTableId> element that references the IDN Table used for the IDN. This corresponds to the "id" attribute of the <idnTableRef> element. This element MUST be present if the domain name is an IDN.
- An OPTIONAL <originalName> element is used to indicate that the domain name is an IDN variant. This element contains the domain name used to generate the IDN variant.
- One or more <status> elements that contain the current status descriptors associated with the domain name.
- Zero or more OPTIONAL <rgpStatus> element to represent "pendingDelete" sub-statuses, including "redemptionPeriod", "pendingRestore", and "pendingDelete", that a domain name can be in as a result of grace period processing as specified in [RFC3915].
- An OPTIONAL <registrant> element that contain the identifier for the human or organizational social information object associated as the holder of the domain name object.
- Zero or more OPTIONAL <contact> elements that contain identifiers for the human or organizational social information objects associated with the domain name object.
- An OPTIONAL <ns> element that contains the fully qualified names of the delegated host objects or host attributes (name servers) associated with the domain name object. See Section 1.1 of

[RFC5731] for a description of the elements used to specify host objects or host attributes.

- A <clID> element that contains the identifier of the sponsoring registrar.

- A <crRr> element that contains the identifier of the registrar that created the domain name object. An OPTIONAL client attribute is used to specify the client that performed the operation.

- An OPTIONAL <crDate> element that contains the date and time of the domain name object creation. This element MUST be present if the domain name has been allocated.

- An OPTIONAL <exDate> element that contains the date and time identifying the end (expiration) of the domain name object's registration period. This element MUST be present if the domain name has been allocated.

- An OPTIONAL <upRr> element that contains the identifier of the registrar that last updated the domain name object. This element MUST NOT be present if the domain has never been modified. An OPTIONAL client attribute is used to specify the client that performed the operation.

- An OPTIONAL <upDate> element that contains the date and time of the most recent domain-name-object modification. This element MUST NOT be present if the domain name object has never been modified.

- An OPTIONAL <secDNS> element that contains the public key information associated with Domain Name System security (DNSSEC) extensions for the domain name as specified in [RFC5910].

- An OPTIONAL <trDate> element that contains the date and time of the most recent domain object successful transfer. This element MUST NOT be present if the domain name object has never been transfered.

- An OPTIONAL <trnData> element that contains the following child elements related to the last transfer request of the domain name object. This element MUST NOT be present if a transfer request for the domain name has never been created.

  * A <trStatus> element that contains the state of the most recent transfer request.

* A <reRr> element that contains the identifier of the registrar that requested the domain name object transfer. An OPTIONAL client attribute is used to specify the client that performed the operation.

* A <reDate> element that contains the date and time that the transfer was requested.

* An <acRr> element that contains the identifier of the registrar that SHOULD act upon a PENDING transfer request. For all other status types, the value identifies the registrar that took the indicated action. An OPTIONAL client attribute is used to specify the client that performed the operation.

* An <acDate> element that contains the date and time of a required or completed response. For a PENDING request, the value identifies the date and time by which a response is required before an automated response action will be taken by the registry. For all other status types, the value identifies the date and time when the request was completed.

* An OPTIONAL <exDate> element that contains the end of the domain name object's validity period (expiry date) if the transfer caused or causes a change in the validity period.

Example of a domain object:

```
...
<rdeDom:domain>
  <rdeDom:name>example1.test</rdeDom:name>
  <rdeDom:roid>Dexample1-TEST</rdeDom:roid>
  <rdeDom:status s="ok"/>
  <rdeDom:registrant>jd1234</rdeDom:registrant>
  <rdeDom:contact type="admin">sh8013</rdeDom:contact>
  <rdeDom:contact type="tech">sh8013</rdeDom:contact>
  <rdeDom:ns>
    <domain:hostObj>ns1.example.com</domain:hostObj>
    <domain:hostObj>ns1.example1.test</domain:hostObj>
  </rdeDom:ns>
  <rdeDom:clID>RegistrarX</rdeDom:clID>
  <rdeDom:crRr client="jdoe">RegistrarX</rdeDom:crRr>
  <rdeDom:crDate>1999-04-03T22:00:00.0Z</rdeDom:crDate>
  <rdeDom:exDate>2015-04-03T22:00:00.0Z</rdeDom:exDate>
</rdeDom:domain>
...
```

5.1.2. <rdeDomain:delete> object

The <rdeDomain:delete> element contains the fully qualified domain name that was deleted and purged.

Example of <rdeDomain:delete> object:

```
...
<rde:deletes>
  ...
  <rdeDomain:delete>
    <rdeDomain:name>foo.test</rdeDomain:name>
    <rdeDomain:name>bar.test</rdeDomain:name>
  </rdeDomain:delete>
  ...
</rde:deletes>
...
```

5.2. RDE Host object

The RDE host object is based on the EPP host name mapping in [RFC5732]. There are two elements used in this format related to hosts: the host object per se, used inside the <contents> element and the <rdeHost:delete> object used inside the <deletes> element.

A <host> element substitutes for the <abstractHost> abstract element to define a concrete definition of a host. The <abstractHost> element can be replaced by other host definitions using the XML schema substitution groups feature.

5.2.1. <host> object

The RDE host object is based on the EPP host <info> response for an authorized client (Section 3.1.2. of [RFC5732]).

The OPTIONAL <host> element contains the following child elements:

- A <name> element that contains the fully qualified name of the host object.
- A <roid> element that contains the repository object identifier assigned to the host object when the object was created.
- One or more <status> elements that describe the status of the host object.

- Zero or more <addr> elements that contain the IP addresses associated with the host object.

- A <clID> element that contains the identifier of the sponsoring registrar.

- A <crRr> element that contains the identifier of the registrar that created the host object. An OPTIONAL client attribute is used to specify the client that performed the operation.

- A <crDate> element that contains the date and time of host-object creation.

- An OPTIONAL <upRr> element that contains the identifier of the registrar that last updated the host object. This element MUST NOT be present if the host object has never been modified. An OPTIONAL client attribute is used to specify the client that performed the operation.

- An OPTIONAL <upDate> element that contains the date and time of the most recent host-object modification. This element MUST NOT be present if the host object has never been modified.

- An OPTIONAL <trDate> element that contains the date and time of the most recent host object successful transfer. This element MUST NOT be present if the domain name object has never been transfered.

Example of <host> object:

```
...
<rdeHost:host>
  <rdeHost:name>ns1.example1.test</rdeHost:name>
  <rdeHost:roid>Hns1_example_test-TEST</rdeHost:roid>
  <rdeHost:status s="ok"/>
  <rdeHost:status s="linked"/>
  <rdeHost:addr ip="v4">192.0.2.2</rdeHost:addr>
  <rdeHost:addr ip="v4">192.0.2.29</rdeHost:addr>
  <rdeHost:addr ip="v6">1080:0:0:0:8:800:200C:417A</rdeHost:addr>
  <rdeHost:clID>RegistrarX</rdeHost:clID>
  <rdeHost:crRr>RegistrarX</rdeHost:crRr>
  <rdeHost:crDate>1999-05-08T12:10:00.0Z</rdeHost:crDate>
  <rdeHost:upRr>RegistrarX</rdeHost:upRr>
  <rdeHost:upDate>2009-10-03T09:34:00.0Z</rdeHost:upDate>
</rdeHost:host>
...
```

5.2.2. <rdeHost:delete> object

The <rdeHost:delete> element contains the fully qualified domain name of a host that was deleted.

Example of <rdeHost:delete> object:

```
...
<rde:deletes>
  ...
  <rdeHost:delete>
    <rdeHost:name>ns1.example.test</rdeHost:name>
  </rdeHost:delete>
  ...
</rde:deletes>
...
```

5.3. RDE Contact object

The RDE contact object is based on the EPP contact name mapping in [RFC5733]. There are two elements used in this format related to contacts: the contact object per se, used inside the <contents> element and the <rdeContact:delete> object used inside the <deletes> element.

A <contact> element substitutes for the <abstractContact> abstract element to define a concrete definition of a contact. The <abstractContact> element can be replaced by other contact definitions using the XML schema substitution groups feature.

5.3.1. <contact> object

The contact object is based on the EPP contact <info> response for an authorized client (Section 3.1.2. of [RFC5733]) with some additions including the data from an EPP <transfer> Query Response, see Section 3.1.3. of [RFC5733].

The OPTIONAL <contact> element contains the following child elements:

- An <id> element that contains the repository object identifier assigned to the contact object when the object was created.
- A <roid> element that contains the repository object identifier assigned to the contact object when it was created.
- One or more <status> elements that describe the status of the contact object.

- One or two <postalInfo> elements that contain postal-address information. Two elements are provided so that address information can be provided in both internationalized and localized forms; a "type" attribute is used to identify the two forms. If an internationalized form (type="int") is provided, element content MUST be represented in a subset of UTF-8 that can be represented in the 7-bit US-ASCII character set. If a localized form (type="loc") is provided, element content MAY be represented in unrestricted UTF-8. The <postalInfo> element contains the following child elements:

  * A <name> element that contains the name of the individual or role represented by the contact.

  * An OPTIONAL <org> element that contains the name of the organization with which the contact is affiliated.

  * An <addr> element that contains address information associated with the contact. An <addr> element contains the following child elements:

    + One, two, or three OPTIONAL <street> elements that contain the contact's street address.

    + A <city> element that contains the contact's city.

    + An OPTIONAL <sp> element that contains the contact's state or province.

    + An OPTIONAL <pc> element that contains the contact's postal code.

    + A <cc> element that contains the contact's two-letter country code.

- An OPTIONAL <voice> element that contains the contact's voice telephone number.

- An OPTIONAL <fax> element that contains the contact's facsimile telephone number.

- An <email> element that contains the contact's email address.

- A <clID> element that contains the identifier of the sponsoring registrar.

- A <crRr> element that contains the identifier of the registrar that created the contact object. An OPTIONAL client attribute is

used to specify the client that performed the operation.

- A <crDate> element that contains the date and time of contact-object creation.

- An OPTIONAL <upRr> element that contains the identifier of the registrar that last updated the contact object. This element MUST NOT be present if the contact has never been modified. An OPTIONAL client attribute is used to specify the client that performed the operation.

- An OPTIONAL <upDate> element that contains the date and time of the most recent contact-object modification. This element MUST NOT be present if the contact object has never been modified.

- An OPTIONAL <trDate> element that contains the date and time of the most recent contact object successful transfer. This element MUST NOT be present if the contact object has never been transferred.

- An OPTIONAL <trnData> element that contains the following child elements related to the last transfer request of the contact object:

  * A <trStatus> element that contains the state of the most recent transfer request.

