

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: July 6, 2019

G. Lozano
ICANN
J. Gould
C. Thippeswamy
VeriSign
Jan 02, 2019

Domain Name Registration Data (DNRD) Objects Mapping
draft-arias-noguchi-dnrd-objects-mapping-10

Abstract

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

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 <https://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 6, 2019.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must 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	3
2. Models	4
3. Terminology	5
4. General Conventions	5
4.1. Date and Time	5
4.2. Country names	5
4.3. Telephone numbers	6
4.4. Checksum	6
4.5. IP addresses	6
4.6. CSV Parent Child Relationship	6
4.7. CSV elements	7
4.8. Internationalized and Localized Elements	12
5. Object Description	14
5.1. Domain Name Object	14
5.2. Host Object	33
5.3. Contact Object	43
5.4. Registrar Object	60
5.5. IDN Table Reference Object	68
5.6. NNDN Object	72
5.7. EPP Parameters Object	77
5.8. Policy Object	79
5.9. Header Object	79
6. RDE IDN Variants handling	82
7. Profile	83
8. Data escrow agent extended verification process	83
9. Formal Syntax	84
9.1. RDE CSV Schema	84
9.2. RDE Domain Object	94
9.3. CSV Domain Object	97
9.4. RDE Host Object	101
9.5. CSV Host Object	104
9.6. RDE Contact Object	107
9.7. CSV Contact Object	110
9.8. RDE Registrar Object	116
9.9. CSV Registrar Object	120
9.10. RDE IDN Table Reference Objects	124
9.11. CSV IDN Language Object	126
9.12. EPP Parameters Object	128
9.13. NNDN Object	130
9.14. CSV NNDN Object	132
9.15. Policy Object	135
9.16. Header Object	136
10. Internationalization Considerations	138
11. IANA Considerations	138
12. Implementation Status	144
12.1. Implementation in the gTLD space	144

13. Security Considerations	145
14. Acknowledgments	145
15. Change History	145
15.1. Changes from draft-arias-noguchi-registry-data-escrow-02 to -dnrd-objects-mapping-00	146
15.2. Changes from version 00 to 01	146
15.3. Changes from version 01 to 02	146
15.4. Changes from version 02 to 03	147
15.5. Changes from version 03 to 04	147
15.6. Changes from version 04 to 05	148
15.7. Changes from version 05 to 06	149
15.8. Changes from version 06 to 07	149
15.9. Changes from version 07 to 08	149
15.10. Changes from version 08 to 09	150
15.11. Changes from version 09 to 10	150
16. Example of a full deposit using the XML model	150
17. Example of differential deposit using the XML model	155
18. Example of a full deposit using the CSV model	157
19. Example of differential deposit using the CSV model	166
20. References	177
20.1. Normative References	177
20.2. Informative References	178
20.3. URIs	178
Authors' Addresses	179

1. Introduction

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

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

- o NNDN (NNDN's not domain name): A lightweight domain-like object that is not linked to a Registrar.

This document defines the following pseudo-objects:

- o IDN Table Reference: Internationalized Domain Names (IDN) included in the Domain Object Data Escrow include references to the IDN Table and Policy used in IDN registration.
- o EPP parameters: Definition of the specific EPP parameters supported by the Registry Operator.
- o Header: Used to specify counters of objects in the database at a certain point in time (watermark).
- o 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:

- o XML: The XML model includes all the deposit information (meta-data and data) in an XML document. The definition of the XML format is fully defined in the XML schemas. As a convention, the objects represented using the XML model are referenced using RDE and an XML namespace that is prefixed with "rde". For example, the Domain Name object represented using the XML model can be referred to as the RDE Domain Name with the XML namespace including rdeDomain (urn:ietf:params:xml:ns:rdeDomain-1.0).
- o CSV: The CSV model uses XML to define the data escrow format of the data contained in referenced Comma-Separated Values (CSV) files. As a convention, the objects represented using the CSV model is referenced using CSV and an XML namespace that is prefixed with "csv". For example, the Domain Name object represented using the CSV model can be referred to as the CSV Domain Name with the XML namespace including csvDomain (urn:ietf:params:xml:ns:csvDomain-1.0).

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 an 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 represent 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 facsimile) 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. Checksum

Checksum of the CSV data escrow files MUST use CRC32, that is the algorithm used in the ISO 3309 standard and in section 8.1.1.6.2 of ITU-T recommendation V.42.

4.5. IP addresses

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

4.6. CSV Parent Child Relationship

The CSV model represents a relational model, where the CSV files represent relational tables, the fields of the CSV files represent columns of the tables, and each line of the CSV file represents a record. As in a relational model, the CSV files can have relationships utilizing primary keys in the parent CSV file definitions and foreign keys in the child CSV file definitions for a 1-to-many relationship. The primary keys are not explicitly defined, but the foreign keys are using the boolean "parent" field attribute in the child CSV file. The relationships between the CSV files are used to support a cascade replace or cascade delete of records starting from the parent record in differential and incremental deposits.

The following is an example of the CSV file definitions for a Sample object consisting of a parent "sample" CSV File Definition and a child "sampleStatuses" CSV File Definition. The primary key for the Sample object is the field <csvSample:fName> that is used as the foreign key in the "sampleStatuses" CSV File Definition by specifying the "parent=true" attribute. If a Sample record is updated or deleted in a differential or incremental deposit, it should cascade replace the data using the records included in the child "sampleStatuses" CSV File Definition or cascade delete the existing records in the child "sampleStatuses" CSV File Definition, respectively.

```

<csvSample:contents>
...
  <rdeCsv:csv name="sample" sep=", ">
    <rdeCsv:fields>
      <csvSample:fName/>
      <rdeCsv:fClID/>
      <rdeCsv:fCrRr/>
      <rdeCsv:fCrID/>
      <rdeCsv:fCrDate/>
      <rdeCsv:fUpRr/>
      <rdeCsv:fUpID/>
      <rdeCsv:fUpDate/>
      <rdeCsv:fExDate/>
    </rdeCsv:fields>
    <rdeCsv:files>
      <rdeCsv:file
        cksum="75E2D22F">
        sample-YYYYMMDD.csv
      </rdeCsv:file>
    </rdeCsv:files>
  </rdeCsv:csv>
  <rdeCsv:csv name="sampleStatuses" sep=", ">
    <rdeCsv:fields>
      <csvSample:fName parent="true"/>
      <csvSample:fStatus/>
      <rdeCsv:fStatusDescription/>
      <rdeCsv:fLang/>
    </rdeCsv:fields>
    <rdeCsv:files>
      <rdeCsv:file
        cksum="EB9C558E">
        sampleStatuses-YYYYMMDD.csv
      </rdeCsv:file>
    </rdeCsv:files>
  </rdeCsv:csv>
...
</csvSample:contents>

```

4.7. CSV elements

4.7.1. <rdeCsv:csv> element

To support a CSV model with the Registry Data Escrow Specification [1], an element is defined for each object that substitutes for the <rde:content> element and for the <rde:delete> element, that contains one or more <rdeCsv:csv> elements. For example, the Domain Name Object (Section 5.1) defines the <csvDomain:contents> element, that substitutes for the <rde:content> element, and the

<csvDomain:deletes> element, that substitutes for the <rde:delete> element. Both the <csvDomain:contents> element and the <csvDomain:deletes> elements contain one or more <rdeCsv:csv> elements. The <rdeCsv:csv> element has the following child elements:

<rdeCsv:fields> Ordered list of CSV fields used in the CSV files. There is one or more child elements that substitute for the <rdeCsv:field> abstract element. Each element defines the format of the CSV field contained in the CSV files. The <rdeCsv:field> elements support the "type" attribute that defines the XML simple data type of the field element. The <rdeCsv:field> elements support the "isRequired" attribute, with a default value of "false", when set to "true" indicates that the field must be non-empty in the CSV files and when set to "false" indicates that the field MAY be empty in the CSV files. The "isRequired" attribute MAY be specifically set for the field elements within the XML schema and MAY be overridden when specifying the fields under the <rdeCsv:fields> element. The <rdeCsv:field> element supports an OPTIONAL "parent" attribute that identifies the field as a reference to a parent object, as defined in CSV Parent Child Relationship (Section 4.6). For example, the <rdeCsv:csv name="domainStatuses"> <csvDomain:fName> field SHOULD set the "parent" attribute to "true" to identify it as the parent domain name of the domain status.

<rdeCsv:files> A list of one or more CSV files using the <rdeCsv:file> child element. The <rdeCsv:file> child element defines a reference to the CSV file name and has the following optional attributes:

compression If the CSV file is compressed, the "compression" attribute defines the compression format like "gzip" or "zip".

encoding Defines the encoding of the CSV file with the default encoding of "UTF-8".

cksum Defines the checksum of the CSV file using CRC32, as defined in Section 4.4. This attribute is used to validate that the full CSV file exists and has not been tampered with.

The <rdeCsv:csv> elements requires a "name" attribute that defines the purpose of the CSV file with values like "domain", "host", "contact". The supported "name" attribute values are defined for each object type. The OPTIONAL "sep" attribute defines the CSV separator character with the default separator character of ",".

The following is an example of the <csvDomain:contents> <rdeCsv:csv> element for domain name records where the <rdeCsv:fRegistrant> is set as required with isRequired="true".

```
<csvDomain:contents>
...
  <rdeCsv:csv name="domain" sep=", ">
    <rdeCsv:fields>
      <csvDomain:fName/>
      <rdeCsv:fRoid/>
      <rdeCsv:fIdnTableId/>
      <csvDomain:fOriginalName/>
      <rdeCsv:fRegistrant isRequired="true"/>
      <rdeCsv:fClID/>
      <rdeCsv:fCrRr/>
      <rdeCsv:fCrID/>
      <rdeCsv:fCrDate/>
      <rdeCsv:fUpRr/>
      <rdeCsv:fUpID/>
      <rdeCsv:fUpDate/>
      <rdeCsv:fExDate/>
    </rdeCsv:fields>
    <rdeCsv:files>
      <rdeCsv:file
        cksum="75E2D01F">
        domain-YYYYMMDD.csv
      </rdeCsv:file>
    </rdeCsv:files>
  </rdeCsv:csv>
...
</csvDomain:contents>
```

The following is example of the "domain-YYYYMMDD.csv" file with one record matching the <rdeCsv:fields> definition.

```
domain1.test,Ddomain2-TEST,,,registrantid,registrarX,registrarX,
clientY,2009-04-03T22:00:00.0Z,registrarX,clientY,
2009-12-03T09:05:00.0Z,2015-04-03T22:00:00.0Z
```

The following is an example of the <csvDomain:deletes> <rdeCsv:csv> element for domain name records.

```
<csvDomain:deletes>
...
  <rdeCsv:csv name="domain">
    <rdeCsv:fields>
      <csvDomain:fName/>
    </rdeCsv:fields>
    <rdeCsv:files>
      <rdeCsv:file
        cksum="6F2B988F">
        domain-delete-YYYYMMDD.csv
      </rdeCsv:file>
    </rdeCsv:files>
  </rdeCsv:csv>
...
</csvDomain:deletes>
```

The following is example of the "domain-delete-YYYYMMDD.csv" file with three records that matches the single <csvDomain:fName> field.

```
domain1.test
domain2.test
domainN.test
```

4.7.2. CSV common field elements

The <rdeCsv:fields> element defined in the <rdeCsv:csv> element (Section 4.7.1) section has child elements that substitute for the abstract <rdeCsv:field> element. By convention <rdeCsv:field> elements include an 'f' prefix to identify them as field definition elements. There are a set of common field elements that are used across multiple data escrow objects. The common field elements are defined using the "urn:ietf:params:xml:ns:rdeCsv-1.0" namespace and using the "rdeCsv" sample namespace prefix. The CSV common field elements include:

<rdeCsv:fUName> UTF-8 encoded name field with
type="eppcom:labelType".

<rdeCsv:fRoid> Repository Object Identifier (ROID) field with
type="eppcom:roidType" and isRequired="true".

<rdeCsv:fRegistrant> Registrant contact identifier with
type="eppcom:clIDType".

<rdeCsv:fStatusDescription> The object status description, which is free form text describing the rationale for the status, with type="normalizedString".

<rdeCsv:fClID> Identifier of client (registrar) that sponsors the object with type="eppcom:clIDType" and isRequired="true".

<rdeCsv:fCrRr> Identifier of the registrar, defined in Section 5.4, of the client that created the object with type="eppcom:clIDType".

<rdeCsv:fCrID> Identifier of client that created the object with type="eppcom:clIDType".

<rdeCsv:fUpRr> Identifier of the registrar, defined in Section 5.4, of the client that updated the object with type="eppcom:clIDType".

<rdeCsv:fUpID> Identifier of client that updated the object with type="eppcom:clIDType".

<rdeCsv:fReRr> Identifier of the registrar, defined in Section 5.4, of the client that requested the transfer with type="eppcom:clIDType" and isRequired="true".

<rdeCsv:fReID> Identifier of client that requested the transfer with type="eppcom:clIDType".

<rdeCsv:fAcRr> Identifier of the registrar, defined in Section 5.4, of the client that should take or took action with type="eppcom:clIDType" and isRequired="true".

<rdeCsv:fAcID> Identifier of client that should take or took action for transfer with type="eppcom:clIDType".

<rdeCsv:fCrDate> Created date of object with type="dateTime" and isRequired="true".

<rdeCsv:fUpDate> Updated date of object with type="dateTime".

<rdeCsv:fExDate> Expiration date of object with type="dateTime".

<rdeCsv:fReDate> Date that transfer was requested with type="dateTime" and isRequired="true".

<rdeCsv:fAcDate> Date that transfer action should be taken or has been taken with type="dateTime" and isRequired="true".

<rdeCsv:fTrDate> Date of last transfer with type="dateTime".

<rdeCsv:fTrStatus> State of the most recent transfer request with type="eppcom:trStatusType" and isRequired="true".

<rdeCsv:fTokenType> General token field with type="token".

<rdeCsv:fLang> General language field with type="language".

<rdeCsv:fIdnTableId> IDN Table Identifier used for IDN domain names with type="token".

<rdeCsv:fPositiveIntegerType> General positive integer field with type="positiveInteger".

<rdeCsv:fUrl> Contains the URL of an object like a registrar object with type="anyURI".

<rdeCsv:fCustom> Custom field with name attribute that defines the custom field name" with type="token".

4.8. Internationalized and Localized Elements

Some elements MAY be provided in either internationalized form ("int") or provided in localized form ("loc"). Those elements use a field value or "isLoc" attribute to specify the form used. If an "isLoc" attribute is used, a value of "true" indicates the use of the localized form and a value of "false" indicates the use of the internationalized form. This MAY override the form specified for a parent element. A value of "int" is used to indicate the internationalized form and a value of "loc" is used to indicate the localized form. When the internationalized form ("int") is provided, the field value MUST be represented in a subset of UTF-8 that can be represented in the 7-bit US-ASCII character set. When the localized form ("loc") is provided, the field value MAY be represented in unrestricted UTF-8.

The field elements below of the "registrar" <rdeCsv:csv">
<rdeCsv:fields> element specify the internationalized form with the
isLoc="false" attribute.

```
...
<csvRegistrar:contents>
...
  <rdeCsv:csv name="registrar" sep=", ">
    <rdeCsv:fields>
      <csvRegistrar:fId/>
      <rdeCsv:fRoid/>
      <csvRegistrar:fName isLoc="false"/>
      <csvRegistrar:fGurid/>
      <csvRegistrar:fStatus/>
      <csvContact:fStreet isLoc="false" index="0"/>
      <csvContact:fStreet isLoc="false" index="1"/>
      <csvContact:fStreet isLoc="false" index="2"/>
      <csvContact:fCity isLoc="false" />
      <csvContact:fSp isLoc="false" />
      <csvContact:fPc isLoc="false" />
      <csvContact:fCc isLoc="false" />
      <csvContact:fVoice/>
      <csvContact:fVoiceExt/>
      <csvContact:fFax/>
      <csvContact:fFaxExt/>
      <csvContact:fEmail/>
      <rdeCsv:fUrl/>
      <csvRegistrar:fWhoisUrl/>
      <rdeCsv:fCrRr/>
      <rdeCsv:fCrID/>
      <rdeCsv:fCrDate/>
      <rdeCsv:fUpRr/>
      <rdeCsv:fUpID/>
      <rdeCsv:fUpDate/>
    </rdeCsv:fields>
    <rdeCsv:files>
      <rdeCsv:file
        cksum="306178BB">
        registrar-YYYYMMDD.csv
      </rdeCsv:file>
    </rdeCsv:files>
  </rdeCsv:csv>
...
</csvRegistrar:contents>
...
```

The following is an example of using the `<csvContact:fPostalType>` field value to define the internationalized or localized form of the remainder of the "contactPostal" field values.

```
...
<csvContact:contents>
...
  <rdeCsv:csv name="contactPostal">
    <rdeCsv:fields>
      <csvContact:fId parent="true"/>
      <csvContact:fPostalType/>
      <csvContact:fName/>
      <csvContact:fOrg/>
      <csvContact:fStreet index="0"/>
      <csvContact:fStreet index="1"/>
      <csvContact:fStreet index="2"/>
      <csvContact:fCity/>
      <csvContact:fSp/>
      <csvContact:fPc/>
      <csvContact:fCc/>
    </rdeCsv:fields>
    <rdeCsv:files>
      <rdeCsv:file
        cksum="02CC2504">
        contactPostal-YYYYMMDD.csv
      </rdeCsv:file>
    </rdeCsv:files>
  </rdeCsv:csv>
...
</csvContact:contents>
...
```

5. Object Description

This section describes the base objects supported by this specification:

5.1. Domain Name Object

The domain name object is based on the EPP domain name mapping specified in [RFC5731]. The domain name object supports both the XML Model and the CSV Model, defined in the Models (Section 2) section. The elements used for both models are defined in the following sections.

5.1.1.1. XML Model

There are two elements used in the data escrow of the domain name objects for the XML model including the `<rdeDomain:domain>`, under the `<rde:contents>` element, and the `<rdeDomain:delete>` element, under the `<rde:deletes>` element.

5.1.1.1.1. `<rdeDomain: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:

- o A `<name>` element that contains the fully qualified name of the domain name object. If the domain name is an IDN, the ASCII Compatible Encoding (ACE) MUST be used.
- o A `<roid>` element that contains the repository object identifier assigned to the domain name object when it was created.
- o An OPTIONAL `<uName>` element that contains the fully qualified name of the domain name in Unicode character set. It MUST be provided if available.
- o 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.
- o 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.
- o One or more `<status>` elements that contain the current status descriptors associated with the domain name.
- o 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].

- o 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.
- o Zero or more OPTIONAL <contact> elements that contain identifiers for the human or organizational social information objects associated with the domain name object.
- o 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.
- o A <clID> element that contains the identifier of the sponsoring registrar.
- o An OPTIONAL <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.
- o 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.
- o 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.
- o 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.
- o 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.
- o 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].

- o 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 transferred.
- o 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>
...
```

5.1.1.2. <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>
...
```

5.1.2. CSV Model

For the CSV Model of the domain name object, the <csvDomain:contents> child element of the <rde:contents> element is used to hold the new or updated domain name objects for the deposit. The <csvDomain:deletes> child element of the <rde:deletes> element is used to hold the deleted or purged domain name objects for the deposit. Both the <csvDomain:contents> and <csvDomain:deletes>

elements contain one or more `<rdeCsv:csv>` elements with a set of named CSV file definitions using the `<rdeCsv:csv>` "name" attribute.

Differential and incremental deposits are based on changes to the domain name objects. The updated domain name object data under the `<csvDomain:contents>` element is a cascade replace down all of the domain name CSV files starting with the parent "domain" CSV File Definition (Section 5.1.2.1.1). The child CSV file definitions include a `<csvDomain:fName parent="true">` field. All the child CSV file definition data for the domain name objects in the parent "domain" CSV File Definition (Section 5.1.2.1.1) MUST first be deleted and then set using the data in the child CSV files. The deleted domain name object data under the `<csvDomain:deletes>` element is a cascade delete starting from the "domain" Deletes CSV File Definition (Section 5.1.2.2.1).

5.1.2.1. `<csvDomain:contents>`

The `<csvDomain:contents>` is used to hold the new or updated domain name object information for the deposit. The `<csvDomain:contents>` is split into separate CSV file definitions using named `<rdeCsv:csv>` elements with the "name" attribute. The following sections include the supported domain name CSV file definitions:

5.1.2.1.1. "domain" CSV File Definition

The "domain" CSV File Definition defines the fields and CSV file references used for the parent domain name object records. All the other domain name CSV file definitions are child CSV files based on the inclusion of the `<csvDomain:fName parent="true">` field.

The following "csvDomain" field elements MUST be used in the "domain" `<rdeCsv:csv>` `<rdeCsv:fields>` element:

`<csvDomain:fName>` Domain name field with `type="eppcom:labelType"` and `isRequired="true"`.

The following "csvDomain" field elements MAY be used in the "domain" `<rdeCsv:csv>` `<rdeCsv:fields>` element:

`<csvDomain:fOriginalName>` Fully qualified name of the original IDN domain name object related to the variant domain name object with `type="eppcom:labelType"`.

The following "rdeCsv" fields, defined in section CSV common field elements (Section 4.7.2), MUST be used in the "domain" `<rdeCsv:csv>` `<rdeCsv:fields>` element:

<rdeCsv:fROID> Registry Object Identifier (ROID) for the domain name object with isRequired="true".

<rdeCsv:fClID> Identifier of client that sponsors the domain name object with isRequired="true".

<rdeCsv:fCrDate> Created date and time of the domain name object with isRequired="true".

<rdeCsv:fUpRr> Identifier of the registrar, defined in Section 5.4, of the client that updated the object.

<rdeCsv:fUpDate> Date and time of the last update to the domain name object.

<rdeCsv:fExDate> Expiration date and time for the domain name object. The attribute isRequired MUST equal "true".

The following "rdeCsv" fields, defined in section CSV common field elements (Section 4.7.2), MAY be used in the "domain" <rdeCsv:csv> <rdeCsv:fields> element:

<rdeCsv:fCrRr> Identifier of the registrar, defined in Section 5.4, of the client that created the object.

<rdeCsv:fUName> UTF8 encoded domain name for the <csvDomain:fName> field element.

<rdeCsv:fIdnTableId> IDN Table Identifier used for the IDN domain name object that MUST match a <rdeCsv:fIdnTableId> field element in the "idnLanguage" CSV files, as defined in Section 5.5.2.

<rdeCsv:fRegistrant> Registrant contact identifier for the domain name object.

<rdeCsv:fCrID> Identifier of client that created the domain name object.

<rdeCsv:fUpID> Identifier of the client that last updated the domain name object.

<rdeCsv:fTrDate> Date and time of the last transfer for the domain name object.

Example of a "domain" <csvDomain:contents> <rdeCsv:csv> element.

```
...
<csvDomain:contents>
...
  <rdeCsv:csv name="domain">
    <rdeCsv:fields>
      <csvDomain:fName/>
      <rdeCsv:fRoid/>
      <rdeCsv:fIdnTableId/>
      <csvDomain:fOriginalName/>
      <rdeCsv:fRegistrant/>
      <rdeCsv:fClID/>
      <rdeCsv:fCrRr/>
      <rdeCsv:fCrID/>
      <rdeCsv:fCrDate/>
      <rdeCsv:fUpRr/>
      <rdeCsv:fUpID/>
      <rdeCsv:fUpDate/>
      <rdeCsv:fExDate isRequired="true"/>
    </rdeCsv:fields>
    <rdeCsv:files>
      <rdeCsv:file
        cksum="5E403BD6">
        domain-YYYYMMDD.csv
      </rdeCsv:file>
    </rdeCsv:files>
  </rdeCsv:csv>
...
</csvDomain:contents>
...
```

Example of the corresponding domain-YYYYMMDD.csv file. The file contains four records (two active ASCII domains, original IDN with LANG-1 language rules, and variant IDN with LANG-1 language rules).

```
domain1.test,Ddomain1-TEST,,,registrantid,registrarX,registrarX,
clientY,2009-04-03T22:00:00.0Z,registrarX,clientY,
2009-12-03T09:05:00.0Z,2015-04-03T22:00:00.0Z
domain2.test,Ddomain2-TEST,,,registrantid,registrarX,registrarX,
clientY,1999-04-03T22:00:00.0Z,registrarX,clientY,
2009-12-03T09:05:00.0Z,2015-04-03T22:00:00.0Z
xn--abc123.test,Dxnabc123-TEST,LANG-1,,,registrantid,registrarX,
registrarX,clientY,2009-04-03T22:00:00.0Z,registrarX,clientY,
2009-12-03T09:05:00.0Z,2015-04-03T22:00:00.0Z
xn--abc321.test,Dxnabc321-TEST,LANG-1,xn--abc123.test,
registrantid,registrarX,registrarX,clientY,2009-04-03T22:00:00.0Z,
registrarX,clientY,2009-12-03T09:05:00.0Z,2015-04-03T22:00:00.0Z
```

5.1.2.1.2. "domainContacts" CSV File Definition

The "domainContacts" CSV File Definition defines the fields and CSV file references used for the domain name object link records to contact objects, as described in Contact Object (Section 5.3).

The following "csvDomain" field elements MUST be used in the "domainContacts" <rdeCsv:csv> <rdeCsv:fields> element:

<csvDomain:fContactType> The contact type for the contact object link with type="domain:contactAttrType" and isRequired="true". The supported contact type values include "admin" for the administration contact, "billing" for the billing contact, and "tech" for the technical contact.

The following "csvDomain" field elements, defined for the "domain" CSV File Definition (Section 5.1.2.1.1), MUST be used in the "domainContacts" <rdeCsv:csv> <rdeCsv:fields> element:

<csvDomain:fName> The name of the domain object that is linked to the contact object with isRequired="true".

The following "csvContact" fields, defined for the "contact" CSV File Definition (Section 5.3.2.1.1), MUST be used in the "domainContacts" <rdeCsv:csv> <rdeCsv:fields> element:

<csvContact:fId> The server-unique contact identifier with isRequired="true".

Example of a "domainContacts" <csvDomain:contents> <rdeCsv:csv> element.

```
...
<csvDomain:contents>
...
  <rdeCsv:csv name="domainContacts">
    <rdeCsv:fields>
      <csvDomain:fName parent="true"/>
      <csvContact:fId parent="true"/>
      <csvDomain:fContactType/>
    </rdeCsv:fields>
    <rdeCsv:files>
      <rdeCsv:file
        cksum="6B976A6C">
        domainContacts-YYYYMMDD.csv
      </rdeCsv:file>
    </rdeCsv:files>
  </rdeCsv:csv>
...
</csvDomain:contents>
...
```

Example of the corresponding domainContacts-YYYYMMDD.csv file. The file contains an admin, tech, and billing contact for the four domain names domain1.test, domain2.test, xn--abc123.test and xn--abc321.test.

```
domain1.test,domain1admin,admin
domain1.test,domain1tech,tech
domain1.test,domain1billing,billing
domain2.test,domain2admin,admin
domain2.test,domain2tech,tech
domain2.test,domain2billing,billing
xn--abc123.test,xnabc123admin,admin
xn--abc123.test,xnabc123tech,tech
xn--abc123.test,xnabc123billing,billing
xn--abc321.test,xnabc123admin,admin
xn--abc321.test,xnabc123tech,tech
xn--abc321.test,xnabc123billing,billing
```

5.1.2.1.3. "domainStatuses" CSV File Definition

The "domainStatuses" CSV File Definition defines the fields and CSV file references used for the domain name object statuses.

The following "csvDomain" field elements MUST be used in the "domainStatuses" <rdeCsv:csv> <rdeCsv:fields> element:

<csvDomain:fStatus> The status of the domain name with
type="domain:statusValueType" and isRequired="true".

<csvDomain:fRgpStatus> The Registry Grace Period (RGP) status, as a
sub-status of the <csvDomain:fStatus> "pendingDelete" status
value, with type="rgp:statusValueType" as defined in [RFC3915].

The following "csvDomain" fields, defined for the "domain" CSV File
Definition (Section 5.1.2.1.1), MUST be used in the "domainStatuses"
<rdeCsv:csv> <rdeCsv:fields> element:

<csvDomain:fName> Domain name of status with isRequired="true".

The following "rdeCsv" fields, defined in section CSV common field
elements (Section 4.7.2), MAY be used in the "domainStatuses"
<rdeCsv:csv> <rdeCsv:fields> element:

<rdeCsv:fStatusDescription> Domain object status description which
is free form text describing the rationale for the status.

<rdeCsv:fLang> Language of the <rdeCsv:fStatusDescription> field.

Example of a "domainStatuses" <csvDomain:contents> <rdeCsv:csv>
element.

```
...
<csvDomain:contents>
...
  <rdeCsv:csv name="domainStatuses">
    <rdeCsv:fields>
      <csvDomain:fName parent="true"/>
      <csvDomain:fStatus/>
      <rdeCsv:fStatusDescription/>
      <rdeCsv:fLang/>
      <csvDomain:fRgpStatus/>
    </rdeCsv:fields>
    <rdeCsv:files>
      <rdeCsv:file
        cksum="98D139A3">
        domainStatuses-YYYYMMDD.csv
      </rdeCsv:file>
    </rdeCsv:files>
  </rdeCsv:csv>
...
</csvDomain:contents>
...
```


Example of the corresponding domainStatuses-YYYYMMDD.csv file. The file contains the statuses for the four domain names domain1.test, domain2.test, xn--abc123.test and xn--abc321.test.

```
domain1.test,clientUpdateProhibited,"Disallow update",
en,
domain1.test,clientDeleteProhibited,"Disallow delete",
en,
domain2.test,ok,,,
xn--abc123.test,ok,,,
xn--abc321.test,ok,,,
```

5.1.2.1.4. "domainNameServers" CSV File Definition

The "domainNameServers" CSV File Definition defines the fields and CSV file references used for the domain name delegated hosts (name servers). The "domainNameServers" CSV files define the relationship between a domain name object and a delegated host. The "domainNameServers" CSV File is used to support the <domain:hostObj> model, defined in [RFC5731].

The following "csvDomain" fields, defined for the "domain" CSV File Definition (Section 5.1.2.1.1), MUST be used in the "domainNameServers" <rdeCsv:csv> <rdeCsv:fields> element:

<csvDomain:fName> Domain name using the delegated host with host <rdeCsv:fRoid> and isRequired="true".

The following "rdeCsv" fields, defined in section CSV common field elements (Section 4.7.2), MUST be used in the "domainNameServers" <rdeCsv:csv> <rdeCsv:fields> element:

<rdeCsv:fRoid> Registry Object Identifier (ROID) used to uniquely reference a delegated host record with isRequired="true".

Example of a "domainNameServers" <csvDomain:contents> <rdeCsv:csv> element.

```
...
<csvDomain:contents>
...
  <rdeCsv:csv name="domainNameServers">
    <rdeCsv:fields>
      <csvDomain:fName parent="true"/>
      <rdeCsv:fRoid parent="true"/>
    </rdeCsv:fields>
    <rdeCsv:files>
      <rdeCsv:file
        cksum="8FE6E9E1">
        domainNameServers-YYYYMMDD.csv
      </rdeCsv:file>
    </rdeCsv:files>
  </rdeCsv:csv>
...
</csvDomain:contents>
...
```

Example of the corresponding domainNameServers-YYYYMMDD.csv file. The file contains the delegated hosts (name servers) for the four domain names domain1.test, domain2.test, xn--abc123.test and xn--abc321.test referenced via the <rdeCsv:fRoid> field element.

```
domain1.test,Hns1_domain1_test-TEST
domain1.test,Hns2_domain1_test-TEST
domain2.test,Hns1_domain2_test-TEST
domain2.test,Hns2_domain2_test-TEST
xn--abc123.test,Hns1_example_test-TEST
xn--abc123.test,Hns2_example_test-TEST
xn--abc321.test,Hns1_example_test-TEST
xn--abc321.test,Hns2_example_test-TEST
```

5.1.2.1.5. "domainNameServersAddresses" CSV File Definition

The "domainNameServersAddresses" CSV File Definition defines the fields and CSV file references used for supporting the host as domain attributes model.

The following "csvDomain" fields, defined for the "domain" CSV File Definition (Section 5.1.2.1.1), MUST be used in the "domainNameServersAddresses" <rdeCsv:csv> <rdeCsv:fields> element:

```
<csvDomain:fName> Domain name using the delegated host with host
  <csvHost:fName> and isRequired="true".
```

The following "rdeCsv" fields, defined in section Host CSV model elements (Section 5.2.2), MUST be used in the "domainNameServersAddresses" <rdeCsv:csv> <rdeCsv:fields> element:

<csvHost:fName> Host name field with type="eppcom:labelType" and isRequired="true".

The following "csvHost" fields, defined in section Host CSV model elements (Section 5.2.2), MAY be used in the "domainNameServersAddresses" <rdeCsv:csv> <rdeCsv:fields> element:

<csvHost:fAddr> IP addresses associated with the host object with type="host:addrStringType".

<csvHost:fAddrVersion> IP addresses version associated with the host object with type="host:ipType". "host:ipType" has the enumerated values of "v4" or "v6".

Example of a "domainNameServersAddresses" <csvDomain:contents> <rdeCsv:csv> element.

```
...
<csvDomain:contents>
...
  <rdeCsv:csv name="domainNameServersAddresses">
    <rdeCsv:fields>
      <csvDomain:fName parent="true"/>
      <csvHost:fName/>
      <csvHost:fAddr/>
      <csvHost:fAddrVersion/>
    </rdeCsv:fields>
    <rdeCsv:files>
      <rdeCsv:file
        cksum="D3B77438">
        domainNameServersAddresses-YYYYMMDD.csv
      </rdeCsv:file>
    </rdeCsv:files>
  </rdeCsv:csv>
...
</csvDomain:contents>
...
```

Example of the corresponding domainNameServersAddresses-YYYYMMDD.csv file. The file contains the delegated hosts (name servers) for the four domain names domain1.test, domain2.test, xn--abc123.test and xn--abc321.test.

```
domain1.test,ns1.domain1.test,192.0.2.1,v4
domain1.test,ns2.domain1.test,2001:DB8::1,v6
domain2.test,ns1.domain2.test2,192.0.2.2,v4
domain2.test,ns2.domain2.test2,2001:DB8::2,v6
xn--abc123.test,ns1.example.example,,
xn--abc123.test,ns2.example.example,,
xn--abc321.test,ns1.example.example,,
xn--abc321.test,ns2.example.example,,
```

5.1.2.1.6. "dnssec" CSV File Definition

The "dnssec" CSV File Definition defines the fields and CSV file references used for the domain name object DNSSEC records (DS or Key Data).

The following "csvDomain" field elements MUST be used in the "dnssec" <rdeCsv:csv> <rdeCsv:fields> element when the DS Data Interface per [RFC5910] is used:

<csvDomain:fKeyTag> Contains the DS key tag value per [RFC5910] with type="unsignedShort" and isRequired="true".

<csvDomain:fDsAlg> Contains the DS algorithm value per [RFC5910] with type="unsignedByte" and isRequired="true".

<csvDomain:fDigestType> Contains the DS digest type value per [RFC5910] with type="unsignedByte" and isRequired="true".

<csvDomain:fDigest> Contains the DS digest value per [RFC5910] with type="hexBinary" and isRequired="true".

The following "csvDomain" field elements MUST be used in the "dnssec" <rdeCsv:csv> <rdeCsv:fields> element when the Key Data Interface per [RFC5910] is used and MAY be used in the "dnssec" <rdeCsv:csv> <rdeCsv:fields> element when the DS Data Interface per [RFC5910] is used:

<csvDomain:fFlags> Contains the flags field value per [RFC5910] with type="unsignedShort" and isRequired="true".

<csvDomain:fProtocol> Contains the Key protocol value per [RFC5910] with type="unsignedByte" and isRequired="true".

<csvDomain:fKeyAlg> Contains the Key algorithm value per [RFC5910] with type="unsignedByte" and isRequired="true".

<csvDomain:fPubKey> Contains the public key value per [RFC5910] with type="secDNS:keyType" and isRequired="true".

The following "csvDomain" field elements MAY be used in the "dnssec" <rdeCsv:csv> <rdeCsv:fields> element:

<csvDomain:fMaxSigLife> Indicates a child's preference for the number of seconds after signature generation when the parent's signature on the DS information provided by the child will expire with type="secDNS:maxSigLifeType" defined in [RFC5910].

The following "domain" fields, defined for the "domain" CSV File Definition (Section 5.1.2.1.1), MUST be used in the "dnssec" <rdeCsv:csv> <rdeCsv:fields> element:

<csvDomain:fName> Domain name of the domain object associated with the DNSSEC record and isRequired="true".

Example of a "dnssec" <csvDomain:contents> <rdeCsv:csv> element with the DS Data Interface of [RFC5910]:

```
<csvDomain:contents>
...
<rdeCsv:csv name="dnssec">
  <rdeCsv:fields>
    <csvDomain:fName parent="true"/>
    <csvDomain:fMaxSigLife/>
    <csvDomain:fKeyTag/>
    <csvDomain:fDsAlg/>
    <csvDomain:fDigestType/>
    <csvDomain:fDigest/>
  </rdeCsv:fields>
  <rdeCsv:files>
    <rdeCsv:file
      cksum="10ED6C42">
      dnssec-ds-YYYYMMDD.csv
    </rdeCsv:file>
  </rdeCsv:files>
</rdeCsv:csv>
...
</csvDomain:contents>
...
```

Example of the corresponding dnssec-ds-YYYYMMDD.csv file. The file contains two DS records for domain1.test.

```
domain1.test,604800,12345,3,1,49FD46E6C4B45C55D4AC
domain1.test,604800,12346,3,1,38EC35D5B3A34B44C39B
```

Example of a "dnssec" <csvDomain:contents> <rdeCsv:csv> element with the Key Data Interface of [RFC5910]:

```
<csvDomain:contents>
...
  <rdeCsv:csv name="dnssec">
    <rdeCsv:fields>
      <csvDomain:fName/>
      <csvDomain:fMaxSigLife/>
      <csvDomain:fFlags/>
      <csvDomain:fProtocol/>
      <csvDomain:fKeyAlg/>
      <csvDomain:fPubKey/>
    </rdeCsv:fields>
    <rdeCsv:files>
      <rdeCsv:file
        cksum="183C3F79">
        dnssec-key-YYYYMMDD.csv
      </rdeCsv:file>
    </rdeCsv:files>
  </rdeCsv:csv>
...
</csvDomain:contents>
...
```

Example of the corresponding dnssec-key-YYYYMMDD.csv file. The file contains two key records for domain1.test.

```
domain1.test,604800,257,3,1,AQPJ///4Q==
domain1.test,604800,257,3,1,AQPJ///4QQQ
```

5.1.2.1.7. "domainTransfer" CSV File Definition

The "domainTransfer" CSV File Definition defines the fields and CSV file references used for the domain name object pending and completed transfer records. No additional field elements were added for use in the "domainTransfer" <rdeCsv:csv> <rdeCsv:fields> element.

The following "rdeCsv" fields, defined in section CSV common field elements (Section 4.7.2), MUST be used in the "domainTransfer" <rdeCsv:csv> <rdeCsv:fields> element:

<rdeCsv:fTrStatus> State of the most recent transfer request with isRequired="true".

<rdeCsv:fReRr> Identifier of the registrar, defined in Section 5.4, of the client that requested the transfer with isRequired="true".

<rdeCsv:fReDate> Date and time that the transfer was requested with isRequired="true".

<rdeCsv:fAcRr> Identifier of the registrar, defined in Section 5.4, of the client that should take or took action with isRequired="true".

<rdeCsv:fAcDate> Date and time that the transfer action should be taken or has been taken with isRequired="true".

The following "rdeCsv" fields, defined in section CSV common field elements (Section 4.7.2), MAY be used in the "domainTransfer" <rdeCsv:csv> <rdeCsv:fields> element:

<rdeCsv:fExDate> Expiration date if the transfer command caused or causes a change in the validity period.

<rdeCsv:fReID> Identifier of client that requested the transfer.

<rdeCsv:fAcID> Identifier of client that should take or took action for transfer.

The following "csvDomain" fields, defined for the "domain" CSV File Definition (Section 5.1.2.1.1), MUST be used in the "domainTransfer" <rdeCsv:csv> <rdeCsv:fields> element:

<csvDomain:fName> Domain name of the domain object involved in the transfer with isRequired="true".

Example of a "domainTransfer" <csvDomain:contents> <rdeCsv:csv> element.

```
...
<csvDomain:contents>
...
  <rdeCsv:csv name="domainTransfer">
    <rdeCsv:fields>
      <csvDomain:fName parent="true"/>
      <rdeCsv:fTrStatus/>
      <rdeCsv:fReRr/>
      <rdeCsv:fReID/>
      <rdeCsv:fReDate/>
      <rdeCsv:fAcRr/>
      <rdeCsv:fAcID/>
      <rdeCsv:fAcDate/>
      <rdeCsv:fExDate/>
    </rdeCsv:fields>
    <rdeCsv:files>
      <rdeCsv:file
        cksum="2E5A9ACD">
        domainTransfer-YYYYMMDD.csv
      </rdeCsv:file>
    </rdeCsv:files>
  </rdeCsv:csv>
...
</csvDomain:contents>
...
```

Example of the corresponding domainTransfer-YYYYMMDD.csv file. The file contains one domain transfer record with a pending status.

```
domain1.test,pending,registrarX,clientY,
2011-03-08T19:38:00.0Z,registrarY,,2011-03-13T23:59:59.0Z,
2016-04-03T22:00:00.0Z
```

5.1.2.2. <csvDomain:deletes>

The <csvDomain:deletes> is used to hold the deleted domain name objects in a differential or incremental deposit. All the domain name object data is deleted as part of a cascade delete. The <csvDomain:deletes> is split into separate CSV file definitions using named <rdeCsv:csv> elements with the "name" attribute. The following section defines the supported domain name deletes CSV file definition.

5.1.2.2.1. "domain" Deletes CSV File Definition

The following "csvDomain" field elements MUST be used in the deletes "domain" <rdeCsv:csv> <rdeCsv:fields> element:

<csvDomain:fName> Domain name field with type="eppcom:labelType" and isRequired="true".

Example of a "domain" <csvDomain:deletes> <rdeCsv:csv> element:

```
...
<csvDomain:deletes>
...
  <rdeCsv:csv name="domain">
    <rdeCsv:fields>
      <csvDomain:fName/>
    </rdeCsv:fields>
    <rdeCsv:files>
      <rdeCsv:file
        cksum="A06D8194">
        domain-delete-YYYYMMDD.csv
      </rdeCsv:file>
    </rdeCsv:files>
  </rdeCsv:csv>
...
</csvDomain:deletes>
...
```

Example of the corresponding domain-delete-YYYYMMDD.csv file. The file contains two domain name records.

```
domain1.test
domain2.test
```

5.2. Host Object

The host object is based on the EPP host name mapping in [RFC5732]. The host object supports both the XML Model and the CSV Model, defined in Models (Section 2) section. The elements used for both models are defined in the following sections. Both the <csvHost:contents> and <csvHost:deletes> elements contain one or more <rdeCsv:csv> elements with a set of named CSV file definitions using the <rdeCsv:csv> "name" attribute.

5.2.1. XML Model

There are two elements used in the data escrow of the host objects for the XML model including the `<rdeHost:host>`, under the `<rdeHost:contents>` element, and the `<rdeHost:delete>` element, under the `<rde:deletes>` element.

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

5.2.1.1. `<rdeHost:host>` element

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:

- o A `<name>` element that contains the fully qualified name of the host object.
- o A `<roid>` element that contains the repository object identifier assigned to the host object when the object was created.
- o One or more `<status>` elements that describe the status of the host object.
- o Zero or more `<addr>` elements that contain the IP addresses associated with the host object.
- o A `<clID>` element that contains the identifier of the sponsoring registrar.
- o 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.
- o A `<crDate>` element that contains the date and time of host-object creation.
- o 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.

- o 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.
- o 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 transferred.

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>
...
```

5.2.1.2. <rdeHost:delete> object

The <rdeHost:delete> element contains the fully qualified domain name of a host that was deleted. The <rdeHost:delete> element also supports host removal based on roid to support SRS systems in which different hosts with the same fully qualified domain name are active at the same time.

Example of <rdeHost:delete> object:

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

5.2.2. CSV Model

For the CSV Model of the host object, the `<csvHost:contents>` child element of the `<rde:contents>` element is used to hold the new or updated host objects for the deposit. The `<csvHost:deletes>` child element of the `<rde:deletes>` element is used to hold the deleted or purged host objects for the deposit.

Differential and incremental deposits are based on changes to the host objects. The updated host object data under the `<csvHost:contents>` element is a cascade replace down all of the host CSV files starting with the parent "host" CSV File Definition (Section 5.2.2.1.1). The child CSV file definitions include a `<rdeCsv:fRoid parent="true">` field. All the child CSV file definition data for the host objects in the parent "host" CSV File Definition (Section 5.2.2.1.1) MUST first be deleted and then set using the data in the child CSV files. The deleted host object data under the `<csvHost:deletes>` element is a cascade delete starting from the "host" Deletes CSV File Definition (Section 5.2.2.2.1).

5.2.2.1. `<csvHost:contents>`

The `<csvHost:contents>` is used to hold the new or updated host object information for the deposit. The `<csvHost:contents>` is split into separate CSV file definitions using named `<rdeCsv:csv>` elements with the "name" attribute. The following sections include the supported host CSV file definitions.

5.2.2.1.1. "host" CSV File Definition

The "host" CSV File Definition defines the fields and CSV file references used for the host object records.

The following "csvHost" field elements MUST be used in the "host" `<rdeCsv:csv>` `<rdeCsv:fields>` element:

`<csvHost:fName>` Host name field with `type="eppcom:labelType"` and `isRequired="true"`.

The following "rdeCsv" fields, defined in section CSV common field elements (Section 4.7.2), MUST be used in the "host" `<rdeCsv:csv>` `<rdeCsv:fields>` element:

`<rdeCsv:fRoid>` Repository Object Identifier (ROID) assigned to the host object with `isRequired="true"`.

The following "rdeCsv" fields, defined in section CSV common field elements (Section 4.7.2), MAY be used in the "host" <rdeCsv:csv> <rdeCsv:fields> element:

<rdeCsv:fClID> Identifier of the sponsoring client with
isRequired="true".

<rdeCsv:fCrRr> Identifier of the registrar, defined in Section 5.4,
of the client that created the object with isRequired="true".

<rdeCsv:fCrID> Identifier of client that created the host object.

<rdeCsv:fCrDate> Date and time that the host object was created with
isRequired="true".

<rdeCsv:fUpRr> Identifier of the registrar, defined in Section 5.4,
of the client that updated the object.

<rdeCsv:fUpID> Identifier of client that last updated the host
object.

<rdeCsv:fUpDate> Date and time that the host object was last
updated.

<rdeCsv:fTrDate> Date and time that the host was last transferred.

Example of a "host" <csvHost:contents> <rdeCsv:csv> element.

```
...
<csvHost:contents>
...
  <rdeCsv:csv name="host">
    <rdeCsv:fields>
      <csvHost:fName/>
      <rdeCsv:fRoid/>
      <rdeCsv:fClID/>
      <rdeCsv:fCrRr/>
      <rdeCsv:fCrID/>
      <rdeCsv:fCrDate/>
      <rdeCsv:fUpRr/>
      <rdeCsv:fUpID/>
      <rdeCsv:fUpDate/>
      <rdeCsv:fTrDate/>
    </rdeCsv:fields>
    <rdeCsv:files>
      <rdeCsv:file
        cksum="6F1E58E5">
        host-YYYYMMDD.csv
      </rdeCsv:file>
    </rdeCsv:files>
  </rdeCsv:csv>
...
</csvHost:contents>
...
```

Example of the corresponding host-YYYYMMDD.csv file. The file contains six host records with four being internal hosts and two being external hosts.

```
ns1.domain1.test,Hns1_example_test-TEST,registrarX,registrarX,
clientY,1999-05-08T12:10:00.0Z,registrarX,
clientY,2009-10-03T09:34:00.0Z,2007-01-08T09:19:00.0Z
ns2.domain1.test,Hns2_domain1_test-TEST,registrarX,registrarX,
clientY,1999-05-08T12:10:00.0Z,registrarX,
clientY,2009-10-03T09:34:00.0Z,2007-01-08T09:19:00.0Z
ns1.domain2.test,Hns1_domain2_test-TEST,registrarX,registrarX,
clientY,1999-05-08T12:10:00.0Z,registrarX,
clientY,2009-10-03T09:34:00.0Z,2007-01-08T09:19:00.0Z
ns2.domain2.test,Hns2_domain2_test-TEST,registrarX,registrarX,
clientY,1999-05-08T12:10:00.0Z,registrarX,
clientY,2009-10-03T09:34:00.0Z,2007-01-08T09:19:00.0Z
ns1.example.example,Hns1_example_test-TEST,registrarX,registrarX,
clientY,1999-05-08T12:10:00.0Z,registrarX,
clientY,2009-10-03T09:34:00.0Z,2007-01-08T09:19:00.0Z
ns2.example.example,Hns2_example_test-TEST,registrarX,registrarX,
clientY,1999-05-08T12:10:00.0Z,registrarX,
clientY,2009-10-03T09:34:00.0Z,2007-01-08T09:19:00.0Z
```

5.2.2.1.2. "hostStatuses" CSV File Definition

The "hostStatuses" CSV File Definition defines the fields and CSV file references used for the host object statuses.

The following "csvHost" fields, defined for the "host" CSV File Definition (Section 5.2.2.1.1), MUST be used in the "hostStatuses" <rdeCsv:csv> <rdeCsv:fields> element:

```
<csvHost:fStatus> The status of the host with
  type="host:statusValueType" and isRequired="true"./>.
```

The following "rdeCsv" fields, defined in section CSV common field elements (Section 4.7.2), MUST be used in the "hostStatuses" <rdeCsv:csv> <rdeCsv:fields> element:

```
<rdeCsv:fRoid> Host object Registry Object IDentifier (ROID)
  assigned to the host object with isRequired="true".
```

The following "rdeCsv" fields, defined in section CSV common field elements (Section 4.7.2), MAY be used in the "hostStatuses" <rdeCsv:csv> <rdeCsv:fields> element:

<rdeCsv:fStatusDescription> Host object status description which is free form text describing the rationale for the status. The attribute isRequired MUST equal "true".

<rdeCsv:fLang> Language of the <rdeCsv:fStatusDescription> field.

Example of a "hostStatuses" <csvHost:contents> <rdeCsv:csv> element.

```
...
<csvHost:contents>
...
  <rdeCsv:csv name="hostStatuses">
    <rdeCsv:fields>
      <rdeCsv:fRoid parent="true"/>
      <csvHost:fStatus/>
      <rdeCsv:fStatusDescription/>
      <rdeCsv:fLang/>
    </rdeCsv:fields>
    <rdeCsv:files>
      <rdeCsv:file
        cksum="0DAE0583">
        hostStatuses-YYYYMMDD.csv
      </rdeCsv:file>
    </rdeCsv:files>
  </rdeCsv:csv>
...
</csvHost:contents>
...
```

Example of the corresponding hostStatuses-YYYYMMDD.csv file. The file contains the statuses for the six host names ns1.domain1.test, ns2.domain1.test, ns1.domain2.test, ns2.domain2.test, ns1.example.example and ns2.example.example.

```
Hns1_domain1_test-TEST,ok,,
Hns2_domain1_test-TEST,ok,,
Hns1_domain2_test-TEST,ok,,
Hns2_domain2_test-TEST,ok,,
Hns1_example_test-TEST,ok,,
Hns2_example_test-TEST,ok,,
```

5.2.2.1.3. "hostAddresses" CSV File Definition

The "hostAddresses" CSV File Definition defines the fields and CSV file references used for the host object IP addresses.

The following "csvHost" field elements MUST be used in the "hostAddresses" <rdeCsv:csv> <rdeCsv:fields> element:

<csvHost:fAddr> IP addresses associated with the host object with type="host:addrStringType". The attribute isRequired MUST equal "true".

<csvHost:fAddrVersion> IP addresses version associated with the host object with type="host:ipType". "host:ipType" has the enumerated values of "v4" or "v6". The attribute isRequired MUST equal "true".

The following "rdeCsv" fields, defined in section CSV common field elements (Section 4.7.2), MUST be used in the "hostAddresses"

<rdeCsv:csv> <rdeCsv:fields> element:

<rdeCsv:fRoid> Host object Registry Object Identifier (ROID) assigned to the host object.

Example of a "hostAddresses" <csvHost:contents> <rdeCsv:csv> element.

```
...
<csvHost:contents>
...
  <rdeCsv:csv name="hostAddresses">
    <rdeCsv:fields>
      <rdeCsv:fRoid parent="true"/>
      <csvHost:fAddr isRequired="true"/>
      <csvHost:fAddrVersion isRequired="true"/>
    </rdeCsv:fields>
    <rdeCsv:files>
      <rdeCsv:file
        cksum="28B194B0">
        hostAddresses-YYYYMMDD.csv
      </rdeCsv:file>
    </rdeCsv:files>
  </rdeCsv:csv>
...
</csvHost:contents>
...
```

Example of the corresponding hostAddressesObj-YYYYMMDD.csv file. The file contains the IP addresses for the host names ns1.domain1.test, ns2.domain1.test, ns1.domain2.test and ns2.domain2.test.

```
Hns1_domain1_test-TEST,192.0.2.1,v4
Hns2_domain1_test-TEST,2001:DB8::1,v6
Hns1_domain2_test-TEST,192.0.2.2,v4
Hns2_domain2_test-TEST,2001:DB8::2,v6
```

5.2.2.2. <csvHost:deletes>

The <csvHost:deletes> is used to hold the deleted host objects in a differential or incremental deposit. All the host object data is deleted as part of a cascade delete. The <csvHost:deletes> is split into separate CSV file definitions using named <rdeCsv:csv> elements with the "name" attribute. The following section defines the supported host deletes CSV file definition.

5.2.2.2.1. "host" Deletes CSV File Definition

The following "rdeCsv" fields, defined in section CSV common field elements (Section 4.7.2), MUST be used in the "host" <rdeCsv:csv> <rdeCsv:fields> element:

<rdeCsv:fROID> Repository Object IDentifier (ROID) assigned to the host object with isRequired="true".

Example of a "host" <csvHost:deletes> <rdeCsv:csv> element.

```
...
<csvHost:deletes>
...
  <rdeCsv:csv name="host">
    <rdeCsv:fields>
      <rdeCsv:fROID/>
    </rdeCsv:fields>
    <rdeCsv:files>
      <rdeCsv:file
        cksum="777F5F0E">
        host-delete-YYYYMMDD.csv
      </rdeCsv:file>
    </rdeCsv:files>
  </rdeCsv:csv>
...
</csvHost:deletes>
...
```

Example of the host-delete-YYYYMMDD.csv file. The file contains four host records.

```
Hns1_domain1_test-TEST
Hns2_domain1_test-TEST
Hns1_domain2_test-TEST
Hns2_domain2_test-TEST
```

5.3. Contact Object

The contact object is based on the EPP contact name mapping in [RFC5733]. The contact object supports both the XML Model and the CSV Model, defined in Models (Section 2) section. The elements used for both models are defined in the following sections.

5.3.1. XML Model

There are two elements used in the data escrow of the contact objects for the XML model including the `<rdeContact:contact>`, under the `<rdeContact:contents>` element, and the `<rdeContact:delete>` element, under the `<rde: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.

5.3.1.1. `<rdeContact: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:

- o A `<id>` element that contains the server-unique identifier of the contact object
- o A `<roid>` element that contains the Repository Object IDentifier assigned to the contact object when the object was created.
- o One or more `<status>` elements that describe the status of the contact object.
- o 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.
- o An OPTIONAL <voice> element that contains the contact's voice telephone number.
- o An OPTIONAL <fax> element that contains the contact's facsimile telephone number.
- o An <email> element that contains the contact's email address.
- o A <clID> element that contains the identifier of the sponsoring registrar.
- o 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.
- o A <crDate> element that contains the date and time of contact-object creation.
- o 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.

- o 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.
- o 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.
- o 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:roid>Csh8013-TEST</rdeContact:roid>
  <rdeContact:id>sh8013</rdeContact:id>
  <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="jstyles">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>
...
```

5.3.1.2. <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>
...
```

5.3.2. CSV Model

For the CSV Model of the contact object, the <csvContact:contents> child element of the <rde:contents> element is used to hold the new or updated contacts objects for the deposit. The <csvContact:deletes> child element of the <rde:deletes> element is used to hold the deleted or purged contact objects for the deposit. Both the <csvContact:contents> and <csvContact:deletes> elements contain one or more <rdeCsv:csv> elements with a set of named CSV file definitions using the <rdeCsv:csv> "name" attribute.

Differential and incremental deposits are based on changes to the contact objects. The updated contact object data under the <csvContact:contents> element is a cascade replace down all of the contact CSV files starting with the parent "contact" CSV File Definition (Section 5.3.2.1.1). The child CSV file definitions include a <csvContact:fId parent="true"> field. All the child CSV file definition data for the contact objects in the parent "contact" CSV File Definition (Section 5.3.2.1.1) MUST first be deleted and then set using the data in the child CSV files. The deleted contact object data under the <csvContact:deletes> element is a cascade delete starting from the "contact" Deletes CSV File Definition (Section 5.3.2.2.1).

5.3.2.1. <csvContact:contents>

The <csvContact:contents> is used to hold the new or updated contact object information for the deposit. The <csvContact:contents> is split into separate CSV file definitions using named <rdeCsv:csv> elements with the "name" attribute. The following sections include the supported contact CSV file definitions.

5.3.2.1.1. "contact" CSV File Definition

The "contact" CSV File Definition defines the fields and CSV file references used for the contact object records.

The following "csvContact" field elements MUST be used in the "contact" <rdeCsv:csv> <rdeCsv:fields> element:

<csvContact:fId> Contains the server-unique contact identifier with type="eppcom:clIDType" and isRequired="true".

<csvContact:fEmail> Contains the contact's email address with type="eppcom:minTokenType" and isRequired="true".

The following field elements MAY be used in the "contact" <rdeCsv:csv> <rdeCsv:fields> element:

<csvContact:fVoice> Contains the contact's voice telephone number with type="contact:el64StringType".

<csvContact:fVoiceExt> Contains the contact's voice telephone number extension with type="token".

<csvContact:fFax> Contains the contact's facsimile telephone number with type="contact:el64StringType".

<csvContact:fFaxExt> Contains the contact's facsimile telephone number extension with type="token".

The following "rdeCsv" fields, defined in section CSV common field elements (Section 4.7.2), MUST be used in the "contact" <rdeCsv:csv> <rdeCsv:fields> element:

<rdeCsv:fRoid> The Registry Object Identifier (ROID) for the contact object with isRequired="true".

<rdeCsv:fClID> Identifier of client (registrar) that sponsors the contact object with isRequired="true".

<rdeCsv:fCrRr> Identifier of the registrar, defined in Section 5.4, of the client that created the object with isRequired="true".

<rdeCsv:fCrDate> Created date and time of the contact object with isRequired="true".

<rdeCsv:fUpRr> Identifier of the registrar, defined in Section 5.4, of the client that updated the object.

<rdeCsv:fUpDate> Date and time of the last update to the contact object.

The following "rdeCsv" fields, defined in section CSV common field elements (Section 4.7.2), MAY be used in the "contact" <rdeCsv:csv> <rdeCsv:fields> element:

<rdeCsv:fCrID> Identifier of client that created the contact object.

<rdeCsv:fUpID> Identifier of the client that last updated the contact object.

<rdeCsv:fTrDate> Date and time of the last transfer for the contact object.

Example of a "contact" <csvContact:contacts> <rdeCsv:csv> element.

```
...
<csvContact:contents>
...
  <rdeCsv:csv name="contact">
    <rdeCsv:fields>
      <csvContact:fId/>
      <rdeCsv:fRoid/>
      <csvContact:fVoice/>
      <csvContact:fVoiceExt/>
      <csvContact:fFax/>
      <csvContact:fFaxExt/>
      <csvContact:fEmail/>
      <rdeCsv:fClID/>
      <rdeCsv:fCrRr/>
      <rdeCsv:fCrID/>
      <rdeCsv:fCrDate/>
      <rdeCsv:fUpRr/>
      <rdeCsv:fUpID/>
      <rdeCsv:fUpDate/>
    </rdeCsv:fields>
    <rdeCsv:files>
      <rdeCsv:file
        cksum="8587AA49">
        contact-YYYYMMDD.csv
      </rdeCsv:file>
    </rdeCsv:files>
  </rdeCsv:csv>
...
</csvContact:contents>
...
```

Example of the contact-YYYYMMDD.csv file. The file contains nine object contact records.

```
domainladmin,Cdomainladmin-TEST,+1.7035555555,1234,
+1.7035555556,,jdoe@example.test,registrarX,registrarX,
clientY,2009-09-13T08:01:00.0Z,registrarX,clientY,
2009-11-26T09:10:00.0Z
domainltech,Cdomainltech-TEST,+1.7035555555,1234,
+1.7035555556,,jdoe@example.test,registrarX,registrarX,
clientY,2009-09-13T08:01:00.0Z,registrarX,clientY,
2009-11-26T09:10:00.0Z
domainlbilling,Cdomainlbilling-TEST,+1.7035555555,1234,
+1.7035555556,,jdoe@example.test,registrarX,registrarX,
clientY,2009-09-13T08:01:00.0Z,registrarX,clientY,
2009-11-26T09:10:00.0Z
domain2admin,Cdomain2admin-TEST,+1.7035555555,1234,
+1.7035555556,,jdoe@example.test,registrarX,registrarX,
clientY,2009-09-13T08:01:00.0Z,registrarX,clientY,
2009-11-26T09:10:00.0Z
domain2tech,Cdomain2tech-TEST,+1.7035555555,1234,
+1.7035555556,,jdoe@example.test,registrarX,registrarX,
clientY,2009-09-13T08:01:00.0Z,registrarX,clientY,
2009-11-26T09:10:00.0Z
domain2billing,Cdomain2billing-TEST,+1.7035555555,1234,
+1.7035555556,,jdoe@example.test,registrarX,registrarX,
clientY,2009-09-13T08:01:00.0Z,registrarX,clientY,
2009-11-26T09:10:00.0Z
xnabc123admin,Cxnabc123admin-TEST,+1.7035555555,1234,
+1.7035555556,,jdoe@example.test,registrarX,registrarX,
clientY,2009-09-13T08:01:00.0Z,registrarX,clientY,
2009-11-26T09:10:00.0Z
xnabc123tech,Cxnabc123tech-TEST,+1.7035555555,1234,
+1.7035555556,,jdoe@example.test,registrarX,registrarX,
clientY,2009-09-13T08:01:00.0Z,registrarX,clientY,
2009-11-26T09:10:00.0Z
xnabc123billing,Cxnabc123billing-TEST,+1.7035555555,1234,
+1.7035555556,,jdoe@example.test,registrarX,registrarX,
clientY,2009-09-13T08:01:00.0Z,registrarX,clientY,
2009-11-26T09:10:00.0Z
```

5.3.2.1.2. "contactStatuses" CSV File Definition

The "contactStatuses" CSV File Definition defines the fields and CSV file references used for the contact object statuses.

The following "csvContact" field elements, defined for the "contact" CSV File Definition (Section 5.3.2.1.1), MUST be used in the "contactStatuses" <rdeCsv:csv> <rdeCsv:fields> element:

<csvContact:fId> Server-unique contact identifier of status with
isRequired="true".

<csvContact:fStatus> The status of the contact with
type="contact:statusValueType" and isRequired="true".

The following "rdeCsv" fields, defined in section CSV common field
elements (Section 4.7.2), MAY be used in the "contactStatuses"

<rdeCsv:csv> <rdeCsv:fields> element:

<rdeCsv:fStatusDescription> The contact object status description
which is free form text describing the rationale for the status.

<rdeCsv:fLang> Language of the <rdeCsv:fStatusDescription> field.

Example of a "contactStatuses" <csvContact:contents> <rdeCsv:csv>
element.

```
...
<csvContact:contents>
...
  <rdeCsv:csv name="contactStatuses">
    <rdeCsv:fields>
      <csvContact:fId parent="true"/>
      <csvContact:fStatus/>
      <rdeCsv:fStatusDescription/>
      <rdeCsv:fLang/>
    </rdeCsv:fields>
    <rdeCsv:files>
      <rdeCsv:file
        cksum="137E13EC">
        contactStatuses-YYYYMMDD.csv
      </rdeCsv:file>
    </rdeCsv:files>
  </rdeCsv:csv>
...
</csvContact:contents>
...
```

Example of the corresponding contactStatuses-YYYYMMDD.csv file. The file contains the statuses for the nine contact identifiers.

```
domain1admin,ok,,
domain1tech,ok,,
domain1billing,ok,,
domain2admin,ok,,
domain2tech,ok,,
domain2billing,ok,,
xnabc123admin,ok,,
xnabc123tech,ok,,
xnabc123billing,ok,,
```

5.3.2.1.3. "contactPostal" CSV File Definition

The "contactPostal" CSV File Definition defines the fields and CSV file references used for the contact postal info object records.

The following "csvContact" field elements MUST be used in the "contactPostal" <rdeCsv:csv> <rdeCsv:fields> element:

<csvContact:fPostalType> Contains the form of the postal-address information with type="contact:postalLineType" and isRequired="true". This field specifies the form ("int" or "loc"), as defined in Section 4.8, of the <csvContact:fName>, <csvContact:fOrg>, <csvContact:fStreet>, <csvContact:fCity>, <csvContact:fSp>, <csvContact:fPc>, <csvContact:fCc> fields.

<csvContact:fName> Contains the contact's name of the individual or role represented by the contact with type="contact:postalLineType" and isRequired="true". An OPTIONAL "isLoc" attribute to used to indicate the localized or internationalized form as defined in section Section 4.8.

<csvContact:fStreet> Contains the contact's contact's street address line with type="contact:fPostalLineType". An index attribute is required to indicate which street address line the field represents with index "0" for the first line and index "2" for the last line. An OPTIONAL "isLoc" attribute to used to indicate the localized or internationalized form as defined in section Section 4.8.

<csvContact:fCity> Contains the contact's city with type="contact:postalLineType" and isRequired="true". An OPTIONAL "isLoc" attribute to used to indicate the localized or internationalized form as defined in section Section 4.8.

<csvContact:fCc> Contains the contact's country code with type="contact:ccType" and isRequired="true". An OPTIONAL "isLoc" attribute to used to indicate the localized or internationalized form as defined in section Section 4.8.

The following "csvContact" field elements MAY be used in the "contactPostal" <rdeCsv:csv> <rdeCsv:fields> element:

<csvContact:fOrg> Contains the name of the organization with which the contact is affiliated with type="contact:optPostalLineType". An OPTIONAL "isLoc" attribute to used to indicate the localized or internationalized form as defined in section Section 4.8.

<csvContact:fSp> Contains the contact's state or province with type="contact:optPostalLineType". An OPTIONAL "isLoc" attribute to used to indicate the localized or internationalized form as defined in section Section 4.8.

<csvContact:fPc> Contains the contact's postal code with type="contact:pcType". An OPTIONAL "isLoc" attribute to used to indicate the localized or internationalized form as defined in section Section 4.8.

The following "csvContact" fields, defined for the "contact" CSV File Definition (Section 5.3.2.1.1), MUST be used in the "contactPostal" <rdeCsv:csv> <rdeCsv:fields> element:

<csvContact:fId> Server-unique contact identifier for the contact object with isRequired="true".

Example of a "contactPostal" <csvContact:contents> <rdeCsv:csv> element.

```
...
<csvContact:contents>
...
  <rdeCsv:csv name="contactPostal">
    <rdeCsv:fields>
      <csvContact:fId parent="true"/>
      <csvContact:fPostalType/>
      <csvContact:fName/>
      <csvContact:fOrg/>
      <csvContact:fStreet index="0"/>
      <csvContact:fStreet index="1"/>
      <csvContact:fStreet index="2"/>
      <csvContact:fCity/>
      <csvContact:fSp/>
      <csvContact:fPc/>
      <csvContact:fCc/>
    </rdeCsv:fields>
    <rdeCsv:files>
      <rdeCsv:file
        cksum="1456A89C">
        contactPostal-YYYYMMDD.csv
      </rdeCsv:file>
    </rdeCsv:files>
  </rdeCsv:csv>
...
</csvContact:contents>
...
```

Example of the contactPostal-YYYYMMDD.csv file. The file contains nine contact postal records.

```
domainladmin,int,"John Doe","Example Inc.",
"123 Example Dr.,"Suite 100",,Reston,VA,20190,US
domainltech,int,"John Doe","Example Inc.",
"123 Example Dr.,"Suite 100",,Reston,VA,20190,US
domainlbilling,int,"John Doe","Example Inc.",
"123 Example Dr.,"Suite 100",,Reston,VA,20190,US
domain2admin,int,"John Doe","Example Inc.",
"123 Example Dr.,"Suite 100",,Reston,VA,20190,US
domain2tech,int,"John Doe","Example Inc.",
"123 Example Dr.,"Suite 100",,Reston,VA,20190,US
domain2billing,int,"John Doe","Example Inc.",
"123 Example Dr.,"Suite 100",,Reston,VA,20190,US
xnabc123admin,int,"John Doe","Example Inc.",
"123 Example Dr.,"Suite 100",,Reston,VA,20190,US
xnabc123tech,int,"John Doe","Example Inc.",
"123 Example Dr.,"Suite 100",,Reston,VA,20190,US
xnabc123billing,int,"John Doe","Example Inc.",
"123 Example Dr.,"Suite 100",,Reston,VA,20190,US
```

5.3.2.1.4. "contactTransfer" CSV File Definition

The "contactTransfer" CSV File Definition defines the fields and CSV file references used for the contact object pending and completed transfer records. No additional field elements were added for use in the "contactTransfer" <rdeCsv:csv> <rdeCsv:fields> element. The following "rdeCsv" fields, defined in section CSV common field elements (Section 4.7.2), MUST be used in the "contactTransfer" <rdeCsv:csv> <rdeCsv:fields> element:

```
<rdeCsv:fTrStatus> State of the most recent transfer request with
    isRequired="true".

<rdeCsv:fReRr> Identifier of the registrar, defined in Section 5.4,
    of the client that requested the transfer with isRequired="true".

<rdeCsv:fReDate> Date and time that the transfer was requested with
    isRequired="true".

<rdeCsv:fAcRr> Identifier of the registrar, defined in Section 5.4,
    of the client that should take or took action with
    isRequired="true".

<rdeCsv:fAcDate> Date and time that the transfer action should be
    taken or has been taken with isRequired="true".
```

The following "rdeCsv" fields, defined in section CSV common field elements (Section 4.7.2), MAY be used in the "contactTransfer" <rdeCsv:csv> <rdeCsv:fields> element:

<rdeCsv:fReID> Identifier of client that requested the transfer.

<rdeCsv:fAcID> Identifier of client that should take or took action for transfer.

The following "csvContact" fields, defined for the "contact" CSV File Definition (Section 5.3.2.1.1), MUST be used in the "contactTransfer" <rdeCsv:csv> <rdeCsv:fields> element:

<csvContact:fId> Server-unique contact identifier for the contact object with isRequired="true".

Example of a "contactTransfer" <csvContact:contents> <rdeCsv:csv> element.

```
...
<csvContact:contents>
...
  <rdeCsv:csv name="contactTransfer">
    <rdeCsv:fields>
      <csvContact:fId parent="true"/>
      <rdeCsv:fTrStatus/>
      <rdeCsv:fReRr/>
      <rdeCsv:fReID/>
      <rdeCsv:fReDate/>
      <rdeCsv:fAcRr/>
      <rdeCsv:fAcID/>
      <rdeCsv:fAcDate/>
    </rdeCsv:fields>
    <rdeCsv:files>
      <rdeCsv:file
        cksum="788D308E">
        contactTransfer-YYYYMMDD.csv
      </rdeCsv:file>
    </rdeCsv:files>
  </rdeCsv:csv>
...
</csvContact:contents>
...
```


Example of the contactTransfer-YYYYMMDD.csv file. The file contains one contact transfer record in pending status.

```
xnabc123admin,clientApproved,registrarX,clientX,  
2011-04-08T19:38:00.0Z,registrarY,clientY,2011-04-09T20:38:00.0Z
```

5.3.2.1.5. "contactDisclose" CSV File Definition

The "contactDisclose" CSV File Definition defines the fields and CSV file references used for the contact disclose object records.

The following "csvContact" field elements MAY be used in the "contactDisclose" <rdeCsv:csv> <rdeCsv:fields> element:

<csvContact:fDiscloseFlag> Contains flag with a value of "true" or "1" (one) notes the preference to allow disclosure of the specified elements as an exception to the stated data-collection policy. A value of "false" or "0" (zero) notes a client preference to not allow disclosure of the specified elements as an exception to the stated data-collection policy with type="boolean". The additional fields define specific exceptional disclosure preferences based on the <csvContact:fDiscloseFlag> field.

<csvContact:fDiscloseNameLoc> Exceptional disclosure preference flag for the localized form of the contact name with type="boolean".

<csvContact:fDiscloseNameInt> Exceptional disclosure preference flag for the internationalized form of the contact name with type="boolean".

<csvContact:fDiscloseOrgLoc> Exceptional disclosure preference flag for the localized form of the contact organization with type="boolean".

<csvContact:fDiscloseOrgInt> Exceptional disclosure preference flag for the internationalized form of the contact organization with type="boolean".

<csvContact:fDiscloseAddrLoc> Exceptional disclosure preference flag for the localized form of the contact address with type="boolean".

<csvContact:fDiscloseAddrInt> Exceptional disclosure preference flag for the internationalized form of the contact address with type="boolean".

<csvContact:fDiscloseVoice> Exceptional disclosure preference flag of the contact voice telephone number with type="boolean".

<csvContact:fDiscloseFax> Exceptional disclosure preference flag of the contact facsimile telephone number with type="boolean".

<csvContact:fDiscloseEmail> Exceptional disclosure preference flag of the contact email address with type="boolean".

The following "csvContact" fields, defined for the "contact" CSV File Definition (Section 5.3.2.1.1), MUST be used in the "contactDisclose" <rdeCsv:csv> <rdeCsv:fields> element:

<csvContact:fId> Server-unique contact identifier for the contact object with isRequired="true".

Example of a "contactDisclose" <csvContact:contents> <rdeCsv:csv> element.

```
...
<csvContact:contents>
...
  <rdeCsv:csv name="contactDisclose">
    <rdeCsv:fields>
      <csvContact:fId parent="true"/>
      <csvContact:fDiscloseFlag/>
      <csvContact:fDiscloseNameLoc/>
      <csvContact:fDiscloseNameInt/>
      <csvContact:fDiscloseOrgLoc/>
      <csvContact:fDiscloseOrgInt/>
      <csvContact:fDiscloseAddrLoc/>
      <csvContact:fDiscloseAddrInt/>
      <csvContact:fDiscloseVoice/>
      <csvContact:fDiscloseFax/>
      <csvContact:fDiscloseEmail/>
    </rdeCsv:fields>
    <rdeCsv:files>
      <rdeCsv:file
        cksum="1141EFD4">
        contactDisclose-YYYYMMDD.csv
      </rdeCsv:file>
    </rdeCsv:files>
  </rdeCsv:csv>
...
</csvContact:contents>
...
```

Example of the contactDisclose-YYYYMMDD.csv file. The file contains one disclosure records, disabling disclosure of voice, fax, and email.