  * A <reRr> element that contains the identifier of the registrar that requested the domain name object transfer. An OPTIONAL client attribute is used to specify the client that performed the operation.

  * An <acRr> element that contains the identifier of the registrar that SHOULD act upon a PENDING transfer request. For all other status types, the value identifies the registrar that took the indicated action. An OPTIONAL client attribute is used to specify the client that performed the operation.

  * A <reDate> element that contains the date and time that the transfer was requested.

  * An <acDate> element that contains the date and time of a required or completed response. For a PENDING request, the value identifies the date and time by which a response is required before an automated response action will be taken by the registry. For all other status types, the value identifies the date and time when the request was completed.

o An OPTIONAL <disclose> element that identifies elements that requiring exceptional server-operator handling to allow or restrict disclosure to third parties. See Section 2.9 of [RFC5733] for a description of the child elements contained within the <disclose> element.

Example <contact> object:

```
...
<rdeContact:contact>
  <rdeContact:id>sh8013</rdeContact:id>
  <rdeContact:roid>Csh8013-TEST</rdeContact:roid>
  <rdeContact:status s="linked"/>
  <rdeContact:status s="clientDeleteProhibited"/>
  <rdeContact:postalInfo type="int">
    <contact:name>John Doe</contact:name>
    <contact:org>Example Inc.</contact:org>
    <contact:addr>
      <contact:street>123 Example Dr.</contact:street>
      <contact:street>Suite 100</contact:street>
      <contact:city>Dulles</contact:city>
      <contact:sp>VA</contact:sp>
      <contact:pc>20166-6503</contact:pc>
      <contact:cc>US</contact:cc>
    </contact:addr>
  </rdeContact:postalInfo>
  <rdeContact:voice x="1234">+1.7035555555</rdeContact:voice>
  <rdeContact:fax>+1.7035555556</rdeContact:fax>
  <rdeContact:email>jdoe@example.test</rdeContact:email>
  <rdeContact:clID>RegistrarX</rdeContact:clID>
  <rdeContact:crRr client="jdoe">RegistrarX</rdeContact:crRr>
  <rdeContact:crDate>2009-09-13T08:01:00.0Z</rdeContact:crDate>
  <rdeContact:upRr client="jdoe">RegistrarX</rdeContact:upRr>
  <rdeContact:upDate>2009-11-26T09:10:00.0Z</rdeContact:upDate>
  <rdeContact:trDate>2009-12-03T09:05:00.0Z</rdeContact:trDate>
  <rdeContact:trnData>
    <rdeContact:trStatus>pending</rdeContact:trStatus>
    <rdeContact:reRr client="jstiles">clientW</rdeContact:reRr>
    <rdeContact:reDate>2011-03-08T19:38:00.0Z</rdeContact:reDate>
    <rdeContact:acRr client="rmiles">RegistrarX</rdeContact:acRr>
    <rdeContact:acDate>2011-03-13T23:59:59.0Z</rdeContact:acDate>
  </rdeContact:trnData>
  <rdeContact:disclose flag="0">
    <contact:voice/>
    <contact:email/>
  </rdeContact:disclose>
</rdeContact:contact>
```

```
...
```

5.3.2. <rdeContact:delete> object

The <rdeContact:delete> element contains the id of a contact that was deleted.

Example of <rdeContact:delete> object:

```
...
<rde:deletes>
  ...
  <rdeContact:delete>
    <rdeContact:id>sh8013-TEST</rdeContact:id>
    <rdeContact:id>co8013-TEST</rdeContact:id>
  </rdeContact:delete>
  ...
</rde:deletes>
...
```

5.4. RDE Registrar object

The RDE registrar object is the sponsoring client of other RDE objects, for operational purposes MAY be the registry operator. There are two elements used in this format related to registrars: the registrar object per se, used inside the <contents> element and the <rdeRegistrar:delete> object used inside the <deletes> element.

A <registrar> element substitutes for the <abstractRegistrar> abstract element to define a concrete definition of a registrar. The <abstractRegistrar> element can be replaced by other domain definitions using the XML schema substitution groups feature.

5.4.1. <registrar> object

The <registrar> element contains the following child elements:

- An <id> element that contains the Registry-unique identifier of the registrar object. This <id> has a superordinate relationship to a subordinate <clID>, <crRr> or <upRr> of domain, contact and host objects.

- An <name> element that contains the name of the registrar.

- An OPTIONAL <gurid> element that contains the ID assigned by ICANN.

- A <status> element that contains the operational status of the registrar. Possible values are: ok, readonly and terminated.

- One or two <postalInfo> elements that contain postal- address information. Two elements are provided so that address information can be provided in both internationalized and localized forms; a "type" attribute is used to identify the two forms. If an internationalized form (type="int") is provided, element content MUST be represented in a subset of UTF-8 that can be represented in the 7-bit US-ASCII character set. If a localized form (type="loc") is provided, element content MAY be represented in unrestricted UTF-8. The <postalInfo> element contains the following child elements:

  * A <addr> element that contains address information associated with the registrar. The <addr> element contains the following child elements:

    + One, two, or three OPTIONAL <street> elements that contain the registrar's street address.

    + A <city> element that contains the registrar's city.

    + An OPTIONAL <sp> element that contains the registrar's state or province.

    + An OPTIONAL <pc> element that contains the registrar's postal code.

    + A <cc> element that contains the registrar's country code.

- An OPTIONAL <voice> element that contains the registrar's voice telephone number.

- An OPTIONAL <fax> element that contains the registrar's facsimile telephone number.

- An <email> element that contains the registrar's email address.

- An OPTIONAL <url> element that contains the registrar's URL.

- An OPTIONAL <whoisInfo> elements that contains whois information. The <whoisInfo> element contains the following child elements:

  * An OPTIONAL <name> element that contains the name of the registrar WHOIS server listening on TCP port 43 as specified in [RFC3912].

    * An OPTIONAL <url> element that contains the name of the registrar WHOIS server listening on TCP port 80/443.

- A <crDate> element that contains the date and time of registrar-object creation.

- An OPTIONAL <upDate> element that contains the date and time of the most recent RDE registrar-object modification. This element MUST NOT be present if the rdeRegistrar object has never been modified.

Example of <registrar> object:

```
...
<rdeRegistrar:registrar>
  <rdeRegistrar:id>RegistrarX</rdeRegistrar:id>
  <rdeRegistrar:name>Registrar X</rdeRegistrar:name>
  <rdeRegistrar:gurid>123</rdeRegistrar:gurid>
  <rdeRegistrar:status>ok</rdeRegistrar:status>
  <rdeRegistrar:postalInfo type="int">
    <rdeRegistrar:addr>
      <rdeRegistrar:street>123 Example Dr.</rdeRegistrar:street>
      <rdeRegistrar:street>Suite 100</rdeRegistrar:street>
      <rdeRegistrar:city>Dulles</rdeRegistrar:city>
      <rdeRegistrar:sp>VA</rdeRegistrar:sp>
      <rdeRegistrar:pc>20166-6503</rdeRegistrar:pc>
      <rdeRegistrar:cc>US</rdeRegistrar:cc>
    </rdeRegistrar:addr>
  </rdeRegistrar:postalInfo>
  <rdeRegistrar:voice x="1234">+1.7035555555</rdeRegistrar:voice>
  <rdeRegistrar:fax>+1.7035555556</rdeRegistrar:fax>
  <rdeRegistrar:email>jdoe@example.test</rdeRegistrar:email>
  <rdeRegistrar:url>http://www.example.test</rdeRegistrar:url>
  <rdeRegistrar:whoisInfo>
    <rdeRegistrar:name>whois.example.test</rdeRegistrar:name>
    <rdeRegistrar:url>http://whois.example.test</rdeRegistrar:url>
  </rdeRegistrar:whoisInfo>
  <rdeRegistrar:crDate>2005-04-23T11:49:00.0Z</rdeRegistrar:crDate>
  <rdeRegistrar:upDate>2009-02-17T17:51:00.0Z</rdeRegistrar:upDate>
</rdeRegistrar:registrar>
...
```

5.4.2. <rdeRegistrar:delete> object

The <rdeRegistrar:delete> element contains the id of a registrar that was deleted.

Example of <rdeRegistrar:delete> object:

```
...
<rde:deletes>
  ...
  <rdeRegistrar:delete>
    <rdeRegistrar:id>agnt0001-TEST</rdeRegistrar:id>
  </rdeRegistrar:delete>
  ...
</rde:deletes>
...
```

5.5. RDE IDN Practices

The RDE Internationalized Domain Names (IDN) Practices reference is a pseudo-object that is used to provide a short reference to the IDN Table and Policy used in IDN registrations. The <idnTableRef> element has an "id" attribute that is used to uniquely identify an IDN Table stored externally.

5.5.1. <idnTableRef> object

The OPTIONAL <idnTableRef> contains the following elements. An id attribute is used to specify an identifier for the IDN table.

- An <url> element that contains the URL of the IDN table that is being referenced.

- A <urlPolicy> element that contains the URL of the IDN policy document. If IDN variants are generated algorithmically, the policy document MUST define the algorithm and the state of the implicit generated IDN variants. For a list of suggested states for implicit IDN variants, please see [variantTLDsReport].

Example of <idnTableRef> object:

```
...
<rdeIDN:idnTableRef id="pt-BR">
  <rdeIDN:url>
    http://www.iana.org/domains/idn-tables/tables/br_pt-br_1.0.html
  </rdeIDN:url>
  <rdeIDN:urlPolicy>
    http://registro.br/dominio/regras.html
  </rdeIDN:urlPolicy>
</rdeIDN:idnTableRef>
...
```

5.6. RDE NNDN

A NNDN (NNDN's not domain name) does not exist as a domain object; it is stored in the SRS database. NNDNs can optionally be used to store registry reserved names or IDN variant handling (blocked and withheld). A NNDN is a lightweight domain object that is not linked to a Registrar. A FQDN can only exist as a domain name or NNDN, but not both.

A <NNDN> element substitutes for the <abstractNNDN> abstract element to define a concrete definition of a NNDN. The <abstractDomain> element can be replaced by other NNDN definitions using the XML schema substitution groups feature.