```
xnabc123admin,0,0,0,0,0,0,0,0,1,1,1
```

5.3.2.2. <csvContact:deletes>

The <csvContact:deletes> is used to hold the deleted contact objects in a differential or incremental deposit. All the contact object data is deleted as part of a cascade delete. The <csvContact:deletes> is split into separate CSV file definitions using named <rdeCsv:csv> elements with the "name" attribute. The following section defines the supported contact deletes CSV file definition.

5.3.2.2.1. "contact" Deletes CSV File Definition

The following "csvContact" field elements MUST be used in the deletes "contact" <rdeCsv:csv> <rdeCsv:fields> element:

<csvContact:fId> Contains the server-unique contact identifier with type="eppcom:clIDType" and isRequired="true".

Example of a "contact" <csvContact:deletes> <rdeCsv:csv> element.

```
...
<csvContact:deletes>
...
  <rdeCsv:csv name="contact">
    <rdeCsv:fields>
      <csvContact:fId/>
    </rdeCsv:fields>
    <rdeCsv:files>
      <rdeCsv:file
        cksum="0C4B70DC">
          contact-delete-YYYYMMDD.csv
        </rdeCsv:file>
      </rdeCsv:files>
    </rdeCsv:csv>
  </rdeCsv:deletes>
...
```

Example of the contact-delete-YYYYMMDD.csv file. The file contains six contact records.

```
domain1admin  
domain1tech  
domain1billing  
domain2admin  
domain2tech  
domain2billing
```

5.4. Registrar Object

The registrar object represents the sponsoring client for other objects, for operational purposes MAY be the registry operator. The registrar object supports both the XML Model and the CSV Model, defined in Section 2. The elements used for both models are defined in the following sections.

5.4.1. XML Model

There are two elements used in the data escrow of the registrar objects for the XML model including the `<rdeRegistrar:registrar>`, under the `<rdeRegistrar:contents>` element, and the `<rdeRegistrar:delete>` element, under the `<rde:deletes>` element.

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

5.4.1.1. `<rdeRegistrar:registrar>` element

The `<registrar>` element contains the following child elements:

- o 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.
- o An `<name>` element that contains the name of the registrar.
- o An OPTIONAL `<gurid>` element that contains the ID assigned by ICANN.
- o An OPTIONAL `<status>` element that contains the operational status of the registrar. Possible values are: ok, readonly and terminated.

- o Zero or two OPTIONAL <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.
- o An OPTIONAL <voice> element that contains the registrar's voice telephone number.
- o An OPTIONAL <fax> element that contains the registrar's facsimile telephone number.
- o An OPTIONAL <email> element that contains the registrar's email address.
- o An OPTIONAL <url> element that contains the registrar's URL.
- o 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.

- o An OPTIONAL <crDate> element that contains the date and time of registrar-object creation.
- o 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>
...
```

5.4.1.2. <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>
...
```

5.4.2. CSV Model

For the CSV Model of the registrar object, the <csvRegistrar:contents> child element of the <rde:contents> element is used to hold the new or updated registrar objects for the deposit. The <csvRegistrar:deletes> child element of the <rde:deletes> element is used to hold the deleted or purged registrar objects for the deposit. Both the <csvRegistrar:contents> and <csvRegistrar:deletes> elements contain one or more <rdeCsv:csv> elements with a set of named CSV file definitions using the <rdeCsv:csv> "name" attribute.

Differential and incremental deposits are based on changes to the registrar objects. The updated registrar object data under the <csvContact:contents> element is a cascade replace down all of the registrar CSV files starting with the parent "registrar" CSV File Definition (Section 5.4.2.1.1). The child CSV file definitions include a <csvRegistrar:fId parent="true"> field. All the child CSV file definition data for the registrar objects in the parent "registrar" CSV File Definition (Section 5.4.2.1.1) MUST first be deleted and then set using the data in the child CSV files. The deleted registrar object data under the <csvRegistrar:deletes> element is a cascade delete starting from the "registrar" Deletes CSV File Definition (Section 5.4.2.2.1).

5.4.2.1. <csvRegistrar:contents>

The <csvRegistrar:contents> is used to hold the new or updated registrar object information for the deposit. The <csvRegistrar:contents> is split into separate CSV file definitions using named <rdeCsv:csv> elements with the "name" attribute. The following sections include the supported contact CSV file definitions.

5.4.2.1.1. "registrar" CSV File Definition

The "registrar" CSV File Definition defines the fields and CSV file references used for the registrar object records.

The following "csvRegistrar" field elements MUST be used in the "registrar" <rdeCsv:csv> <rdeCsv:fields> element:

<csvRegistrar:fId> Contains the server-unique registrar identifier with type="eppcom:clIDType" and isRequired="true".

<csvRegistrar:fName> Contains the name of the registrar with type="normalizedString" and isRequired="true".

The following field elements MAY be used in the "registrar" <rdeCsv:csv> <rdeCsv:fields> element:

<csvRegistrar:fStatus> Contains the status of the registrar with type="csvRegistrar:statusValueType".

<csvRegistrar:fGurid> Contains the ID assigned by ICANN with type="positiveInteger".

<csvRegistrar:fWhoisUrl> Contains the Whois URL of the registrar with type="anyURI".

The following "rdeCsv" fields, defined in section CSV common field elements (Section 4.7.2), MAY be used in the "registrar" <rdeCsv:csv> <rdeCsv:fields> element:

<rdeCsv:fCrDate> Created date and time of the registrar object.

<rdeCsv:fUpDate> Date and time of the last update to the registrar object.

<rdeCsv:fUrl> URL for the registrar web home page.

The following "csvContact" fields, defined in section Contact Object (Section 5.3), MAY be used in the "registrar" <rdeCsv:csv> <rdeCsv:fields> element:

<csvContact:fStreet> Registrar street address line with an "index" attribute that represents the order of the street address line from "0" to "2". An OPTIONAL "isLoc" attribute that is used to indicate the localized or internationalized form, as defined in Section 4.8.

<csvContact:fCity> Registrar city with an OPTIONAL "isLoc" attribute that is used to indicate the localized or internationalized form, as defined in Section 4.8.

<csvContact:fCc> Registrar country code with an OPTIONAL "isLoc" attribute that is used to indicate the localized or internationalized form, as defined in Section 4.8.

<csvContact:fEmail> Registrar email address.

<csvContact:fSp> Registrar state or province with an OPTIONAL "isLoc" attribute that is used to indicate the localized or internationalized form, as defined in Section 4.8.

<csvContact:fPc> Registrar postal code with an OPTIONAL "isLoc" attribute that is used to indicate the localized or internationalized form, as defined in Section 4.8.

<csvContact:fVoice> Registrar voice telephone number.

<csvContact:fVoiceExt> Registrar voice telephone number extension.

<csvContact:fFax> Registrar facsimile telephone number.

<csvContact:fFaxExt> Registrar facsimile telephone number extension.

Example of a "registrar" <csvRegistrar:contents> <rdeCsv:csv> element.

```
...
<csvRegistrar:contents>
...
  <rdeCsv:csv name="registrar">
    <rdeCsv:fields>
      <csvRegistrar:fId/>
      <csvRegistrar:fName isLoc="false"/>
      <csvRegistrar:fGurid/>
      <csvRegistrar:fStatus/>
      <csvContact:fStreet isLoc="false" index="0"/>
      <csvContact:fStreet isLoc="false" index="1"/>
      <csvContact:fStreet isLoc="false" index="2"/>
      <csvContact:fCity isLoc="false"/>
      <csvContact:fSp isLoc="false" />
      <csvContact:fPc isLoc="false" />
      <csvContact:fCc isLoc="false"/>
      <csvContact:fVoice/>
      <csvContact:fVoiceExt/>
      <csvContact:fFax/>
      <csvContact:fFaxExt/>
      <csvContact:fEmail/>
      <rdeCsv:fUrl/>
      <csvRegistrar:fWhoisUrl/>
      <rdeCsv:fCrDate/>
      <rdeCsv:fUpDate/>
    </rdeCsv:fields>
    <rdeCsv:files>
      <rdeCsv:file
        cksum="57F6856F">
        registrar-YYYYMMDD.csv
      </rdeCsv:file>
    </rdeCsv:files>
  </rdeCsv:csv>
...
</csvRegistrar:contents>
...
```

Example of the registrar-YYYYMMDD.csv file. The file contains three registrar records.

```
registrarX,"Example Inc.",1234,ok,"123 Example Dr.",  
"Suite 100",,Dulles,VA,20166-6503,US,+1.7035555555,1234,  
+1.7035555556,,jdoe@example.test,http://www.example.test,  
http://whois.example.test,2005-04-23T11:49:00.0Z,  
2009-02-17T17:51:00.0Z  
registrarY,"Example2 Inc.",1234,ok,"123 Example Dr.",  
"Suite 100",,Dulles,VA,20166-6503,US,+1.7035555555,1234,  
+1.7035555556,,jdoe@example.test,http://www.example.test,  
http://whois.example.test,2005-04-23T11:49:00.0Z,  
2009-02-17T17:51:00.0Z  
registrarZ,"Example2 Inc.",1234,ok,"123 Example Dr.",  
"Suite 100",,Dulles,VA,20166-6503,US,+1.7035555555,1234,  
+1.7035555556,,jdoe@example.test,http://www.example.test,  
http://whois.example.test,2005-04-23T11:49:00.0Z,  
2009-02-17T17:51:00.0Z
```

5.4.2.2. <csvRegistrar:deletes>

The <csvRegistrar:deletes> is used to hold the deleted registrar objects in a differential or incremental deposit. All the registrar object data is deleted as part of a cascade delete. The <csvRegistrar:deletes> is split into separate CSV file definitions using named <rdeCsv:csv> elements with the "name" attribute. The following section defines the supported registrar deletes CSV file definition.

5.4.2.2.1. "registrar" Deletes CSV File Definition

The following "csvRegistrar" field elements MUST be used in the deletes "registrar" <rdeCsv:csv> <rdeCsv:fields> element:

<csvRegistrar:fId> Contains the server-unique registrar identifier with type="eppcom:clIDType" and isRequired="true".

Example of a "registrar" <csvRegistrar:deletes> <rdeCsv:csv> element.

```
...
<csvRegistrar:deletes>
...
  <rdeCsv:csv name="registrar">
    <rdeCsv:fields>
      <csvRegistrar:fId/>
    </rdeCsv:fields>
    <rdeCsv:files>
      <rdeCsv:file
        cksum="5CB20A52">
        registrar-delete-YYYYMMDD.csv
      </rdeCsv:file>
    </rdeCsv:files>
  </rdeCsv:csv>
...
</csvRegistrar:deletes>
...
```

Example of the registrar-delete-YYYYMMDD.csv file. The file contains one registrar record.

registrarZ

5.5. IDN Table Reference Object

The Internationalized Domain Names (IDN) table reference object is a pseudo-object that is used to provide a short reference to the IDN Table and Policy used in IDN registrations. The IDN reference object supports both the XML and the CSV Model, defined in the Models (Section 2) section. The elements used for both models are defined in the following sections.

5.5.1. XML Model

There is one element used in the data escrow of the IDN table reference objects for the XML model that is the <rdeIDN:idnTableRef>, under the <rde:contents> element.

5.5.1.1. <rdeIDN:idnTableRef> object

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

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

- o 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>
...
```

5.5.2. CSV Model

The IDN domain names, defined in Section 5.1, MAY have references to the IDN language identifier using the `<rdeCsv:fIdnTableId>` field element. The IDN table reference object defines the mapping of a language identifier to a language table URL. The language table URL defines the character code points that can be used for the language identifier. The elements used for the IDN table reference object is defined in this section. The `<csvIDN:contents>` child element of the `<rde:contents>` element is used to hold the new or updated IDN table reference objects for the deposit. The `<csvIDN:deletes>` child element of the `<rde:deletes>` element is used to hold the deleted or purged IDN table reference objects for the deposit. Both the `<csvIDN:contents>` and `<csvIDN:deletes>` elements contain one or more `<rdeCsv:csv>` elements with a set of named CSV file definitions using the `<rdeCsv:csv>` "name" attribute.

5.5.2.1. `<csvIDN:contents>`

The `<csvIDN:contents>` is used to hold the new or updated IDN table reference object information for the deposit. The `<csvIDN:contents>` is split into separate CSV file definitions using named `<rdeCsv:csv>` elements with the "name" attribute. The following sections include the supported IDN table reference CSV file definitions.

5.5.2.1.1. "idnLanguage" CSV File Definition

The "idnLanguage" CSV File Definition defines the fields and CSV file references used for the IDN table reference object records.

The following "rdeCsv" fields, defined in Section 4.7.2, MUST be used in the "idnLanguage" <rdeCsv:csv> <rdeCsv:fields> element:

<rdeCsv:fIdnTableId> The language identifier that matches the values for the <rdeCsv:fIdnTableId> field element in the "domain" CSV File Definition (Section 5.1.2.1.1) files. The attribute isRequired MUST equal "true".

<rdeCsv:fUrl> URL that defines the character code points that can be used for the language defined by the <rdeCsv:fLang> field element. The attribute isRequired MUST equal "true".

Example of a "idnLanguage" <csvIDN:contents> <rdeCsv:csv> element.

```
...
<csvIDN:contents>
...
  <rdeCsv:csv name="idnLanguage">
    <rdeCsv:sep>,</rdeCsv:sep>
    <rdeCsv:fields>
      <rdeCsv:fIdnTableId isRequired="true"/>
      <rdeCsv:fUrl isRequired="true"/>
    </rdeCsv:fields>
    <rdeCsv:files>
      <rdeCsv:file
        cksum="D6B0424F">
        idnLanguage-YYYYMMDD.csv
      </rdeCsv:file>
    </rdeCsv:files>
  </rdeCsv:csv>
...
</csvIDN:contents>
...
```

Example of the corresponding idnLanguage-YYYYMMDD.csv file. The file contains two IDN language records.

```
LANG-1,
http://www.iana.org/domains/idn-tables/tables/test_tab1_1.1.txt
LANG-2,
http://www.iana.org/domains/idn-tables/tables/test_tab2_1.1.txt
```

5.5.2.2. <csvIDN:deletes>

The <csvIDN:deletes> is used to hold the deleted IDN table reference objects in a differential or incremental deposit. The <csvIDN:deletes> is split into separate CSV file definitions using named <rdeCsv:csv> elements with the "name" attribute. The following section defines the supported IDN table reference deletes CSV file definition.

5.5.2.2.1. "idnLanguage" Deletes CSV File Definition

The following "idnLanguage" field elements MUST be used in the deletes "idnLanguage" <rdeCsv:csv> <rdeCsv:fields> element:

<rdeCsv:fIdnTableId> The language identifier that matches the values for the <rdeCsv:fIdnTableId> field element in the "domain" CSV File Definition (Section 5.1.2.1.1) files. The attribute isRequired MUST equal "true".

Example of a "idnLanguage" <csvIDN:deletes> <rdeCsv:csv> element.

```
...
<csvIDN:deletes>
...
  <rdeCsv:csv name="idnLanguage">
    <rdeCsv:fields>
      <rdeCsv:fIdnTableId isRequired="true"/>
    </rdeCsv:fields>
    <rdeCsv:files>
      <rdeCsv:file
        cksum="4A28A569">
        idnLanguage-delete-YYYYMMDD.csv
      </rdeCsv:file>
    </rdeCsv:files>
  </rdeCsv:csv>
...
</csvIDN:deletes>
...
```

Example of the idnLanguage-delete-YYYYMMDD.csv file. The file contains one IDN language record.

LANG-2

5.6. NNDN Object

A NNDN (NNDN's not domain name) can be used to store registry reserved names or (blocked, withheld or mirrored) IDN variants.

A NNDN is a lightweight domain-like object that is not linked directly to a Registrar (a mirroring NNDN is linked to a Registrar via the original name).

A domain name can only exist as a domain name object or NNDN object, but not both.

The NNDN object supports both the XML and the CSV Model, defined in the Models (Section 2) section. The elements used for both models are defined in the following sections.

5.6.1. XML Model

There are two elements used in the data escrow of the NNDN objects for the XML model including the `<rdeNNDN:NNDN>`, under the `<rde:contents>` element, and the `<rdeNNDN:delete>` element, under the `<rde:deletes>` element.

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

5.6.1.1. `<rdeNNDN:NNDN>` object

The `<rdeNNDN:NNDN>` element contains the following child elements:

- o An `<aName>` element that contains the fully qualified name of the NNDN. If the NNDN is an IDN, the ASCII Compatible Encoding (ACE) MUST be used.
- o An OPTIONAL `<uName>` element that contains the fully qualified name of the NNDN in Unicode character set. It MUST be provided if available.
- o 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.
- o An OPTIONAL `<originalName>` element is used to indicate that the NNDN is used for an IDN variant. This element contains the domain name used to generate the IDN variant.

- o A <nameState> element that indicates the state of the NNDN: blocked, withheld or mirrored.
 - * 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 considered a potential registration of a domain object for a registrant, then the NNDN will be tagged as "withheld". This status is only used when the NNDN is used for an IDN variant.
 - * If a NNDN is considered a mirrored IDN variant of a domain object, then the NNDN will be tagged as "mirrored". A mirroringNS attribute is used to specify if the mirrored IDN variant use the NS mirror mechanism. The default value of mirroringNS is true. If another mechanism such as DNAME is used, the value of mirroringNS attribute MUST be false.
- o A <crDate> element that contains the date and time of the NNDN object creation.

Example of <rdeNNDN:NNDN> object:

```
...
<rdeNNDN:NNDN>
  <rdeNNDN:aName>xn--exempl-gva.test</rdeNNDN:aName>
  <rdeNNDN:idnTableId>pt-BR</rdeNNDN:idnTableId>
  <rdeNNDN:originalName>example1.test</rdeNNDN:originalName>
  <rdeNNDN:nameState>withheld</rdeNNDN:nameState>
  <rdeNNDN:crDate>2005-04-23T11:49:00.0Z</rdeNNDN:crDate>
</rdeNNDN:NNDN>
...
```

5.6.1.2. <rdeNNDN:delete> object

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

Example of <rdeNNDN::delete> object:

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

5.6.2. CSV Model

For the CSV Model of the NNDN object, the <csvNNDN:contents> child element of the <rde:contents> element is used to hold the new or updated NNDN objects for the deposit. The <csvNNDN:deletes> child element of the <rde:deletes> element is used to hold the deleted or purged NNDN objects for the deposit. Both the <csvNNDN:contents> and <csvNNDN:deletes> elements contain one or more <rdeCsv:csv> elements with a set of named CSV file definitions using the <rdeCsv:csv> "name" attribute.

5.6.2.1. <csvNNDN:contents>

The <csvNNDN:contents> is used to hold the new or updated NNDN object information for the deposit. The <csvNNDN:contents> is split into separate CSV file definitions using named <rdeCsv:csv> elements with the "name" attribute. The following sections include the supported NNDN CSV file definitions.

5.6.2.1.1. "NNDN" CSV File Definition

The "NNDN" CSV File Definition defines the fields and CSV file references used for the NNDN object records.

The following "csvNNDN" field elements MUST be used in the "NNDN" <rdeCsv:csv> <rdeCsv:fields> element:

<csvNNDN:fName> ASCII Compatible Encoding (ACE) field of the NNDN with type="eppcom:labelType" and isRequired="true".

<csvNNDN:fNameState> State of the NNDN: blocked or withheld with type="rdeNNDN:nameState" and isRequired="true". See Section 5.6.1.1 for a description of the possible values for the <rdeNNDN:nameState> element.

The following field elements MAY be used in the "NNDN" <rdeCsv:csv> <rdeCsv:fields> element:

<csvNNDN:fOriginalName> Domain name used to generate the IDN variant with type="eppcom:labelType".

<csvNNDN:fMirroringNS> Defines whether the "mirroring" <csvNNDN:fNameState> uses the NS mirror mechanism, as described for the <rdeNNDN:nameState> "mirroringNS" attribute in Section 5.6.1.1, with type="boolean". If the field element is not defined the default value is "true".

The following "rdeCsv" fields, defined in section CSV common field elements (Section 4.7.2), MUST be used in the "domain" <rdeCsv:csv> <rdeCsv:fields> element:

<rdeCsv:fCrDate> Created date and time of the NNDN object with isRequired="true".

The following "rdeCsv" fields, defined in section CSV common field elements (Section 4.7.2), MAY be used in the "domain" <rdeCsv:csv> <rdeCsv:fields> element:

<rdeCsv:fUName> Name of the NNDN in Unicode character set for the <csvNNDN:fAName> field element.

<rdeCsv:fIdnTableId> IDN Table Identifier for the NNDN that matches an IDN Table Reference Object record, as defined in Section 5.5.2.

Example of a "NNDN" <csvNNDN:contents> <rdeCsv:csv> element:

```
...
<csvNNDN:contents>
...
  <rdeCsv:csv name="NNDN" sep=",">
    <rdeCsv:fields>
      <csvNNDN:fAName/>
      <rdeCsv:fIdnTableId/>
      <csvNNDN:fOriginalName/>
      <csvNNDN:fNameState/>
      <csvNNDN:fMirroringNS/>
      <rdeCsv:fCrDate/>
    </rdeCsv:fields>
    <rdeCsv:files>
      <rdeCsv:file
        cksum="085A7CE4">
        NNDN-YYYYMMDD.csv
      </rdeCsv:file>
    </rdeCsv:files>
  </rdeCsv:csv>
...
</csvNNDN:contents>
...
```

Example of the corresponding NNDN-YYYYMMDD.csv file. The file contains two NNDN records for an IDN with one blocked variant and one mirrored variant.

```
xn--abc456.test,LANG-1,xn--abc123.test,
blocked,,2005-04-23T11:49:00.0Z
xn--abc789.test,LANG-1,xn--abc123.test,
mirrored,1,2005-04-23T11:49:00.0Z
```

5.6.2.2. <csvNNDN:deletes>

The <csvNNDN:deletes> is used to hold the deleted NNDN objects in a differential or incremental deposit. The <csvNNDN:deletes> is split into separate CSV file definitions using named <rdeCsv:csv> elements with the "name" attribute. The following section defines the supported NNDN deletes CSV file definition.

5.6.2.2.1. "NNDN" Deletes CSV File Definition

The following "NNDN" field elements MUST be used in the deletes "NNDN" <rdeCsv:csv> <rdeCsv:fields> element:

<csvNNDN:fAName> ASCII Compatible Encoding (ACE) field of the NNDN with type="eppcom:labelType" and isRequired="true".

Example of a "NNDN" <csvNNDN:deletes> <rdeCsv:csv> element.

```
...
<csvNNDN:deletes>
...
  <rdeCsv:csv name="NNDN">
    <rdeCsv:fields>
      <csvNNDN:fAName/>
    </rdeCsv:fields>
    <rdeCsv:files>
      <rdeCsv:file
        cksum="A41F1D9B">
          NNDN-delete-YYYYMMDD.csv
        </rdeCsv:file>
      </rdeCsv:files>
    </rdeCsv:csv>
  ...
</csvNNDN:deletes>
...
```

Example of the corresponding NNDN-delete-YYYYMMDD.csv file. The file contains one NNDN records.

```
xn--abc456.test
```

5.7. EPP Parameters Object

The EPP Parameters Object is a pseudo-object that defines the set of object and object extension services supported by the registry, as defined in [RFC5730]. The EPP Parameters Object is only defined as XML but could be used in the XML model or CSV model. The EPP Parameters Object is defined using the <rdeEppParams:eppParams> element. The EPP Parameters Object SHOULD be included if the registry supports EPP. Only one EPP Parameters Object MUST exist at a certain point in time (watermark).

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

- o One or more <version> elements that indicate the EPP versions supported by the registry.
- o One or more <lang> elements that indicate the identifiers of the text response languages supported by the registry's EPP server.

- o One or more <objURI> elements that contain namespace URIs representing the objects that the registry's EPP server is capable of managing.
- o 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.
- o 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. Policy Object

The Policy object is a pseudo-object that is used to specify which OPTIONAL elements from the XML Model are REQUIRED based on the business model of the registry. For the CSV Model, the OPTIONAL "isRequired" attribute of the <rdeCsv:field> elements, defined in Section 4.7.1, is used to specify which OPTIONAL fields are REQUIRED based on the business model of the registry.

5.8.1. <rdePolicy:policy> object

The OPTIONAL <policy> contains the following attributes:

- o An <element> that defines that the referenced <element> is REQUIRED.
- o <scope> that defines the XPath of the element referenced by <element>.

Example of <rdePolicy:policy> object:

```
...
<rdePolicy:policy scope="//rde:deposit/rde:contents/rdeDomain:domain"
  element="rdeDom:registrant" />
...
```

5.9. Header Object

The Header Object is a pseudo-object that is used to specify the number of objects in the repository at a specific point in time (watermark) regardless of the type of deposit: differential, full or incremental. The Header Object is only defined as XML but one header object MUST always be present per escrow deposit regardless of using XML Model or CSV Model. The Header Object is defined using the <rdeHeader:header> element.

5.9.1. <rdeHeader:header> object

The <rdeHeader:header> contains the following elements:

- o A choice of one of the elements defined in the "repositoryTypeGroup" group element that indicates the unique identifier for the repository being escrowed. Possible elements are:
 - * A <rdeHeader:tld> element that defines TLD or the REGISTRY-CLASS DOMAIN NAME (RCDN) being escrowed in the case of a

Registry data escrow deposit. For IDNs the A-Label is used [RFC5891].

- * A `<rdeHeader:registrar>` element that defines the Registrar ID corresponding to a Registrar data escrow deposit. In the case of an ICANN-accredited Registrar, the `<rdeHeader:registrar>` element MUST be the IANA Registrar ID assigned by ICANN.
- * A `<rdeHeader:ppsp>` element that defines the provider ID corresponding to a Privacy and Proxy Services Provider data escrow deposit. In the case of an ICANN-accredited Privacy and Proxy Services Provider, the `<rdeHeader:ppsp>` element MUST be the unique ID assigned by ICANN.
- o A `<count>` element that contains the number of objects in the SRS at a specific point in time (watermark) regardless of the type of deposit: differential, full or incremental. The `<count>` element supports the following attributes:
 - * A "uri" attribute reflects the XML namespace URI of the primary objects for the XML Model and CSV Model. For example, the "uri" is set to "urn:ietf:params:xml:ns:rdeDomain-1.0" for domain name objects using the XML Model, and the "uri" is set to "urn:ietf:params:xml:ns:csvDomain-1.0" for domain name objects using the CSV Model.
 - * An OPTIONAL "rcdn" attribute indicates the REGISTRY-CLASS DOMAIN NAME (RCDN) of the objects included in the `<count>` element. For IDNs the A-Label is used [RFC5891]. If the "rcdn" attribute is present, the value of the `<count>` element must include only objects related to registrations in the same and lower levels. For example in a data escrow deposit for the .EXAMPLE TLD, a value of "example" in the "rcdn" attribute within the `<count>` element indicates the number of objects in the TLD including objects in other RCDNs within the TLD, whereas a value of "com.example" indicates the number of elements for objects under "com.example" and lower levels. Omitting the "rcdn" attribute indicates that the total includes all objects of the specified "uri" in the repository (e.g. the TLD, Registrar, or PPSP).
 - * An OPTIONAL "registrarId" attribute indicates the identifier of the sponsoring Registrar of the objects included in the `<count>` element. In the case of an ICANN-accredited Registrar, the value MUST be the IANA Registrar ID assigned by ICANN.

Example of <rdeHeader:header> object referencing only the XML Model objects:

```
...
<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>
...
```

Example of <rdeHeader:header> object referencing the CSV and XML Model objects:

```
...
<rdeHeader:header>
  <rdeHeader:tld>test</rdeHeader:tld>
  <rdeHeader:count
    uri="urn:ietf:params:xml:ns:csvDomain-1.0">2</rdeHeader:count>
  <rdeHeader:count
    uri="urn:ietf:params:xml:ns:csvHost-1.0">1</rdeHeader:count>
  <rdeHeader:count
    uri="urn:ietf:params:xml:ns:csvContact-1.0">1</rdeHeader:count>
  <rdeHeader:count
    uri="urn:ietf:params:xml:ns:csvRegistrar-1.0">1
  </rdeHeader:count>
  <rdeHeader:count
    uri="urn:ietf:params:xml:ns:csvIDN-1.0">1</rdeHeader:count>
  <rdeHeader:count
    uri="urn:ietf:params:xml:ns:csvNNDN-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 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 specific IDN variant can be represented in the escrow deposit, as a domain or as a NNDN object, but not both.

If using domain objects to represent IDN variants, the normal behavior during restoration of a SRS based on an escrow deposit is to restore the IDN variants as a mirrored variant. If the registration data of the IDN variant is different from the original name, the details of this specific implementation MUST be described in the IDN policy document.

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 is 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 support 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. Data escrow agent extended verification process

The Data Escrow Agent MUST perform a extended verification process using the contents of the 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:

- o 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.
- o Count the objects and validate that the number of objects is equal to the number objects reported in the <header> element of the escrow deposit of that point in time (watermark).

- o All contact objects linked to domain names MUST be present.
- o All registrars objects linked to other objects MUST be present.
- o A domain name exists as a domain name or NNDN, but not both.
- o The elements listed as required in the <policy> element MUST be present.
- o All idnTableRef definitions linked from other objects MUST be present.
- o If an EPP Parameters Object was escrowed in the past, one and only one EPP Parameters Object MUST be present.
- o The watermark is not in the future.

9. Formal Syntax

Schemas are presented here.

9.1. RDE CSV Schema

Copyright (c) 2012 IETF Trust and the persons identified as authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- o Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- o Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- o Neither the name of Internet Society, IETF or IETF Trust, nor the names of specific contributors, may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,

SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

BEGIN

```
<?xml version="1.0" encoding="UTF-8"?>

<schema targetNamespace="urn:ietf:params:xml:ns:rdeCsv-1.0"
  xmlns:rdeCsv="urn:ietf:params:xml:ns:rdeCsv-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 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:rde-1.0"
    schemaLocation="rde-1.0.xsd"/>

  <annotation>
    <documentation>
      Registry Data Escrow Comma-Seperated Values (CSV)
    </documentation>
  </annotation>

  <!-- csv content element -->
  <element name="csv" type="rdeCsv:csvType"/>

  <!-- Definition of CSV file -->
  <complexType name="csvType">
    <sequence>
      <element name="fields" type="rdeCsv:fieldsType"/>
      <element name="files" type="rdeCsv:filesType"/>
    </sequence>
    <attribute name="name" type="token" use="required"/>
    <attribute name="sep" type="rdeCsv:sepType" default=","/>
  </complexType>

  <!-- field seperator must be a single character -->
  <simpleType name="sepType">
    <restriction base="string">
      <minLength value="1"/>
    </restriction>
  </simpleType>
</schema>
```

```
<maxLength value="1"/>
</restriction>
</simpleType>

<!-- Abstract field type -->
<element name="field" type="rdeCsv:fieldType"
  abstract="true"/>

<complexType name="fieldType">
  <sequence/>
</complexType>

<!-- fieldType with optional value (isRequired=false) -->
<complexType name="fieldOptionalType">
  <complexContent>
    <extension base="rdeCsv:fieldType">
      <sequence/>
      <attribute name="isRequired" type="boolean"
        default="false"/>
      <attribute name="parent" type="boolean"
        default="false"/>
    </extension>
  </complexContent>
</complexType>

<!-- fieldType with required value (isRequired=false) -->
<complexType name="fieldRequiredType">
  <complexContent>
    <extension base="rdeCsv:fieldType">
      <sequence/>
      <attribute name="isRequired" type="boolean"
        default="true"/>
      <attribute name="parent" type="boolean"
        default="false"/>
    </extension>
  </complexContent>
</complexType>

<!-- Concrete field types -->

<!-- UTF-8 Name field (e.g. domain name) -->
<element name="fUName" type="rdeCsv:fNameType"
  substitutionGroup="rdeCsv:field"/>
<complexType name="fNameType">
  <complexContent>
    <extension base="rdeCsv:fieldOptionalType">
      <sequence/>
    </extension>
  </complexContent>
</complexType>
```

```
        <attribute name="type" type="token"
            default="eppcom\:labelType"/>
    </extension>
</complexContent>
</complexType>
<complexType name="fNameRequiredType">
    <complexContent>
        <extension base="rdeCsv:fieldRequiredType">
            <sequence/>
            <attribute name="type" type="token"
                default="eppcom\:labelType"/>
        </extension>
    </complexContent>
</complexType>

<!-- Registry Object Identifier (roid) field -->
<element name="fRoid" type="rdeCsv:fRoidType"
    substitutionGroup="rdeCsv:field"/>
<complexType name="fRoidType">
    <complexContent>
        <extension base="rdeCsv:fieldRequiredType">
            <sequence/>
            <attribute name="type" type="token"
                default="eppcom\:roidType"/>
        </extension>
    </complexContent>
</complexType>

<!-- Registrant field -->
<element name="fRegistrant" type="rdeCsv:fRegistrantType"
    substitutionGroup="rdeCsv:field"/>
<complexType name="fRegistrantType">
    <complexContent>
        <extension base="rdeCsv:fieldOptionalType">
            <sequence/>
            <attribute name="type" type="token"
                default="eppcom\:clIDType"/>
        </extension>
    </complexContent>
</complexType>

<!-- Object Status Description -->
<element name="fStatusDescription"
    type="rdeCsv:fNormalizedStringType"
    substitutionGroup="rdeCsv:field"/>

<!-- clID fields (fClID, fCrID, fUpID) -->
```

```
<!-- Identifier of client that sponsors the object -->
<element name="fClID" type="rdeCsv:fClIDRequiredType"
  substitutionGroup="rdeCsv:field"/>
<!-- Identifier of registrar of client
that created the object -->
<element name="fCrRr" type="rdeCsv:fClIDRequiredType"
  substitutionGroup="rdeCsv:field"/>
<!-- Identifier of client that created the object -->
<element name="fCrID" type="rdeCsv:fClIDType"
  substitutionGroup="rdeCsv:field"/>
<!-- Identifier of registrar of client that
updated the object -->
<element name="fUpRr" type="rdeCsv:fClIDType"
  substitutionGroup="rdeCsv:field"/>
<!-- Identifier of client that updated the object -->
<element name="fUpID" type="rdeCsv:fClIDType"
  substitutionGroup="rdeCsv:field"/>
<!-- Identifier of registrar of client that
requested the transfer -->
<element name="fReRr" type="rdeCsv:fClIDType"
  substitutionGroup="rdeCsv:field"/>
<!-- Identifier of client that requested
the transfer -->
<element name="fReID" type="rdeCsv:fClIDType"
  substitutionGroup="rdeCsv:field"/>
<!-- Identifier of registrar client that
should take or took action -->
<element name="fAcRr" type="rdeCsv:fClIDRequiredType"
  substitutionGroup="rdeCsv:field"/>
<!-- Identifier of client that should take or
took action -->
<element name="fAcID" type="rdeCsv:fClIDType"
  substitutionGroup="rdeCsv:field"/>
<complexType name="fClIDType">
  <complexContent>
    <extension base="rdeCsv:fieldOptionalType">
      <sequence/>
      <attribute name="type" type="token"
        default="eppcom\:clIDType"/>
    </extension>
  </complexContent>
</complexType>
<complexType name="fClIDRequiredType">
  <complexContent>
    <extension base="rdeCsv:fieldRequiredType">
      <sequence/>
      <attribute name="type" type="token"
        default="eppcom\:clIDType"/>
    </extension>
  </complexContent>
</complexType>
```



```

    </extension>
  </complexContent>
</complexType>

<!-- dateTime fields (fCrDate, fUpDate, fExDate) -->
<element name="fCrDate" type="rdeCsv:fRequiredDateTimeType"
  substitutionGroup="rdeCsv:field"/>
<element name="fUpDate" type="rdeCsv:fDateTimeType"
  substitutionGroup="rdeCsv:field"/>
<element name="fExDate" type="rdeCsv:fDateTimeType"
  substitutionGroup="rdeCsv:field"/>
<!-- Date and time that transfer was requested -->
<element name="fReDate" type="rdeCsv:fRequiredDateTimeType"
  substitutionGroup="rdeCsv:field"/>
<!-- Date and time of a required or completed response -->
<element name="fAcDate" type="rdeCsv:fRequiredDateTimeType"
  substitutionGroup="rdeCsv:field"/>
<element name="fTrDate" type="rdeCsv:fDateTimeType"
  substitutionGroup="rdeCsv:field"/>
<complexType name="fDateTimeType">
  <complexContent>
    <extension base="rdeCsv:fieldOptionalType">
      <sequence/>
      <attribute name="type" type="token"
        default="dateTime"/>
    </extension>
  </complexContent>
</complexType>
<complexType name="fRequiredDateTimeType">
  <complexContent>
    <extension base="rdeCsv:fieldRequiredType">
      <sequence/>
      <attribute name="type" type="token"
        default="dateTime"/>
    </extension>
  </complexContent>
</complexType>

<!-- boolean type -->
<complexType name="fBooleanType">
  <complexContent>
    <extension base="rdeCsv:fieldOptionalType">
      <sequence/>
      <attribute name="type" type="token"
        default="boolean"/>
    </extension>
  </complexContent>
</complexType>

```