5.6.1. <NNDN> object

The OPTIONAL <NNDN> element contains the following child elements:

- An <aName> element that contains the ASCII Compatible Encoding (ACE) of the NNDN.
- An OPTIONAL <uName> element that contains the name of the NNDN in Unicode character set. It MUST be provided if available.
- An OPTIONAL <idnTableId> element that references the IDN Table used for the NNDN. This corresponds to the "id" attribute of the <idnTableRef> element. This element MUST be present if the NNDN is an IDN.
- An OPTIONAL <originalName> element is used to indicate that the NNDN is an IDN variant. This element contains the domain name used to generate the IDN variant.
- A <nameState> element that indicates the state of the NNDN: blocked or withheld.
  - If a NNDN is considered undesirable for registration (i.e., unavailable for allocation to anyone), then the NNDN will be tagged as "blocked".
  - If a NNDN is created to allow the registration of a domain object to a particular registrant then the NNDN will be tagged as "withheld".
- A <crDate> element that contains the date and time of the NNDN object creation.

Example of <NNDN> object:

```
...
<rdeNNDN:NNDN>
  <rdeNNDN:aName>xn--exampl-gva.test</rdeNNDN:aName>
  <rdeNNDN:idnTableId>pt-BR</rdeNNDN:idnTableId>
  <rdeNNDN:originalName>Dexample1-TEST</rdeNNDN:originalName>
  <rdeNNDN:nameState>withheld</rdeNNDN:nameState>
  <rdeNNDN:crDate>2005-04-23T11:49:00.0Z</rdeNNDN:crDate>
</rdeNNDN:NNDN>
...
```

5.6.2. <rdeNNDN:delete> object

The <rdeNNDN:delete> element contains the ACE of a NNDN that was deleted, i.e., the <aName>.

Example of <rdeNDN:delete> object:

```
...
<rde:deletes>
  ...
  <rdeNNDN:delete>
    <rdeNNDN:aName>xn--pingino-q2a.test</rdeNDN:aName>
  </rdeNNDN:delete>
  ...
</rde:deletes>
...
```

5.7. RDE EPP Parameters object

An OPTIONAL <eppParams> element contains some EPP parameters that may be helpful when rebuilding a registry from the escrow deposits. The element SHOULD be included in Deposits if the registry uses EPP.

The syntax and content of the <eppParams> children elements is as explained in section 2.4 of [RFC5730]. The children of the <eppParams> are as follows:

- One or more <version> elements that indicate the EPP versions supported by the registry.
- One or more <lang> elements that indicate the identifiers of the text response languages supported by the registry's EPP server.
- One or more <objURI> elements that contain namespace URIs representing the objects that the registry's EPP server is capable of managing.

- An OPTIONAL <svcExtension> element that contains one or more <extURI> elements that contain namespace URIs representing object extensions supported by the registry's EPP server.

- A <dcp> element that contains child elements used to describe the server's privacy policy for data collection and management. See section 2.4 of [RFC5730] for more details.

Example of <eppParams> element object:

```
...
<rdeEppParams:eppParams>
  <rdeEppParams:version>1.0</rdeEppParams:version>
  <rdeEppParams:lang>en</rdeEppParams:lang>
  <rdeEppParams:objURI>urn:ietf:params:xml:ns:domain-1.0
    </rdeEppParams:objURI>
  <rdeEppParams:objURI>urn:ietf:params:xml:ns:contact-1.0
    </rdeEppParams:objURI>
  <rdeEppParams:objURI>urn:ietf:params:xml:ns:host-1.0
    </rdeEppParams:objURI>
  <rdeEppParams:svcExtension>
    <epp:extURI>urn:ietf:params:xml:ns:rgp-1.0</epp:extURI>
    <epp:extURI>urn:ietf:params:xml:ns:secDNS-1.1</epp:extURI>
  </rdeEppParams:svcExtension>
  <rdeEppParams:dcp>
  <epp:access><epp:all/></epp:access>
    <epp:statement>
      <epp:purpose>
        <epp:admin/>
        <epp:prov/>
      </epp:purpose>
      <epp:recipient>
        <epp:ours/>
        <epp:public/>
      </epp:recipient>
      <epp:retention>
        <epp:stated/>
      </epp:retention>
    </epp:statement>
  </rdeEppParams:dcp>
</rdeEppParams:eppParams>
...
```

5.8. RDE Policy object

The RDE Policy is a pseudo-object that is used to specify which OPTIONAL elements from this specification are REQUIRED based on the business model of the registry.

5.8.1. <policy> object

The OPTIONAL <policy> contains the following attributes:

- An <element> that defines that the referenced <element> is REQUIRED.

Example of <policy> object:

```
...
<rdePolicy:policy element="rdeDom:registrant" />
...
```

5.9. Header object

The RDE Header is a pseudo-object that is used to specify the number of objects in the SRS at a specific point in time (watermark) regardless of the type of deposit: differential, full or incremental.

5.9.1. <header> object

The <header> contains the following attributes:

- A <tld> element that defines TLD being escrowed.

- A <count> element that number of objects being escrowed. An uri attribute is used to define the type of object.

Example of <header> object:

```
...
   <rdeHeader:header>
      <rdeHeader:tld>test</rdeHeader:tld>
      <rdeHeader:count
        uri="urn:ietf:params:xml:ns:rdeDomain-1.0">2</rdeHeader:count>
      <rdeHeader:count
        uri="urn:ietf:params:xml:ns:rdeHost-1.0">1</rdeHeader:count>
      <rdeHeader:count
        uri="urn:ietf:params:xml:ns:rdeContact-1.0">1</rdeHeader:count>
      <rdeHeader:count
        uri="urn:ietf:params:xml:ns:rdeRegistrar-1.0">1
          </rdeHeader:count>
      <rdeHeader:count
        uri="urn:ietf:params:xml:ns:rdeIDN-1.0">1</rdeHeader:count>
      <rdeHeader:count
        uri="urn:ietf:params:xml:ns:rdeNNDN-1.0">1</rdeHeader:count>
      <rdeHeader:count
        uri="urn:ietf:params:xml:ns:rdeEppParams-1.0">1
          </rdeHeader:count>
    </rdeHeader:header>
...
```

6. RDE IDN Variants handling

Depending on the Registration Policy of the Registry; for a particular domain name there may be multiple variant names. See [variantTLDsReport] for further detail on IDN variants.

A registry could choose to escrow IDN variants as domains or NNDN objects.

A NNDN or a domain name are explicit representations of an IDN variant while an IDN variant computed based on an algorithm is an implicit representation. Explicit representation of an IDN variant takes precedence over an implicit representation.

7. Profile

Different business models of registries exist, therefore the registry is responsible to define a profile that matches its particular business model. The profile mechanism allows a registry to extend this specification.

A profile is the process of:

1. Extending base objects with the mechanisms defined for XML and CSV models.

   * In the case of the XML model, abstract elements could be use to extend the following objects: <domain>, <host>, <contact>, <NNDN> and <registrar> using XML schema substitution groups feature.

2. Defining a <policy> object to specify which OPTIONAL elements of this base specification are required based on the business model of the registry. An example is the <registrant> element that is usually REQUIRED but it is specified as OPTIONAL in this specification to accomodate existing business models.