```
</complexContent>
</complexType>
<complexType name="fRequiredBooleanType">
  <complexContent>
    <extension base="rdeCsv:fieldRequiredType">
      <sequence/>
      <attribute name="type" type="token"
        default="boolean"/>
    </extension>
  </complexContent>
</complexType>

<!-- unsignedByte type -->
<complexType name="fUnsignedByteType">
  <complexContent>
    <extension base="rdeCsv:fieldOptionalType">
      <sequence/>
      <attribute name="type" type="token"
        default="unsignedByte"/>
    </extension>
  </complexContent>
</complexType>
<complexType name="fRequiredUnsignedByteType">
  <complexContent>
    <extension base="rdeCsv:fieldRequiredType">
      <sequence/>
      <attribute name="type" type="token"
        default="unsignedByte"/>
    </extension>
  </complexContent>
</complexType>

<!-- unsignedShort type -->
<complexType name="fUnsignedShortType">
  <complexContent>
    <extension base="rdeCsv:fieldOptionalType">
      <sequence/>
      <attribute name="type" type="token"
        default="unsignedShort"/>
    </extension>
  </complexContent>
</complexType>
<complexType name="fRequiredUnsignedShortType">
  <complexContent>
    <extension base="rdeCsv:fieldRequiredType">
      <sequence/>
      <attribute name="type" type="token"
        default="unsignedShort"/>
    </extension>
  </complexContent>
</complexType>
```

```
</extension>
</complexContent>
</complexType>

<!-- hexBinary type -->
<complexType name="fHexBinaryType">
  <complexContent>
    <extension base="rdeCsv:fieldOptionalType">
      <sequence/>
      <attribute name="type" type="token"
        default="hexBinary"/>
    </extension>
  </complexContent>
</complexType>
<complexType name="fRequiredHexBinaryType">
  <complexContent>
    <extension base="rdeCsv:fieldRequiredType">
      <sequence/>
      <attribute name="type" type="token"
        default="hexBinary"/>
    </extension>
  </complexContent>
</complexType>

<!-- language type -->
<element name="fLang" type="rdeCsv:fLangType"
  substitutionGroup="rdeCsv:field"/>
<complexType name="fLangType">
  <complexContent>
    <extension base="rdeCsv:fieldOptionalType">
      <sequence/>
      <attribute name="type" type="token"
        default="language"/>
    </extension>
  </complexContent>
</complexType>

<!-- IDN Table Identifier -->
<element name="fIdnTableId" type="rdeCsv:fTokenType"
  substitutionGroup="rdeCsv:field"/>

<!-- State of the most recent transfer request -->
<element name="fTrStatus" type="rdeCsv:fTrStatusType"
  substitutionGroup="rdeCsv:field"/>
<complexType name="fTrStatusType">
  <complexContent>
```

```
<extension base="rdeCsv:fieldRequiredType">
  <sequence/>
  <attribute name="type" type="token"
    default="eppcom\:trStatusType"/>
</extension>
</complexContent>
</complexType>

<!-- General token type -->
<complexType name="fTokenType">
  <complexContent>
    <extension base="rdeCsv:fieldOptionalType">
      <sequence/>
      <attribute name="type" type="token"
        default="token"/>
    </extension>
  </complexContent>
</complexType>

<!-- General normalizedString type -->
<complexType name="fNormalizedStringType">
  <complexContent>
    <extension base="rdeCsv:fieldOptionalType">
      <sequence/>
      <attribute name="type" type="token"
        default="normalizedString"/>
    </extension>
  </complexContent>
</complexType>

<!-- positive integer type -->
<complexType name="fPositiveIntegerType">
  <complexContent>
    <extension base="rdeCsv:fieldOptionalType">
      <sequence/>
      <attribute name="type" type="token"
        default="positiveInteger"/>
    </extension>
  </complexContent>
</complexType>

<!-- Custom / extension field type -->
<element name="fCustom" type="rdeCsv:fCustomType"
  substitutionGroup="rdeCsv:field"/>
<complexType name="fCustomType">
  <complexContent>
    <extension base="rdeCsv:fieldOptionalType">
      <sequence/>
```

```
        <attribute name="name" type="token"/>
        <attribute name="type" type="token"
            default="token"/>
    </extension>
</complexContent>
</complexType>

<!-- Ordered list of field definitions for the csv -->
<complexType name="fieldsType">
    <sequence maxOccurs="unbounded">
        <element ref="rdeCsv:field"/>
    </sequence>
</complexType>

<!-- List of files -->
<complexType name="filesType">
    <sequence>
        <element name="file" type="rdeCsv:fileType"
            maxOccurs="unbounded"/>
    </sequence>
</complexType>

<!-- File definition -->
<complexType name="fileType">
    <simpleContent>
        <extension base="token">
            <attribute name="compression" type="token"/>
            <attribute name="encoding" type="token"
                default="UTF-8"/>
            <attribute name="cksum" type="token"/>
        </extension>
    </simpleContent>
</complexType>

<!-- URL fields -->
<element name="fUrl" type="rdeCsv:anyURIType"
    substitutionGroup="rdeCsv:field"/>
<complexType name="anyURIType">
    <complexContent>
        <extension base="rdeCsv:fieldOptionalType">
            <sequence/>
            <attribute name="type" type="token"
                default="anyURI"/>
        </extension>
    </complexContent>
</complexType>

<!--
```

```
End of schema.  
-->  
</schema>  
END
```

9.2. RDE Domain Object

Copyright (c) 2011 IETF Trust and the persons identified as authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- o Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- o Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- o Neither the name of Internet Society, IETF or IETF Trust, nor the names of specific contributors, may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

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

<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="rdeIDN:idType" 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"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

```
<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" minOccurs="0"/>
<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="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" minOccurs="0"
        maxOccurs="unbounded"/>
    </sequence>
  </extension>
</complexContent>
</complexType>
</schema>
END
```

9.3. CSV Domain Object

Copyright (c) 2012 IETF Trust and the persons identified as authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- o Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- o Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- o Neither the name of Internet Society, IETF or IETF Trust, nor the names of specific contributors, may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

BEGIN

```
<?xml version="1.0" encoding="UTF-8"?>

<schema targetNamespace="urn:ietf:params:xml:ns:csvDomain-1.0"
  xmlns:csvDomain="urn:ietf:params:xml:ns:csvDomain-1.0"
  xmlns:rde="urn:ietf:params:xml:ns:rde-1.0"
  xmlns:rdeCsv="urn:ietf:params:xml:ns:rdeCsv-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 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: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"/>
  <import namespace="urn:ietf:params:xml:ns:rdeCsv-1.0"
    schemaLocation="rdeCsv-1.0.xsd"/>

  <annotation>
    <documentation>
      Domain Name Comma-Separated Values (CSV) Object
    </documentation>
  </annotation>

  <!--
  Child elements of the <rde:contents> object
  -->
  <element name="contents" type="csvDomain:contentType"
    substitutionGroup="rde:content"/>

  <complexType name="contentType">
    <complexContent>
      <extension base="rde:contentType">
        <sequence>
          <element ref="rdeCsv:csv" maxOccurs="unbounded"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
</schema>
```

```
</complexContent>
</complexType>

<!--
Child elements of the <rde:deletes> object
-->
<element name="deletes" type="csvDomain:deleteType"
substitutionGroup="rde:delete"/>

<complexType name="deleteType">
  <complexContent>
    <extension base="rde:deleteType">
      <sequence>
        <element ref="rdeCsv:csv" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<!-- Domain name field -->
<element name="fName" type="rdeCsv:fNameRequiredType"
substitutionGroup="rdeCsv:field"/>

<!-- RGP status field -->
<element name="fRgpStatus"
type="csvDomain:fRgpStatusType"
substitutionGroup="rdeCsv:field"/>
<complexType name="fRgpStatusType">
  <complexContent>
    <extension base="rdeCsv:fieldOptionalType">
      <sequence/>
      <attribute name="type" type="token"
default="rgp\:statusValueType"/>
    </extension>
  </complexContent>
</complexType>

<!-- Contact type field -->
<element name="fContactType" type="csvDomain:fContactsTypeType"
substitutionGroup="rdeCsv:field"/>

<complexType name="fContactsTypeType">
  <complexContent>
    <extension base="rdeCsv:fieldRequiredType">
      <sequence/>
    </extension>
  </complexContent>
</complexType>
```

```
        <attribute name="type" type="token"
          default="domain\:contactAttrType"/>
      </extension>
    </complexContent>
  </complexType>

<!-- DNSSEC field types -->

<!-- Maximum signature lifetime field -->
<element name="fMaxSigLife" type="csvDomain:fMaxSigLifeType"
  substitutionGroup="rdeCsv:field"/>
<complexType name="fMaxSigLifeType">
  <complexContent>
    <extension base="rdeCsv:fieldOptionalType">
      <sequence/>
      <attribute name="type" type="token"
        default="secDNS\:maxSigLifeType"/>
    </extension>
  </complexContent>
</complexType>

<!-- Key tag field -->
<element name="fKeyTag" type="rdeCsv:fRequiredUnsignedShortType"
  substitutionGroup="rdeCsv:field"/>

<!-- DS Algorithm field -->
<element name="fDsAlg" type="rdeCsv:fRequiredUnsignedByteType"
  substitutionGroup="rdeCsv:field"/>

<!-- Digest type field -->
<element name="fDigestType" type="rdeCsv:fRequiredUnsignedByteType"
  substitutionGroup="rdeCsv:field"/>

<!-- Digest field -->
<element name="fDigest" type="rdeCsv:fRequiredHexBinaryType"
  substitutionGroup="rdeCsv:field"/>

<!-- Flags field -->
<element name="fFlags" type="rdeCsv:fRequiredUnsignedShortType"
  substitutionGroup="rdeCsv:field"/>

<!-- Protocol field -->
<element name="fProtocol" type="rdeCsv:fRequiredUnsignedByteType"
  substitutionGroup="rdeCsv:field"/>

<!-- Key Algorithm field -->
<element name="fKeyAlg" type="rdeCsv:fRequiredUnsignedByteType"
  substitutionGroup="rdeCsv:field"/>
```

```
<!-- Public Key field -->
<element name="fPubKey" type="csvDomain:fPubKeyType"
  substitutionGroup="rdeCsv:field"/>
<complexType name="fPubKeyType">
  <complexContent>
    <extension base="rdeCsv:fieldRequiredType">
      <sequence/>
      <attribute name="type" type="token"
        default="secDNS\:keyType"/>
    </extension>
  </complexContent>
</complexType>

<!-- Original Domain Name for Variant field -->
<element name="fOriginalName" type="rdeCsv:fNameType"
  substitutionGroup="rdeCsv:field"/>

<!-- Variant group / tag field -->
<element name="fVariantGroup"
  type="rdeCsv:fTokenType"
  substitutionGroup="rdeCsv:field"/>

<!-- Domain status field -->
<element name="fStatus" type="csvDomain:fStatusType"
  substitutionGroup="rdeCsv:field"/>

<!-- Domain status based on domain-1.0.xsd -->
<complexType name="fStatusType">
  <complexContent>
    <extension base="rdeCsv:fieldRequiredType">
      <sequence/>
      <attribute name="type" type="token"
        default="domain\:statusValueType"/>
    </extension>
  </complexContent>
</complexType>

<!--
End of schema.
-->
</schema>
END
```

9.4. RDE Host Object

Copyright (c) 2011 IETF Trust and the persons identified as authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- o Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- o Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- o Neither the name of Internet Society, IETF or IETF Trust, nor the names of specific contributors, may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

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

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

```
<element name="abstractHost" type="rdeHost:abstractContentType"
  substitutionGroup="rde:content" abstract="true"/>
<element name="host" substitutionGroup="rdeHost:abstractHost" />
<element name="delete" type="rdeHost: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="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">
      <choice minOccurs="0" maxOccurs="unbounded">
        <element name="name"
          type="eppcom:labelType"/>
        <element name="roid"
          type="eppcom:roidType"/>
      </choice>
    </extension>
  </complexContent>
</complexType>
```

```
    </complexContent>
  </complexType>
</schema>
END
```

9.5. CSV Host Object

Copyright (c) 2012 IETF Trust and the persons identified as authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- o Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- o Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- o Neither the name of Internet Society, IETF or IETF Trust, nor the names of specific contributors, may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

BEGIN

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<schema targetNamespace="urn:ietf:params:xml:ns:csvHost-1.0"
  xmlns:csvHost="urn:ietf:params:xml:ns:csvHost-1.0"
  xmlns:rde="urn:ietf:params:xml:ns:rde-1.0"
  xmlns:rdeCsv="urn:ietf:params:xml:ns:rdeCsv-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 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:host-1.0"
  schemaLocation="host-1.0.xsd"/>
<import namespace="urn:ietf:params:xml:ns:rde-1.0"
  schemaLocation="rde-1.0.xsd"/>
<import namespace="urn:ietf:params:xml:ns:rdeCsv-1.0"
  schemaLocation="rdeCsv-1.0.xsd"/>

<annotation>
  <documentation>
    Host Comma-Separated Values (CSV) Object
  </documentation>
</annotation>

<!--
Child elements of the <rde:contents> object
-->
<element name="contents" type="csvHost:contentType"
  substitutionGroup="rde:content"/>

<complexType name="contentType">
  <complexContent>
    <extension base="rde:contentType">
      <sequence>
        <element ref="rdeCsv:csv" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<!--
Child elements of the <rde:deletes> object
-->
<element name="deletes" type="csvHost:deleteType"
  substitutionGroup="rde:delete"/>

<complexType name="deleteType">
  <complexContent>
    <extension base="rde:deleteType">
      <sequence>
        <element ref="rdeCsv:csv" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

```
        </sequence>
      </extension>
    </complexContent>
  </complexType>

  <!-- Host name field -->
  <element name="fName" type="rdeCsv:fNameRequiredType"
    substitutionGroup="rdeCsv:field"/>

  <!-- IP address field -->
  <element name="fAddr" type="csvHost:fAddrType"
    substitutionGroup="rdeCsv:field"/>
  <complexType name="fAddrType">
    <complexContent>
      <extension base="rdeCsv:fieldOptionalType">
        <sequence/>
        <attribute name="type" type="token"
          default="host\:addrStringType"/>
      </extension>
    </complexContent>
  </complexType>

  <!-- IP address version field -->
  <element name="fAddrVersion" type="csvHost:fAddrVersionType"
    substitutionGroup="rdeCsv:field"/>
  <complexType name="fAddrVersionType">
    <complexContent>
      <extension base="rdeCsv:fieldOptionalType">
        <sequence/>
        <attribute name="type" type="token"
          default="host\:ipType"/>
      </extension>
    </complexContent>
  </complexType>

  <!-- Host status field -->
  <element name="fStatus" type="csvHost:fStatusType"
    substitutionGroup="rdeCsv:field"/>

  <!-- Host status based on host-1.0.xsd -->
  <complexType name="fStatusType">
    <complexContent>
      <extension base="rdeCsv:fieldRequiredType">
        <sequence/>
        <attribute name="type" type="token"
          default="host\:statusValueType"/>
      </extension>
    </complexContent>
  </complexType>
```

```
        </extension>
      </complexContent>
    </complexType>

    <!--
      End of schema.
    -->
  </schema>
END
```

9.6. RDE Contact Object

Copyright (c) 2011 IETF Trust and the persons identified as authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- o Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- o Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- o Neither the name of Internet Society, IETF or IETF Trust, nor the names of specific contributors, may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

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

<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:el64Type" minOccurs="0"/>
        <element name="fax"
          type="contact:el64Type" 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

```

9.7. CSV Contact Object

Copyright (c) 2012 IETF Trust and the persons identified as authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- o Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- o Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- o Neither the name of Internet Society, IETF or IETF Trust, nor the names of specific contributors, may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

BEGIN

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<schema targetNamespace="urn:ietf:params:xml:ns:csvContact-1.0"
  xmlns:csvContact="urn:ietf:params:xml:ns:csvContact-1.0"
  xmlns:rde="urn:ietf:params:xml:ns:rde-1.0"
  xmlns:rdeCsv="urn:ietf:params:xml:ns:rdeCsv-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"/>
<import namespace="urn:ietf:params:xml:ns:rdeCsv-1.0"
  schemaLocation="rdeCsv-1.0.xsd"/>

<annotation>
  <documentation>
    Contact Comma-Separated Values (CSV) Object
  </documentation>
</annotation>

<!--
Child elements of the <rde:contents> object
-->
<element name="contents" type="csvContact:contentType"
  substitutionGroup="rde:content"/>

<complexType name="contentType">
  <complexContent>
    <extension base="rde:contentType">
      <sequence>
        <element ref="rdeCsv:csv" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<!--
Child elements of the <rde:deletes> object
-->
<element name="deletes" type="csvContact:deleteType"
  substitutionGroup="rde:delete"/>

<complexType name="deleteType">
  <complexContent>
    <extension base="rde:deleteType">
      <sequence>
        <element ref="rdeCsv:csv" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

```
<!-- Server-unique contact identifier field -->
<element name="fId" type="csvContact:fIdType"
  substitutionGroup="rdeCsv:field"/>
<complexType name="fIdType">
  <complexContent>
    <extension base="rdeCsv:fieldRequiredType">
      <sequence/>
      <attribute name="type" type="token"
        default="eppcom\:clIDType"/>
    </extension>
  </complexContent>
</complexType>

<!-- Is Registrar Contact field -->
<element name="fIsRegistrarContact"
  type="rdeCsv:fBooleanType"
  substitutionGroup="rdeCsv:field"/>

<!-- voice and fax telephone number fields -->
<element name="fVoice" type="csvContact:fE164StringType"
  substitutionGroup="rdeCsv:field"/>
<element name="fFax" type="csvContact:fE164StringType"
  substitutionGroup="rdeCsv:field"/>
<complexType name="fE164StringType">
  <complexContent>
    <extension base="rdeCsv:fieldOptionalType">
      <sequence/>
      <attribute name="type" type="token"
        default="contact\:e164StringType"/>
    </extension>
  </complexContent>
</complexType>

<!-- voice and fax telephone extension fields -->
<element name="fVoiceExt" type="rdeCsv:fTokenType"
  substitutionGroup="rdeCsv:field"/>
<element name="fFaxExt" type="rdeCsv:fTokenType"
  substitutionGroup="rdeCsv:field"/>

<!-- contact email address field -->
<element name="fEmail" type="csvContact:fEmailType"
  substitutionGroup="rdeCsv:field"/>
<complexType name="fEmailType">
  <complexContent>
    <extension base="rdeCsv:fieldRequiredType">
      <sequence/>
    </extension>
  </complexContent>
</complexType>
```



```
        <attribute name="type" type="token"
        default="eppcom\:minTokenType"/>
    </extension>
</complexContent>
</complexType>

<!--
    Postal type field
    ("loc" = localized, "int" = internationalized)
-->
<element name="fPostalType" type="csvContact:fPostalTypeType"
    substitutionGroup="rdeCsv:field"/>
<complexType name="fPostalTypeType">
    <complexContent>
        <extension base="rdeCsv:fieldRequiredType">
            <sequence/>
            <attribute name="type" type="token"
            default="contact\:postalInfoEnumType"/>
        </extension>
    </complexContent>
</complexType>

<!-- Standard postal line field -->
<complexType name="fPostalLineType">
    <complexContent>
        <extension base="rdeCsv:fieldRequiredType">
            <sequence/>
            <attribute name="type" type="token"
            default="contact\:postalLineType"/>
            <attribute name="isLoc" type="boolean"/>
        </extension>
    </complexContent>
</complexType>

<!-- Standard optional postal line field -->
<complexType name="fOptPostalLineType">
    <complexContent>
        <extension base="rdeCsv:fieldOptionalType">
            <sequence/>
            <attribute name="type" type="token"
            default="contact\:optPostalLineType"/>
            <attribute name="isLoc" type="boolean"/>
        </extension>
    </complexContent>
</complexType>
```

```
<!-- Name of the individual or role field -->
<element name="fName" type="csvContact:fPostalLineType"
  substitutionGroup="rdeCsv:field"/>

<!-- Name organization field -->
<element name="fOrg" type="csvContact:fOptPostalLineType"
  substitutionGroup="rdeCsv:field"/>

<!-- Street address line field with required index attribute -->
<!-- starting with index 0. -->
<element name="fStreet" type="csvContact:fStreetType"
  substitutionGroup="rdeCsv:field"/>
<complexType name="fStreetType">
  <complexContent>
    <extension base="csvContact:fOptPostalLineType">
      <sequence/>
      <attribute name="index" type="int"
        use="required"/>
    </extension>
  </complexContent>
</complexType>

<!-- Contact's city field -->
<element name="fCity" type="csvContact:fPostalLineType"
  substitutionGroup="rdeCsv:field"/>

<!-- Contact's state or province field -->
<element name="fSp" type="csvContact:fOptPostalLineType"
  substitutionGroup="rdeCsv:field"/>

<!-- Contact's postal code field -->
<element name="fPc" type="csvContact:fPcType"
  substitutionGroup="rdeCsv:field"/>
<complexType name="fPcType">
  <complexContent>
    <extension base="rdeCsv:fieldOptionalType">
      <sequence/>
      <attribute name="type" type="token"
        default="contact\:pcType"/>
      <attribute name="isLoc" type="boolean"/>
    </extension>
  </complexContent>
</complexType>

<!-- Contact's country code field -->
<element name="fCc" type="csvContact:fCcType"
```

```
    substitutionGroup="rdeCsv:field"/>
<complexType name="fCcType">
  <complexContent>
    <extension base="rdeCsv:fieldRequiredType">
      <sequence/>
      <attribute name="type" type="token"
        default="contact\:ccType"/>
      <attribute name="isLoc" type="boolean"/>
    </extension>
  </complexContent>
</complexType>

<!-- Disclosure element fields -->
<!-- Flag of "1" to allow disclosure
    and "0" to disallow disclosure -->
<element name="fDiscloseFlag" type="csvContact:fBoolean"
  substitutionGroup="rdeCsv:field"/>
<!-- Disclosure of localized name
    based on fDiscloseFlag? -->
<element name="fDiscloseNameLoc" type="csvContact:fBoolean"
  substitutionGroup="rdeCsv:field"/>
<!-- Disclosure of internationalized name
    based on fDiscloseFlag? -->
<element name="fDiscloseNameInt" type="csvContact:fBoolean"
  substitutionGroup="rdeCsv:field"/>
<!-- Disclosure of localized org
    based on fDiscloseFlag? -->
<element name="fDiscloseOrgLoc" type="csvContact:fBoolean"
  substitutionGroup="rdeCsv:field"/>
<!-- Disclosure of internationalized org
    based on fDiscloseFlag? -->
<element name="fDiscloseOrgInt" type="csvContact:fBoolean"
  substitutionGroup="rdeCsv:field"/>
<!-- Disclosure of localized address
    based on fDiscloseFlag? -->
<element name="fDiscloseAddrLoc" type="csvContact:fBoolean"
  substitutionGroup="rdeCsv:field"/>
<!-- Disclosure of internationalized address
    based on fDiscloseFlag? -->
<element name="fDiscloseAddrInt" type="csvContact:fBoolean"
  substitutionGroup="rdeCsv:field"/>
<!-- Disclosure voice telephone number
    based on fDiscloseFlag? -->
<element name="fDiscloseVoice" type="csvContact:fBoolean"
  substitutionGroup="rdeCsv:field"/>
<!-- Disclosure facsimile telephone number
    based on fDiscloseFlag? -->
<element name="fDiscloseFax" type="csvContact:fBoolean"
```

```
    substitutionGroup="rdeCsv:field"/>
<!-- Disclosure email address
    based on fDiscloseFlag? -->
<element name="fDiscloseEmail" type="csvContact:fBoolean"
    substitutionGroup="rdeCsv:field"/>
<complexType name="fBoolean">
  <complexContent>
    <extension base="rdeCsv:fieldOptionalType">
      <sequence/>
      <attribute name="type" type="token"
        default="boolean"/>
    </extension>
  </complexContent>
</complexType>

<!-- Contact status field -->
<element name="fStatus" type="csvContact:fStatusType"
    substitutionGroup="rdeCsv:field"/>

<!-- Host status based on contact-1.0.xsd -->
<complexType name="fStatusType">
  <complexContent>
    <extension base="rdeCsv:fieldRequiredType">
      <sequence/>
      <attribute name="type" type="token"
        default="contact\:statusValueType"/>
    </extension>
  </complexContent>
</complexType>

<!--
End of schema.
-->
</schema>
END
```

9.8. RDE Registrar Object

Copyright (c) 2011 IETF Trust and the persons identified as authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- o Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- o Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- o Neither the name of Internet Society, IETF or IETF Trust, nor the names of specific contributors, may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

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"/>
  <import namespace="urn:ietf:params:xml:ns:domain-1.0"/>
  <import namespace="urn:ietf:params:xml:ns:contact-1.0"/>
  <import namespace="urn:ietf:params:xml:ns:rde-1.0"/>

  <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" minOccurs="0"/>
        <element name="postalInfo"
          type="rdeRegistrar:postalInfoType"
          minOccurs="0" maxOccurs="2"/>
        <element name="voice"
          type="contact:el64Type" minOccurs="0"/>
        <element name="fax"
          type="contact:el64Type" minOccurs="0"/>
        <element name="email"
          type="eppcom:minTokenType" minOccurs="0"/>
        <element name="url"
          type="anyURI" minOccurs="0"/>
        <element name="whoisInfo"
          type="rdeRegistrar:whoisInfoType" minOccurs="0"/>
        <element name="crDate"
          type="dateTime" minOccurs="0"/>
        <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
```

9.9. CSV Registrar Object

Copyright (c) 2012 IETF Trust and the persons identified as authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- o Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- o Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- o Neither the name of Internet Society, IETF or IETF Trust, nor the names of specific contributors, may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

BEGIN

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<schema targetNamespace="urn:ietf:params:xml:ns:csvRegistrar-1.0"
  xmlns:csvRegistrar="urn:ietf:params:xml:ns:csvRegistrar-1.0"
  xmlns:rde="urn:ietf:params:xml:ns:rde-1.0"
  xmlns:rdeCsv="urn:ietf:params:xml:ns:rdeCsv-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"/>
<import namespace="urn:ietf:params:xml:ns:rdeCsv-1.0"
        schemaLocation="rdeCsv-1.0.xsd"/>

<annotation>
  <documentation>
    Registrar Comma-Separated Values (CSV) Object
  </documentation>
</annotation>

<!--
Child elements of the <rde:contents> object
-->
<element name="contents" type="csvRegistrar:contentType"
substitutionGroup="rde:content"/>

<complexType name="contentType">
  <complexContent>
    <extension base="rde:contentType">
      <sequence>
        <element ref="rdeCsv:csv" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<!--
Child elements of the <rde:deletes> object
-->
<element name="deletes" type="csvRegistrar:deleteType"
substitutionGroup="rde:delete"/>

<complexType name="deleteType">
  <complexContent>
    <extension base="rde:deleteType">
      <sequence>
        <element ref="rdeCsv:csv" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<!-- Registrar unique identifier (short name / id) -->
```

```
<element name="fId" type="rdeCsv:fClIDRequiredType"
  substitutionGroup="rdeCsv:field"/>

<!-- Registrar name (full name) -->
<element name="fName" type="csvRegistrar:fNameType"
  substitutionGroup="rdeCsv:field"/>

<!-- Registrar name field -->
<complexType name="fNameType">
  <complexContent>
    <extension base="rdeCsv:fieldRequiredType">
      <sequence/>
      <attribute name="type" type="token"
        default="normalizedString"/>
      <attribute name="isLoc" type="boolean" default="false"/>
    </extension>
  </complexContent>
</complexType>

<!-- Registrar GURID field -->
<element name="fGurid"
  type="rdeCsv:fPositiveIntegerType"
  substitutionGroup="rdeCsv:field"/>

<!-- Registrar status field -->
<element name="fStatus" type="csvRegistrar:fStatusType"
  substitutionGroup="rdeCsv:field"/>
<element name="fStatusName" type="rdeCsv:fTokenType"
  substitutionGroup="rdeCsv:field"/>
<complexType name="fStatusType">
  <complexContent>
    <extension base="rdeCsv:fieldRequiredType">
      <sequence/>
      <attribute name="type" type="token"
        default="csvRegistrar\:statusType"/>
    </extension>
  </complexContent>
</complexType>

<!-- Registrar status type with optional name attr -->
<complexType name="statusType">
  <simpleContent>
    <extension base="csvRegistrar:statusValueType">
      <attribute name="name" type="token"/>
    </extension>
  </simpleContent>
</complexType>
```

```
<!-- Registrar status enumerated values -->
<simpleType name="statusValueType">
  <restriction base="token">
    <enumeration value="ok"/>
    <enumeration value="readonly"/>
    <enumeration value="terminated"/>
  </restriction>
</simpleType>

<!-- Whois URL field -->
<element name="fWhoisUrl"
  type="rdeCsv:anyURIType"
  substitutionGroup="rdeCsv:field"/>

<!--
End of schema.
-->
</schema>
END
```

9.10. RDE IDN Table Reference Objects

Copyright (c) 2011 IETF Trust and the persons identified as authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- o Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- o Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- o Neither the name of Internet Society, IETF or IETF Trust, nor the names of specific contributors, may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,

SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

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"/>

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

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

  <!-- Content Types -->
  <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>

  <complexType name="deleteType">
    <complexContent>
      <extension base="rde:deleteType">
        <sequence>
          <element name="id" type="rdeIDN:idType"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
```

```
    </complexContent>
  </complexType>

  <!-- Simple Types -->
  <simpleType name="idType">
    <restriction base="token">
      <minLength value="1"/>
      <maxLength value="64"/>
    </restriction>
  </simpleType>

</schema>
END
```

9.11. CSV IDN Language Object

Copyright (c) 2012 IETF Trust and the persons identified as authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- o Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- o Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- o Neither the name of Internet Society, IETF or IETF Trust, nor the names of specific contributors, may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

```
BEGIN
<?xml version="1.0" encoding="UTF-8"?>

<schema targetNamespace="urn:ietf:params:xml:ns:csvIDN-1.0"
  xmlns:csvIDN="urn:ietf:params:xml:ns:csvIDN-1.0"
  xmlns:rde="urn:ietf:params:xml:ns:rde-1.0"
  xmlns:rdeCsv="urn:ietf:params:xml:ns:rdeCsv-1.0"
  xmlns="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">

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

  <annotation>
    <documentation>
      IDN Language Comma-Separated Values (CSV) Object
    </documentation>
  </annotation>

  <!--
  Child elements of the <rde:contents> object
  -->
  <element name="contents" type="csvIDN:contentType"
    substitutionGroup="rde:content"/>

  <complexType name="contentType">
    <complexContent>
      <extension base="rde:contentType">
        <sequence>
          <element ref="rdeCsv:csv" maxOccurs="unbounded"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>

  <!--
  Child elements of the <rde:deletes> object
  -->
  <element name="deletes" type="csvIDN:deleteType"
    substitutionGroup="rde:delete"/>

  <complexType name="deleteType">
    <complexContent>
```

```
<extension base="rde:deleteType">
  <sequence>
    <element ref="rdeCsv:csv" maxOccurs="unbounded"/>
  </sequence>
</extension>
</complexContent>
</complexType>

<!--
End of schema.
-->
</schema>
END
```

9.12. EPP Parameters Object

Copyright (c) 2011 IETF Trust and the persons identified as authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- o Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- o Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- o Neither the name of Internet Society, IETF or IETF Trust, nor the names of specific contributors, may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.


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

  <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

9.13. NNDN Object

Copyright (c) 2011 IETF Trust and the persons identified as authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- o Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- o Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- o Neither the name of Internet Society, IETF or IETF Trust, nor the names of specific contributors, may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

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:rdeIDN="urn:ietf:params:xml:ns:rdeIDN-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"/>
  <import namespace="urn:ietf:params:xml:ns:rde-1.0"/>
```

```
<import namespace="urn:ietf:params:xml:ns:rdeIDN-1.0"/>

<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="rdeIDN:idType" 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="nameStateValue">
  <restriction base="token">
    <enumeration value="withheld" />
    <enumeration value="blocked" />
    <enumeration value="mirrored" />
  </restriction>
</simpleType>

<complexType name="nameState">
  <simpleContent>
    <extension base="rdeNNDN:nameStateValue">
      <attribute name="mirroringNS" />
    </extension>
  </simpleContent>
</complexType>
```

```

        type="boolean" default="true"/>
      </extension>
    </simpleContent>
  </complexType>

  <!-- 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

```

9.14. CSV NNDN Object

Copyright (c) 2013 IETF Trust and the persons identified as authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- o Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- o Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- o Neither the name of Internet Society, IETF or IETF Trust, nor the names of specific contributors, may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT

LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

BEGIN

```
<?xml version="1.0" encoding="UTF-8"?>

<schema targetNamespace="urn:ietf:params:xml:ns:csvNNDN-1.0"
  xmlns:csvNNDN="urn:ietf:params:xml:ns:csvNNDN-1.0"
  xmlns:rde="urn:ietf:params:xml:ns:rde-1.0"
  xmlns:rdeCsv="urn:ietf:params:xml:ns:rdeCsv-1.0"
  xmlns:rdeNNDN="urn:ietf:params:xml:ns:rdeNNDN-1.0"
  xmlns="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">

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

  <annotation>
    <documentation>
      NNDN (NNDN's not domain name) (CSV) Object
    </documentation>
  </annotation>

  <!--
  Child elements of the <rde:contents> object
  -->
  <element name="contents" type="csvNNDN:contentType"
    substitutionGroup="rde:content"/>

  <complexType name="contentType">
    <complexContent>
      <extension base="rde:contentType">
        <sequence>
          <element ref="rdeCsv:csv" maxOccurs="unbounded"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
```

```
<!--
Child elements of the <rde:deletes> object
-->
<element name="deletes" type="csvNNDN:deleteType"
substitutionGroup="rde:delete"/>

<complexType name="deleteType">
  <complexContent>
    <extension base="rde:deleteType">
      <sequence>
        <element ref="rdeCsv:csv" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<!-- ASCII Compatible Encoding (ACE) name field -->
<element name="fAName" type="rdeCsv:fNameRequiredType"
substitutionGroup="rdeCsv:field"/>

<!-- domain name used to generate the IDN variant field -->
<element name="fOriginalName" type="rdeCsv:fNameType"
substitutionGroup="rdeCsv:field"/>

<!-- Variant group / tag field -->
<element name="fVariantGroup"
type="rdeCsv:fTokenType"
substitutionGroup="rdeCsv:field"/>

<!-- RGP status field -->
<element name="fNameState"
type="csvNNDN:fNameStateType"
substitutionGroup="rdeCsv:field"/>
<complexType name="fNameStateType">
  <complexContent>
    <extension base="rdeCsv:fieldRequiredType">
      <sequence/>
      <attribute name="type" type="token"
default="rdeNNDN\:nameState"/>
    </extension>
  </complexContent>
</complexType>

<!-- Mirroring uses NS mirror mechanism? -->
<element name="fMirroringNS"
type="rdeCsv:fBooleanType"
substitutionGroup="rdeCsv:field"/>
```

```
<!--  
End of schema.  
-->  
</schema>  
END
```

9.15. Policy Object

Copyright (c) 2011 IETF Trust and the persons identified as authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- o Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- o Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- o Neither the name of Internet Society, IETF or IETF Trust, nor the names of specific contributors, may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

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

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

9.16. Header Object

Copyright (c) 2011 IETF Trust and the persons identified as authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- o Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- o Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- o Neither the name of Internet Society, IETF or IETF Trust, nor the names of specific contributors, may be used to endorse or promote

products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

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

  <annotation>
    <documentation>
      Data Escrow Deposit 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>
          <group ref="rdeHeader:repositoryTypeGroup"/>
          <element name="count" type="rdeHeader:countType"
            maxOccurs="unbounded"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
```

```
</complexType>

<group name="repositoryTypeGroup">
  <choice>
    <element name="tld" type="eppcom:labelType" />
    <element name="registrar" type="positiveInteger"/>
    <element name="ppsp" type="token"/>
  </choice>
</group>

<complexType name="countType">
  <simpleContent>
    <extension base="long">
      <attribute name="uri" type="anyURI" use="required"/>
      <attribute name="rcdn" type="eppcom:labelType"/>
      <attribute name="registrarId" type="positiveInteger"/>
    </extension>
  </simpleContent>
</complexType>

</schema>
END
```

10. 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.

11. 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 CSV namespace:

URI: urn:ietf:params:xml:ns:rdeCsv-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 CSV XML schema:

URI: urn:ietf:params:xml:schema:rdeCsv-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 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 CSV domain namespace:

URI: urn:ietf:params:xml:ns:csvDomain-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 CSV domain XML schema:

URI: urn:ietf:params:xml:schema:csvDomain-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 CSV host namespace:

URI: urn:ietf:params:xml:ns:csvHost-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 CSV host XML schema:

URI: urn:ietf:params:xml:schema:csvHost-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 CSV contact namespace:

URI: urn:ietf:params:xml:ns:csvContact-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 CSV contact XML schema:

URI: urn:ietf:params:xml:schema:csvContact-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 CSV registrar namespace:

URI: urn:ietf:params:xml:ns:csvRegistrar-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 CSV registrar XML schema:

URI: urn:ietf:params:xml:schema:csvRegistrar-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 CSV IDN namespace:

URI: urn:ietf:params:xml:ns:csvIDN-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 CSV IDN XML schema:

URI: urn:ietf:params:xml:schema:csvIDN-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 NNDN namespace:

URI: urn:ietf:params:xml:ns:rdeNNDN-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 NNDN XML schema:

URI: urn:ietf:params:xml:schema:rdeNNDN-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 CSV NNDN namespace:

URI: urn:ietf:params:xml:ns:csvNNDN-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 CSV NNDN XML schema:

URI: urn:ietf:params:xml:schema:csvNNDN-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.

12. Implementation Status

Note to RFC Editor: Please remove this section and the reference to RFC 7942 [RFC7942] before publication.

This section records the status of known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft, and is based on a proposal described in RFC 7942 [RFC7942]. The description of implementations in this section is intended to assist the IETF in its decision processes in progressing drafts to RFCs. Please note that the listing of any individual implementation here does not imply endorsement by the IETF. Furthermore, no effort has been spent to verify the information presented here that was supplied by IETF contributors. This is not intended as, and must not be construed to be, a catalog of available implementations or their features. Readers are advised to note that other implementations may exist.

According to RFC 7942 [RFC7942], "this will allow reviewers and working groups to assign due consideration to documents that have the benefit of running code, which may serve as evidence of valuable experimentation and feedback that have made the implemented protocols more mature. It is up to the individual working groups to use this information as they see fit".

12.1. Implementation in the gTLD space

Organization: ICANN

Name: ICANN Registry Agreement

Description: the ICANN Base Registry Agreement requires Registries, Data Escrow Agents, and ICANN to implement this specification. ICANN receives daily notifications from Data Escrow Agents confirming that more than 1,200 gTLDs are sending deposits that comply with this specification. ICANN receives on a weekly basis per gTLD, from more than 1,200 gTLD registries, a Bulk Registration Data Access file that also complies with this specification. In addition, ICANN is aware of Registry Service Provider transitions using data files that conform to this specification.

Level of maturity: production.

Coverage: all aspects of this specification are implemented.

Version compatibility: versions 03 - 08 are known to be implemented.

Contact: gustavo.lozano@icann.org

URL: <https://www.icann.org/resources/pages/registries/registries-agreements-en>

13. 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".

14. Acknowledgments

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

Shoji Noguchi and Francisco Arias participated as co-authors until version 05 providing invaluable support for this document.

15. Change History

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

- 15.1. 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.
- 15.2. 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.
- 15.3. 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.
- 15.4. Changes from version 02 to 03
1. Remove authinfo from the XML Schema.
 2. Resend attribute is now an element
 3. Scope attribute added to policy object.
- 15.5. Changes from version 03 to 04
1. Merged draft-gould-thippeswamy-dnr-d-csv-mapping-03 into draft-arias-noguchi-dnr-d-objects-mapping-02.
 2. Changed the cksum attribute of <rdeCsv:file> to use CRC32 and changed all of the sample cksum values to use CRC32, based on feedback from David Kipling.
 3. Changed the optional <rdeCsv:sep> element to be an optional "sep" attribute value of the <rdeCsv:csv> element with a default value of "," based on feedback from David Kipling.
 4. Added support for the optional "parent" attribute for the to the CSV fields to indicate a field as a reference to a parent object, based on feedback from David Kipling.
 5. Added support for the CSV model for the NNDN.
 6. Added support to delete hosts based on roid.
 7. Added mirrored state to NNDN
 8. Minor fixes to XML XSDs.
 9. The Report and Notification objects were moved to draft-lozano-icann-registry-interfaces
 10. The section Data escrow notifications was moved to draft-lozano-icann-registry-interfaces

11. Removed references to the `<rdeCsv:fCrRr>`, `<rdeCsv:fCrID>`, and `<rdeCsv:fCrDate>` from the "hostStatuses" and "hostAddresses" CSV files.
 12. Removed references to the `<rdeCsv:fCrRr>`, `<rdeCsv:fCrID>`, and `<rdeCsv:fCrDate>` from the "contactStatuses" CSV file.
 13. Removed references to the `<rdeCsv:fCrRr>`, `<rdeCsv:fCrID>`, and `<rdeCsv:fCrDate>` from the "domainContacts", "domainStatuses", and "domainNameServers" CSV files.
 14. Changed `<rdeCsv:fLanguage>` to `<rdeCsv:fLang>`.
 15. Replaced use of `<rdeCsv:fLang>` to new `<rdeCsv:fIdnTableId>` field in the "domain", "idnLanguage", and "NNDN" CSV files.
 16. Replaced use of `<csvHost:fName>` with `<rdeCsv:fRoid>` in the "host" `<csvHost:deletes>` `<rdeCsv:csv>` element.
 17. Changed the foreign key of the hosts to use `<rdeCsv:fRoid>` instead of `<csvHost:fName>` and removed use of `<csvHost:fName>` in the "domainNameServers", "hostStatuses", and "hostAddresses" CSV files.
 18. Added use of the MUST keyword for CSV fields that are required to be supported in an EPP based system.
 19. Removed use of the `<rdeCsv:fRoid>` field element for the "registrar" CSV file.
 20. Added definition of `<csvNNDN:fMirroringNS>` field element.
- 15.6. Changes from version 04 to 05
1. Updated the examples of the full and differential deposits using the CSV and XML model.
 2. Made `<rdeCsv:fExDate>` optional for the "domainTransfer" CSV file to match the XML definition.
 3. Made `<csvDomain:fOriginalName>` optional for the "domain" CSV file to match the XML definition.
 4. Made `<rdeCsv:fTrDate>` optional for the "domain" and "contact" CSV files to match the XML definition.
 5. Change `<idnTableId>` from IDREF to idType.

6. Minor editorial changes.

15.7. Changes from version 05 to 06

1. Revised the differential and incremental deposits for the CSV format to use cascade update / replace and delete from the parent object to be consistent with the XML format.
2. Revised the structure of the CSV format sections to utilize sub-sections instead of a list for the CSV file definitions.
3. Added the "CSV Parent Child Relationship" section to describe the concept of parent child relationships across CSV file definitions.
4. Added the "domainNameServersAddresses" CSV File Definition section to support the domain host attributes model of [RFC5731].
5. Made the required fields in the CSV format consistent with the XML format. The CSV fields updated to be required include:
<rdeCsv:fCrDate>, <csvDomain:fContactType>, <csvDomain:fStatus>, <csvDomain:fKeyTag>, <csvDomain:fDsAlg>, <csvDomain:fDigestType>, <csvDomain:fDigest>, <csvDomain:fFlags>, <csvDomain:fProtocol>, <csvDomain:fKeyAlg>, <csvDomain:fPubKey>, <rdeCsv:fTrStatus>, <rdeCsv:fReRr>, <rdeCsv:fReDate>, <rdeCsv:fAcRr>, <rdeCsv:fAcDate>, <csvHost:fStatus>, <csvContact:fCc>, <csvContact:fStatus>, <csvContact:fPostalType>, <csvRegistrar:fStatus>, and <csvNNDN:fNameState>.
6. Revised the CSV examples to use a more realistic set of records.

15.8. Changes from version 06 to 07

1. Created "repositoryTypeGroup" group element in the rdeHeader including the <rdeHeader:registrar>, <rdeHeader:ppsp> and <rdeHeader:tld> elements.
2. Added the optional "rcdn" and "registrarId" attributes to the <rdeHeader:count> element

15.9. Changes from version 07 to 08

1. The following registrar elements were made optional to support greater flexibility for the implementation of policies: status, postalInfo, email and crDate.
2. The following domain name elements were made optional to support greater flexibility for the implementation of policies: crRr.

15.10. Changes from version 08 to 09

1. Implementation Status section was added

15.11. Changes from version 09 to 10

1. Editorial changes in section Section 5.1.2.1.6.
2. Added MAY clause when the DS Data Interface is used in section Section 5.1.2.1.6.

16. Example of a full deposit using the XML model

Example of a full deposit using the XML model:

```
<?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--exempl-gva.test</rdeNNDN:aName>
  <rdeNNDN:idnTableId>pt-BR</rdeNNDN:idnTableId>
  <rdeNNDN:originalName>example1.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
  scope="//rde:deposit/rde:contents/rdeDomain:domain"
  element="rdeDom:registrant" />
</rde:contents>
</rde:deposit>

```

17. Example of differential deposit using the XML model

Example of a differential deposit using the XML model:

```

<?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>

```

18. Example of a full deposit using the CSV model

Example of a full deposit using the CSV model:

```

<?xml version="1.0" encoding="UTF-8"?>
<rde:deposit
  xmlns:epp="urn:ietf:params:xml:ns:epp-1.0"
  xmlns:rde="urn:ietf:params:xml:ns:rde-1.0"
  xmlns:rdeCsv="urn:ietf:params:xml:ns:rdeCsv-1.0"
  xmlns:csvDomain="urn:ietf:params:xml:ns:csvDomain-1.0"
  xmlns:csvHost="urn:ietf:params:xml:ns:csvHost-1.0"
  xmlns:csvContact="urn:ietf:params:xml:ns:csvContact-1.0"
  xmlns:csvRegistrar="urn:ietf:params:xml:ns:csvRegistrar-1.0"
  xmlns:csvIDN="urn:ietf:params:xml:ns:csvIDN-1.0"
  xmlns:rdeHeader="urn:ietf:params:xml:ns:rdeHeader-1.0"
  xmlns:csvNNDN="urn:ietf:params:xml:ns:csvNNDN-1.0"
  xmlns:rdeEppParams="urn:ietf:params:xml:ns:rdeEppParams-1.0"
  type="FULL"
  id="20101017001" prevId="20101010001">
  <rde:watermark>2010-10-18T00:00:00Z</rde:watermark>
  <rde:rdeMenu>
    <rde:version>1.0</rde:version>
    <rde:objURI>urn:ietf:params:xml:ns:csvDomain-1.0</rde:objURI>
    <rde:objURI>urn:ietf:params:xml:ns:csvHost-1.0</rde:objURI>
    <rde:objURI>urn:ietf:params:xml:ns:csvContact-1.0</rde:objURI>
    <rde:objURI>urn:ietf:params:xml:ns:csvRegistrar-1.0</rde:objURI>
    <rde:objURI>urn:ietf:params:xml:ns:csvIDN-1.0</rde:objURI>
    <rde:objURI>urn:ietf:params:xml:ns:csvNNDN-1.0</rde:objURI>
    <rde:objURI>urn:ietf:params:xml:ns:rdeEppParams-1.0</rde:objURI>
  </rde:rdeMenu>
  <rde:contents>
    <rdeHeader:header>
      <rdeHeader:tld>test</rdeHeader:tld>
      <rdeHeader:count uri="urn:ietf:params:xml:ns:csvDomain-1.0">
4

```

```

</rdeHeader:count>
<rdeHeader:count uri="urn:ietf:params:xml:ns:csvHost-1.0">
6
</rdeHeader:count>
<rdeHeader:count uri="urn:ietf:params:xml:ns:csvContact-1.0">
9
</rdeHeader:count>
<rdeHeader:count uri="urn:ietf:params:xml:ns:csvRegistrar-1.0">
3
</rdeHeader:count>
<rdeHeader:count uri="urn:ietf:params:xml:ns:csvIDN-1.0">
2
</rdeHeader:count>
<rdeHeader:count uri="urn:ietf:params:xml:ns:csvNNDN-1.0">
2
</rdeHeader:count>
<rdeHeader:count uri="urn:ietf:params:xml:ns:rdeEppParams-1.0">
1
</rdeHeader:count>
</rdeHeader:header>
<csvDomain:contents>
  <rdeCsv:csv name="domain" sep=", ">
    <rdeCsv:fields>
      <csvDomain:fName/>
      <rdeCsv:fRoid/>
      <rdeCsv:fIdnTableId/>
      <csvDomain:fOriginalName/>
      <rdeCsv:fRegistrant/>
      <rdeCsv:fClID/>
      <rdeCsv:fCrRr/>
      <rdeCsv:fCrID/>
      <rdeCsv:fCrDate/>
      <rdeCsv:fUpRr/>
      <rdeCsv:fUpID/>
      <rdeCsv:fUpDate/>
      <rdeCsv:fExDate isRequired="true"/>
    </rdeCsv:fields>
    <rdeCsv:files>
      <rdeCsv:file
        cksum="75E2D01F">
        domain-YYYYMMDD.csv
      </rdeCsv:file>
    </rdeCsv:files>
  </rdeCsv:csv>
  <rdeCsv:csv name="domainContacts" sep=", ">
    <rdeCsv:fields>
      <csvDomain:fName parent="true"/>
      <csvContact:fId/>

```

```
        <csvDomain:fContactType/>
    </rdeCsv:fields>
    <rdeCsv:files>
        <rdeCsv:file
            cksum="70A7C17B">
            domainContacts-YYYYMMDD.csv
        </rdeCsv:file>
    </rdeCsv:files>
</rdeCsv:csv>
<rdeCsv:csv name="domainStatuses" sep=", ">
    <rdeCsv:fields>
        <csvDomain:fName parent="true"/>
        <csvDomain:fStatus/>
        <rdeCsv:fStatusDescription/>
        <rdeCsv:fLang/>
        <csvDomain:fRgpStatus/>
    </rdeCsv:fields>
    <rdeCsv:files>
        <rdeCsv:file
            cksum="EB8C548E">
            domainStatuses-YYYYMMDD.csv
        </rdeCsv:file>
    </rdeCsv:files>
</rdeCsv:csv>
<rdeCsv:csv name="domainNameServers" sep=", ">
    <rdeCsv:fields>
        <csvDomain:fName parent="true"/>
        <csvHost:fName parent="true"/>
    </rdeCsv:fields>
    <rdeCsv:files>
        <rdeCsv:file
            cksum="984C3097">
            domainNameServers-name-YYYYMMDD.csv
        </rdeCsv:file>
    </rdeCsv:files>
</rdeCsv:csv>
<rdeCsv:csv name="domainNameServers" sep=", ">
    <rdeCsv:fields>
        <csvDomain:fName parent="true"/>
        <rdeCsv:fRoid/>
    </rdeCsv:fields>
    <rdeCsv:files>
        <rdeCsv:file
            cksum="569D4638">
            domainNameServers-roid-YYYYMMDD.csv
        </rdeCsv:file>
    </rdeCsv:files>
</rdeCsv:csv>
```

```
<rdeCsv:csv name="dnssec" sep=", ">
  <rdeCsv:fields>
    <csvDomain:fName parent="true"/>
    <csvDomain:fMaxSigLife/>
    <csvDomain:fKeyTag/>
    <csvDomain:fDsAlg/>
    <csvDomain:fDigestType/>
    <csvDomain:fDigest/>
  </rdeCsv:fields>
  <rdeCsv:files>
    <rdeCsv:file
      cksum="AA15CB43">
        dnssec-ds-YYYYMMDD.csv
      </rdeCsv:file>
    </rdeCsv:files>
  </rdeCsv:csv>
<rdeCsv:csv name="dnssec" sep=", ">
  <rdeCsv:fields>
    <csvDomain:fName parent="true"/>
    <csvDomain:fMaxSigLife/>
    <csvDomain:fFlags/>
    <csvDomain:fProtocol/>
    <csvDomain:fKeyAlg/>
    <csvDomain:fPubKey/>
  </rdeCsv:fields>
  <rdeCsv:files>
    <rdeCsv:file
      cksum="1B16F334">
        dnssec-key-YYYYMMDD.csv
      </rdeCsv:file>
    </rdeCsv:files>
  </rdeCsv:csv>
<rdeCsv:csv name="domainTransfer" sep=", ">
  <rdeCsv:fields>
    <csvDomain:fName parent="true"/>
    <rdeCsv:fTrStatus/>
    <rdeCsv:fReRr/>
    <rdeCsv:fReID/>
    <rdeCsv:fReDate/>
    <rdeCsv:fAcRr/>
    <rdeCsv:fAcID/>
    <rdeCsv:fAcDate/>
    <rdeCsv:fExDate/>
  </rdeCsv:fields>
  <rdeCsv:files>
    <rdeCsv:file
      cksum="71170194">
        domainTransfer-YYYYMMDD.csv
      </rdeCsv:file>
    </rdeCsv:files>
  </rdeCsv:csv>
```



```
        </rdeCsv:file>
      </rdeCsv:files>
    </rdeCsv:csv>
  </csvDomain:contents>
  <csvHost:contents>
    <rdeCsv:csv name="host" sep=",">
      <rdeCsv:fields>
        <csvHost:fName/>
        <rdeCsv:fRoid/>
        <rdeCsv:fClID/>
        <rdeCsv:fCrRr/>
        <rdeCsv:fCrID/>
        <rdeCsv:fCrDate/>
        <rdeCsv:fUpRr/>
        <rdeCsv:fUpID/>
        <rdeCsv:fUpDate/>
        <rdeCsv:fTrDate/>
      </rdeCsv:fields>
      <rdeCsv:files>
        <rdeCsv:file
          cksum="120938E3">
          host-YYYYMMDD.csv
        </rdeCsv:file>
      </rdeCsv:files>
    </rdeCsv:csv>
    <rdeCsv:csv name="hostStatuses" sep=",">
      <rdeCsv:fields>
        <rdeCsv:fRoid parent="true"/>
        <csvHost:fStatus/>
        <rdeCsv:fStatusDescription/>
        <rdeCsv:fLang/>
      </rdeCsv:fields>
      <rdeCsv:files>
        <rdeCsv:file
          cksum="0BA504FC">
          hostStatuses-YYYYMMDD.csv
        </rdeCsv:file>
      </rdeCsv:files>
    </rdeCsv:csv>
    <rdeCsv:csv name="hostAddresses" sep=",">
      <rdeCsv:fields>
        <rdeCsv:fRoid parent="true"/>
        <csvHost:fAddr isRequired="true"/>
        <csvHost:fAddrVersion isRequired="true"/>
      </rdeCsv:fields>
      <rdeCsv:files>
        <rdeCsv:file
          cksum="17888F02">
```

```
        hostAddresses-YYYYMMDD.csv
      </rdeCsv:file>
    </rdeCsv:files>
  </rdeCsv:csv>
</csvHost:contents>
<csvContact:contents>
  <rdeCsv:csv name="contact" sep=", ">
    <rdeCsv:fields>
      <csvContact:fId/>
      <rdeCsv:fRoid/>
      <csvContact:fVoice/>
      <csvContact:fVoiceExt/>
      <csvContact:fFax/>
      <csvContact:fFaxExt/>
      <csvContact:fEmail/>
      <rdeCsv:fClID/>
      <rdeCsv:fCrRr/>
      <rdeCsv:fCrID/>
      <rdeCsv:fCrDate/>
      <rdeCsv:fUpRr/>
      <rdeCsv:fUpID/>
      <rdeCsv:fUpDate/>
    </rdeCsv:fields>
    <rdeCsv:files>
      <rdeCsv:file
        cksum="D7F106A5">
        contact-YYYYMMDD.csv
      </rdeCsv:file>
    </rdeCsv:files>
  </rdeCsv:csv>
  <rdeCsv:csv name="contactStatuses" sep=", ">
    <rdeCsv:fields>
      <csvContact:fId parent="true"/>
      <csvContact:fStatus/>
      <rdeCsv:fStatusDescription/>
      <rdeCsv:fLang/>
    </rdeCsv:fields>
    <rdeCsv:files>
      <rdeCsv:file
        cksum="2AAF99D4">
        contactStatuses-YYYYMMDD.csv
      </rdeCsv:file>
    </rdeCsv:files>
  </rdeCsv:csv>
  <rdeCsv:csv name="contactPostal" sep=", ">
    <rdeCsv:fields>
      <csvContact:fId parent="true"/>
      <csvContact:fPostalType/>
```

```
<csvContact:fName/>
<csvContact:fOrg/>
<csvContact:fStreet index="0"/>
<csvContact:fStreet index="1"/>
<csvContact:fStreet index="2"/>
<csvContact:fCity/>
<csvContact:fSp/>
<csvContact:fPc/>
<csvContact:fCc/>
</rdeCsv:fields>
<rdeCsv:files>
  <rdeCsv:file
    cksum="02CC2504">
    contactPostal-YYYYMMDD.csv
  </rdeCsv:file>
</rdeCsv:files>
</rdeCsv:csv>
<rdeCsv:csv name="contactTransfer" sep=", ">
  <rdeCsv:fields>
    <csvContact:fId parent="true"/>
    <rdeCsv:fTrStatus/>
    <rdeCsv:fReRr/>
    <rdeCsv:fReID/>
    <rdeCsv:fReDate/>
    <rdeCsv:fAcRr/>
    <rdeCsv:fAcID/>
    <rdeCsv:fAcDate/>
  </rdeCsv:fields>
  <rdeCsv:files>
    <rdeCsv:file
      cksum="D0929632">
      contactTransfer-YYYYMMDD.csv
    </rdeCsv:file>
  </rdeCsv:files>
</rdeCsv:csv>
<rdeCsv:csv name="contactDisclose" sep=", ">
  <rdeCsv:fields>
    <csvContact:fId parent="true"/>
    <csvContact:fDiscloseFlag/>
    <csvContact:fDiscloseNameLoc/>
    <csvContact:fDiscloseNameInt/>
    <csvContact:fDiscloseOrgLoc/>
    <csvContact:fDiscloseOrgInt/>
    <csvContact:fDiscloseAddrLoc/>
    <csvContact:fDiscloseAddrInt/>
    <csvContact:fDiscloseVoice/>
    <csvContact:fDiscloseFax/>
    <csvContact:fDiscloseEmail/>
```

```
</rdeCsv:fields>
<rdeCsv:files>
  <rdeCsv:file
    cksum="89043A90">
    contactDisclose-YYYYMMDD.csv
  </rdeCsv:file>
</rdeCsv:files>
</rdeCsv:csv>
</csvContact:contents>
<csvRegistrar:contents>
  <rdeCsv:csv name="registrar" sep=", ">
    <rdeCsv:fields>
      <csvRegistrar:fId/>
      <csvRegistrar:fName isLoc="false"/>
      <csvRegistrar:fGurid/>
      <csvRegistrar:fStatus/>
      <csvContact:fStreet isLoc="false" index="0"/>
      <csvContact:fStreet isLoc="false" index="1"/>
      <csvContact:fStreet isLoc="false" index="2"/>
      <csvContact:fCity isLoc="false" />
      <csvContact:fSp isLoc="false" />
      <csvContact:fPc isLoc="false" />
      <csvContact:fCc isLoc="false" />
      <csvContact:fVoice/>
      <csvContact:fVoiceExt/>
      <csvContact:fFax/>
      <csvContact:fFaxExt/>
      <csvContact:fEmail/>
      <rdeCsv:fUrl/>
      <csvRegistrar:fWhoisUrl/>
      <rdeCsv:fCrDate/>
      <rdeCsv:fUpDate/>
    </rdeCsv:fields>
    <rdeCsv:files>
      <rdeCsv:file
        cksum="306178BB">
        registrar-YYYYMMDD.csv
      </rdeCsv:file>
    </rdeCsv:files>
  </rdeCsv:csv>
</csvRegistrar:contents>
<csvIDN:contents>
  <rdeCsv:csv name="idnLanguage" sep=", ">
    <rdeCsv:fields>
      <rdeCsv:fIdnTableId isRequired="true"/>
      <rdeCsv:fUrl isRequired="true"/>
    </rdeCsv:fields>
  <rdeCsv:files>
```

```
<rdeCsv:file
  cksum="D462EAD0">
  idnLanguage-YYYYMMDD.csv
</rdeCsv:file>
</rdeCsv:files>
</rdeCsv:csv>
</csvIDN:contents>
<csvNNDN:contents>
  <rdeCsv:csv name="NNDN" sep=", ">
    <rdeCsv:fields>
      <csvNNDN:fAName/>
      <rdeCsv:fIdnTableId/>
      <csvNNDN:fOriginalName/>
      <csvNNDN:fNameState/>
      <csvNNDN:fMirroringNS/>
      <rdeCsv:fCrDate/>
    </rdeCsv:fields>
    <rdeCsv:files>
      <rdeCsv:file
        cksum="11C80D60">
        NNDN-YYYYMMDD.csv
      </rdeCsv:file>
    </rdeCsv:files>
  </rdeCsv:csv>
</csvNNDN:contents>
<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:host-1.0
</rdeEppParams:objURI>
  <rdeEppParams:objURI>urn:ietf:params:xml:ns:contact-1.0
</rdeEppParams:objURI>
  <rdeEppParams:svcExtension>
    <epp:extURI>urn:ietf:params:xml:ns:secDNS-1.1
    </epp:extURI>
    <epp:extURI>urn:ietf:params:xml:ns:rgp-1.0
    </epp:extURI>
  </rdeEppParams:svcExtension>
  <rdeEppParams:dcP>
    <epp:access>
      <epp:all/>
    </epp:access>
    <epp:statement>
      <epp:purpose>
        <epp:admin/>
        <epp:other/>
```

```

        <epp:prov/>
      </epp:purpose>
    <epp:recipient>
      <epp:ours/>
      <epp:public/>
      <epp:unrelated/>
    </epp:recipient>
    <epp:retention>
      <epp:indefinite/>
    </epp:retention>
  </epp:statement>
</rdeEppParams:dcp>
</rdeEppParams:eppParams>
</rde:contents>
</rde:deposit>

```

19. Example of differential deposit using the CSV model

Example of a differential deposit using the CSV model:

```

<?xml version="1.0" encoding="UTF-8"?>
<rde:deposit
  xmlns:epp="urn:ietf:params:xml:ns:epp-1.0"
  xmlns:rde="urn:ietf:params:xml:ns:rde-1.0"
  xmlns:rdeCsv="urn:ietf:params:xml:ns:rdeCsv-1.0"
  xmlns:csvDomain="urn:ietf:params:xml:ns:csvDomain-1.0"
  xmlns:csvHost="urn:ietf:params:xml:ns:csvHost-1.0"
  xmlns:csvContact="urn:ietf:params:xml:ns:csvContact-1.0"
  xmlns:csvRegistrar="urn:ietf:params:xml:ns:csvRegistrar-1.0"
  xmlns:csvIDN="urn:ietf:params:xml:ns:csvIDN-1.0"
  xmlns:rdeHeader="urn:ietf:params:xml:ns:rdeHeader-1.0"
  xmlns:csvNNDN="urn:ietf:params:xml:ns:csvNNDN-1.0"
  xmlns:rdeEppParams="urn:ietf:params:xml:ns:rdeEppParams-1.0"
  type="DIFF"
  id="20101017001" prevId="20101010001">
  <rde:watermark>2010-10-18T00:00:00Z</rde:watermark>
  <rde:rdeMenu>
    <rde:version>1.0</rde:version>
    <rde:objURI>urn:ietf:params:xml:ns:csvDomain-1.0</rde:objURI>
    <rde:objURI>urn:ietf:params:xml:ns:csvHost-1.0</rde:objURI>
    <rde:objURI>urn:ietf:params:xml:ns:csvContact-1.0</rde:objURI>
    <rde:objURI>urn:ietf:params:xml:ns:csvRegistrar-1.0</rde:objURI>
    <rde:objURI>urn:ietf:params:xml:ns:csvIDN-1.0</rde:objURI>
  </rde:rdeMenu>
  <rde:deletes>
    <csvDomain:deletes>
      <rdeCsv:csv name="domain">
        <rdeCsv:fields>

```

```
        <csvDomain:fName/>
    </rdeCsv:fields>
    <rdeCsv:files>
        <rdeCsv:file
            cksum="6F2B988F">
                domain-delete-YYYYMMDD.csv
            </rdeCsv:file>
        </rdeCsv:files>
    </rdeCsv:csv>
</csvDomain:deletes>
<csvHost:deletes>
    <rdeCsv:csv name="host">
        <rdeCsv:fields>
            <rdeCsv:fRoid/>
        </rdeCsv:fields>
        <rdeCsv:files>
            <rdeCsv:file
                cksum="E3408F5E">
                    host-delete-YYYYMMDD.csv
                </rdeCsv:file>
            </rdeCsv:files>
        </rdeCsv:csv>
    </csvHost:deletes>
</csvContact:deletes>
<csvRegistrar:deletes>
    <rdeCsv:csv name="registrar">
        <rdeCsv:fields>
            <csvRegistrar:fId/>
        </rdeCsv:fields>
        <rdeCsv:files>
            <rdeCsv:file
                cksum="6F2B988F">
                    contact-delete-YYYYMMDD.csv
                </rdeCsv:file>
            </rdeCsv:files>
        </rdeCsv:csv>
    </csvRegistrar:deletes>
</csvRegistrar:deletes>
    <rdeCsv:csv name="registrar">
        <rdeCsv:fields>
            <csvRegistrar:fId/>
        </rdeCsv:fields>
        <rdeCsv:files>
            <rdeCsv:file
                cksum="307B87AE">
                    registrar-delete-YYYYMMDD.csv
                </rdeCsv:file>
            </rdeCsv:files>
        </rdeCsv:csv>
    </rdeCsv:csv>
```

```

</csvRegistrar:deletes>
<csvIDN:deletes>
  <rdeCsv:csv name="idnLanguage">
    <rdeCsv:fields>
      <rdeCsv:fIdnTableId/>
    </rdeCsv:fields>
    <rdeCsv:files>
      <rdeCsv:file
        cksum="757B573A">
        idnLanguage-delete-YYYYMMDD.csv
      </rdeCsv:file>
    </rdeCsv:files>
  </rdeCsv:csv>
</csvIDN:deletes>
<csvNNDN:deletes>
  <rdeCsv:csv name="NNDN">
    <rdeCsv:fields>
      <csvNNDN:fAName/>
    </rdeCsv:fields>
    <rdeCsv:files>
      <rdeCsv:file
        cksum="FF104E83">
        NNDN-delete-YYYYMMDD.csv
      </rdeCsv:file>
    </rdeCsv:files>
  </rdeCsv:csv>
</csvNNDN:deletes>
</rde:deletes>
<rde:contents>
  <rdeHeader:header>
    <rdeHeader:tld>test</rdeHeader:tld>
    <rdeHeader:count uri="urn:ietf:params:xml:ns:csvDomain-1.0">
      2
    </rdeHeader:count>
    <rdeHeader:count uri="urn:ietf:params:xml:ns:csvHost-1.0">
      2
    </rdeHeader:count>
    <rdeHeader:count uri="urn:ietf:params:xml:ns:csvContact-1.0">
      3
    </rdeHeader:count>
    <rdeHeader:count uri="urn:ietf:params:xml:ns:csvRegistrar-1.0">
      1
    </rdeHeader:count>
    <rdeHeader:count uri="urn:ietf:params:xml:ns:csvIDN-1.0">
      1
    </rdeHeader:count>
    <rdeHeader:count uri="urn:ietf:params:xml:ns:csvNNDN-1.0">
      1

```



```
</rdeHeader:count>
<rdeHeader:count uri="urn:ietf:params:xml:ns:rdeEppParams-1.0">
1
</rdeHeader:count>
</rdeHeader:header>
<csvDomain:contents>
  <rdeCsv:csv name="domain" sep=", ">
    <rdeCsv:fields>
      <csvDomain:fName/>
      <rdeCsv:fRoid/>
      <rdeCsv:fIdnTableId/>
      <csvDomain:fOriginalName/>
      <rdeCsv:fRegistrant/>
      <rdeCsv:fClID/>
      <rdeCsv:fCrRr/>
      <rdeCsv:fCrID/>
      <rdeCsv:fCrDate/>
      <rdeCsv:fUpRr/>
      <rdeCsv:fUpID/>
      <rdeCsv:fUpDate/>
      <rdeCsv:fExDate isRequired="true"/>
    </rdeCsv:fields>
    <rdeCsv:files>
      <rdeCsv:file
        cksum="75E2D01F">
        domain-YYYYMMDD.csv
      </rdeCsv:file>
    </rdeCsv:files>
  </rdeCsv:csv>
  <rdeCsv:csv name="domainContacts" sep=", ">
    <rdeCsv:fields>
      <csvDomain:fName parent="true"/>
      <csvContact:fId/>
      <csvDomain:fContactType/>
    </rdeCsv:fields>
    <rdeCsv:files>
      <rdeCsv:file
        cksum="70A7C17B">
        domainContacts-YYYYMMDD.csv
      </rdeCsv:file>
    </rdeCsv:files>
  </rdeCsv:csv>
  <rdeCsv:csv name="domainStatuses" sep=", ">
    <rdeCsv:fields>
      <csvDomain:fName parent="true"/>
      <csvDomain:fStatus/>
      <rdeCsv:fStatusDescription/>
      <rdeCsv:fLang/>
```

```
        <csvDomain:fRgpStatus/>
    </rdeCsv:fields>
    <rdeCsv:files>
        <rdeCsv:file
            cksum="EB8C548E">
            domainStatuses-YYYYMMDD.csv
        </rdeCsv:file>
    </rdeCsv:files>
</rdeCsv:csv>
<rdeCsv:csv name="domainNameServers" sep=", ">
    <rdeCsv:fields>
        <csvDomain:fName parent="true"/>
        <csvHost:fName parent="true"/>
    </rdeCsv:fields>
    <rdeCsv:files>
        <rdeCsv:file
            cksum="984C3097">
            domainNameServers-name-YYYYMMDD.csv
        </rdeCsv:file>
    </rdeCsv:files>
</rdeCsv:csv>
<rdeCsv:csv name="domainNameServers" sep=", ">
    <rdeCsv:fields>
        <csvDomain:fName parent="true"/>
        <rdeCsv:fRoid/>
    </rdeCsv:fields>
    <rdeCsv:files>
        <rdeCsv:file
            cksum="569D4638">
            domainNameServers-roid-YYYYMMDD.csv
        </rdeCsv:file>
    </rdeCsv:files>
</rdeCsv:csv>
<rdeCsv:csv name="dnssec" sep=", ">
    <rdeCsv:fields>
        <csvDomain:fName parent="true"/>
        <csvDomain:fMaxSigLife/>
        <csvDomain:fKeyTag/>
        <csvDomain:fDsAlg/>
        <csvDomain:fDigestType/>
        <csvDomain:fDigest/>
    </rdeCsv:fields>
    <rdeCsv:files>
        <rdeCsv:file
            cksum="AA15CB43">
            dnssec-ds-YYYYMMDD.csv
        </rdeCsv:file>
    </rdeCsv:files>
```

```
</rdeCsv:csv>
<rdeCsv:csv name="dnssec" sep=", ">
  <rdeCsv:fields>
    <csvDomain:fName parent="true"/>
    <csvDomain:fMaxSigLife/>
    <csvDomain:fFlags/>
    <csvDomain:fProtocol/>
    <csvDomain:fKeyAlg/>
    <csvDomain:fPubKey/>
  </rdeCsv:fields>
  <rdeCsv:files>
    <rdeCsv:file
      cksum="1B16F334">
      dnssec-key-YYYYMMDD.csv
    </rdeCsv:file>
  </rdeCsv:files>
</rdeCsv:csv>
<rdeCsv:csv name="domainTransfer" sep=", ">
  <rdeCsv:fields>
    <csvDomain:fName parent="true"/>
    <rdeCsv:fTrStatus/>
    <rdeCsv:fReRr/>
    <rdeCsv:fReID/>
    <rdeCsv:fReDate/>
    <rdeCsv:fAcRr/>
    <rdeCsv:fAcID/>
    <rdeCsv:fAcDate/>
    <rdeCsv:fExDate/>
  </rdeCsv:fields>
  <rdeCsv:files>
    <rdeCsv:file
      cksum="71170194">
      domainTransfer-YYYYMMDD.csv
    </rdeCsv:file>
  </rdeCsv:files>
</rdeCsv:csv>
</csvDomain:contents>
<csvHost:contents>
  <rdeCsv:csv name="host" sep=", ">
    <rdeCsv:fields>
      <csvHost:fName/>
      <rdeCsv:fRoid/>
      <rdeCsv:fClID/>
      <rdeCsv:fCrRr/>
      <rdeCsv:fCrID/>
      <rdeCsv:fCrDate/>
      <rdeCsv:fUpRr/>
      <rdeCsv:fUpID/>
    </rdeCsv:fields>
  </rdeCsv:csv>
</csvHost:contents>
```

```
<rdeCsv:fUpDate/>
<rdeCsv:fTrDate/>
</rdeCsv:fields>
<rdeCsv:files>
  <rdeCsv:file
    cksum="120938E3">
    host-YYYYMMDD.csv
  </rdeCsv:file>
</rdeCsv:files>
</rdeCsv:csv>
<rdeCsv:csv name="hostStatuses" sep=", ">
  <rdeCsv:fields>
    <rdeCsv:fRoid parent="true"/>
    <csvHost:fStatus/>
    <rdeCsv:fStatusDescription/>
    <rdeCsv:fLang/>
  </rdeCsv:fields>
  <rdeCsv:files>
    <rdeCsv:file
      cksum="0BA504FC">
      hostStatuses-YYYYMMDD.csv
    </rdeCsv:file>
  </rdeCsv:files>
</rdeCsv:csv>
<rdeCsv:csv name="hostAddresses" sep=", ">
  <rdeCsv:fields>
    <rdeCsv:fRoid parent="true"/>
    <csvHost:fAddr isRequired="true"/>
    <csvHost:fAddrVersion isRequired="true"/>
  </rdeCsv:fields>
  <rdeCsv:files>
    <rdeCsv:file
      cksum="17888F02">
      hostAddresses-YYYYMMDD.csv
    </rdeCsv:file>
  </rdeCsv:files>
</rdeCsv:csv>
</csvHost:contents>
<csvContact:contents>
  <rdeCsv:csv name="contact" sep=", ">
    <rdeCsv:fields>
      <csvContact:fId/>
      <rdeCsv:fRoid/>
      <csvContact:fVoice/>
      <csvContact:fVoiceExt/>
      <csvContact:fFax/>
      <csvContact:fFaxExt/>
      <csvContact:fEmail/>
```