3. Adding new escrowed objects using the <rde:contents> and <rde: deletes> elements.

4. Providing the XML schemas to third parties that require them to validate the escrow deposits.

8. Appendix A. Example of a full deposit using the XML model only

Example of a full deposit using the XML model only:

```
<?xml version="1.0" encoding="UTF-8"?>
<rde:deposit type="FULL" id="20101017001" prevId="20101010001"
  xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
  xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
  xmlns:secDNS="urn:ietf:params:xml:ns:secDNS-1.1"
  xmlns:rde="urn:ietf:params:xml:ns:rde-1.0"
  xmlns:rdeHeader="urn:ietf:params:xml:ns:rdeHeader-1.0"
  xmlns:rdeDom="urn:ietf:params:xml:ns:rdeDomain-1.0"
  xmlns:rdeHost="urn:ietf:params:xml:ns:rdeHost-1.0"
  xmlns:rdeContact="urn:ietf:params:xml:ns:rdeContact-1.0"
  xmlns:rdeRegistrar="urn:ietf:params:xml:ns:rdeRegistrar-1.0"
  xmlns:rdeIDN="urn:ietf:params:xml:ns:rdeIDN-1.0"
  xmlns:rdeNNDN="urn:ietf:params:xml:ns:rdeNNDN-1.0"
  xmlns:rdeEppParams="urn:ietf:params:xml:ns:rdeEppParams-1.0"
  xmlns:rdePolicy="urn:ietf:params:xml:ns:rdePolicy-1.0"
  xmlns:epp="urn:ietf:params:xml:ns:epp-1.0">

  <rde:watermark>2010-10-17T00:00:00Z</rde:watermark>
  <rde:rdeMenu>
    <rde:version>1.0</rde:version>
    <rde:objURI>urn:ietf:params:xml:ns:rdeHeader-1.0</rde:objURI>
    <rde:objURI>urn:ietf:params:xml:ns:rdeContact-1.0</rde:objURI>
```

```
    <rde:objURI>urn:ietf:params:xml:ns:rdeHost-1.0</rde:objURI>
    <rde:objURI>urn:ietf:params:xml:ns:rdeDomain-1.0</rde:objURI>
    <rde:objURI>urn:ietf:params:xml:ns:rdeRegistrar-1.0</rde:objURI>
    <rde:objURI>urn:ietf:params:xml:ns:rdeIDN-1.0</rde:objURI>
    <rde:objURI>urn:ietf:params:xml:ns:rdeNNDN-1.0</rde:objURI>
    <rde:objURI>urn:ietf:params:xml:ns:rdeEppParams-1.0</rde:objURI>
  </rde:rdeMenu>

  <!-- Contents -->
  <rde:contents>
    <!-- Header -->
    <rdeHeader:header>
      <rdeHeader:tld>test</rdeHeader:tld>
      <rdeHeader:count
        uri="urn:ietf:params:xml:ns:rdeDomain-1.0">2</rdeHeader:count>
      <rdeHeader:count
        uri="urn:ietf:params:xml:ns:rdeHost-1.0">1</rdeHeader:count>
      <rdeHeader:count
        uri="urn:ietf:params:xml:ns:rdeContact-1.0">1</rdeHeader:count>
      <rdeHeader:count
        uri="urn:ietf:params:xml:ns:rdeRegistrar-1.0">1
          </rdeHeader:count>
      <rdeHeader:count
        uri="urn:ietf:params:xml:ns:rdeIDN-1.0">1</rdeHeader:count>
      <rdeHeader:count
        uri="urn:ietf:params:xml:ns:rdeNNDN-1.0">1</rdeHeader:count>
      <rdeHeader:count
        uri="urn:ietf:params:xml:ns:rdeEppParams-1.0">1
          </rdeHeader:count>
    </rdeHeader:header>

    <!-- Domian: example1.test -->
    <rdeDom:domain>
      <rdeDom:name>example1.test</rdeDom:name>
      <rdeDom:roid>Dexample1-TEST</rdeDom:roid>
      <rdeDom:status s="ok"/>
      <rdeDom:registrant>jd1234</rdeDom:registrant>
      <rdeDom:contact type="admin">sh8013</rdeDom:contact>
      <rdeDom:contact type="tech">sh8013</rdeDom:contact>
      <rdeDom:ns>
        <domain:hostObj>ns1.example.com</domain:hostObj>
        <domain:hostObj>ns1.example1.test</domain:hostObj>
      </rdeDom:ns>
      <rdeDom:clID>RegistrarX</rdeDom:clID>
      <rdeDom:crRr client="jdoe">RegistrarX</rdeDom:crRr>
      <rdeDom:crDate>1999-04-03T22:00:00.0Z</rdeDom:crDate>
      <rdeDom:exDate>2015-04-03T22:00:00.0Z</rdeDom:exDate>
    </rdeDom:domain>
```

```
<!-- Domian: example2.test -->
<rdeDom:domain>
  <rdeDom:name>example2.test</rdeDom:name>
  <rdeDom:roid>Dexample2-TEST</rdeDom:roid>
  <rdeDom:status s="ok"/>
  <rdeDom:status s="clientUpdateProhibited"/>
  <rdeDom:registrant>jd1234</rdeDom:registrant>
  <rdeDom:contact type="admin">sh8013</rdeDom:contact>
  <rdeDom:contact type="tech">sh8013</rdeDom:contact>
  <rdeDom:clID>RegistrarX</rdeDom:clID>
  <rdeDom:crRr>RegistrarX</rdeDom:crRr>
  <rdeDom:crDate>1999-04-03T22:00:00.0Z</rdeDom:crDate>
  <rdeDom:exDate>2015-04-03T22:00:00.0Z</rdeDom:exDate>
</rdeDom:domain>

<!-- Host: ns1.example.test -->
<rdeHost:host>
  <rdeHost:name>ns1.example1.test</rdeHost:name>
  <rdeHost:roid>Hns1_example_test-TEST</rdeHost:roid>
  <rdeHost:status s="ok"/>
  <rdeHost:status s="linked"/>
  <rdeHost:addr ip="v4">192.0.2.2</rdeHost:addr>
  <rdeHost:addr ip="v4">192.0.2.29</rdeHost:addr>
  <rdeHost:addr ip="v6">1080:0:0:0:8:800:200C:417A</rdeHost:addr>
  <rdeHost:clID>RegistrarX</rdeHost:clID>
  <rdeHost:crRr>RegistrarX</rdeHost:crRr>
  <rdeHost:crDate>1999-05-08T12:10:00.0Z</rdeHost:crDate>
  <rdeHost:upRr>RegistrarX</rdeHost:upRr>
  <rdeHost:upDate>2009-10-03T09:34:00.0Z</rdeHost:upDate>
</rdeHost:host>


<!-- Contact: sh8013 -->
<rdeContact:contact>
  <rdeContact:id>sh8013</rdeContact:id>
  <rdeContact:roid>Csh8013-TEST</rdeContact:roid>
  <rdeContact:status s="linked"/>
  <rdeContact:status s="clientDeleteProhibited"/>
  <rdeContact:postalInfo type="int">
    <contact:name>John Doe</contact:name>
    <contact:org>Example Inc.</contact:org>
    <contact:addr>
      <contact:street>123 Example Dr.</contact:street>
      <contact:street>Suite 100</contact:street>
      <contact:city>Dulles</contact:city>
      <contact:sp>VA</contact:sp>
      <contact:pc>20166-6503</contact:pc>
      <contact:cc>US</contact:cc>
```

```
        </contact:addr>
      </rdeContact:postalInfo>
      <rdeContact:voice x="1234">+1.7035555555</rdeContact:voice>
      <rdeContact:fax>+1.7035555556</rdeContact:fax>
      <rdeContact:email>jdoe@example.test</rdeContact:email>
      <rdeContact:clID>RegistrarX</rdeContact:clID>
      <rdeContact:crRr client="jdoe">RegistrarX</rdeContact:crRr>
      <rdeContact:crDate>2009-09-13T08:01:00.0Z</rdeContact:crDate>
      <rdeContact:upRr client="jdoe">RegistrarX</rdeContact:upRr>
      <rdeContact:upDate>2009-11-26T09:10:00.0Z</rdeContact:upDate>
      <rdeContact:trDate>2009-12-03T09:05:00.0Z</rdeContact:trDate>
      <rdeContact:disclose flag="0">
        <contact:voice/>
        <contact:email/>
      </rdeContact:disclose>
    </rdeContact:contact>

    <!-- Registrar: RegistrarX -->
    <rdeRegistrar:registrar>
      <rdeRegistrar:id>RegistrarX</rdeRegistrar:id>
      <rdeRegistrar:name>Registrar X</rdeRegistrar:name>
      <rdeRegistrar:gurid>123</rdeRegistrar:gurid>
      <rdeRegistrar:status>ok</rdeRegistrar:status>
      <rdeRegistrar:postalInfo type="int">
        <rdeRegistrar:addr>
          <rdeRegistrar:street>123 Example Dr.</rdeRegistrar:street>
          <rdeRegistrar:street>Suite 100</rdeRegistrar:street>
          <rdeRegistrar:city>Dulles</rdeRegistrar:city>
          <rdeRegistrar:sp>VA</rdeRegistrar:sp>
          <rdeRegistrar:pc>20166-6503</rdeRegistrar:pc>
          <rdeRegistrar:cc>US</rdeRegistrar:cc>
        </rdeRegistrar:addr>
      </rdeRegistrar:postalInfo>
      <rdeRegistrar:voice x="1234">+1.7035555555</rdeRegistrar:voice>
      <rdeRegistrar:fax>+1.7035555556</rdeRegistrar:fax>
      <rdeRegistrar:email>jdoe@example.test</rdeRegistrar:email>
      <rdeRegistrar:url>http://www.example.test</rdeRegistrar:url>
      <rdeRegistrar:whoisInfo>
        <rdeRegistrar:name>whois.example.test</rdeRegistrar:name>
        <rdeRegistrar:url>http://whois.example.test</rdeRegistrar:url>
      </rdeRegistrar:whoisInfo>
      <rdeRegistrar:crDate>2005-04-23T11:49:00.0Z</rdeRegistrar:crDate>
      <rdeRegistrar:upDate>2009-02-17T17:51:00.0Z</rdeRegistrar:upDate>
    </rdeRegistrar:registrar>

    <!-- IDN Table -->
    <rdeIDN:idnTableRef id="pt-BR">
      <rdeIDN:url>
```

```
        http://www.iana.org/domains/idn-tables/tables/br_pt-br_1.0.html
      </rdeIDN:url>
      <rdeIDN:urlPolicy>
        http://registro.br/dominio/regras.html
      </rdeIDN:urlPolicy>
    </rdeIDN:idnTableRef>

    <!-- NNDN: pinguino.test -->
    <rdeNNDN:NNDN>
      <rdeNNDN:aName>xn--exampl-gva.test</rdeNNDN:aName>
      <rdeNNDN:idnTableId>pt-BR</rdeNNDN:idnTableId>
      <rdeNNDN:originalName>Dexample1-TEST</rdeNNDN:originalName>
      <rdeNNDN:nameState>withheld</rdeNNDN:nameState>
      <rdeNNDN:crDate>2005-04-23T11:49:00.0Z</rdeNNDN:crDate>
    </rdeNNDN:NNDN>

    <!-- EppParams -->
    <rdeEppParams:eppParams>
      <rdeEppParams:version>1.0</rdeEppParams:version>
      <rdeEppParams:lang>en</rdeEppParams:lang>
      <rdeEppParams:objURI>
        urn:ietf:params:xml:ns:domain-1.0
      </rdeEppParams:objURI>
      <rdeEppParams:objURI>
        urn:ietf:params:xml:ns:contact-1.0
      </rdeEppParams:objURI>
      <rdeEppParams:objURI>
        urn:ietf:params:xml:ns:host-1.0
      </rdeEppParams:objURI>
      <rdeEppParams:svcExtension>
        <epp:extURI>urn:ietf:params:xml:ns:rgp-1.0</epp:extURI>
        <epp:extURI>urn:ietf:params:xml:ns:secDNS-1.1</epp:extURI>
      </rdeEppParams:svcExtension>
      <rdeEppParams:dcp>
      <epp:access><epp:all/></epp:access>
        <epp:statement>
          <epp:purpose>
            <epp:admin/>
            <epp:prov/>
          </epp:purpose>
          <epp:recipient>
            <epp:ours/>
            <epp:public/>
          </epp:recipient>
          <epp:retention>
            <epp:stated/>
          </epp:retention>
        </epp:statement>
```

```
      </rdeEppParams:dcp>
    </rdeEppParams:eppParams>
        <rdePolicy:policy element="rdeDom:registrant" />
  </rde:contents>
</rde:deposit>
```