```
<rdeCsv:fClID/>
<rdeCsv:fCrRr/>
<rdeCsv:fCrID/>
<rdeCsv:fCrDate/>
<rdeCsv:fUpRr/>
<rdeCsv:fUpID/>
<rdeCsv:fUpDate/>
</rdeCsv:fields>
<rdeCsv:files>
  <rdeCsv:file
    cksum="D7F106A5">
    contact-YYYYMMDD.csv
  </rdeCsv:file>
</rdeCsv:files>
</rdeCsv:csv>
<rdeCsv:csv name="contactStatuses" sep=", ">
  <rdeCsv:fields>
    <csvContact:fId parent="true"/>
    <csvContact:fStatus/>
    <rdeCsv:fStatusDescription/>
    <rdeCsv:fLang/>
  </rdeCsv:fields>
  <rdeCsv:files>
    <rdeCsv:file
      cksum="2AAF99D4">
      contactStatuses-YYYYMMDD.csv
    </rdeCsv:file>
  </rdeCsv:files>
</rdeCsv:csv>
<rdeCsv:csv name="contactPostal" sep=", ">
  <rdeCsv:fields>
    <csvContact:fId parent="true"/>
    <csvContact:fPostalType/>
    <csvContact:fName/>
    <csvContact:fOrg/>
    <csvContact:fStreet index="0"/>
    <csvContact:fStreet index="1"/>
    <csvContact:fStreet index="2"/>
    <csvContact:fCity/>
    <csvContact:fSp/>
    <csvContact:fPc/>
    <csvContact:fCc/>
  </rdeCsv:fields>
  <rdeCsv:files>
    <rdeCsv:file
      cksum="02CC2504">
      contactPostal-YYYYMMDD.csv
    </rdeCsv:file>
```

```
</rdeCsv:files>
</rdeCsv:csv>
<rdeCsv:csv name="contactTransfer" sep=", ">
  <rdeCsv:fields>
    <csvContact:fId parent="true"/>
    <rdeCsv:fTrStatus/>
    <rdeCsv:fReRr/>
    <rdeCsv:fReID/>
    <rdeCsv:fReDate/>
    <rdeCsv:fAcRr/>
    <rdeCsv:fAcID/>
    <rdeCsv:fAcDate/>
  </rdeCsv:fields>
  <rdeCsv:files>
    <rdeCsv:file
      cksum="D0929632">
      contactTransfer-YYYYMMDD.csv
    </rdeCsv:file>
  </rdeCsv:files>
</rdeCsv:csv>
<rdeCsv:csv name="contactDisclose" sep=", ">
  <rdeCsv:fields>
    <csvContact:fId parent="true"/>
    <csvContact:fDiscloseFlag/>
    <csvContact:fDiscloseNameLoc/>
    <csvContact:fDiscloseNameInt/>
    <csvContact:fDiscloseOrgLoc/>
    <csvContact:fDiscloseOrgInt/>
    <csvContact:fDiscloseAddrLoc/>
    <csvContact:fDiscloseAddrInt/>
    <csvContact:fDiscloseVoice/>
    <csvContact:fDiscloseFax/>
    <csvContact:fDiscloseEmail/>
  </rdeCsv:fields>
  <rdeCsv:files>
    <rdeCsv:file
      cksum="89043A90">
      contactDisclose-YYYYMMDD.csv
    </rdeCsv:file>
  </rdeCsv:files>
</rdeCsv:csv>
</csvContact:contents>
<csvRegistrar:contents>
  <rdeCsv:csv name="registrar" sep=", ">
    <rdeCsv:fields>
      <csvRegistrar:fId/>
      <csvRegistrar:fName isLoc="false"/>
      <csvRegistrar:fGurid/>
```

```
<csvRegistrar:fStatus/>
<csvContact:fStreet isLoc="false" index="0"/>
<csvContact:fStreet isLoc="false" index="1"/>
<csvContact:fStreet isLoc="false" index="2"/>
<csvContact:fCity isLoc="false" />
<csvContact:fSp isLoc="false" />
<csvContact:fPc isLoc="false" />
<csvContact:fCc isLoc="false" />
<csvContact:fVoice/>
<csvContact:fVoiceExt/>
<csvContact:fFax/>
<csvContact:fFaxExt/>
<csvContact:fEmail/>
<rdeCsv:fUrl/>
<csvRegistrar:fWhoisUrl/>
<rdeCsv:fCrDate/>
<rdeCsv:fUpDate/>
</rdeCsv:fields>
<rdeCsv:files>
  <rdeCsv:file
    cksum="306178BB">
    registrar-YYYYMMDD.csv
  </rdeCsv:file>
</rdeCsv:files>
</rdeCsv:csv>
</csvRegistrar:contents>
<csvIDN:contents>
  <rdeCsv:csv name="idnLanguage" sep=", ">
    <rdeCsv:fields>
      <rdeCsv:fIdnTableId isRequired="true"/>
      <rdeCsv:fUrl isRequired="true"/>
    </rdeCsv:fields>
    <rdeCsv:files>
      <rdeCsv:file
        cksum="D462EAD0">
        idnLanguage-YYYYMMDD.csv
      </rdeCsv:file>
    </rdeCsv:files>
  </rdeCsv:csv>
</csvIDN:contents>
<csvNNDN:contents>
  <rdeCsv:csv name="NNDN" sep=", ">
    <rdeCsv:fields>
      <csvNNDN:fAName/>
      <rdeCsv:fIdnTableId/>
      <csvNNDN:fOriginalName/>
      <csvNNDN:fNameState/>
      <csvNNDN:fMirroringNS/>
```

```
        <rdeCsv:fCrDate/>
      </rdeCsv:fields>
    <rdeCsv:files>
      <rdeCsv:file
        cksum="11C80D60">
        NNDN-YYYYMMDD.csv
      </rdeCsv:file>
    </rdeCsv:files>
  </rdeCsv:csv>
</csvNNDN:contents>
<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:host-1.0
</rdeEppParams:objURI>
  <rdeEppParams:objURI>urn:ietf:params:xml:ns:contact-1.0
</rdeEppParams:objURI>
  <rdeEppParams:svcExtension>
    <epp:extURI>urn:ietf:params:xml:ns:secDNS-1.1
    </epp:extURI>
    <epp:extURI>urn:ietf:params:xml:ns:rgp-1.0
    </epp:extURI>
  </rdeEppParams:svcExtension>
  <rdeEppParams:dcp>
    <epp:access>
      <epp:all/>
    </epp:access>
    <epp:statement>
      <epp:purpose>
        <epp:admin/>
        <epp:other/>
        <epp:prov/>
      </epp:purpose>
      <epp:recipient>
        <epp:ours/>
        <epp:public/>
        <epp:unrelated/>
      </epp:recipient>
      <epp:retention>
        <epp:indefinite/>
      </epp:retention>
    </epp:statement>
  </rdeEppParams:dcp>
</rdeEppParams:eppParams>
</rde:contents>
</rde:deposit>
```


20. References

20.1. 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, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3339] Klyne, G. and C. Newman, "Date and Time on the Internet: Timestamps", RFC 3339, DOI 10.17487/RFC3339, July 2002, <<https://www.rfc-editor.org/info/rfc3339>>.
- [RFC3915] Hollenbeck, S., "Domain Registry Grace Period Mapping for the Extensible Provisioning Protocol (EPP)", RFC 3915, DOI 10.17487/RFC3915, September 2004, <<https://www.rfc-editor.org/info/rfc3915>>.
- [RFC5730] Hollenbeck, S., "Extensible Provisioning Protocol (EPP)", STD 69, RFC 5730, DOI 10.17487/RFC5730, August 2009, <<https://www.rfc-editor.org/info/rfc5730>>.
- [RFC5731] Hollenbeck, S., "Extensible Provisioning Protocol (EPP) Domain Name Mapping", STD 69, RFC 5731, DOI 10.17487/RFC5731, August 2009, <<https://www.rfc-editor.org/info/rfc5731>>.
- [RFC5732] Hollenbeck, S., "Extensible Provisioning Protocol (EPP) Host Mapping", STD 69, RFC 5732, DOI 10.17487/RFC5732, August 2009, <<https://www.rfc-editor.org/info/rfc5732>>.
- [RFC5733] Hollenbeck, S., "Extensible Provisioning Protocol (EPP) Contact Mapping", STD 69, RFC 5733, DOI 10.17487/RFC5733, August 2009, <<https://www.rfc-editor.org/info/rfc5733>>.

- [RFC5910] Gould, J. and S. Hollenbeck, "Domain Name System (DNS) Security Extensions Mapping for the Extensible Provisioning Protocol (EPP)", RFC 5910, DOI 10.17487/RFC5910, May 2010, <<https://www.rfc-editor.org/info/rfc5910>>.

20.2. Informative References

- [RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <<https://www.rfc-editor.org/info/rfc791>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC3912] Daigle, L., "WHOIS Protocol Specification", RFC 3912, DOI 10.17487/RFC3912, September 2004, <<https://www.rfc-editor.org/info/rfc3912>>.
- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, <<https://www.rfc-editor.org/info/rfc4291>>.
- [RFC5891] Klensin, J., "Internationalized Domain Names in Applications (IDNA): Protocol", RFC 5891, DOI 10.17487/RFC5891, August 2010, <<https://www.rfc-editor.org/info/rfc5891>>.
- [RFC7942] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", BCP 205, RFC 7942, DOI 10.17487/RFC7942, July 2016, <<https://www.rfc-editor.org/info/rfc7942>>.
- [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>>.

20.3. URIs

- [1] <http://tools.ietf.org/id/draft-arias-noguchi-registry-data-escrow-05.txt>

Authors' Addresses

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

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

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: July 6, 2019

G. Lozano
ICANN
Jan 2, 2019

Registry Data Escrow Specification
draft-arias-noguchi-registry-data-escrow-11

Abstract

This document specifies the format and contents of data escrow deposits targeted primarily for domain name registries. However, the specification was designed to be independent of the underlying objects that are being escrowed, therefore it could be used for purposes other than domain name registries.

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 <https://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 6, 2019.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must 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	2
2. Terminology	3
3. Problem Scope	4
4. General Conventions	5
4.1. Date and Time	5
5. Protocol Description	5
5.1. Root element <deposit>	5
5.2. Child <watermark> element	6
5.3. Child <rdeMenu> element	7
5.4. Child <deletes> element	7
5.5. Child <contents> element	8
6. Formal Syntax	9
6.1. RDE Schema	9
7. Internationalization Considerations	12
8. IANA Considerations	12
9. Implementation Status	13
9.1. Implementation in the gTLD space	13
10. Security Considerations	14
11. Acknowledgments	14
12. Change History	15
12.1. Changes from version 00 to 01	15
12.2. Changes from version 01 to 02	16
12.3. Changes from version 02 to 03	16
12.4. Changes from version 03 to 04	17
12.5. Changes from version 04 to 05	17
12.6. Changes from version 05 to 06	17
12.7. Changes from version 06 to 07	17
12.8. Changes from version 07 to 08	17
12.9. Changes from version 08 to 09	17
12.10. Changes from version 09 to 10	17
12.11. Changes from version 10 to 11	17
13. References	17
13.1. Normative References	18
13.2. Informative References	18
Author's Address	18

1. Introduction

Registry Data Escrow is the process by which an Registry periodically submits data deposits to a third party called an Escrow Agent. These deposits comprise the minimum data needed by a third party to resume operations if the registry can not function and is unable or unwilling to facilitate an orderly transfer of service. For example, for a domain name registry or registrar the data to be deposited would include all the objects related to registered domain names, e.g., names, contacts, name servers, etc.

The goal of data escrow is higher resiliency of registration services, for the benefit of Internet users. The beneficiaries of a registry are not just those registering information there, but all relying parties that need to identify the owners of objects.

In the context of domain name registries, registration data escrow is a requirement for generic top-level domains and some country code top-level domain managers are also currently escrowing data. There is also a similar requirement for ICANN-accredited domain registrars.

This document specifies a format for data escrow deposits independent of the objects being escrowed. A specification is required for each type of registry/set of objects that is expected to be escrowed.

2. 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].

DEPOSIT. Deposits can be of three kinds: Full, Differential or Incremental. For all kinds of Deposits, the Universe of Registry objects to be considered for data escrow are those objects necessary in order to offer the Registry Services.

DIFFERENTIAL DEPOSIT. Contains data that reflects all transactions involving the database that were not reflected in the last previous Full, Incremental or Differential Deposit, as the case may be. Differential deposit files will contain information from all database objects that were added, modified or deleted since the previous Deposit was completed as of its defined Timeline Watermark.

ESCROW AGENT. The organization designated by the Registry or the Third-Party Beneficiary to receive and guard Data Escrow Deposits from the Registry.

FULL DEPOSIT. Contains the Registry Data that reflects the current and complete Registry Database and will consist of data that reflects the state of the registry as of a defined Timeline Watermark for the deposit.

INCREMENTAL DEPOSIT. Contains data that reflects all transactions involving the database that were not reflected in the last previous Full Deposit. Incremental Deposit files will contain information from all database objects that were added, modified or deleted since the previous Full Deposit was completed as of its defined Timeline Watermark. If the Timeline Watermark of an Incremental Deposit were to cover the Watermark of another (Incremental or Differential)

Deposit since the last Full Deposit, the more recent Deposit MUST contain all the transactions of the earlier Deposit.

REGISTRY. A registration organization providing registration services for a certain type of objects, e.g., domain names, IP number resources, routing information.

THIRD-PARTY BENEFICIARY. Is the organization that, under extraordinary circumstances, would receive the escrow Deposits the Registry transferred to the Escrow Agent. This organization could be a backup Registry, Registry regulator, contracting party of the Registry, etc.

TIMELINE WATERMARK. Point in time on which to base the collecting of database objects for a Deposit. Deposits are expected to be consistent to that point in time.

3. Problem Scope

In the past few years, the issue of Registry continuity has been carefully considered in the gTLD and ccTLD space. Various organizations have carried out risk analyses and developed business continuity plans to deal with those risks, should they materialize.

One of the solutions considered and used, especially in the gTLD space, is Registry Data Escrow as a way to ensure the Continuity of Registry Services in the extreme case of Registry failure.

So far, almost every Registry that uses Registry Data Escrow has its own specification. It is anticipated that more Registries will be implementing escrow especially with an increasing number of domain registries coming into service, adding complexity to this issue.

It would seem beneficial to have a standardized specification for Registry Data Escrow that can be used by any Registry to submit its deposits.

While the main motivation for developing this solution is rooted on the domain name industry, the specification has been designed to be as general as possible. This allows other types of registries to use the base specification and develop their own specifications covering the objects used by other registration organizations.

A solution to the problem at hand SHALL clearly identify the format and contents of the deposits a Registry has to make, such that a different Registry would be able to rebuild the registration services of the former, without its help, in a timely manner, with minimum disruption to its users.

Since the details of the registration services provided vary from Registry to Registry, the solution SHALL provide mechanisms that allow its extensibility to accommodate variations and extensions of the registration services.

Given the requirement for confidentiality and the importance of accuracy of the information that is handled in order to offer registration services, the solution SHALL define confidentiality and integrity mechanisms for handling the registration data.

The solution SHALL NOT include in the specification transient objects that can be recreated by the new Registry, particularly those of delicate confidentiality, e.g., DNSSEC KSK/ZSK private keys.

Details that are a matter of policy SHOULD be identified as such for the benefit of the implementers.

Non-technical issues concerning Data Escrow, such as whether to escrow data and under which purposes the data may be used, are outside of scope of this document.

4. General Conventions

4.1. Date and Time

Numerous fields indicate "dates", such as the creation and expiry dates for objects. These fields SHALL contain timestamps indicating the date and time in UTC, specified in Internet Date/Time Format (see [RFC3339], Section 5.6) with the time-offset specified as "Z".

5. Protocol Description

The following is a format for Data Escrow deposits as produced by a Registry. The deposits are represented in XML. Only the format of the objects deposited is defined, nothing is prescribed about the method used to transfer such deposits between the Registry and the Escrow Agent or vice versa.

The protocol intends to be object agnostic allowing the "overload" of abstract elements using the "substitutionGroup" attribute to define the actual elements of an object to be escrowed.

5.1. Root element <deposit>

The container or root element for a Registry Data Escrow deposits is <deposit>. This element contains the following child elements: watermark, deletes and contents. This element also contains the following attributes:

- o A REQUIRED "type" attribute that is used to identify the kind of deposit: FULL, INCR (Incremental) or DIFF (Differential).
- o A REQUIRED "id" attribute that is used to uniquely identify the escrow deposit. Each registry is responsible for maintaining its own escrow deposits identifier space to ensure uniqueness, e.g., using identifiers as described in Section 2.8 of [RFC5730].
- o An OPTIONAL "prevId" attribute that can be used to identify the previous incremental, differential or full escrow deposit. This attribute MUST be used in Differential Deposits ("DIFF" type).
- o An OPTIONAL "resend" attribute that is incremented each time the escrow deposit failed the verification procedure at the receiving party and a new escrow deposit needs to be generated by the Registry for that specific date. The first time a deposit is generated the attribute is either omitted or MUST be "0". If a deposit needs to be generated again, the attribute MUST be set to "1", and so on.

Example of root element object:

```
<?xml version="1.0" encoding="UTF-8"?>
<rde:deposit
  xmlns:rde="urn:ietf:params:xml:ns:rde-1.0"
  ...
  type="FULL"
  id="20101017001" prevId="20101010001">
  <rde:watermark>2010-10-18T00:00:00Z</rde:watermark>
  <rde:deletes>
    ...
  </rde:deletes>
  <rde:contents>
    ...
  </rde:contents>
</rde:deposit>
```

5.2. Child <watermark> element

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

Example of <watermark> element object:

```
<?xml version="1.0" encoding="UTF-8"?>
<rde:deposit
  xmlns:rde="urn:ietf:params:xml:ns:rde-1.0"
  ...
  type="FULL"
  id="20101017001" prevId="20101010001">
  <rde:watermark>2010-10-18T00:00:00Z</rde:watermark>
  ...
</rde:deposit>
```

5.3. Child <rdeMenu> element

This element contains auxiliary information of the data escrow deposit.

A REQUIRED <rdeMenu> element contains the following child elements:

- o A REQUIRED <version> element that identifies the RDE protocol version.
- o One or more <objURI> elements that contain namespace URIs representing the <contents> and <deletes> element objects.

Example of <rdeMenu> element object:

```
<?xml version="1.0" encoding="UTF-8"?>
<rde:deposit
  xmlns:rde="urn:ietf:params:xml:ns:rde-1.0"
  ...
  <rde:rdeMenu>
    <rde:version>1.0</rde:version>
    <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>
  ...
</rde:deposit>
```

5.4. Child <deletes> element

This element SHOULD be present in deposits of type Incremental or Differential. It contains the list of objects that were deleted since the base previous deposit. Each object in this section SHALL contain an ID for the object deleted.

This section of the deposit SHOULD NOT be present in Full deposits. When rebuilding a registry it SHOULD be ignored if present in a Full deposit.

The specification for each object to be escrowed MUST declare the identifier to be used to reference the object to be deleted.

Example of <deletes> element object:

```
<?xml version="1.0" encoding="UTF-8"?>
<rde:deposit
  xmlns:rde="urn:ietf:params:xml:ns:rde-1.0"
  ...
  <rde:deletes>
    <rdeObj1:delete>
      <rdeObj1:name>foo.test</rdeObj1:name>
      <rdeObj1:name>bar.test</rdeObj1:name>
    </rdeObj1:delete>
    <rdeObj2:delete>
      <rdeObj2:id>sh8013-TEST</rdeObj2:id>
      <rdeObj2:id>co8013-TEST</rdeObj2:id>
    </rdeObj2:delete>
  </rde:deletes>
  ...
</rde:deposit>
```

5.5. Child <contents> element

This element of the deposit contains the objects in the deposit. It MUST be present in all type of deposits. It contains the data for the objects to be escrowed. The actual objects have to be specified individually.

In the case of Incremental or Differential deposits, the objects indicate whether the object was added or modified after the base previous deposit. In order to distinguish between one and the other, it will be sufficient to check existence of the referenced object in the base previous deposit.

When applying Incremental or Differential deposits (when rebuilding the registry from data escrow deposits) the relative order of the <deletes> elements is important, as is the relative order of the <contents> elements. All the <deletes> elements MUST be applied first, in the order that they appear. All the <contents> elements MUST be applied next, in the order that they appear.

If an object is present in the <contents> section of several Deposits (e.g. Full and Differential) the registry data from the latest

Deposit (as defined by the Timeline Watermark) SHOULD be used when rebuilding the registry.

Example of <contents> element object:

```
<?xml version="1.0" encoding="UTF-8"?>
<rde:deposit
  xmlns:rde="urn:ietf:params:xml:ns:rde-1.0"
  ...
  <rde:contents>
    ...
    <rdeObj1:contents>
      <rdeObj1:element1>
        <rdeObj1:child1>Object1 specific.</rdeObj1:child1>
        ...
      </rdeObj1:element1>
      <rdeObj2:element2>
        <rdeObj2:field1>Object2 specific.</rdeObj2:field1>
        ...
      </rdeObj2:element2>
    </rdeObj1:contents>
    ...
  </rde:contents>
  ...
</rde:deposit>
```

6. Formal Syntax

6.1. RDE Schema

Copyright (c) 2011 IETF Trust and the persons identified as authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- o Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- o Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- o Neither the name of Internet Society, IETF or IETF Trust, nor the names of specific contributors, may be used to endorse or promote

products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

BEGIN

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="urn:ietf:params:xml:ns:rde-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">

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

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

  <!-- Root element -->
  <element name="deposit" type="rde:escrowDepositType"/>

  <!-- RDE types -->
  <complexType name="escrowDepositType">
    <sequence>
      <element name="watermark" type="dateTime"/>
      <element name="rdeMenu" type="rde:rdeMenuType"/>
      <element name="deletes" type="rde:deletesType" minOccurs="0"/>
      <element name="contents" type="rde:contentsType"/>
    </sequence>
    <attribute name="type" type="rde:depositTypeType" use="required"/>
    <attribute name="id" type="rde:depositIdType" use="required"/>
    <attribute name="prevId" type="rde:depositIdType"/>
    <attribute name="resend" type="unsignedShort" default="0"/>
  </complexType>
```

```
<!-- Menu type -->
<complexType name="rdeMenuType">
  <sequence>
    <element name="version" type="rde:versionType"/>
    <element name="objURI" type="anyURI" maxOccurs="unbounded"/>
  </sequence>
</complexType>

<!-- Deletes Type -->
<complexType name="deletesType">
  <sequence minOccurs="0" maxOccurs="unbounded">
    <element ref="rde:delete"/>
  </sequence>
</complexType>

<element name="delete" type="rde:deleteType" abstract="true" />
<complexType name="deleteType">
  <complexContent>
    <restriction base="anyType"/>
  </complexContent>
</complexType>

<!-- Contents Type -->
<complexType name="contentsType">
  <sequence maxOccurs="unbounded">
    <element ref="rde:content"/>
  </sequence>
</complexType>

<element name="content" type="rde:contentType" abstract="true" />
<complexType name="contentType">
  <complexContent>
    <restriction base="anyType"/>
  </complexContent>
</complexType>

<!-- Type of deposit -->
<simpleType name="depositTypeType">
  <restriction base="token">
    <enumeration value="FULL"/>
    <enumeration value="INCR"/>
    <enumeration value="DIFF"/>
  </restriction>
</simpleType>

<!-- Deposit identifier type -->
<simpleType name="depositIdType">
  <restriction base="token">
```

```
        <pattern value="\w{1,13}"/>
      </restriction>
    </simpleType>

    <!-- A RDE version number is a dotted pair of decimal numbers -->
    <simpleType name="versionType">
      <restriction base="token">
        <pattern value="[1-9]+\.[0-9]+"/>
        <enumeration value="1.0"/>
      </restriction>
    </simpleType>

    <complexType name="rrType">
      <simpleContent>
        <extension base="eppcom:clIDType">
          <attribute name="client" type="eppcom:clIDType"/>
        </extension>
      </simpleContent>
    </complexType>
  </schema>

  END
```

7. 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.

8. IANA Considerations

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

Registration request for the RDE namespace:

URI: urn:ietf:params:xml:ns:rde-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 XML schema:

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

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

See the "Formal Syntax" section of this document.

9. Implementation Status

Note to RFC Editor: Please remove this section and the reference to RFC 7942 [RFC7942] before publication.

This section records the status of known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft, and is based on a proposal described in RFC 7942 [RFC7942]. The description of implementations in this section is intended to assist the IETF in its decision processes in progressing drafts to RFCs. Please note that the listing of any individual implementation here does not imply endorsement by the IETF. Furthermore, no effort has been spent to verify the information presented here that was supplied by IETF contributors. This is not intended as, and must not be construed to be, a catalog of available implementations or their features. Readers are advised to note that other implementations may exist.

According to RFC 7942 [RFC7942], "this will allow reviewers and working groups to assign due consideration to documents that have the benefit of running code, which may serve as evidence of valuable experimentation and feedback that have made the implemented protocols more mature. It is up to the individual working groups to use this information as they see fit".

9.1. Implementation in the gTLD space

Organization: ICANN

Name: ICANN Registry Agreement

Description: the ICANN Base Registry Agreement requires Registries, Data Escrow Agents, and ICANN to implement this specification. ICANN receives daily notifications from Data Escrow Agents confirming that more than 1,200 gTLDs are sending deposits that comply with this specification. ICANN receives on a weekly basis per gTLD, from more than 1,200 gTLD registries, a Bulk Registration Data Access file that also complies with this specification. In addition, ICANN is aware of Registry Service Provider transitions using data files that conform to this specification.

Level of maturity: production.

Coverage: all aspects of this specification are implemented.

Version compatibility: versions 03 - 08 are known to be implemented.

Contact: gustavo.lozano@icann.org

URL: <https://www.icann.org/resources/pages/registries/registries-agreements-en>

10. 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 must take all the necessary precautions like encrypting the data itself and/or the transport channel to avoid inadvertent disclosure of private data.

Mutual authentication of both parties passing data escrow deposit files is of the utmost importance. 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. It is recommended that specifications defining format and semantics for particular business models define an algorithm that Escrow Agents and Third-Party Beneficiaries could use to validate the contents of the data escrow deposit.

11. Acknowledgments

Special suggestions that have been incorporated into this document were provided by James Gould, Edward Lewis, Jaap Akkerhuis, Lawrence Conroy, Marc Groeneweg, Michael Young, Chris Wright, Patrick Mevzek, Stephen Morris, Scott Hollenbeck, Stephane Bortzmeyer, Warren Kumari, Paul Hoffman, Vika Mpisane, Bernie Hoeneisen, Jim Galvin, Andrew Sullivan, Hiro Hotta, Christopher Browne, Daniel Kalchev, David

Conrad, James Mitchell, Francisco Obispo, Bhadres Modi and Alexander Mayrhofer.

Shoji Noguchi and Francisco Arias participated as co-authors until version 07 providing invaluable support for this document.

12. Change History

12.1. Changes from version 00 to 01

1. Included DNSSEC elements as part of the basic <domain> element as defined in RFC 5910.
2. Included RGP elements as part of the basic <domain> element as defined in RFC 3915.
3. Added support for IDNs and IDN variants.
4. Eliminated the <summary> element and all its subordinate objects, except <watermarkDate>.
5. Renamed <watermarkDate> to <watermark> and included it directly under root element.
6. Renamed root element to <deposit>.
7. Added <authinfo> element under <registrar> element.
8. Added <roid> element under <registrar> element.
9. Reversed the order of the <deletes> and <contents> elements.
10. Removed <rdeDomain:status> minOccurs="0".
11. Added <extension> element under root element.
12. Added <extension> element under <contact> element.
13. Removed <period> element from <domain> element.
14. Populated the "Security Considerations" section.
15. Populated the "Internationalization Considerations" section.
16. Populated the "Extension Example" section.
17. Added <deDate> element under <domain> element.

18. Added <icannID> element under <registrar> element.

19. Added <eppParams> element under root element.

20. Fixed some typographical errors and omissions.

12.2. Changes from version 01 to 02

1. Added definition for "canonical" in the "IDN variants Handling" section.

2. Clarified that "blocked" and "reserved" IDN variants are optional.

3. Made <rdeRegistrar:authInfo> optional.

4. Introduced substitutionGroup as the mechanism for extending the protocol.

5. Moved <eppParams> element to be child of <contents>

6. Text improvements in the Introduction, Terminology, and Problem Scope per Jay's suggestion.

7. Removed <trDate> from <rdeDomain> and added <trnData> instead, which include all the data from the last (pending/processed) transfer request

8. Removed <trDate> from <rdeContact> and added <trnData> instead, which include all the data from the last (pending/processed) transfer request

9. Fixed some typographical errors and omissions.

12.3. Changes from version 02 to 03

1. Separated domain name objects from protocol.

2. Moved <extension> elements to be child of <deletes> and <contents>, additionally removed <extension> element from <rdeDomain>, <rdeHost>, <rdeContact>, <rdeRegistrar> and <rdeIDN> elements.

3. Modified the definition of <rde:id> and <rde:prevId>.

4. Added <rdeMenu> element under <deposit> element.

5. Fixed some typographical errors and omissions.

12.4. Changes from version 03 to 04

1. Removed <eppParams> objects.
2. Populated the "Extension Guidelines" section.
3. Fixed some typographical errors and omissions.

12.5. Changes from version 04 to 05

1. Fixes to the XSD
2. Extension Guidelines moved to dnrd-mappings draft
3. Fixed some typographical errors and omissions.

12.6. Changes from version 05 to 06

1. Fix resend definition.

12.7. Changes from version 06 to 07

1. Editorial updates.
2. schemaLocation removed from RDE Schema.

12.8. Changes from version 07 to 08

1. Ping update

12.9. Changes from version 08 to 09

1. Ping update.

12.10. Changes from version 09 to 10

1. Implementation Status section was added

12.11. Changes from version 10 to 11

1. Ping update.

13. References

13.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3339] Klyne, G. and C. Newman, "Date and Time on the Internet: Timestamps", RFC 3339, DOI 10.17487/RFC3339, July 2002, <<https://www.rfc-editor.org/info/rfc3339>>.

13.2. Informative References

- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC5730] Hollenbeck, S., "Extensible Provisioning Protocol (EPP)", STD 69, RFC 5730, DOI 10.17487/RFC5730, August 2009, <<https://www.rfc-editor.org/info/rfc5730>>.
- [RFC7942] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", BCP 205, RFC 7942, DOI 10.17487/RFC7942, July 2016, <<https://www.rfc-editor.org/info/rfc7942>>.

Author's Address

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

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: July 19, 2019

J. Gould
M. Pozun
VeriSign, Inc.
January 15, 2019

Login Security Extension for the Extensible Provisioning Protocol (EPP)
draft-gould-regext-login-security-03

Abstract

The Extensible Provisioning Protocol (EPP) includes a client authentication scheme that is based on a user identifier and password. The structure of the password field is defined by an XML Schema data type that specifies minimum and maximum password length values, but there are no other provisions for password management other than changing the password. This document describes an EPP extension that allows longer passwords to be created and adds additional security features to the EPP login command and response.

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 19, 2019.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must

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	2
1.1. Conventions Used in This Document	3
2. Migrating to Newer Versions of This Extension	3
3. Object Attributes	4
3.1. Event	4
3.2. "[LOGIN-SECURITY]" Password	5
3.3. Dates and Times	6
4. EPP Command Mapping	6
4.1. EPP <login> Command	6
5. Formal Syntax	13
5.1. Login Security Extension Schema	13
6. IANA Considerations	15
6.1. XML Namespace	15
6.2. EPP Extension Registry	16
7. Implementation Status	16
8. Security Considerations	16
9. Acknowledgements	17
10. References	17
10.1. Normative References	17
10.2. Informative References	17
10.3. URIs	17
Appendix A. Change History	18
A.1. Change from 00 to 01	18
A.2. Change from 01 to 02	18
A.3. Change from 02 to 03	18
Authors' Addresses	18

1. Introduction

This document describes an Extensible Provisioning Protocol (EPP) extension for enhancing the security of the EPP login command in EPP RFC 5730. The enhancements include supporting longer passwords (or passphrases) than the 16-character maximum and providing a list of security events in the login response. The password (current and new) in EPP RFC 5730 can be overridden by the password included in the extension to extend past the 16-character maximum. The security events supported include: password expiry, client certificate expiry, insecure cipher, insecure TLS protocol, new password complexity, login security statistical warning, and a custom event. The attributes supported by the security events include identifying the event type or sub-type, indicating the security level of warning or error, a future or past-due expiration date, the value that resulted in the

event, the duration of the statistical event, and a free-form description with an optional language.

1.1. Conventions Used in This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

XML is case sensitive. Unless stated otherwise, XML specifications and examples provided in this document MUST be interpreted in the character case presented in order to develop a conforming implementation.

In examples, "C:" represents lines sent by a protocol client and "S:" represents lines returned by a protocol server. Indentation and white space in examples are provided only to illustrate element relationships and are not a REQUIRED feature of this protocol.

"loginSec-0.3" is used as an abbreviation for "urn:ietf:params:xml:ns:epp:loginSec-0.3". The XML namespace prefix "loginSec" is used, but implementations MUST NOT depend on it and instead employ a proper namespace-aware XML parser and serializer to interpret and output the XML documents.

2. Migrating to Newer Versions of This Extension

(Note to RFC Editor: remove this section before publication as an RFC.)

Servers which implement this extension SHOULD provide a way for clients to progressively update their implementations when a new version of the extension is deployed.

Servers SHOULD (for a temporary migration period) provide support for older versions of the extension in parallel to the newest version, and allow clients to select their preferred version via the <svcExtension> element of the <login> command.

If a client requests multiple versions of the extension at login, then, when preparing responses to commands which do not include extension elements, the server SHOULD only include extension elements in the namespace of the newest version of the extension requested by the client.

When preparing responses to commands which do include extension elements, the server SHOULD only include extension elements for the extension versions present in the command.

3. Object Attributes

This extension adds additional elements to [RFC5730] login command and response. Only those new elements are described here.

3.1. Event

A security event, using the <loginSec:event> element, represents either a warning or error identified by the server after the client has connected and submitted the login command. There MAY be multiple events returned that provides information for the client to address. The <loginSec:event> MAY include a free form description. All of the security events use a consistent set of attributes, where the exact set of applicable attributes is based on the event type. The supported set of <loginSec:event> element attributes include:

"type": A REQUIRED attribute that defines the type of security event. The enumerated list of "type" values include:

- "password": Identifies a password expiry event, where the password expires in the future or has expired based on the "exDate" date and time.
- "certificate": Identifies a client certificate expiry event, where the client certificate will expire at the "exDate" date and time.
- "cipher": Identifies the use of an insecure or deprecated TLS cipher suite.
- "tlsProtocol": Identifies the use of an insecure or deprecated TLS protocol.
- "newPW": The new password does not meet the server password complexity requirements.
- "stat": Provides a login security statistical warning that MUST set the "name" of the statistic.
- "custom": Custom event type that MUST set the "name" attribute with the custom event type name.
- "name": Used to define a sub-type or the type name when the "type" attribute is "custom".
- "level": Defines the level of the event as either "warning" for a warning event that needs action, or "error" for an error event that requires immediate action.
- "exDate": Contains the date and time that a "warning" level has or will become an "error" level. At expiry there MAY be an error to connect or MAY be an error to login. An example is an expired certificate that will result in a error to connect or an expired password that may result in a failed login.

"value": Identifies the value that resulted in the login security event. An example is the negotiated insecure cipher suite or the negotiated insecure TLS protocol.

"duration": Defines the duration that a statistical event is associated with.

"lang": Identifies the language of the free form description if the negotiated language is something other than the default value of "en" (English).

Example login security event for a password expiring in a week:

```
<loginSec:event
  type="password"
  level="warning"
  exDate="2018-04-01T22:00:00.0Z"
  lang="en">
  Password expiration soon
</loginSec:event>
```

Example login security event for identifying 100 failed logins over the last day, using the "stat" sub-type of "failedLogins":

```
<loginSec:event
  type="stat"
  name="failedLogins"
  level="warning"
  value="100"
  duration="P1D">
  Excessive invalid daily logins
</loginSec:event>
```

3.2. "[LOGIN-SECURITY]" Password

The <loginSec:pw> element MUST override the [RFC5730] <pw> element only if the <pw> contains the predefined value of "[LOGIN-SECURITY]", which is a constant value for the server to use the <loginSec:pw> element for the password. Similarly, the <loginSec:newPW> element MUST override the [RFC5730] <newPW> element only if the <newPW> contains the predefined value of "[LOGIN-SECURITY]", which is a constant value for the server to use the <loginSec:newPW> element for the new password. The "[LOGIN-SECURITY]" pre-defined string MUST be supported by the server for the client to explicitly indicate to the server whether to use <loginSec:pw> element in place of the [RFC5730] <pw> element or to use the <loginSec:newPW> in place of the [RFC5730] <newPW> element.

3.3. Dates and Times

Date and time attribute values MUST be represented in Universal Coordinated Time (UTC) using the Gregorian calendar. The extended date-time form using upper case "T" and "Z" characters defined in [W3C.REC-xmlschema-2-20041028] MUST be used to represent date-time values, as XML Schema does not support truncated date-time forms or lower case "T" and "Z" characters.

4. EPP Command Mapping

A detailed description of the EPP syntax and semantics can be found in the EPP core protocol specification [RFC5730].

4.1. EPP <login> Command

This extension defines additional elements to extend the EPP <login> command and response to be used in conjunction with [RFC5730].

The EPP <login> command is used to establish a session with an EPP server. This extension overrides the password that is passed with the [RFC5730] <pw> or the <newPW> element as defined in Section 3.2. A <loginSec:loginSec> element is sent along with the [RFC5730] <login> command and MUST contain at least one of the following child elements:

<loginSec:userAgent>: OPTIONAL client user agent that identifies the client software and platform used by the server to identify functional or security constraints, current security issues, and potential future functional or security issues for the client.

<loginSec:pw>: OPTIONAL plain text password that is case sensitive, has a minimum length of 6 characters, and has a maximum length that is up to server policy. All leading and trailing whitespace is removed, and all internal contiguous whitespace that includes #x9 (tab), #xA (linefeed), #xD (carriage return), and #x20 (space) is replaced with a single #x20 (space). This element MUST only be used if the [RFC5730] <pw> element is set to the "[LOGIN-SECURITY]" value.

<loginSec:newPW>: OPTIONAL plain text new password that is case sensitive, has a minimum length of 6 characters, and has a maximum length that is up to server policy. All leading and trailing whitespace is removed, and all internal contiguous whitespace that includes #x9 (tab), #xA (linefeed), #xD (carriage return), and #x20 (space) is replaced with a single #x20 (space). This element MUST only be used if the [RFC5730] <newPW> element is set to the "[LOGIN-SECURITY]" value.

Example login command that uses the <loginSec:pw> element instead of the [RFC5730] <pw> element to establish the session and includes the <loginSec:userAgent> element:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <login>
C:      <clID>ClientX</clID>
C:      <pw>[LOGIN-SECURITY]</pw>
C:      <options>
C:        <version>1.0</version>
C:        <lang>en</lang>
C:      </options>
C:      <svcs>
C:        <objURI>urn:ietf:params:xml:ns:obj1</objURI>
C:        <objURI>urn:ietf:params:xml:ns:obj2</objURI>
C:        <objURI>urn:ietf:params:xml:ns:obj3</objURI>
C:        <svcExtension>
C:          <extURI>urn:ietf:params:xml:ns:epp:loginSec-0.3</extURI>
C:        </svcExtension>
C:      </svcs>
C:    </login>
C:    <extension>
C:      <loginSec:loginSec
C:        xmlns:loginSec=
C:          "urn:ietf:params:xml:ns:epp:loginSec-0.3">
C:        <loginSec:userAgent>EPP SDK/1.0.0
C:          (Java 1.7.0_15; x86_64 Mac OS X 10.11.6)
C:        </loginSec:userAgent>
C:        <loginSec:pw>this is a long password</loginSec:pw>
C:      </loginSec:loginSec>
C:    </extension>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

Example login command that uses the <loginSec:pw> element instead of the [RFC5730] <pw> element to establish the session, and uses the <loginSec:newPW> element instead of the [RFC5730] <newPW> element to set the new password:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <login>
C:      <clID>ClientX</clID>
C:      <pw>[LOGIN-SECURITY]</pw>
C:      <newPW>[LOGIN-SECURITY]</newPW>
C:      <options>
C:        <version>1.0</version>
C:        <lang>en</lang>
C:      </options>
C:      <svcs>
C:        <objURI>urn:ietf:params:xml:ns:obj1</objURI>
C:        <objURI>urn:ietf:params:xml:ns:obj2</objURI>
C:        <objURI>urn:ietf:params:xml:ns:obj3</objURI>
C:        <svcExtension>
C:          <extURI>urn:ietf:params:xml:ns:epp:loginSec-0.3</extURI>
C:        </svcExtension>
C:      </svcs>
C:    </login>
C:    <extension>
C:      <loginSec:loginSec
C:        xmlns:loginSec=
C:          "urn:ietf:params:xml:ns:epp:loginSec-0.3">
C:        <loginSec:pw>this is a long password
C:      </loginSec:pw>
C:      <loginSec:newPW>new password that is still long
C:    </loginSec:newPW>
C:    </loginSec:loginSec>
C:  </extension>
C:  <clTRID>ABC-12345</clTRID>
C: </command>
C:</epp>
```

Example login command that uses the [RFC5730] <pw> element to establish the session, and uses the <loginSec:newPW> element instead of the [RFC5730] <newPW> element to set the new password:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <login>
C:      <clID>ClientX</clID>
C:      <pw>shortpassword</pw>
C:      <newPW>[LOGIN-SECURITY]</newPW>
C:      <options>
C:        <version>1.0</version>
C:        <lang>en</lang>
C:      </options>
C:      <svcs>
C:        <objURI>urn:ietf:params:xml:ns:obj1</objURI>
C:        <objURI>urn:ietf:params:xml:ns:obj2</objURI>
C:        <objURI>urn:ietf:params:xml:ns:obj3</objURI>
C:        <svcExtension>
C:          <extURI>urn:ietf:params:xml:ns:epp:loginSec-0.3</extURI>
C:        </svcExtension>
C:      </svcs>
C:    </login>
C:    <extension>
C:      <loginSec:loginSec
C:        xmlns:loginSec=
C:          "urn:ietf:params:xml:ns:epp:loginSec-0.3">
C:        <loginSec:newPW>new password that is still long
C:      </loginSec:newPW>
C:    </loginSec:loginSec>
C:  </extension>
C:  <clTRID>ABC-12345</clTRID>
C: </command>
C:</epp>
```

Upon a completed login command (success or failed), the extension MUST be included in the response based on the following conditions:

Client supports extension: client supports the extension based on the <svcExtension> element of the <login> command.

At least one login security event: The server has identified at least one login security event to communicate to the client.

The extension to the EPP response uses the <loginSec:loginSecData> element that contains the following child elements:

<loginSec:event>: One or more <loginSec:event> elements defined in Section 3.1.

Example EPP response to a successful login command where the password will expire in a week:

```
S:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S:  <response>
S:    <result code="1000">
S:      <msg>Command completed successfully</msg>
S:    </result>
S:    <extension>
S:      <loginSec:loginSecData
S:        xmlns:loginSec=
S:          "urn:ietf:params:xml:ns:epp:loginSec-0.3">
S:        <loginSec:event
S:          type="password"
S:          level="warning"
S:          exDate="2018-04-01T22:00:00.0Z"
S:          lang="en">
S:          Password expiring in a week
S:        </loginSec:event>
S:      </loginSec:loginSecData>
S:    </extension>
S:    <trID>
S:      <clTRID>ABC-12345</clTRID>
S:      <svTRID>54321-XYZ</svTRID>
S:    </trID>
S:  </response>
S:</epp>
```

Example EPP response to a failed login command where the password has expired and the new password does not meet the server complexity requirements:

```
S:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S:  <response>
S:    <result code="2200">
S:      <msg>Authentication error</msg>
S:    </result>
S:    <extension>
S:      <loginSec:loginSecData
S:        xmlns:loginSec=
S:          "urn:ietf:params:xml:ns:epp:loginSec-0.3">
S:        <loginSec:event
S:          type="password"
S:          level="error"
S:          exDate="2018-03-26T22:00:00.0Z">
S:          Password has expired
S:        </loginSec:event>
S:        <loginSec:event
S:          type="newPW"
S:          level="error">
S:          New password does not meet complexity requirements
S:        </loginSec:event>
S:      </loginSec:loginSecData>
S:    </extension>
S:    <trID>
S:      <clTRID>ABC-12345</clTRID>
S:      <svTRID>54321-XYZ</svTRID>
S:    </trID>
S:  </response>
S:</epp>
```

Example EPP response to a successful login command where there is a set of login security events:

```
S:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S:  <response>
S:    <result code="1000">
S:      <msg>Command completed successfully</msg>
S:    </result>
S:    <extension>
S:      <loginSec:loginSecData
S:        xmlns:loginSec=
S:          "urn:ietf:params:xml:ns:epp:loginSec-0.3">
S:        <loginSec:event
```



```
S:         type="password"
S:         level="warning"
S:         exDate="2018-04-01T22:00:00.0Z"
S:         lang="en">
S:         Password expiration soon
S:     </loginSec:event>
S:     <loginSec:event
S:         type="certificate"
S:         level="warning"
S:         exDate="2018-04-02T22:00:00.0Z"/>
S:     <loginSec:event
S:         type="cipher"
S:         level="warning"
S:         value="TLS_RSA_WITH_AES_128_CBC_SHA">
S:         Non-PFS Cipher negotiated
S:     </loginSec:event>
S:     <loginSec:event
S:         type="tlsProtocol"
S:         level="warning"
S:         value="TLSv1.0">
S:         Insecure TLS protocol negotiated
S:     </loginSec:event>
S:     <loginSec:event
S:         type="stat"
S:         name="failedLogins"
S:         level="warning"
S:         value="100"
S:         duration="P1D">
S:         Excessive invalid daily logins
S:     </loginSec:event>
S:     <loginSec:event
S:         type="custom"
S:         name="myCustomEvent"
S:         level="warning">
S:         A custom login security event occurred
S:     </loginSec:event>
S: </loginSec:loginSecData>
S: </extension>
S: <trID>
S:     <clTRID>ABC-12345</clTRID>
S:     <svTRID>54321-XYZ</svTRID>
S: </trID>
S: </response>
S: </epp>
```

5. Formal Syntax

One schema is presented here that is the EPP Login Security Extension schema.

The formal syntax presented here is a complete schema representation of the object mapping suitable for automated validation of EPP XML instances. The BEGIN and END tags are not part of the schema; they are used to note the beginning and ending of the schema for URI registration purposes.

5.1. Login Security Extension Schema

```
BEGIN
<?xml version="1.0" encoding="UTF-8"?>
  <schema targetNamespace="urn:ietf:params:xml:ns:epp:loginSec-0.3"
    xmlns:eppcom="urn:ietf:params:xml:ns:eppcom-1.0"
    xmlns:epp="urn:ietf:params:xml:ns:epp-1.0"
    xmlns:loginSec="urn:ietf:params:xml:ns:epp:loginSec-0.3"
    xmlns="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="qualified">

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

    <annotation>
      <documentation>
        Extensible Provisioning Protocol v1.0
        Login Security Extension Schema.
      </documentation>
    </annotation>

    <!-- login command extension elements -->
    <element name="loginSec" type="loginSec:loginSecType"/>

    <!--
    Attributes associated with the login command extension.
    -->
    <complexType name="loginSecType">
      <sequence>
        <element name="userAgent" type="token"
          minOccurs="0"/>
        <element name="pw" type="loginSec:pwType"
          minOccurs="0"/>
      </sequence>
    </complexType>
  </schema>
END
```

```
        <element name="newPW" type="loginSec:pwType"
          minOccurs="0"/>
      </sequence>
    </complexType>

    <simpleType name="pwType">
      <restriction base="token">
        <minLength value="6"/>
      </restriction>
    </simpleType>

    <!-- login response extension elements -->
    <element name="loginSecData" type="loginSec:loginSecDataType"/>

    <!--
      Attributes associated with the change.
    -->
    <complexType name="loginSecDataType">
      <sequence>
        <element name="event" type="loginSec:eventType"
          minOccurs="1" maxOccurs="unbounded"/>
      </sequence>
    </complexType>

    <complexType name="eventType">
      <simpleContent>
        <extension base="normalizedString">
          <attribute name="type"
            type="loginSec:typeEnum" use="required"/>
          <attribute name="name" type="token"/>
          <attribute name="level"
            type="loginSec:levelEnum" use="required"/>
          <attribute name="exDate" type="dateTime"/>
          <attribute name="value" type="token"/>
          <attribute name="duration"
            type="duration"/>
          <attribute name="lang"
            type="language" default="en"/>
        </extension>
      </simpleContent>
    </complexType>

    <!--
      Enumerated list of event types, with extensibility via "custom".
    -->
    <simpleType name="typeEnum">
      <restriction base="token">
```

```
        <enumeration value="password"/>
        <enumeration value="certificate"/>
        <enumeration value="cipher"/>
        <enumeration value="tlsProtocol"/>
        <enumeration value="newPW"/>
        <enumeration value="stat"/>
        <enumeration value="custom"/>
    </restriction>
</simpleType>

<!--
Enumerated list of levels.
-->
<simpleType name="levelEnum">
    <restriction base="token">
        <enumeration value="warning"/>
        <enumeration value="error"/>
    </restriction>
</simpleType>

<!--
End of schema.
-->
</schema>
END
```

6. IANA Considerations

6.1. XML Namespace

This document uses URNs to describe XML namespaces and XML schemas conforming to a registry mechanism described in [RFC3688]. The following URI assignment is requested of IANA:

Registration request for the loginSec namespace:

URI: urn:ietf:params:xml:ns:epp:loginSec-0.3
Registrant Contact: IESG
XML: None. Namespace URIs do not represent an XML specification.

Registration request for the loginSec XML schema:

URI: urn:ietf:params:xml:schema:epp:loginSec-0.3
Registrant Contact: IESG
XML: See the "Formal Syntax" section of this document.

6.2. EPP Extension Registry

The EPP extension described in this document should be registered by the IANA in the EPP Extension Registry described in [RFC7451]. The details of the registration are as follows:

Name of Extension: "Login Security Extension for the Extensible Provisioning Protocol (EPP)"

Document status: Standards Track

Reference: (insert reference to RFC version of this document)

Registrant Name and Email Address: IESG, <iesg@ietf.org>

TLDs: Any

IPR Disclosure: None

Status: Active

Notes: None

7. Implementation Status

Note to RFC Editor: Please remove this section and the reference to RFC 7942 [RFC7942] before publication.

TBD

8. Security Considerations

The extension leaves the password (<pw> element) and new password (<newPW> element) minimum length beyond 6 characters and the maximum length up to sever policy. The server SHOULD enforce minimum and maximum length requirements that are appropriate for their operating environment. One example of a guideline for password length policies can be found in section 5 of NIST Special Publication 800-63B [1].

The client SHOULD NOT decrease the security of a new password by decreasing the length of the current password. For example, a client with a 20 character password set using the extension, should not use the login command in [RFC5730] without using the extension, to set a new password that is less than or equal to 16 characters.

The extension provides an extensible list of login security events to inform clients of connection and login warnings and errors.

9. Acknowledgements

The authors wish to thank the following persons for their feedback and suggestions:

- o Patrick Mevzek
- o Scott Hollenbeck

10. References

10.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC5730] Hollenbeck, S., "Extensible Provisioning Protocol (EPP)", STD 69, RFC 5730, DOI 10.17487/RFC5730, August 2009, <<https://www.rfc-editor.org/info/rfc5730>>.
- [RFC7942] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", BCP 205, RFC 7942, DOI 10.17487/RFC7942, July 2016, <<https://www.rfc-editor.org/info/rfc7942>>.
- [W3C.REC-xmlschema-2-20041028] Biron, P. and A. Malhotra, "XML Schema Part 2: Datatypes Second Edition", World Wide Web Consortium Recommendation REC-xmlschema-2-20041028, October 2004, <<http://www.w3.org/TR/2004/REC-xmlschema-2-20041028>>.

10.2. Informative References

- [RFC7451] Hollenbeck, S., "Extension Registry for the Extensible Provisioning Protocol", RFC 7451, DOI 10.17487/RFC7451, February 2015, <<https://www.rfc-editor.org/info/rfc7451>>.

10.3. URIs

- [1] <https://pages.nist.gov/800-63-3/sp800-63b.html>

Appendix A. Change History

A.1. Change from 00 to 01

1. Based on the feedback from Patrick Mevzek and a proposal from Scott Hollenbeck, changed the minimum length of the password from 8 to 6, revised the description of the password, and added text in the Security Considerations section for the server password length policy.

A.2. Change from 01 to 02

1. Changed the XML namespace from urn:ietf:params:xml:ns:loginSec-0.3 to urn:ietf:params:xml:ns:epp:loginSec-0.3, and changed the XML schema registration from urn:ietf:params:xml:ns:loginSec-0.3 to urn:ietf:params:xml:schema:epp:loginSec-0.3 based on a request from IANA with draft-ietf-regext-allocation-token.

A.3. Change from 02 to 03

Updates based on the review by Patrick Mevzek, that include:

1. Fix the inconsistent case for newPW, that required a global change in the draft text and an update to the XML schema to "urn:ietf:params:xml:ns:loginSec-0.3".
2. Changed "contains the following child elements" to "MUST contain at least one of the following child elements", section "EPP <login> Command" to ensure that an empty <loginSec:loginSec> element is not passed.
3. Add "The client SHOULD NOT decrease the security of a new password by decreasing the length of the current password." along with an example to the "Security Considerations" section.

Authors' Addresses

James Gould
VeriSign, Inc.
12061 Bluemont Way
Reston, VA 20190
US

Email: jgould@verisign.com
URI: <http://www.verisign.com>

Matthew Pozun
VeriSign, Inc.
12061 Bluemont Way
Reston, VA 20190
US

Email: mpozun@verisign.com
URI: <http://www.verisign.com>

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 7, 2019

J. Gould
VeriSign, Inc.
K. Feher
Neustar
October 04, 2018

Allocation Token Extension for the Extensible Provisioning Protocol
(EPP)
draft-ietf-regext-allocation-token-12

Abstract

This document describes an Extensible Provisioning Protocol (EPP) extension for including an Allocation Token in "query" and "transform" commands. The Allocation Token is used as a credential that authorizes a client to request the allocation of a specific object from the server, using one of the EPP transform commands including create and transfer.

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 <https://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 April 7, 2019.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must

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	3
1.1. Conventions Used in This Document	3
2. Object Attributes	4
2.1. Allocation Token	4
3. EPP Command Mapping	5
3.1. EPP Query Commands	5
3.1.1. EPP <check> Command	5
3.1.2. EPP <info> Command	9
3.1.3. EPP <transfer> Query Command	11
3.2. EPP Transform Commands	12
3.2.1. EPP <create> Command	12
3.2.2. EPP <delete> Command	13
3.2.3. EPP <renew> Command	13
3.2.4. EPP <transfer> Command	13
3.2.5. EPP <update> Command	14
4. Formal Syntax	15
4.1. Allocation Token Extension Schema	15
5. IANA Considerations	16
5.1. XML Namespace	16
5.2. EPP Extension Registry	16
6. Implementation Status	16
6.1. Verisign EPP SDK	17
6.2. Neustar EPP SDK	17
6.3. Neustar gTLD SRS	18
6.4. Net::DRI	18
7. Security Considerations	19
8. Acknowledgements	19
9. References	19
9.1. Normative References	19
9.2. Informative References	20
Appendix A. Change History	20
A.1. Change from 00 to 01	20
A.2. Change from 01 to 02	20
A.3. Change from 02 to 03	21
A.4. Change from 03 to 04	21
A.5. Change from 04 to REGEXT 00	21
A.6. Change from REGEXT 00 to REGEXT 01	21
A.7. Change from REGEXT 01 to REGEXT 02	21
A.8. Change from REGEXT 02 to REGEXT 03	21
A.9. Change from REGEXT 03 to REGEXT 04	21
A.10. Change from REGEXT 04 to REGEXT 05	21
A.11. Change from REGEXT 05 to REGEXT 06	22

A.12. Change from REGEXT 06 to REGEXT 07	23
A.13. Change from REGEXT 07 to REGEXT 08	23
A.14. Change from REGEXT 08 to REGEXT 09	24
A.15. Change from REGEXT 09 to REGEXT 10	24
A.16. Change from REGEXT 10 to REGEXT 11	25
A.17. Change from REGEXT 11 to REGEXT 12	26
Authors' Addresses	26

1. Introduction

This document describes an extension mapping for version 1.0 of the Extensible Provisioning Protocol (EPP) [RFC5730]. This mapping, an extension to EPP object mappings like the EPP domain name mapping [RFC5731], supports passing an Allocation Token as a credential that authorizes a client to request the allocation of a specific object from the server, using one of the EPP transform commands including create and transfer.

Allocation is when a server assigns the sponsoring client of an object based on the use of an Allocation Token credential. Examples include allocating a registration based on a pre-eligibility Allocation Token, allocating a premium domain name registration based on an auction Allocation Token, allocating a registration based on a founders Allocation Token, and allocating an existing domain name held by the server or by a different sponsoring client based on an Allocation Token passed with a transfer command.

Clients pass an Allocation Token to the server for validation, and the server determines if the supplied Allocation Token is one supported by the server. It is up to server policy which EPP transform commands and which objects require the Allocation Token. The Allocation Token MAY be returned to an authorized client for passing out-of-band to a client that uses it with an EPP transform command.

1.1. Conventions Used in This Document

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

XML is case sensitive. Unless stated otherwise, XML specifications and examples provided in this document MUST be interpreted in the character case presented in order to develop a conforming implementation.

In examples, "C:" represents lines sent by a protocol client and "S:" represents lines returned by a protocol server. Indentation and white space in the examples are provided only to illustrate element relationships and are not REQUIRED in the protocol.

The XML namespace prefix "allocationToken" is used for the namespace "urn:ietf:params:xml:ns:allocationToken-1.0", but implementations MUST NOT depend on it and instead employ a proper namespace-aware XML parser and serializer to interpret and output the XML documents.

The "abc123" token value is used as a placeholder value in the examples. The server MUST support token values that follow the Security Considerations (Section 7) section.

The domain object attribute values, including the "2fooBAR" <domain:pw> value, in the examples are provided for illustration purposes only. Refer to [RFC5731] for details on the domain object attributes.

2. Object Attributes

This extension adds additional elements to EPP object mappings like the EPP domain name mapping [RFC5731]. Only those new elements are described here.

2.1. Allocation Token

The Allocation Token is a simple XML "token" type. The exact format of the Allocation Token is up to server policy. The server MAY have the Allocation Token for each object to match against the Allocation Token passed by the client to authorize the allocation of the object. The <allocationToken:allocationToken> element is used for all of the supported EPP commands as well as the info response. If the supplied Allocation Token passed to the server does not apply to the object, the server MUST return an EPP error result code of 2201.

Authorization information, like what is defined in the EPP domain name mapping [RFC5731], is associated with objects to facilitate transfer operations. The authorization information is assigned when an object is created. The Allocation Token and the authorization information are both credentials, but used for different purposes and used in different ways. The Allocation Token is used to facilitate the allocation of an object instead of transferring the sponsorship of the object. The Allocation Token is not managed by the client, but is validated by the server to authorize assigning the initial sponsoring client of the object.

An example <allocationToken:allocationToken> element with value of "abc123":

```
<allocationToken:allocationToken xmlns:allocationToken=
    "urn:ietf:params:xml:ns:allocationToken-1.0">
  abc123
</allocationToken:allocationToken>
```

3. EPP Command Mapping

A detailed description of the EPP syntax and semantics can be found in the EPP core protocol specification [RFC5730].

3.1. EPP Query Commands

EPP provides three commands to retrieve object information: <check> to determine if an object can be provisioned, <info> to retrieve information associated with an object, and <transfer> to retrieve object transfer status information.

3.1.1. EPP <check> Command

This extension defines additional elements to extend the EPP <check> command of an object mapping like [RFC5731].

This extension allows clients to check the availability of an object with an Allocation Token, as described in Section 2.1. Clients can check if an object can be created using the Allocation Token. The Allocation Token is applied to all object names included in the EPP <check> command.

Example <check> command for the allocation.example domain name using the <allocationToken:allocationToken> extension with the allocation token of 'abc123':

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <check>
C:      <domain:check
C:        xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
C:          <domain:name>allocation.example</domain:name>
C:        </domain:check>
C:      </check>
C:    <extension>
C:      <allocationToken:allocationToken
C:        xmlns:allocationToken=
C:          "urn:ietf:params:xml:ns:allocationToken-1.0">
C:        abc123
C:      </allocationToken:allocationToken>
C:    </extension>
C:  <clTRID>ABC-12345</clTRID>
C: </command>
C:</epp>
```

If the query was successful, the server replies with a <check> response providing the availability status of the queried object based on the following Allocation Token cases, where the object is otherwise available:

1. If an object requires an Allocation Token and the Allocation Token does apply to the object, then the server MUST return the availability status as available (e.g., "avail" attribute is "1" or "true").
2. If an object requires an Allocation Token and the Allocation Token does not apply to the object, then the server SHOULD return the availability status as unavailable (e.g., "avail" attribute is "0" or "false").
3. If an object does not require an Allocation Token, the server MAY return the availability status as available (e.g., "avail" attribute is "1" or "true").

Example <check> domain response for a <check> command using the <allocationToken:allocationToken> extension:

```
S:<?xml version="1.0" encoding="UTF-8"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S: <response>
S:   <result code="1000">
S:     <msg lang="en-US">Command completed successfully</msg>
S:   </result>
S:   <resData>
S:     <domain:chkData
S:       xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
S:       <domain:cd>
S:         <domain:name avail="1">allocation.example</domain:name>
S:       </domain:cd>
S:     </domain:chkData>
S:   </resData>
S:   <trID>
S:     <clTRID>ABC-DEF-12345</clTRID>
S:     <svTRID>54321-XYZ</svTRID>
S:   </trID>
S: </response>
S:</epp>
```

Example <check> command with the <allocationToken:allocationToken> extension for the allocation.example and allocation2.example domain names. Availability of allocation.example and allocation2.example domain names are based on the Allocation Token 'abc123':

```
C:<?xml version="1.0" encoding="UTF-8"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <check>
C:      <domain:check
C:        xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
C:          <domain:name>allocation.example</domain:name>
C:          <domain:name>allocation2.example</domain:name>
C:        </domain:check>
C:      </check>
C:    <extension>
C:      <allocationToken:allocationToken
C:        xmlns:allocationToken=
C:          "urn:ietf:params:xml:ns:allocationToken-1.0">
C:        abc123
C:      </allocationToken:allocationToken>
C:    </extension>
C:  <clTRID>ABC-DEF-12345</clTRID>
C: </command>
C:</epp>
```


Example <check> domain response for multiple domain names in the <check> command using the <allocationToken:allocationToken> extension, where the Allocation Token 'abc123' matches allocation.example but does not match allocation2.example:

```
S:<?xml version="1.0" encoding="UTF-8"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S: <response>
S:   <result code="1000">
S:     <msg lang="en-US">Command completed successfully</msg>
S:   </result>
S:   <resData>
S:     <domain:chkData
S:       xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
S:       <domain:cd>
S:         <domain:name avail="1">allocation.example</domain:name>
S:       </domain:cd>
S:       <domain:cd>
S:         <domain:name avail="0">allocation2.example</domain:name>
S:         <domain:reason>Allocation Token mismatch</domain:reason>
S:       </domain:cd>
S:     </domain:chkData>
S:   </resData>
S:   <trID>
S:     <clTRID>ABC-DEF-12345</clTRID>
S:     <svTRID>54321-XYZ</svTRID>
S:   </trID>
S: </response>
S:</epp>
```

This extension does not add any elements to the EPP <check> response described in the [RFC5730].

3.1.2. EPP <info> Command

This extension defines additional elements to extend the EPP <info> command of an object mapping like [RFC5731].

The EPP <info> command allows a client to request information associated with an existing object. Authorized clients MAY retrieve the Allocation Token (Section 2.1) along with the other object information by supplying the <allocationToken:info> element in the command. The <allocationToken:info> element is an empty element that serves as a marker to the server to return the <allocationToken:allocationToken> element in the info response. If the client is not authorized to receive the Allocation Token, the server MUST return an EPP error result code of 2201. If the client

is authorized to receive the Allocation Token, but there is no Allocation Token associated with the object, the server MUST return an EPP error result code of 2303. The authorization is subject to server policy.

Example <info> command with the allocationToken:info extension for the allocation.example domain name:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <info>
C:      <domain:info
C:        xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
C:        xsi:schemaLocation="urn:ietf:params:xml:ns:domain-1.0
C:        domain-1.0.xsd">
C:        <domain:name>allocation.example</domain:name>
C:      </domain:info>
C:    </info>
C:    <extension>
C:      <allocationToken:info
C:        xmlns:allocationToken=
C:        "urn:ietf:params:xml:ns:allocationToken-1.0/>
C:    </extension>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

If the query was successful, the server replies with an <allocationToken:allocationToken> element along with the regular EPP <resData>. The <allocationToken:allocationToken> element is described in Section 2.1.

Example <info> domain response using the
<allocationToken:allocationToken> extension:

```
S:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S:  <response>
S:    <result code="1000">
S:      <msg>Command completed successfully</msg>
S:    </result>
S:    <resData>
S:      <domain:infData
S:        xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
S:        <domain:name>allocation.example</domain:name>
S:        <domain:roid>EXAMPLE1-REP</domain:roid>
S:        <domain:status s="pendingCreate"/>
S:        <domain:registrant>jd1234</domain:registrant>
S:        <domain:contact type="admin">sh8013</domain:contact>
S:        <domain:contact type="tech">sh8013</domain:contact>
S:        <domain:clID>ClientX</domain:clID>
S:        <domain:crID>ClientY</domain:crID>
S:        <domain:crDate>2012-04-03T22:00:00.0Z</domain:crDate>
S:        <domain:authInfo>
S:          <domain:pw>2fooBAR</domain:pw>
S:        </domain:authInfo>
S:      </domain:infData>
S:    </resData>
S:    <extension>
S:      <allocationToken:allocationToken
S:        xmlns:allocationToken=
S:          "urn:ietf:params:xml:ns:allocationToken-1.0">
S:        abc123
S:      </allocationToken:allocationToken>
S:    </extension>
S:    <trID>
S:      <clTRID>ABC-12345</clTRID>
S:      <svTRID>54321-XYZ</svTRID>
S:    </trID>
S:  </response>
S:</epp>
```

3.1.3. EPP <transfer> Query Command

This extension does not add any elements to the EPP <transfer> query command or <transfer> query response described in [RFC5730].

3.2. EPP Transform Commands

EPP provides five commands to transform objects: <create> to create an instance of an object, <delete> to delete an instance of an object, <renew> to extend the validity period of an object, <transfer> to manage object sponsorship changes, and <update> to change information associated with an object.

3.2.1. EPP <create> Command

This extension defines additional elements to extend the EPP <create> command of an object mapping like [RFC5731].

The EPP <create> command provides a transform operation that allows a client to create an instance of an object. In addition to the EPP command elements described in an object mapping like [RFC5731], the command MUST contain a child <allocationToken:allocationToken> element for the client to be authorized to create and allocate the object. If the Allocation Token does not apply to the object, the server MUST return an EPP error result code of 2201.

Example <create> command to create a domain object with an Allocation Token:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <create>
C:      <domain:create
C:        xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
C:        <domain:name>allocation.example</domain:name>
C:        <domain:registrant>jd1234</domain:registrant>
C:        <domain:contact type="admin">sh8013</domain:contact>
C:        <domain:contact type="tech">sh8013</domain:contact>
C:        <domain:authInfo>
C:          <domain:pw>2fooBAR</domain:pw>
C:        </domain:authInfo>
C:      </domain:create>
C:    </create>
C:    <extension>
C:      <allocationToken:allocationToken
C:        xmlns:allocationToken=
C:        "urn:ietf:params:xml:ns:allocationToken-1.0">
C:        abc123
C:      </allocationToken:allocationToken>
C:    </extension>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

This extension does not add any elements to the EPP <create> response described in the [RFC5730].

3.2.2. EPP <delete> Command

This extension does not add any elements to the EPP <delete> command or <delete> response described in the [RFC5730].

3.2.3. EPP <renew> Command

This extension does not add any elements to the EPP <renew> command or <renew> response described in the [RFC5730].

3.2.4. EPP <transfer> Command

This extension defines additional elements to extend the EPP <transfer> request command of an object mapping like [RFC5731].

The EPP <transfer> request command provides a transform operation that allows a client to request the transfer of an object. In addition to the EPP command elements described in an object mapping like [RFC5731], the command MUST contain a child <allocationToken:allocationToken> element for the client to be authorized to transfer and allocate the object. The authorization associated with the Allocation Token is in addition to and does not replace the authorization mechanism defined for the object's <transfer> request command. If the Allocation Token is invalid or not required for the object, the server MUST return an EPP error result code of 2201.

Example <transfer> request command to allocate the domain object with the Allocation Token:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <transfer op="request">
C:      <domain:transfer
C:        xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
C:        <domain:name>example1.tld</domain:name>
C:        <domain:period unit="y">1</domain:period>
C:        <domain:authInfo>
C:          <domain:pw>2fooBAR</domain:pw>
C:        </domain:authInfo>
C:      </domain:transfer>
C:    </transfer>
C:    <extension>
C:      <allocationToken:allocationToken
C:        xmlns:allocationToken=
C:          "urn:ietf:params:xml:ns:allocationToken-1.0">
C:        abc123
C:      </allocationToken:allocationToken>
C:    </extension>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

This extension does not add any elements to the EPP <transfer> response described in the [RFC5730].

3.2.5. EPP <update> Command

This extension does not add any elements to the EPP <update> command or <update> response described in the [RFC5730].

4. Formal Syntax

One schema is presented here that is the EPP Allocation Token Extension schema.

The formal syntax presented here is a complete schema representation of the object mapping suitable for automated validation of EPP XML instances. The BEGIN and END tags are not part of the schema; they are used to note the beginning and ending of the schema for URI registration purposes.

4.1. Allocation Token Extension Schema

```
BEGIN
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:allocationToken="urn:ietf:params:xml:ns:allocationToken-1.0"
  targetNamespace="urn:ietf:params:xml:ns:allocationToken-1.0"
  elementFormDefault="qualified">
  <annotation>
    <documentation>
      Extensible Provisioning Protocol v1.0
      Allocation Token Extension
    </documentation>
  </annotation>

  <!-- Element used in info command to get allocation token. -->
  <element name="info">
    <complexType>
      <complexContent>
        <restriction base="anyType" />
      </complexContent>
    </complexType>
  </element>

  <!-- Allocation Token used in transform
  commands and info response -->
  <element name="allocationToken"
    type="allocationToken:allocationTokenType" />
  <simpleType name="allocationTokenType">
    <restriction base="token">
      <minLength value="1" />
    </restriction>
  </simpleType>