9. Appendix B. Example of differential deposit using the XML model only

Example of a differential deposit using the XML model only:

```
<?xml version="1.0" encoding="UTF-8"?>
<rde:deposit type="DIFF" id="20101017002" prevId="20101017001"
  xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
  xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
  xmlns:secDNS="urn:ietf:params:xml:ns:secDNS-1.1"
  xmlns:rde="urn:ietf:params:xml:ns:rde-1.0"
  xmlns:rdeHeader="urn:ietf:params:xml:ns:rdeHeader-1.0"
  xmlns:rdeDom="urn:ietf:params:xml:ns:rdeDomain-1.0"
  xmlns:rdeHost="urn:ietf:params:xml:ns:rdeHost-1.0"
  xmlns:rdeContact="urn:ietf:params:xml:ns:rdeContact-1.0"
  xmlns:rdeRegistrar="urn:ietf:params:xml:ns:rdeRegistrar-1.0"
  xmlns:rdeIDN="urn:ietf:params:xml:ns:rdeIDN-1.0"
  xmlns:rdeNNDN="urn:ietf:params:xml:ns:rdeNNDN-1.0"
  xmlns:rdeEppParams="urn:ietf:params:xml:ns:rdeEppParams-1.0"
  xmlns:epp="urn:ietf:params:xml:ns:epp-1.0">

  <rde:watermark>2010-10-17T00:00:00Z</rde:watermark>
  <rde:rdeMenu>
    <rde:version>1.0</rde:version>
    <rde:objURI>urn:ietf:params:xml:ns:rdeHeader-1.0</rde:objURI>
    <rde:objURI>urn:ietf:params:xml:ns:rdeContact-1.0</rde:objURI>
    <rde:objURI>urn:ietf:params:xml:ns:rdeHost-1.0</rde:objURI>
    <rde:objURI>urn:ietf:params:xml:ns:rdeDomain-1.0</rde:objURI>
    <rde:objURI>urn:ietf:params:xml:ns:rdeRegistrar-1.0</rde:objURI>
    <rde:objURI>urn:ietf:params:xml:ns:rdeIDN-1.0</rde:objURI>
    <rde:objURI>urn:ietf:params:xml:ns:rdeNNDN-1.0</rde:objURI>
    <rde:objURI>urn:ietf:params:xml:ns:rdeEppParams-1.0</rde:objURI>
  </rde:rdeMenu>

  <!-- Deletes -->
  <rde:deletes>
    <rdeDom:delete>
      <rdeDom:name>example2.test</rdeDom:name>
    </rdeDom:delete>
  </rde:deletes>
```

```
   <!-- Contents -->
   <rde:contents>
     <!-- Header -->
     <rdeHeader:header>
       <rdeHeader:tld>test</rdeHeader:tld>
       <rdeHeader:count
         uri="urn:ietf:params:xml:ns:rdeDomain-1.0">1</rdeHeader:count>
       <rdeHeader:count
         uri="urn:ietf:params:xml:ns:rdeHost-1.0">1</rdeHeader:count>
       <rdeHeader:count
         uri="urn:ietf:params:xml:ns:rdeContact-1.0">1</rdeHeader:count>
       <rdeHeader:count
         uri="urn:ietf:params:xml:ns:rdeRegistrar-1.0">1
           </rdeHeader:count>
       <rdeHeader:count
         uri="urn:ietf:params:xml:ns:rdeIDN-1.0">1</rdeHeader:count>
       <rdeHeader:count
         uri="urn:ietf:params:xml:ns:rdeNNDN-1.0">1</rdeHeader:count>
       <rdeHeader:count
         uri="urn:ietf:params:xml:ns:rdeEppParams-1.0">1
           </rdeHeader:count>
     </rdeHeader:header>
   </rde:contents>
 </rde:deposit>
```

10. Appendix C. Data escrow agent extended verification process

The Data Escrow Agent MAY perform a extended verification process using the contents of data escrow deposits to a point in time (watermark), last full plus all differentials or last full plus last incremental escrow deposits. The following are the minimum suggested tests:

- Validate the escrow deposits using the definition agreed with the registry.
  - In the case of the XML model, the contents of the escrow deposits MUST be validated using the XML schemas of the profile.
- Count the objects and validate that number of objects is equal to the number objects reported in the <header> element of the escrow deposit of that point in time (watermark).
- All contacts linked to domain names are present.

- All registrars linked to other objects are present.
- An FQDN exists only as a domain name or NNDN.
- The elements listed in the <policy> element are present.
- All idnTableRef definitions linked from other objects are present.

11. Appendix D. Data escrow notifications

Data escrowing involves several parties interacting with the objective of restoring the operations of a Domain Registry in case of an emergency. The following section defines several notifications that are suggested to be sent between the interacting parties. The parties based on the notification can know the status of the data escrow deposit even if no access to the data escrow deposit file is available.

11.1. Notifications from Registry Operators to Third Parties

Registry Operators MAY notify Third Parties that a data escrow deposit file was sent to the Data Escrow Agent.

11.1.1. <report> object

The <report> object is used by Registry Operator to notify Third Parties about successful delivery of a data escrow deposit to a Data Escrow Agent.

The <report> element contains the following child elements:

- An <id> element contains the identifier assigned to this report. An OPTIONAL resend attribute is used to specify the number of retries needed for a successful reception/validator of the data escrow deposit by the data escrow agent. It is recommended that the report identifier be the same as the data escrow deposit identifier.
- A <reDate> element contains the date and time that the data escrow deposit was successful received by the data escrow agent.
- An OPTIONAL <vaDate> element contains the date and time that the data escrow deposit was successfuly validated by the data escrow agent.
- A <kind> element is used to identify the kind of deposit: FULL, INCR (Incremental) or DIFF (Differential).

- A <lastFullDate> element contains the date and time of the last FULL data escrow deposit that was successfuly validated by the data escrow agent.

- A <watermark> element contains the data-time corresponding to the Timeline Watermark of the deposit.

- A <header> element contains the header of the data escrow deposit.

Example <report> object:

```
<?xml version="1.0" encoding="UTF-8"?>
  <rdeReport:report
    xmlns:rdeReport="urn:ietf:params:xml:ns:rdeReport-1.0"
        xmlns:rdeHeader="urn:ietf:params:xml:ns:rdeHeader-1.0">
    <rdeReport:id>20101017001</rdeReport:id>
    <rdeReport:reDate>2010-10-17T01:51:10.0Z</rdeReport:reDate>
    <rdeReport:vaDate>2010-10-17T02:51:10.0Z</rdeReport:vaDate>
    <rdeReport:kind>FULL</rdeReport:kind>
    <rdeReport:lastFullDate>2010-10-16</rdeReport:lastFullDate>
    <rdeReport:watermark>2010-10-17T00:00:00Z</rdeReport:watermark>
    <rdeHeader:header>
      <rdeHeader:tld>test</rdeHeader:tld>
      <rdeHeader:count
        uri="urn:ietf:params:xml:ns:rdeDomain-1.0">2</rdeHeader:count>
      <rdeHeader:count
        uri="urn:ietf:params:xml:ns:rdeHost-1.0">1</rdeHeader:count>
      <rdeHeader:count
        uri="urn:ietf:params:xml:ns:rdeContact-1.0">1</rdeHeader:count>
      <rdeHeader:count
        uri="urn:ietf:params:xml:ns:rdeRegistrar-1.0">1
          </rdeHeader:count>
      <rdeHeader:count
        uri="urn:ietf:params:xml:ns:rdeIDN-1.0">1</rdeHeader:count>
      <rdeHeader:count
        uri="urn:ietf:params:xml:ns:rdeNNDN-1.0">1</rdeHeader:count>
      <rdeHeader:count
        uri="urn:ietf:params:xml:ns:rdeEppParams-1.0">1
          </rdeHeader:count>
    </rdeHeader:header>
  </rdeReport:report>
```

11.2. Notifications from Data Escrow Agents to Third Parties

Data Escrow Agents MAY notify Third Parties that a data escrow deposit file was received or it is missing for a specific date.

11.2.1. <notification> object

The <notification> object is used by Data Escrow Agents to notify Third Parties about sucessful reception/validation of a data escrow deposit for a specific date. If multiple deposits are received in a day, the latest received deposit MUST be used to generate the notification.

The <notification> element contains the following child elements:

- An <reDate> element contains the reported date.
- A <status> element is used to specify the status of <reDate>. The possible values of status are: valid, invalid and missing.
  - Valid: The last received data escrow deposit for the specified date in <reDate> was successfully validated.
  - Invalid: The last received data escrow deposit for the specified date in <reDate> was not successfully validated.
  - Missing: No data escrow deposit was received on the date specified in <reDate>.
- A <report> element it is used by the data escrow agent to provide extended information about the data escrow deposit. The <header> element MUST be generated by the data escrow agent for a certain point in time (watermark) based on the contents of the escrow deposits. The last full plus all differentials or last full plus last incremental escrow deposits MUST be used to generate <header> element.

Example <notification> object:

```
<?xml version="1.0" encoding="UTF-8"?>
<rdeNotification:notification
  xmlns:rdeNotification="urn:ietf:params:xml:ns:rdeNotification-1.0"
  xmlns:rdeReport="urn:ietf:params:xml:ns:rdeReport-1.0"
  xmlns:rdeHeader="urn:ietf:params:xml:ns:rdeHeader-1.0">
  <rdeNotification:reDate>2010-10-17</rdeNotification:reDate>
  <rdeNotification:status>valid</rdeNotification:status>
  <rdeReport:report>
    <rdeReport:id>20101017001</rdeReport:id>
    <rdeReport:reDate>2010-10-17T02:51:10.0Z</rdeReport:reDate>
    <rdeReport:vaDate>2010-10-17T01:51:10.0Z</rdeReport:vaDate>
    <rdeReport:kind>FULL</rdeReport:kind>
    <rdeReport:lastFullDate>2010-10-16</rdeReport:lastFullDate>
    <rdeReport:watermark>2010-10-17T00:00:00Z</rdeReport:watermark>
    <rdeHeader:header>
      <rdeHeader:tld>test</rdeHeader:tld>
      <rdeHeader:count
        uri="urn:ietf:params:xml:ns:rdeDomain-1.0">2</rdeHeader:count>
      <rdeHeader:count
        uri="urn:ietf:params:xml:ns:rdeHost-1.0">1</rdeHeader:count>
      <rdeHeader:count
        uri="urn:ietf:params:xml:ns:rdeContact-1.0">1</rdeHeader:count>
      <rdeHeader:count
        uri="urn:ietf:params:xml:ns:rdeRegistrar-1.0">1
          </rdeHeader:count>
      <rdeHeader:count
        uri="urn:ietf:params:xml:ns:rdeIDN-1.0">1</rdeHeader:count>
      <rdeHeader:count
        uri="urn:ietf:params:xml:ns:rdeNNDN-1.0">1</rdeHeader:count>
      <rdeHeader:count
        uri="urn:ietf:params:xml:ns:rdeEppParams-1.0">1
          </rdeHeader:count>
    </rdeHeader:header>
  </rdeReport:report>
</rdeNotification:notification>
```

11.3. Formal Syntax

Seven schemas are presented here. The first schema is the base RDE schema. The second schema defines domain object for RDE. The third schema defines host object for RDE. The fourth schema defines contact object for RDE. The fifth schema defines registrar object for RDE. The sixth schema defines the idnTableRef and IDN objects. The last schema defines the eppParams objects.

11.3.1. RDE Domain Object


```
BEGIN
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="urn:ietf:params:xml:ns:rdeDomain-1.0"
  xmlns:rdeDomain="urn:ietf:params:xml:ns:rdeDomain-1.0"
  xmlns:rde="urn:ietf:params:xml:ns:rde-1.0"
  xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0"
  xmlns:secDNS="urn:ietf:params:xml:ns:secDNS-1.1"
  xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
  xmlns:eppcom="urn:ietf:params:xml:ns:eppcom-1.0"
  xmlns="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">

  <import namespace="urn:ietf:params:xml:ns:eppcom-1.0"
    schemaLocation="eppcom-1.0.xsd" />
```

```
  <import namespace="urn:ietf:params:xml:ns:domain-1.0"
    schemaLocation="domain-1.0.xsd"/>
  <import namespace="urn:ietf:params:xml:ns:secDNS-1.1"
    schemaLocation="secdns-1.1.xsd"/>
  <import namespace="urn:ietf:params:xml:ns:rgp-1.0"
    schemaLocation="rgp-1.0.xsd"/>
  <import namespace="urn:ietf:params:xml:ns:rde-1.0"
    schemaLocation="rde-1.0.xsd"/>

  <annotation>
    <documentation>
      Registry Data Escrow Domain provisioning schema
    </documentation>
  </annotation>

  <element name="abstractDomain" type="rdeDomain:abstractContentType"
    substitutionGroup="rde:content" abstract="true"/>
  <element name="domain" substitutionGroup="rdeDomain:abstractDomain"/>
  <element name="delete" type="rdeDomain:deleteType"
    substitutionGroup="rde:delete"/>

  <!-- Content Type -->
  <complexType name="abstractContentType">
    <complexContent>
      <extension base="rde:contentType">
        <sequence>
          <element name="name" type="eppcom:labelType"/>
          <element name="roid" type="eppcom:roidType"/>
          <element name="uName" type="eppcom:labelType" minOccurs="0"/>
          <element name="idnTableId" type="IDREF" minOccurs="0"/>
          <element name="originalName" type="eppcom:labelType"
            minOccurs="0"/>
          <element name="status" type="domain:statusType"
            maxOccurs="11"/>
          <element name="rgpStatus" type="rgp:statusType" minOccurs="0"
            maxOccurs="unbounded"/>
          <element name="registrant" type="eppcom:clIDType"
            minOccurs="0"/>
          <element name="contact" type="domain:contactType"
            minOccurs="0" maxOccurs="unbounded"/>
          <element name="ns" type="domain:nsType" minOccurs="0"/>
          <element name="clID" type="eppcom:clIDType"/>
          <element name="crRr" type="rde:rrType"/>
          <element name="crDate" type="dateTime" minOccurs="0"/>
          <element name="exDate" type="dateTime" minOccurs="0"/>
          <element name="upRr" type="rde:rrType" minOccurs="0"/>
          <element name="upDate" type="dateTime" minOccurs="0"/>
          <element name="secDNS" type="secDNS:dsOrKeyType"
```

```
            minOccurs="0"/>
          <element name="trDate" type="dateTime" minOccurs="0"/>
          <element name="authInfo" type="domain:authInfoType"
            minOccurs="0"/>
          <element name="trnData" type="rdeDomain:transferDataType"
            minOccurs="0"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>

  <complexType name="transferDataType">
    <sequence>
      <element name="trStatus" type="eppcom:trStatusType"/>
      <element name="reRr" type="rde:rrType"/>
      <element name="reDate" type="dateTime"/>
      <element name="acRr" type="rde:rrType"/>
      <element name="acDate" type="dateTime"/>
      <element name="exDate" type="dateTime" minOccurs="0"/>
    </sequence>
  </complexType>

  <!-- Delete Type -->
  <complexType name="deleteType">
    <complexContent>
      <extension base="rde:deleteType">
        <sequence>
          <element name="name" type="eppcom:labelType"
            maxOccurs="unbounded"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
</schema>
END
```

11.3.2. RDE Host Object

```
BEGIN
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="urn:ietf:params:xml:ns:rdeHost-1.0"
  xmlns:rdeHost="urn:ietf:params:xml:ns:rdeHost-1.0"
  xmlns:rde="urn:ietf:params:xml:ns:rde-1.0"
  xmlns:host="urn:ietf:params:xml:ns:host-1.0"
  xmlns:eppcom="urn:ietf:params:xml:ns:eppcom-1.0"
  xmlns="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">

  <import namespace="urn:ietf:params:xml:ns:eppcom-1.0"
    schemaLocation="eppcom-1.0.xsd" />
  <import namespace="urn:ietf:params:xml:ns:host-1.0"
    schemaLocation="host-1.0.xsd"/>
  <import namespace="urn:ietf:params:xml:ns:rde-1.0"
    schemaLocation="rde-1.0.xsd"/>

  <annotation>
    <documentation>
      Registry Data Escrow Host provisioning schema
    </documentation>
  </annotation>

  <element name="host" type="rdeHost:contentType"
    substitutionGroup="rde:content"/>
  <element name="delete" type="rdeHost:deleteType"
```

```
    substitutionGroup="rde:delete"/>

  <!-- Content Type -->
  <complexType name="contentType">
    <complexContent>
      <extension base="rde:contentType">
        <sequence>
          <element name="name" type="eppcom:labelType"/>
          <element name="roid" type="eppcom:roidType"/>
          <element name="status" type="host:statusType" maxOccurs="7"/>
          <element name="addr" type="host:addrType" minOccurs="0"
            maxOccurs="unbounded"/>
          <element name="clID" type="eppcom:clIDType"/>
          <element name="crRr" type="rde:rrType"/>
          <element name="crDate" type="dateTime"/>
          <element name="upRr" type="rde:rrType" minOccurs="0"/>
          <element name="upDate" type="dateTime" minOccurs="0"/>
          <element name="trDate" type="dateTime" minOccurs="0"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>

  <!-- Delete Type -->
  <complexType name="deleteType">
    <complexContent>
      <extension base="rde:deleteType">
        <sequence>
          <element name="name" type="eppcom:labelType"
            maxOccurs="unbounded"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
</schema>
END
```

11.3.3. RDE Contact Object

```
BEGIN
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="urn:ietf:params:xml:ns:rdeContact-1.0"
  xmlns:rdeContact="urn:ietf:params:xml:ns:rdeContact-1.0"
  xmlns:rde="urn:ietf:params:xml:ns:rde-1.0"
  xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
  xmlns:eppcom="urn:ietf:params:xml:ns:eppcom-1.0"
  xmlns="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">

  <!-- Import common element types. -->
  <import namespace="urn:ietf:params:xml:ns:eppcom-1.0"
    schemaLocation="eppcom-1.0.xsd"/>
  <import namespace="urn:ietf:params:xml:ns:contact-1.0"
    schemaLocation="contact-1.0.xsd"/>
  <import namespace="urn:ietf:params:xml:ns:rde-1.0"
    schemaLocation="rde-1.0.xsd"/>

  <annotation>
    <documentation>
      Registry Data Escrow contact provisioning schema
    </documentation>
  </annotation>
```

```
   <element name="abstractContact" type="rdeContact:abstractContentType"
     substitutionGroup="rde:content" abstract="true"/>
   <element name="contact"
     substitutionGroup="rdeContact:abstractContact"/>
   <element name="delete" type="rdeContact:deleteType"
     substitutionGroup="rde:delete"/>

   <!-- Contact Type -->
   <complexType name="abstractContentType">
     <complexContent>
       <extension base="rde:contentType">
         <sequence>
           <element name="id" type="eppcom:clIDType"/>
           <element name="roid" type="eppcom:roidType"/>
           <element name="status" type="contact:statusType"
             maxOccurs="7"/>
           <element name="postalInfo" type="contact:postalInfoType"
             maxOccurs="2"/>
           <element name="voice" type="contact:e164Type" minOccurs="0"/>
           <element name="fax" type="contact:e164Type" minOccurs="0"/>
           <element name="email" type="eppcom:minTokenType"/>
           <element name="clID" type="eppcom:clIDType"/>
           <element name="crRr" type="rde:rrType"/>
           <element name="crDate" type="dateTime"/>
           <element name="upRr" type="rde:rrType" minOccurs="0"/>
           <element name="upDate" type="dateTime" minOccurs="0"/>
           <element name="trDate" type="dateTime" minOccurs="0"/>
           <element name="trnData" type="rdeContact:transferDataType"
             minOccurs="0"/>
           <element name="disclose" type="contact:discloseType"
             minOccurs="0"/>
         </sequence>
       </extension>
     </complexContent>
   </complexType>

   <complexType name="transferDataType">
     <sequence>
       <element name="trStatus" type="eppcom:trStatusType"/>
       <element name="reRr" type="rde:rrType"/>
       <element name="reDate" type="dateTime"/>
       <element name="acRr" type="rde:rrType"/>
       <element name="acDate" type="dateTime"/>
     </sequence>
   </complexType>


   <!-- Delete Type -->
```

```
  <complexType name="deleteType">
    <complexContent>
      <extension base="rde:deleteType">
        <sequence>
          <element name="id" type="eppcom:clIDType" minOccurs="0"
            maxOccurs="unbounded"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
</schema>
END
```

11.3.4. RDE Registrar Object


```
BEGIN
```

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="urn:ietf:params:xml:ns:rdeRegistrar-1.0"
  xmlns:rdeRegistrar="urn:ietf:params:xml:ns:rdeRegistrar-1.0"
  xmlns:rde="urn:ietf:params:xml:ns:rde-1.0"
  xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
  xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
  xmlns:eppcom="urn:ietf:params:xml:ns:eppcom-1.0"
  xmlns="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">

  <!-- Import common element types. -->
  <import namespace="urn:ietf:params:xml:ns:eppcom-1.0"
    schemaLocation="eppcom-1.0.xsd"/>
  <import namespace="urn:ietf:params:xml:ns:domain-1.0"
    schemaLocation="domain-1.0.xsd"/>
  <import namespace="urn:ietf:params:xml:ns:contact-1.0"
    schemaLocation="contact-1.0.xsd"/>
  <import namespace="urn:ietf:params:xml:ns:rde-1.0"
    schemaLocation="rde-1.0.xsd"/>

  <annotation>
    <documentation>
      Registry Data Escrow registrar provisioning schema
    </documentation>
  </annotation>

  <element name="abstractRegistrar"
    type="rdeRegistrar:abstractContentType"
    substitutionGroup="rde:content" abstract="true"/>
  <element name="registrar"
    substitutionGroup="rdeRegistrar:abstractRegistrar"/>
  <element name="delete" type="rdeRegistrar:deleteType"
    substitutionGroup="rde:delete"/>

  <!-- Content Type -->
  <complexType name="abstractContentType">
    <complexContent>
      <extension base="rde:contentType">
        <sequence>
          <element name="id" type="eppcom:clIDType"/>
          <element name="name" type="rdeRegistrar:nameType"/>
          <element name="gurid" type="positiveInteger" minOccurs="0"/>
          <element name="status" type="rdeRegistrar:statusType"/>
          <element name="postalInfo" type="rdeRegistrar:postalInfoType"
            maxOccurs="2"/>
          <element name="voice" type="contact:e164Type" minOccurs="0"/>
          <element name="fax" type="contact:e164Type" minOccurs="0"/>
          <element name="email" type="eppcom:minTokenType"/>
```

```
          <element name="url" type="anyURI" minOccurs="0"/>
          <element name="whoisInfo" type="rdeRegistrar:whoisInfoType"
            minOccurs="0"/>
          <element name="crDate" type="dateTime"/>
          <element name="upDate" type="dateTime" minOccurs="0"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>

  <simpleType name="nameType">
    <restriction base="normalizedString">
      <minLength value="1" />
      <maxLength value="255" />
    </restriction>
  </simpleType>

  <simpleType name="statusType">
    <restriction base="token">
      <enumeration value="ok"/>
      <enumeration value="readonly"/>
      <enumeration value="terminated"/>
    </restriction>
  </simpleType>

  <complexType name="postalInfoType">
    <sequence>
      <element name="addr" type="rdeRegistrar:addrType" />
    </sequence>
    <attribute name="type" type="rdeRegistrar:postalInfoEnumType"
      use="required" />
  </complexType>

  <simpleType name="postalInfoEnumType">
    <restriction base="token">
      <enumeration value="loc" />
      <enumeration value="int" />
    </restriction>
  </simpleType>

  <complexType name="addrType">
    <sequence>
      <element name="street" type="rdeRegistrar:optPostalLineType"
        minOccurs="0" maxOccurs="3" />
      <element name="city" type="rdeRegistrar:postalLineType" />
      <element name="sp" type="rdeRegistrar:optPostalLineType"
        minOccurs="0" />
      <element name="pc" type="rdeRegistrar:pcType" minOccurs="0" />
```

```
      <element name="cc" type="rdeRegistrar:ccType" />
    </sequence>
  </complexType>

  <simpleType name="postalLineType">
    <restriction base="normalizedString">
      <minLength value="1" />
      <maxLength value="255" />
    </restriction>
  </simpleType>

  <simpleType name="optPostalLineType">
    <restriction base="normalizedString">
      <maxLength value="255" />
    </restriction>
  </simpleType>

  <simpleType name="pcType">
    <restriction base="token">
      <maxLength value="16" />
    </restriction>
  </simpleType>

  <simpleType name="ccType">
    <restriction base="token">
      <length value="2" />
    </restriction>
  </simpleType>

  <complexType name="whoisInfoType">
    <sequence>
      <element name="name" type="eppcom:labelType" minOccurs="0"/>
      <element name="url" type="anyURI" minOccurs="0"/>
    </sequence>
  </complexType>

  <!-- Delete Type -->
  <complexType name="deleteType">
    <complexContent>
      <extension base="rde:deleteType">
        <sequence>
          <element name="id" type="eppcom:clIDType" minOccurs="0"
            maxOccurs="unbounded"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
</schema>
```

```
END
```

11.3.5. RDE IDN and IDN Table Reference Objects

```
BEGIN
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="urn:ietf:params:xml:ns:rdeIDN-1.0"
  xmlns:rdeIDN="urn:ietf:params:xml:ns:rdeIDN-1.0"
  xmlns:rde="urn:ietf:params:xml:ns:rde-1.0"
  xmlns="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">

  <import namespace="urn:ietf:params:xml:ns:rde-1.0"
    schemaLocation="rde-1.0.xsd"/>

  <annotation>
    <documentation>
      Registry Data Escrow IDN provisioning schema
    </documentation>
  </annotation>

  <element name="idnTableRef" type="rdeIDN:contentType"
    substitutionGroup="rde:content"/>

  <!-- Content Type -->
  <complexType name="contentType">
    <complexContent>
      <extension base="rde:contentType">
        <sequence>
          <element name="url" type="anyURI"/>
          <element name="urlPolicy" type="anyURI"/>
        </sequence>
        <attribute name="id" type="rdeIDN:IdType" use="required"/>
      </extension>
    </complexContent>
  </complexType>

  <simpleType name="IdType">
    <restriction base="ID">
      <whiteSpace value="collapse"/>
    </restriction>
  </simpleType>

</schema>
END
```

11.3.6. EPP Parameters Object

```
BEGIN
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="urn:ietf:params:xml:ns:rdeEppParams-1.0"
  xmlns:rdeEppParams="urn:ietf:params:xml:ns:rdeEppParams-1.0"
  xmlns:rde="urn:ietf:params:xml:ns:rde-1.0"
  xmlns:epp="urn:ietf:params:xml:ns:epp-1.0"
  xmlns:eppcom="urn:ietf:params:xml:ns:eppcom-1.0"
  xmlns="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">

  <import namespace="urn:ietf:params:xml:ns:epp-1.0"
    schemaLocation="epp-1.0.xsd"/>
  <import namespace="urn:ietf:params:xml:ns:eppcom-1.0"
    schemaLocation="eppcom-1.0.xsd"/>
  <import namespace="urn:ietf:params:xml:ns:rde-1.0"
    schemaLocation="rde-1.0.xsd"/>

  <annotation>
    <documentation>
      Registry Data Escrow EPP Parameters schema
    </documentation>
  </annotation>

  <!-- Content Type -->
  <element name="eppParams"
    substitutionGroup="rdeEppParams:abstractEppParams"/>

  <!-- Abstract Content Type -->
  <element name="abstractEppParams"
    type="rdeEppParams:abstractContentType"
    substitutionGroup="rde:content" abstract="true"/>
  <complexType name="abstractContentType">
    <complexContent>
      <extension base="rde:contentType">
        <sequence>
          <element name="version" type="epp:versionType"
            maxOccurs="unbounded"/>
          <element name="lang" type="language" maxOccurs="unbounded"/>
          <element name="objURI" type="anyURI" maxOccurs="unbounded"/>
          <element name="svcExtension" type="epp:extURIType"
            minOccurs="0"/>
          <element name="dcp" type="epp:dcpType"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
</schema>
END
```

11.3.7. NNDN Object


```
BEGIN
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="urn:ietf:params:xml:ns:rdeNNDN-1.0"
  xmlns:rdeNNDN="urn:ietf:params:xml:ns:rdeNNDN-1.0"
  xmlns:rde="urn:ietf:params:xml:ns:rde-1.0"
  xmlns:eppcom="urn:ietf:params:xml:ns:eppcom-1.0"
  xmlns="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">

  <import namespace="urn:ietf:params:xml:ns:eppcom-1.0"
    schemaLocation="eppcom-1.0.xsd"/>
  <import namespace="urn:ietf:params:xml:ns:rde-1.0"
    schemaLocation="rde-1.0.xsd"/>
```

```
  <annotation>
    <documentation>
      Registry Data Escrow NNDN provisioning schema
    </documentation>
  </annotation>

  <element name="abstractNNDN" type="rdeNNDN:abstractContentType"
    substitutionGroup="rde:content" abstract="true"/>
  <element name="NNDN" substitutionGroup="rdeNNDN:abstractNNDN"/>
  <element name="delete" type="rdeNNDN:deleteType"
    substitutionGroup="rde:delete"/>

  <!-- Content Type -->
  <complexType name="abstractContentType">
    <complexContent>
      <extension base="rde:contentType">
        <sequence>
          <element name="aName" type="eppcom:labelType"/>
          <element name="uName" type="eppcom:labelType" minOccurs="0"/>
          <element name="idnTableId" type="IDREF" minOccurs="0"/>
          <element name="originalName" type="eppcom:labelType"
            minOccurs="0"/>
          <element name="nameState" type="rdeNNDN:nameState"/>
          <element name="crDate" type="dateTime"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>

  <simpleType name="nameState">
    <restriction base="token">
      <enumeration value="withheld"/>
      <enumeration value="blocked"/>
    </restriction>
  </simpleType>

  <!-- Delete Type -->
  <complexType name="deleteType">
    <complexContent>
      <extension base="rde:deleteType">
        <sequence>
          <element name="aName" type="eppcom:labelType" minOccurs="0"
            maxOccurs="unbounded"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
</schema>
```

END

11.3.8. Header Object

```
BEGIN
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="urn:ietf:params:xml:ns:rdeHeader-1.0"
  xmlns:rdeHeader="urn:ietf:params:xml:ns:rdeHeader-1.0"
  xmlns:rde="urn:ietf:params:xml:ns:rde-1.0"
  xmlns:eppcom="urn:ietf:params:xml:ns:eppcom-1.0"
  xmlns="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">

  <import namespace="urn:ietf:params:xml:ns:eppcom-1.0"
    schemaLocation="eppcom-1.0.xsd" />
  <import namespace="urn:ietf:params:xml:ns:rde-1.0"
    schemaLocation="rde-1.0.xsd"/>

  <annotation>
    <documentation>
      Registry Data Escrow Header schema
    </documentation>
  </annotation>

  <!-- Root Element -->
  <element name="header" type="rdeHeader:contentType"
    substitutionGroup="rde:content"/>

  <!-- Content Type -->
  <complexType name="contentType">
    <complexContent>
      <extension base="rde:contentType">
        <sequence>
          <element name="tld" type="eppcom:labelType"/>
          <element name="count" type="rdeHeader:countType"
            maxOccurs="unbounded"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>

  <complexType name="countType">
    <simpleContent>
      <extension base="long">
        <attribute name="uri" type="anyURI"/>
      </extension>
    </simpleContent>
  </complexType>
</schema>
END
```

11.3.9. Report Object

```
BEGIN
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="urn:ietf:params:xml:ns:rdeReport-1.0"
  xmlns:rdeReport="urn:ietf:params:xml:ns:rdeReport-1.0"
  xmlns:rdeHeader="urn:ietf:params:xml:ns:rdeHeader-1.0"
  xmlns:rde="urn:ietf:params:xml:ns:rde-1.0"
  xmlns="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">

  <import namespace="urn:ietf:params:xml:ns:rde-1.0"
    schemaLocation="rde-1.0.xsd"/>
  <import namespace="urn:ietf:params:xml:ns:rdeHeader-1.0"
    schemaLocation="rde-header-1.0.xsd" />

  <annotation>
    <documentation>
      Registry Data Escrow Report schema
    </documentation>
  </annotation>

  <!-- Root Element -->
  <element name="report" type="rdeReport:reportType"/>

  <!-- Report Type -->
  <complexType name="reportType">
    <sequence>
      <element name="id" type="rde:depositIdType"/>
      <element name="reDate" type="dateTime"/>
      <element name="vaDate" type="dateTime" minOccurs="0"/>
      <element name="kind" type="rde:depositTypeType"/>
      <element name="lastFullDate" type="date"/>
      <element name="watermark" type="dateTime"/>
      <element ref="rdeHeader:header"/>
    </sequence>
  </complexType>
</schema>
END
```

11.3.10. Notifiaction Object

```
BEGIN
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="urn:ietf:params:xml:ns:rdeNotification-1.0"
  xmlns:rdeNotification="urn:ietf:params:xml:ns:rdeNotification-1.0"
  xmlns:rdeReport="urn:ietf:params:xml:ns:rdeReport-1.0"
  xmlns="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">

  <import namespace="urn:ietf:params:xml:ns:rdeReport-1.0"
    schemaLocation="rde-report-1.0.xsd"/>

  <annotation>
    <documentation>
      Registry Data Escrow Notification schema
    </documentation>
  </annotation>

  <!-- Root Element -->
  <element name="notification" type="rdeNotification:notificationType"/>

  <!-- Notification -->
  <complexType name="notificationType">
    <sequence>
      <element name="reDate" type="date"/>
      <element name="status" type="rdeNotification:statusType"/>
      <element ref="rdeReport:report"/>
    </sequence>
  </complexType>

  <simpleType name="statusType">
    <restriction base="token">
      <enumeration value="valid"/>
      <enumeration value="invalid"/>
      <enumeration value="missing"/>
    </restriction>
  </simpleType>
</schema>
END
```

11.4. Policy Object

```
BEGIN
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="urn:ietf:params:xml:ns:rdePolicy-1.0"
  xmlns:rdePolicy="urn:ietf:params:xml:ns:rdePolicy-1.0"
  xmlns:rde="urn:ietf:params:xml:ns:rde-1.0"
  xmlns="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">

  <annotation>
    <documentation>
      Registry Data Escrow Policy schema
    </documentation>
  </annotation>

  <import namespace="urn:ietf:params:xml:ns:rde-1.0"
    schemaLocation="rde-1.0.xsd"/>

  <element name="policy" type="rdePolicy:policyType"
    substitutionGroup="rde:content"/>

  <complexType name="policyType">
    <complexContent>
      <extension base="rde:contentType">
        <attribute name="element" type="anyURI" use="required"/>
      </extension>
    </complexContent>
  </complexType>
</schema>
END
```

12. Internationalization Considerations

Data Escrow deposits are represented in XML, which provides native support for encoding information using the Unicode character set and its more compact representations including UTF-8. Conformant XML processors recognize both UTF-8 and UTF-16. Though XML includes provisions to identify and use other character encodings through use of an "encoding" attribute in an <?xml?> declaration, use of UTF-8 is RECOMMENDED.

13. IANA Considerations

This document uses URNs to describe XML namespaces and XML schemas conforming to a registry mechanism described in [RFC3688]. Fourteen URI assignments have been registered by the IANA.

Registration request for the RDE domain namespace:

URI: urn:ietf:params:xml:ns:rdeDomain-1.0

Registrant Contact: See the "Author's Address" section of this document.

XML: None. Namespace URIs do not represent an XML specification.

Registration request for the RDE domain XML schema:

URI: urn:ietf:params:xml:schema:rdeDomain-1.0

Registrant Contact: See the "Author's Address" section of this document.

See the "Formal Syntax" section of this document.

Registration request for the RDE host namespace:

URI: urn:ietf:params:xml:ns:rdeHost-1.0

Registrant Contact: See the "Author's Address" section of this document.

XML: None. Namespace URIs do not represent an XML specification.

Registration request for the RDE host XML schema:

URI: urn:ietf:params:xml:schema:rdeHost-1.0

Registrant Contact: See the "Author's Address" section of this document.

See the "Formal Syntax" section of this document.

Registration request for the RDE contact namespace:

URI: urn:ietf:params:xml:ns:rdeContact-1.0

Registrant Contact: See the "Author's Address" section of this document.

XML: None. Namespace URIs do not represent an XML specification.

Registration request for the RDE contact XML schema:

URI: urn:ietf:params:xml:schema:rdeContact-1.0

Registrant Contact: See the "Author's Address" section of this document.

See the "Formal Syntax" section of this document.

Registration request for the RDE registrar namespace:

URI: urn:ietf:params:xml:ns:rdeRegistrar-1.0

Registrant Contact: See the "Author's Address" section of this document.

XML: None. Namespace URIs do not represent an XML specification.

Registration request for the RDE registrar XML schema:

URI: urn:ietf:params:xml:schema:rdeRegistrar-1.0

Registrant Contact: See the "Author's Address" section of this document.

See the "Formal Syntax" section of this document.

Registration request for the RDE IDN namespace:

URI: urn:ietf:params:xml:ns:rdeIDN-1.0

Registrant Contact: See the "Author's Address" section of this document.

XML: None. Namespace URIs do not represent an XML specification.

Registration request for the RDE IDN XML schema:

URI: urn:ietf:params:xml:schema:rdeIDN-1.0

Registrant Contact: See the "Author's Address" section of this document.

See the "Formal Syntax" section of this document.

Registration request for the RDE EPP parameters namespace:

URI: urn:ietf:params:xml:ns:rdeEppParams-1.0

Registrant Contact: See the "Author's Address" section of this document.

XML: None. Namespace URIs do not represent an XML specification.

Registration request for the RDE EPP parameters XML schema:

URI: urn:ietf:params:xml:schema:rdeEppParams-1.0

Registrant Contact: See the "Author's Address" section of this document.

See the "Formal Syntax" section of this document.

14. Security Considerations

This specification does not define the security mechanisms to be used in the transmission of the data escrow deposits, since it only specifies the minimum necessary to enable the rebuilding of a Registry from deposits without intervention from the original Registry.

Depending on local policies, some elements or most likely, the whole deposit will be considered confidential. As such the Registry transmitting the data to the Escrow Agent SHOULD take all the necessary precautions like encrypting the data itself and/or the transport channel to avoid inadvertent disclosure of private data.

It is also of the utmost importance the authentication of the parties passing data escrow deposit files. The Escrow Agent SHOULD properly authenticate the identity of the Registry before accepting data escrow deposits. In a similar manner, the Registry SHOULD authenticate the identity of the Escrow Agent before submitting any data.

Additionally, the Registry and the Escrow Agent SHOULD use integrity checking mechanisms to ensure the data transmitted is what the source intended. Validation of the contents by the Escrow Agent is RECOMMENDED to ensure not only the file was transmitted correctly from the Registry, but also the contents are also "meaningful".

15. Acknowledgments

Parts of this document are based on EPP [RFC5730] and related RFCs by Scott Hollenbeck.

TBD

16. Change History

[[RFC Editor: Please remove this section.]]

16.1. Changes from draft-arias-noguchi-registry-data-escrow-02 to -dnrd-objects-mapping-00

1. Added definition for child elements under the <domain> element.
2. Added definition for child elements under the <host> element.
3. Added definition for child elements under the <contact> element.
4. Rewrote the IDN Variants Handling section to use the variant states as described in ICANN's Study of Issues Related to the Management of IDN Variant TLDs.
5. Renamed <icannID> to <gurid> in the <rdeRegistrar>.
6. Renamed <dnssec> to <secDNS> in the <domain> element.
7. Renamed <transfData> to <trnData> in the <domain> element.
8. Added <whoisInfo> element under <rdeRegistrar> element.
9. Fixed some typographical errors and omissions.

16.2. Changes from version 00 to 01

1. Specify OPTIONAL elements in the draft.
2. Added NNDN object to support list of reserved names and different IDN variants models.
3. Removed subordinated host element from the domain object.
4. Added eppParams object.
5. Added variantGenerator element to the domain object.
6. Added lgr to the IDN table object.

16.3. Changes from version 01 to 02

1. Updates to the all objects based on feedback from the list.
2. Start of XML and CSV drafts merge.
3. Added header object.
4. Added report object.
5. Added notification object.
6. Added Data Escrow Agent Extended Verification Process section.
7. Added Notifications from Registries to Third Parties.
8. Added Notifications from Data Escrow Agents to Third Parties.
9. Added FULL, DIFF deposit examples using the XML model only.

17. References

17.1. Normative References

[ISO-3166-1] International Organization for Standardization, "Codes for the representation of names of countries and their subdivisions -- Part 1: Country codes", ISO Standard 3166, November 2006.

[ITU-E164] International Telecommunication Union, "The international public telecommunication numbering plan", ITU-T Recommendation E.164, February 2005.

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

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

[RFC3915] Hollenbeck, S., "Domain Registry Grace Period Mapping for the Extensible Provisioning Protocol (EPP)", RFC 3915, September 2004.

[RFC5731] Hollenbeck, S., "Extensible Provisioning Protocol (EPP) Domain Name Mapping", STD 69, RFC 5731, August 2009.

[RFC5732] Hollenbeck, S., "Extensible Provisioning Protocol (EPP) Host Mapping", STD 69, RFC 5732, August 2009.

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

[RFC5910] Gould, J. and S. Hollenbeck, "Domain Name System (DNS) Security Extensions Mapping for the Extensible Provisioning Protocol (EPP)", RFC 5910, May 2010.

17.2. Informative References

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

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, January 2004.

[RFC3743] Konishi, K., Huang, K., Qian, H., and Y. Ko, "Joint Engineering Team (JET) Guidelines for Internationalized Domain Names (IDN) Registration and Administration for Chinese, Japanese, and Korean", RFC 3743, April 2004.

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

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

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

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

[variantTLDsReport] Internet Corporation for Assigned Names and Numbers (ICANN), "A Study of Issues Related to the Management of IDN Variant TLDs", February 2012, <http://www.icann.org/en/topics/idn/idn-vip-integrated-issues-final-clean-20feb12-en.pdf>.

Authors' Addresses

Francisco Arias
Internet Corporation for Assigned Names and Numbers
12025 Waterfront Drive, Suite 300
Los Angeles 90292
United States of America

Phone: +1.310.823.9358
Email: francisco.arias@icann.org

Gustavo Lozano
Internet Corporation for Assigned Names and Numbers
12025 Waterfront Drive, Suite 300
Los Angeles 90292
United States of America

Phone: +1.310.823.9358
Email: gustavo.lozano@icann.org

Shoji Noguchi
Japan Registry Services Co., Ltd.
Chiyoda First Bldg. East 13F, 3-8-1 Nishi-Kanda
Chiyoda-ku, Tokyo 101-0065
Japan

Phone: +81.3.5215.8451
Email: noguchi@jprs.co.jp

James Gould
VeriSign, Inc.
12061 Bluemont Way
Reston 20190
United States of America

Email: jgould@verisign.com

Chethan Thippeswamy
VeriSign, Inc.
12061 Bluemont Way
Reston 20190
United States of America

Email: cthippeswamy@verisign.com