  <!-- End of schema. -->
</schema>
END
```

5. IANA Considerations

5.1. XML Namespace

This document uses URNs to describe XML namespaces and XML schemas conforming to a registry mechanism described in [RFC3688].

Registration request for the allocationToken namespace:

URI: urn:ietf:params:xml:ns:allocationToken-1.0
Registrant Contact: IESG
XML: None. Namespace URIs do not represent an XML specification.

Registration request for the allocationToken XML schema:

URI: urn:ietf:params:xml:schema:allocationToken-1.0
Registrant Contact: IESG
XML: See the "Formal Syntax" section of this document.

5.2. EPP Extension Registry

The following registration of the EPP Extension Registry, described in [RFC7451], is requested:

Name of Extension: "Allocation Token Extension for the Extensible Provisioning Protocol (EPP)"

Document status: Standards Track

Reference: (insert reference to RFC version of this document)

Registrant Name and Email Address: IESG, <iesg@ietf.org>

TLDs: Any

IPR Disclosure: None

Status: Active

Notes: None

6. Implementation Status

Note to RFC Editor: Please remove this section and the reference to RFC 7942 [RFC7942] before publication.

This section records the status of known implementations of the protocol defined by this specification at the time of posting of this

Internet-Draft, and is based on a proposal described in RFC 7942 [RFC7942]. The description of implementations in this section is intended to assist the IETF in its decision processes in progressing drafts to RFCs. Please note that the listing of any individual implementation here does not imply endorsement by the IETF. Furthermore, no effort has been spent to verify the information presented here that was supplied by IETF contributors. This is not intended as, and must not be construed to be, a catalog of available implementations or their features. Readers are advised to note that other implementations may exist.

According to RFC 7942 [RFC7942], "this will allow reviewers and working groups to assign due consideration to documents that have the benefit of running code, which may serve as evidence of valuable experimentation and feedback that have made the implemented protocols more mature. It is up to the individual working groups to use this information as they see fit".

6.1. Verisign EPP SDK

Organization: Verisign Inc.

Name: Verisign EPP SDK

Description: The Verisign EPP SDK includes both a full client implementation and a full server stub implementation of draft-ietf-regext-allocation-token.

Level of maturity: Production

Coverage: All aspects of the protocol are implemented.

Licensing: GNU Lesser General Public License

Contact: jgould@verisign.com

URL: https://www.verisign.com/en_US/channel-resources/domain-registry-products/epp-sdks

6.2. Neustar EPP SDK

Organisation: Neustar Inc.

Name: Neustar EPP SDK

Description: The Neustar EPP SDK includes a full client implementation of draft-ietf-regext-allocation-token.

Level of maturity: Production

Coverage: All aspects of the protocol are implemented.

Licensing: GNU Lesser General Public License

Contact: quoc-anh.np@team.neustar

URL: <http://registrytoolkit.neustar>

6.3. Neustar gTLD SRS

Organisation: Neustar Inc.

Name: Neustar generic Top Level Domain (gTLD) Shared Registry System (SRS).

Description: The Neustar gTLD SRS implements the server side of draft-ietf-regext-allocation-token for several Top Level Domains.

Level of maturity: Production

Coverage: All server side aspects of the protocol are implemented.

Licensing: Proprietary

Contact: quoc-anh.np@team.neustar

6.4. Net::DRI

Organization: Dot and Co

Name: Net::DRI

Description: Net::DRI implements the client-side of draft-ietf-regext-allocation-token.

Level of maturity: Production

Coverage: All client-side aspects of the protocol are implemented.

Licensing: GNU Lesser General Public License

Contact: netdri@dotandco.com

7. Security Considerations

The mapping described in this document does not provide any security services beyond those described by EPP [RFC5730] and protocol layers used by EPP. The security considerations described in these other specifications apply to this specification as well.

The mapping acts as a conduit for the passing of Allocation Tokens between a client and a server. The definition of the Allocation Token SHOULD be defined outside of this mapping. The following are security considerations in the definition and use of an Allocation Token:

1. An Allocation Token should be considered secret information by the client and SHOULD be protected at rest and MUST be protected in transit.
2. An Allocation Token should be single use, meaning it should be unique per object and per allocation operation.
3. An Allocation Token should have a limited life with some form of expiry in the Allocation Token if generated by a trusted 3rd third party, or with a server-side expiry if generated by the server.
4. An Allocation Token should use a strong random value if it is based on an unsigned code.
5. An Allocation Token should leverage digital signatures to confirm its authenticity if generated by a trusted 3rd party.
6. An Allocation Token that is signed XML should be encoded (e.g., base64 [RFC4648]) to mitigate server validation issues.

8. Acknowledgements

The authors wish to acknowledge the original concept for this draft and the efforts in the initial versions of this draft by Trung Tran and Sharon Wodjenski.

Special suggestions that have been incorporated into this document were provided by Ben Campbell, Scott Hollenbeck, Benjamin Kaduk, Mirja Kuehlewind, Rubens Kuhl, Alexander Mayrhofer, Patrick Mevzek, Eric Rescoria, and Adam Roach.

9. References

9.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC5730] Hollenbeck, S., "Extensible Provisioning Protocol (EPP)", STD 69, RFC 5730, DOI 10.17487/RFC5730, August 2009, <<https://www.rfc-editor.org/info/rfc5730>>.
- [RFC5731] Hollenbeck, S., "Extensible Provisioning Protocol (EPP) Domain Name Mapping", STD 69, RFC 5731, DOI 10.17487/RFC5731, August 2009, <<https://www.rfc-editor.org/info/rfc5731>>.
- [RFC7942] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", BCP 205, RFC 7942, DOI 10.17487/RFC7942, July 2016, <<https://www.rfc-editor.org/info/rfc7942>>.

9.2. Informative References

- [RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, DOI 10.17487/RFC4648, October 2006, <<https://www.rfc-editor.org/info/rfc4648>>.
- [RFC7451] Hollenbeck, S., "Extension Registry for the Extensible Provisioning Protocol", RFC 7451, DOI 10.17487/RFC7451, February 2015, <<https://www.rfc-editor.org/info/rfc7451>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

Appendix A. Change History

A.1. Change from 00 to 01

1. Amended XML Namespace section of IANA Considerations, added EPP Extension Registry section.
2. Moved Change History to the back section as an Appendix.

A.2. Change from 01 to 02

1. Ping update.

A.3. Change from 02 to 03

1. Ping update.

A.4. Change from 03 to 04

1. Updated the authors for the draft.

A.5. Change from 04 to REGEXT 00

1. Changed to regext working group draft by changing draft-gould-allocation-token to draft-ietf-regext-allocation-token.

A.6. Change from REGEXT 00 to REGEXT 01

1. Ping update.

A.7. Change from REGEXT 01 to REGEXT 02

1. Added the Implementation Status section.

A.8. Change from REGEXT 02 to REGEXT 03

1. Changed Neustar author to Kal Feher.

A.9. Change from REGEXT 03 to REGEXT 04

1. Added Neustar implementation to the Implementation Status section.

A.10. Change from REGEXT 04 to REGEXT 05

1. Updates based on feedback from Patrick Mevzek, that include:
 1. Remove "or code" from the Abstract section.
 2. Add a missing "to" in "an allocation token TO one of the EPP..." in the Introduction section.
 3. Reword the "The allocation token is known to the server..." sentence in the Introduction section.
 4. Modify the "The allocation token MAY be returned to an authorized client for passing out-of-band to a client that uses it with an EPP transform command" to clarify who the two separate clients are.
 5. Removed an unneeded ":" from the EPP <transfer> Command and EPP <update> Command sections.

A.11. Change from REGEXT 05 to REGEXT 06

1. Fix description of Neustar gTLD SRS based on feedback from Rubens Kuhl.
2. Updates based on feedback from Alexander Mayrhofer, that include:
 1. Making all references to Allocation Token to use the upper case form.
 2. Revise the language of the abstract to include "for including an Allocation Token in query and transform commands. The Allocation Token is used as a credential that authorizes a client to request the allocation of a specific object from the server, using one of the EPP transform commands..."
 3. Replace the title "EPP <transfer> Command" with "EPP <transfer> Query Command" for section 3.1.3.
 4. Revise the second sentence of the Introduction to "The mapping, ..., supports passing an Allocation Token..."
 5. Change "support" to "require" in the Introduction sentence "It is up to server policy which EPP transform commands and which objects support the Allocation Token."
 6. Add the definition of Allocation to the Introduction.
 7. Removed "transform" from "all of the supported EPP transform commands" in the "Allocation Token" section, since the Allocation Token can be used with the "check" command as well.
 8. Remove the word "same" from "The same <allocationToken:allocationToken> element is used for all..." in the "Allocation Token" section.
 9. Change the description of the use of the 2201 error in the "Allocation Token" section, the "EPP <create> Command" section, the "EPP <transfer> Command" section, and the "EPP <update> Command" section.
 10. Revise "<check> to determine if an object is known to the server..." to "<check> to determine if an object can be provisioned..." and remove "detailed" in the description of the <info> in the "EPP Query Commands" section.
 11. Add missing description of the expected <check> response behavior.
 12. Replaced the example reason "Invalid domain-token pair" with "Allocation Token mismatch".
 13. Replace "information on" with "information associated with" in the "EPP <info> Command" section.
 14. Removed the "that identifies the extension namespace", the ", defined in...", the Allocation Token links from the error response sentences, and the "object referencing the <allocationToken:info> element" in the "EPP <info> Command" section.

15. Added "The authorization is subject to server policy." to the "EPP <info> Command" section.
 16. Replace "or <transfer> response" with "or <transfer> query response" in the "EPP <transfer> Query Command" section.
 17. Replace "create an object" with "create an instance of an object" in the "EPP <create> Command" section.
 18. Revised the sentence to include "the command MUST contain a child <allocationToken:allocationToken> element for the client to be authorized to create and allocate the object" in the "EPP <create> Command" section.
 19. Removed the reference to section 2.1 and the namespace identification text in the "EPP <transfer> Command" section.
 20. Added "The authorization associated with the Allocation Token is in addition to and does not replace the authorization mechanism defined for the object's <transfer> request command." to the "EPP <transfer> Command" section.
 21. Modified the first sentence of the "EPP Extension Registry" section to read "The following registration of the EPP Extension Registry, described in RFC7451, is requested"
 22. Removed support with using the Allocation Token with an empty extension of update (e.g., release command), based on the confusion and lack of known applicability.
3. Updates based on feedback from Scott Hollenbeck, that include:
 1. Revised XML schema to included a minimum length of 1 for the allocationTokenType.
 2. Revised the "IANA Considerations" section to include the registration of the XML schema.
 3. Revised the "Security Considerations" section to include considerations for the definition of the Allocation Tokens.
- A.12. Change from REGEXT 06 to REGEXT 07
1. Updates based on feedback from Patrick Mevzek:
 1. Updated obsoleted RFC 7942 to RFC 7942.
 2. Moved RFC 7451 to an informational reference.
- A.13. Change from REGEXT 07 to REGEXT 08
1. Changed Kal Feher's contact e-mail address.
 2. Changed Neustar's Implementation Status contact e-mail address.
 3. Added the Net::DRI sub-section to the Implementation Status section.

A.14. Change from REGEXT 08 to REGEXT 09

1. Updates based on the AD review by Adam Roach, that include:
 1. In "Abstract", set "query" and "transform" off in some way (e.g., using quotation marks)
 2. In "Conventions Used in This Document", please update to use the boilerplate from RFC 8174.
 3. Remove "allocationToken-1.0" is used as an abbreviation for "urn:ietf:params:xml:ns:allocationToken-1.0".
 4. In "Allocation Token", change "The server MUST have the Allocation Token" to "The server MAY have the Allocation Token".
 5. In "EPP <check> Command", change "This extension allow clients" to "This extension allows clients".
 6. Use domains reserved by RFC 2026 for the examples. The example domain "example.tld" was changed to "allocation.example" and the example domain "example2.tld" was changed to "allocation2.example".
 7. In "EPP <info> Command", change "...the server MUST return an EPP error result code of 2303 object referencing the <allocationToken:info> element." to "...the server MUST return an EPP error result code of 2303."
 8. In "EPP <transfer> Query Command", remove "the" before "RFC5730".
 9. In "EPP <transfer> Command", change "If the Allocation Token does not apply to the object..." to "If the Allocation Token is invalid or not required for the object...".
 10. In "XML Namespace", remove the sentence "The following URI assignment is requested of IANA:"
 11. In "Security Considerations", change "An Allocation Token should is" to "An Allocation Token that is". Also informatively cite RFC 4648 for the base64 reference.
2. Change "ietf:params:xml:ns:allocationToken-1.0" to "ietf:params:xml:schema:allocationToken-1.0" for the XML schema IANA registration.

A.15. Change from REGEXT 09 to REGEXT 10

1. Changed "auhorization" to "authorization" in the "EPP <info> Command" section.
2. Added 'If an object does not require an Allocation Token, the server MAY return the availability status as available (e.g., "avail" attribute is "1" or "true").' to the check response cases, based on feedback by Mirja Kuehlewind.
3. Changed the definition of the <info> element in the XML schema to only allow an empty element, based on IANA's expert review.

4. Added normative language to the storage and transport of the Allocation Token, in the "Security Considerations" section, based on feedback from Eric Rescoria.
5. Changed "The definition of the Allocation Token is defined outside of this mapping" to "The definition of the Allocation Token SHOULD be defined outside of this mapping", in the "Security Considerations" section, based on feedback from Eric Rescoria.
6. Added the missing "urn:" prefix with the IANA URI registrations.
7. The URL for the BCP 14 was removed based on feedback from Alissa Cooper.
8. Updates based on review by Benjamin Kaduk, that include:
 1. Added the second paragraph to the "Allocation Token" section to describe the difference (motivation) of using the Allocation Token versus the EPP RFC authorization mechanism.
 2. Added a paragraph to the "Conventions Used in This Document" section for the use of the "abc123" token value and the use of domain object "2fooBAR" password value in the examples.
 3. Changed the "A client MUST pass an Allocation Token known to the server to be authorized to use one of the supported EPP transform commands." sentence in the "Introduction" section to "Clients pass an Allocation Token to the server for validation, and the server determines if the supplied Allocation Token is one supported by the server."
 4. Changed the "Indentation and white space in the examples are provided only to illustrate element relationships and are not REQUIRED in the protocol." sentence in the "Conventions Used in This Document" section to "Indentation and white space in the examples are provided only to illustrate element relationships and are not REQUIRED in the protocol."
 5. Changed the "Authorized clients MAY retrieve..." sentence in the "EPP <info> Command" section.
 6. Changed the "If the query was successful..." sentence in the "EPP <info> Command" section.
 7. Added "supplied" to the "If the supplied Allocation Token passed..." sentence in the "Allocation Token" section.
 8. Removed an extra newline in the <annotation> element in the "Allocation Token Extension Schema" section.

A.16. Change from REGEXT 10 to REGEXT 11

1. Removed the old duplicate "Authorized clients MAY retrieve..." sentence from section 3.1.2 "EPP <info> Command".

A.17. Change from REGEXT 11 to REGEXT 12

1. Revised the example <check> domain response to first include the positive case for allocation.example, and to second include the negative case for allocation2.example, based on feedback from Ben Campbell. The caption was revised for the example to include the text ", where the Allocation Token 'abc123' matches allocation.example but does not match allocation2.example".

Authors' Addresses

James Gould
VeriSign, Inc.
12061 Bluemont Way
Reston, VA 20190
US

Email: jgould@verisign.com
URI: <http://www.verisigninc.com>

Kal Feher
Neustar
lvl 8/10 Queens Road
Melbourne, VIC 3004
AU

Email: ietf@feherfamily.org
URI: <http://www.neustar.biz>

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: July 8, 2019

J. Gould
VeriSign, Inc.
K. Feher
Neustar
January 4, 2019

Change Poll Extension for the Extensible Provisioning Protocol (EPP)
draft-ietf-regext-change-poll-12

Abstract

This document describes an Extensible Provisioning Protocol (EPP) extension for notifying clients of operations on client-sponsored objects that were not initiated by the client through EPP. These operations may include contractual or policy requirements including but not limited to regular batch processes, customer support actions, Uniform Domain-Name Dispute-Resolution Policy (UDRP) or Uniform Rapid Suspension (URS) actions, court-directed actions, and bulk updates based on customer requests. Since the client is not directly involved or knowledgeable of these operations, the extension is used along with an EPP object mapping to provide the resulting state of the post-operation object, and optionally a pre-operation object, with the operation meta-data of what, when, who, and why.

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 8, 2019.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must 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	3
1.1. Conventions Used in This Document	3
2. Object Attributes	4
2.1. Operation	4
2.2. State	5
2.3. Who	5
2.4. Dates and Times	6
3. EPP Command Mapping	6
3.1. EPP Query Commands	6
3.1.1. EPP <check> Command	6
3.1.2. EPP <info> Command	6
3.1.3. EPP <transfer> Command	16
3.2. EPP Transform Commands	16
3.2.1. EPP <create> Command	16
3.2.2. EPP <delete> Command	16
3.2.3. EPP <renew> Command	16
3.2.4. EPP <transfer> Command	16
3.2.5. EPP <update> Command	16
4. Formal Syntax	16
4.1. Change Poll Extension Schema	17
5. IANA Considerations	19
5.1. XML Namespace	19
5.2. EPP Extension Registry	20
6. Implementation Status	20
6.1. Verisign EPP SDK	21
6.2. Verisign Consolidated Top Level Domain (CTLD) SRS	21
6.3. Verisign .COM / .NET SRS	22
6.4. Neustar EPP SDK	22
7. Security Considerations	22
8. Acknowledgements	22
9. References	23
9.1. Normative References	23
9.2. Informative References	24
Appendix A. Change History	24
A.1. Change from 00 to 01	24
A.2. Change from 01 to 02	24

A.3.	Change from 02 to 03	24
A.4.	Change from 03 to 04	24
A.5.	Change from 04 to 05	24
A.6.	Change from 05 to REGEXT 00	24
A.7.	Change from REGEXT 00 to REGEXT 01	24
A.8.	Change from REGEXT 01 to REGEXT 02	25
A.9.	Change from REGEXT 02 to REGEXT 03	25
A.10.	Change from REGEXT 03 to REGEXT 04	25
A.11.	Change from REGEXT 04 to REGEXT 05	25
A.12.	Change from REGEXT 05 to REGEXT 06	25
A.13.	Change from REGEXT 06 to REGEXT 07	25
A.14.	Change from REGEXT 07 to REGEXT 08	26
A.15.	Change from REGEXT 08 to REGEXT 09	26
A.16.	Change from REGEXT 09 to REGEXT 10	26
A.17.	Change from REGEXT 10 to REGEXT 11	27
A.18.	Change from REGEXT 11 to REGEXT 12	27
Authors' Addresses		27

1. Introduction

This document describes an extension mapping for version 1.0 of the Extensible Provisioning Protocol (EPP) [RFC5730]. This mapping, an extension to EPP object mappings like the EPP domain name mapping [RFC5731], is used to notify clients of operations they are not directly involved in, on objects that the client sponsors. It is up to server policy to determine what transform operations and clients to notify. Using this extension, clients can more easily keep their systems in-sync with the objects stored in the server. When a change occurs that a client needs to be notified of, a poll message can be inserted by the server for consumption by the client using the EPP <poll> command and response defined in [RFC5730]. The extension supports including a "before" operation poll message and an "after" operation poll message. The extension only extends the EPP <poll> response in [RFC5730] and does not extend the EPP <poll> command. Please refer to [RFC5730] for information and examples of the EPP <poll> command.

1.1. Conventions Used in This Document

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

XML is case sensitive. Unless stated otherwise, XML specifications and examples provided in this document MUST be interpreted in the

character case presented in order to develop a conforming implementation.

In examples, "C:" represents lines sent by a protocol client and "S:" represents lines returned by a protocol server. Indentation and white space in examples are provided only to illustrate element relationships and are not a REQUIRED feature of this protocol.

The XML namespace prefix "changePoll" is used for the namespace "urn:ietf:params:xml:ns:changePoll-1.0", but implementations MUST NOT depend on it and instead employ a proper namespace-aware XML parser and serializer to interpret and output the XML documents.

2. Object Attributes

This extension adds additional elements to EPP object mappings like the EPP domain name mapping [RFC5731]. Only those new elements are described here.

2.1. Operation

An operation consists of any transform operation that impacts objects that the client sponsors and should be notified of. The <changePoll:operation> element defines the operation. The OPTIONAL "op" attribute is an identifier, represented in the 7-bit US-ASCII character set defined in [RFC0020], that is used to define a sub-operation or the name of a "custom" operation. The enumerated list of <changePoll:operation> values is:

- "create": Create operation as defined in [RFC5730].
- "delete": Delete operation as defined in [RFC5730]. If the delete operation results in an immediate purge of the object, then the "op" attribute MUST be set to "purge".
- "renew": Renew operation as defined in [RFC5730].
- "transfer": Transfer operation as defined in [RFC5730] that MUST set the "op" attribute with one of the possible transfer type values that include "request", "approve", "cancel", or "reject".
- "update": Update operation as defined in [RFC5730].
- "restore": Restore operation as defined in [RFC3915] that MUST set the "op" attribute with one of the possible restore type values that include "request" or "report".
- "autoRenew": Auto renew operation executed by the server.
- "autoDelete": Auto delete operation executed by the server. If the "autoDelete" operation results in an immediate purge of the object, then the "op" attribute MUST be set to "purge".
- "autoPurge": Auto purge operation executed by the server when removing the object after it had the "pendingDelete" status.

"custom": Custom operation that MUST set the "op" attribute with the custom operation name. The custom operations supported is up to server policy.

2.2. State

The state attribute reflects the state of the object "before" or "after" the operation. The state is defined using the OPTIONAL "state" attribute of the <changePoll:changeData> element, with the possible values "before" or "after" and with a default value of "after". The server MAY support both the "before" state and the "after" state of the operation, by using one poll message for the "before" state and one poll message for the "after" state. The "before" state poll message MUST be inserted into the message queue prior to the "after" state poll message.

For operations in Section 2.1 that don't have an "after" state, the server MUST use the "before" state poll message. For example, for the "delete" operation with the "op" attribute set to "purge", or the "autoPurge" operation, the server includes the state of the object prior to being purged in the "before" state poll message.

For operations in Section 2.1 that don't have a "before" state, the server MUST use the "after" state poll message. For example, for the "create" operation, the server includes the state of the object after creation in the "after" state poll message.

2.3. Who

The <changePoll:who> element defines who executed the operation for audit purposes. It is a freeform value that is strictly meant for audit purposes and not meant to drive client-side logic. The scheme used for the possible set of <changePoll:who> element values is up to server policy. The server MAY identify the <changePoll:who> element value based on:

"Identifier": Unique user identifier of the user that executed the operation. An example is "ClientX".

"Name": Name of the user that executed the operation. An example is "John Doe".

"Role": Role of the user that executed operation. An example is "CSR" for a Customer Support Representative or "Batch" for a server batch.

2.4. Dates and Times

Date and time attribute values MUST be represented in Universal Coordinated Time (UTC) using the Gregorian calendar. The extended date-time form using upper case "T" and "Z" characters defined in [W3C.REC-xmlschema-2-20041028] MUST be used to represent date-time values, as XML Schema does not support truncated date-time forms or lower case "T" and "Z" characters.

3. EPP Command Mapping

A detailed description of the EPP syntax and semantics can be found in the EPP core protocol specification [RFC5730].

3.1. EPP Query Commands

EPP provides three commands to retrieve object information: <check> to determine if an object is known to the server, <info> to retrieve detailed information associated with an object, and <transfer> to retrieve object transfer status information.

3.1.1. EPP <check> Command

This extension does not add any elements to the EPP <check> command or <check> response described in the [RFC5730].

3.1.2. EPP <info> Command

This extension does not add any elements to the EPP <info> command described in the [RFC5730].

This extension adds operation detail of EPP object mapping operations Section 2.1 to an EPP poll response, as described in [RFC5730]. The extension is an extension of the EPP object mapping info response. Any transform operation to an object defined in an EPP object mapping by a client other than the sponsoring client MAY result in extending the <info> response of the object for inserting an EPP poll message with the operation detail. The sponsoring client will then receive the state of the object with operation detail like what, who, when, and why the object was changed. The <changePoll:changeData> element contains the operation detail along with an indication of whether the object reflects the state before or after the operation as defined in Section 2.2. The <changePoll:changeData> element includes the operation detail with the following child elements:

<changePoll:operation>: Transform operation executed on the object as defined in Section 2.1.

<changePoll:date>: Date and time when the operation was executed.

<changePoll:svTRID>: Server transaction identifier of the operation.
<changePoll:who>: Who executed the operation as defined in
Section 2.3.

<changePoll:caseId>: OPTIONAL case identifier associated with the operation. The required "type" attribute defines the type of case. The OPTIONAL "name" attribute is an identifier, represented in the 7-bit US-ASCII character set defined in [RFC0020], that is used to define the name of the "custom" case type. The enumerated list of case types is:

udrp: a Uniform Domain-Name Dispute-Resolution Policy (UDRP) case.

urs: a Uniform Rapid Suspension (URS) case.

custom: A custom case that is defined using the "name" attribute.

<changePoll:reason>: OPTIONAL reason for executing the operation. If present, this element contains the server-specific text to help explain the reason the operation was executed. This text MUST be represented in the response language previously negotiated with the client; an OPTIONAL "lang" attribute MAY be present to identify the language if the negotiated value is something other than the default value of "en" (English).

Example poll <info> response with the <changePoll:changeData> extension for a URS lock transaction on the domain.example domain name, with the "before" state. The "before" state is reflected in the <resData> block:

```
S:<?xml version="1.0" encoding="UTF-8"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S:  <response>
S:    <result code="1301">
S:      <msg lang="en-US">
S:        Command completed successfully; ack to dequeue</msg>
S:      </result>
S:    <msgQ id="201" count="1">
S:      <qDate>2013-10-22T14:25:57.0Z</qDate>
S:      <msg>Registry initiated update of domain.</msg>
S:    </msgQ>
S:  <resData>
S:    <domain:infData
S:      xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
S:      <domain:name>domain.example</domain:name>
S:      <domain:roid>EXAMPLE1-REP</domain:roid>
S:      <domain:status s="ok"/>
S:      <domain:registrant>jd1234</domain:registrant>
S:      <domain:contact type="admin">sh8013</domain:contact>
S:      <domain:contact type="tech">sh8013</domain:contact>
S:      <domain:clID>ClientX</domain:clID>
S:      <domain:crID>ClientY</domain:crID>
S:      <domain:crDate>2012-04-03T22:00:00.0Z</domain:crDate>
S:      <domain:exDate>2014-04-03T22:00:00.0Z</domain:exDate>
S:    </domain:infData>
S:  </resData>
S:  <extension>
S:    <changePoll:changeData
S:      xmlns:changePoll="urn:ietf:params:xml:ns:changePoll-1.0"
S:      state="before">
S:      <changePoll:operation>update</changePoll:operation>
S:      <changePoll:date>2013-10-22T14:25:57.0Z</changePoll:date>
S:      <changePoll:svTRID>12345-XYZ</changePoll:svTRID>
S:      <changePoll:who>URS Admin</changePoll:who>
S:      <changePoll:caseId type="urs">urs123</changePoll:caseId>
S:      <changePoll:reason>URS Lock</changePoll:reason>
S:    </changePoll:changeData>
S:  </extension>
S:  <trID>
S:    <clTRID>ABC-12345</clTRID>
S:    <svTRID>54321-XYZ</svTRID>
S:  </trID>
S: </response>
S:</epp>
```

Example poll <info> response with the <changePoll:changeData> extension for a URS lock transaction on the domain.example domain name, with the "after" state. The "after" state is reflected in the

<resData> block:

```
S:<?xml version="1.0" encoding="UTF-8"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S:  <response>
S:    <result code="1301">
S:      <msg lang="en-US">
S:        Command completed successfully; ack to dequeue</msg>
S:      </result>
S:    <msgQ id="202" count="1">
S:      <qDate>2013-10-22T14:25:57.0Z</qDate>
S:      <msg>Registry initiated update of domain.</msg>
S:    </msgQ>
S:  <resData>
S:    <domain:infData
S:      xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
S:      <domain:name>domain.example</domain:name>
S:      <domain:roid>EXAMPLE1-REP</domain:roid>
S:      <domain:status s="serverUpdateProhibited"/>
S:      <domain:status s="serverDeleteProhibited"/>
S:      <domain:status s="serverTransferProhibited"/>
S:      <domain:registrant>jdl234</domain:registrant>
S:      <domain:contact type="admin">sh8013</domain:contact>
S:      <domain:contact type="tech">sh8013</domain:contact>
S:      <domain:clID>ClientX</domain:clID>
S:      <domain:crID>ClientY</domain:crID>
S:      <domain:crDate>2012-04-03T22:00:00.0Z</domain:crDate>
S:      <domain:upID>ClientZ</domain:upID>
S:      <domain:upDate>2013-10-22T14:25:57.0Z</domain:upDate>
S:      <domain:exDate>2014-04-03T22:00:00.0Z</domain:exDate>
S:    </domain:infData>
S:  </resData>
S:  <extension>
S:    <changePoll:changeData
S:      xmlns:changePoll="urn:ietf:params:xml:ns:changePoll-1.0"
S:      state="after">
S:      <changePoll:operation>update</changePoll:operation>
S:      <changePoll:date>2013-10-22T14:25:57.0Z</changePoll:date>
S:      <changePoll:svTRID>12345-XYZ</changePoll:svTRID>
S:      <changePoll:who>URS Admin</changePoll:who>
S:      <changePoll:caseId type="urs">urs123</changePoll:caseId>
S:      <changePoll:reason>URS Lock</changePoll:reason>
S:    </changePoll:changeData>
S:  </extension>
S:  <trID>
S:    <clTRID>ABC-12345</clTRID>
S:    <svTRID>54321-XYZ</svTRID>
S:  </trID>
S: </response>
S:</epp>
```

Example poll <info> response with the <changePoll:changeData> extension for a custom "sync" operation on the domain.example domain name, with the default "after" state. The "after" state is reflected in the <resData> block:

```
S:<?xml version="1.0" encoding="UTF-8"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S:  <response>
S:    <result code="1301">
S:      <msg>Command completed successfully; ack to dequeue</msg>
S:    </result>
S:    <msgQ id="201" count="1">
S:      <qDate>2013-10-22T14:25:57.0Z</qDate>
S:    <msg>Registry initiated Sync of Domain Expiration Date</msg>
S:    </msgQ>
S:    <resData>
S:      <domain:infData
S:        xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
S:        <domain:name>domain.example</domain:name>
S:        <domain:roid>EXAMPLE1-REP</domain:roid>
S:        <domain:status s="ok"/>
S:        <domain:registrant>jd1234</domain:registrant>
S:        <domain:contact type="admin">sh8013</domain:contact>
S:        <domain:contact type="tech">sh8013</domain:contact>
S:        <domain:clID>ClientX</domain:clID>
S:        <domain:crID>ClientY</domain:crID>
S:        <domain:crDate>2012-04-03T22:00:00.0Z</domain:crDate>
S:        <domain:upID>ClientZ</domain:upID>
S:        <domain:upDate>2013-10-22T14:25:57.0Z</domain:upDate>
S:        <domain:exDate>2014-04-03T22:00:00.0Z</domain:exDate>
S:      </domain:infData>
S:    </resData>
S:    <extension>
S:      <changePoll:changeData
S:        xmlns:changePoll="urn:ietf:params:xml:ns:changePoll-1.0">
S:        <changePoll:operation op="sync">custom
S:      </changePoll:operation>
S:      <changePoll:date>2013-10-22T14:25:57.0Z</changePoll:date>
S:      <changePoll:svTRID>12345-XYZ</changePoll:svTRID>
S:      <changePoll:who>CSR</changePoll:who>
S:      <changePoll:reason lang="en">Customer sync request
S:    </changePoll:reason>
S:  </changePoll:changeData>
S:    </extension>
S:  <trID>
S:    <clTRID>ABC-12345</clTRID>
S:    <svTRID>54321-XYZ</svTRID>
S:  </trID>
S:    </response>
S:</epp>
```

Example poll <info> response with the <changePoll:changeData> extension for a "delete" operation on the domain.example domain name that is immediately purged, with the "before" state. The "before" state is reflected in the <resData> block:

```
S:<?xml version="1.0" encoding="UTF-8"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S:  <response>
S:    <result code="1301">
S:      <msg>Command completed successfully; ack to dequeue</msg>
S:    </result>
S:    <msgQ id="200" count="1">
S:      <qDate>2013-10-22T14:25:57.0Z</qDate>
S:      <msg>Registry initiated delete of
S:        domain resulting in immediate purge.</msg>
S:    </msgQ>
S:    <resData>
S:      <domain:infData
S:        xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
S:        <domain:name>domain.example</domain:name>
S:        <domain:roid>EXAMPLE1-REP</domain:roid>
S:        <domain:clID>ClientX</domain:clID>
S:      </domain:infData>
S:    </resData>
S:    <extension>
S:      <changePoll:changeData
S:        xmlns:changePoll="urn:ietf:params:xml:ns:changePoll-1.0"
S:        state="before">
S:        <changePoll:operation op="purge">delete
S:        </changePoll:operation>
S:        <changePoll:date>2013-10-22T14:25:57.0Z
S:        </changePoll:date>
S:        <changePoll:svTRID>12345-XYZ
S:        </changePoll:svTRID>
S:        <changePoll:who>ClientZ
S:        </changePoll:who>
S:        <changePoll:reason>Court order
S:        </changePoll:reason>
S:      </changePoll:changeData>
S:    </extension>
S:    <trID>
S:      <clTRID>ABC-12345</clTRID>
S:      <svTRID>54321-XYZ</svTRID>
S:    </trID>
S:  </response>
S:</epp>
```

Example poll <info> response with the <changePoll:changeData> extension for an "autoPurge" operation on the domain.example domain name that previously had the "pendingDelete" status, with the "before" state. The "before" state is reflected in the <resData> block:

```
S:<?xml version="1.0" encoding="UTF-8"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S:  <response>
S:    <result code="1301">
S:      <msg>Command completed successfully; ack to dequeue
S:    </msg>
S:    </result>
S:    <msgQ id="200" count="1">
S:      <qDate>2013-10-22T14:25:57.0Z</qDate>
S:      <msg>Registry purged domain with pendingDelete status.
S:    </msg>
S:    </msgQ>
S:    <resData>
S:      <domain:infData
S:        xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
S:        <domain:name>domain.example</domain:name>
S:        <domain:roid>EXAMPLE1-REP</domain:roid>
S:        <domain:clID>ClientX</domain:clID>
S:      </domain:infData>
S:    </resData>
S:    <extension>
S:      <changePoll:changeData
S:        xmlns:changePoll="urn:ietf:params:xml:ns:changePoll-1.0"
S:        state="before">
S:        <changePoll:operation>autoPurge
S:      </changePoll:operation>
S:      <changePoll:date>2013-10-22T14:25:57.0Z
S:    </changePoll:date>
S:    <changePoll:svTRID>12345-XYZ
S:  </changePoll:svTRID>
S:    <changePoll:who>Batch
S:  </changePoll:who>
S:    <changePoll:reason>Past pendingDelete 5 day period
S:  </changePoll:reason>
S:    </changePoll:changeData>
S:  </extension>
S:  <trID>
S:    <clTRID>ABC-12345</clTRID>
S:    <svTRID>54321-XYZ</svTRID>
S:  </trID>
S: </response>
S:</epp>
```


Example poll <info> response with the <changePoll:changeData> extension for an "update" operation on the ns1.domain.example host, with the default "after" state. The "after" state is reflected in the <resData> block:

```
S:<?xml version="1.0" encoding="UTF-8"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S:  <response>
S:    <result code="1301">
S:      <msg>Command completed successfully; ack to dequeue</msg>
S:    </result>
S:    <msgQ id="201" count="1">
S:      <qDate>2013-10-22T14:25:57.0Z</qDate>
S:      <msg>Registry initiated update of host.</msg>
S:    </msgQ>
S:    <resData>
S:      <host:infData
S:        xmlns:host="urn:ietf:params:xml:ns:host-1.0">
S:        <host:name>ns1.domain.example</host:name>
S:        <host:roid>NS1_EXAMPLE1-REP</host:roid>
S:        <host:status s="linked"/>
S:        <host:status s="serverUpdateProhibited"/>
S:        <host:status s="serverDeleteProhibited"/>
S:        <host:addr ip="v4">192.0.2.2</host:addr>
S:        <host:addr ip="v6">2001:db8:0:0:1:0:0:1</host:addr>
S:        <host:clID>ClientX</host:clID>
S:        <host:crID>ClientY</host:crID>
S:        <host:crDate>2012-04-03T22:00:00.0Z</host:crDate>
S:        <host:upID>ClientY</host:upID>
S:        <host:upDate>2013-10-22T14:25:57.0Z</host:upDate>
S:      </host:infData>
S:    </resData>
S:    <extension>
S:      <changePoll:changeData
S:        xmlns:changePoll="urn:ietf:params:xml:ns:changePoll-1.0">
S:        <changePoll:operation>update</changePoll:operation>
S:        <changePoll:date>2013-10-22T14:25:57.0Z</changePoll:date>
S:        <changePoll:svTRID>12345-XYZ</changePoll:svTRID>
S:        <changePoll:who>ClientZ</changePoll:who>
S:        <changePoll:reason>Host Lock</changePoll:reason>
S:      </changePoll:changeData>
S:    </extension>
S:    <trID>
S:      <clTRID>ABC-12345</clTRID>
S:      <svTRID>54321-XYZ</svTRID>
S:    </trID>
S:  </response>
S:</epp>
```

3.1.3. EPP <transfer> Command

This extension does not add any elements to the EPP <transfer> query command or <transfer> response described in the [RFC5730].

3.2. EPP Transform Commands

EPP provides five commands to transform objects: <create> to create an instance of an object, <delete> to delete an instance of an object, <renew> to extend the validity period of an object, <transfer> to manage object sponsorship changes, and <update> to change information associated with an object.

3.2.1. EPP <create> Command

This extension does not add any elements to the EPP <create> command or <create> response described in the [RFC5730].

3.2.2. EPP <delete> Command

This extension does not add any elements to the EPP <delete> command or <delete> response described in the [RFC5730].

3.2.3. EPP <renew> Command

This extension does not add any elements to the EPP <renew> command or <renew> response described in the [RFC5730].

3.2.4. EPP <transfer> Command

This extension does not add any elements to the EPP <transfer> command or <transfer> response described in the [RFC5730].

3.2.5. EPP <update> Command

This extension does not add any elements to the EPP <update> command or <update> response described in the [RFC5730].

4. Formal Syntax

One schema is presented here that is the EPP Change Poll Extension schema.

The formal syntax presented here is a complete schema representation of the object mapping suitable for automated validation of EPP XML instances. The BEGIN and END tags are not part of the schema; they are used to note the beginning and ending of the schema for URI registration purposes.

4.1. Change Poll Extension Schema

```
BEGIN
<?xml version="1.0" encoding="UTF-8"?>
  <schema targetNamespace="urn:ietf:params:xml:ns:changePoll-1.0"
    xmlns:eppcom="urn:ietf:params:xml:ns:eppcom-1.0"
    xmlns:epp="urn:ietf:params:xml:ns:epp-1.0"
    xmlns:changePoll="urn:ietf:params:xml:ns:changePoll-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"/>
    <import namespace="urn:ietf:params:xml:ns:epp-1.0"/>

    <annotation>
      <documentation>
        Extensible Provisioning Protocol v1.0
        Change Poll Mapping Schema.
      </documentation>
    </annotation>

    <!--
    Change element.
    -->
    <element name="changeData" type="changePoll:changeDataType"/>

    <!--
    Attributes associated with the change.
    -->
    <complexType name="changeDataType">
      <sequence>
        <element name="operation" type="changePoll:operationType"/>
        <element name="date" type="dateTime"/>
        <element name="svTRID" type="epp:trIDStringType"/>
        <element name="who" type="changePoll:whoType"/>
        <element name="caseId" type="changePoll:caseIdType"
          minOccurs="0"/>
        <element name="reason" type="eppcom:reasonType"
          minOccurs="0"/>
      </sequence>
      <attribute name="state" type="changePoll:stateType"
        default="after"/>
    </complexType>
```

```
<!--
  Enumerated list of operations, with extensibility via "custom".
-->
<simpleType name="operationEnum">
  <restriction base="token">
    <enumeration value="create"/>
    <enumeration value="delete"/>
    <enumeration value="renew"/>
    <enumeration value="transfer"/>
    <enumeration value="update"/>
    <enumeration value="restore"/>
    <enumeration value="autoRenew"/>
    <enumeration value="autoDelete"/>
    <enumeration value="autoPurge"/>
    <enumeration value="custom"/>
  </restriction>
</simpleType>

<!--
  Enumerated of state of the object in the poll message.
-->
<simpleType name="stateType">
  <restriction base="token">
    <enumeration value="before"/>
    <enumeration value="after"/>
  </restriction>
</simpleType>

<!--
  Transform operation type
-->
<complexType name="operationType">
  <simpleContent>
    <extension base="changePoll:operationEnum">
      <attribute name="op" type="token"/>
    </extension>
  </simpleContent>
</complexType>

<!--
  Case identifier type
-->
<complexType name="caseIdType">
  <simpleContent>
    <extension base="token">
      <attribute name="type" type="changePoll:caseTypeEnum"
        use="required"/>
      <attribute name="name" type="token">
```

```
        use="optional"/>
      </extension>
    </simpleContent>
  </complexType>

  <!--
    Enumerated list of case identifier types
  -->
  <simpleType name="caseTypeEnum">
    <restriction base="token">
      <enumeration value="udrp"/>
      <enumeration value="urs"/>
      <enumeration value="custom"/>
    </restriction>
  </simpleType>

  <!--
    Who type
  -->
  <simpleType name="whoType">
    <restriction base="normalizedString">
      <minLength value="1"/>
      <maxLength value="255"/>
    </restriction>
  </simpleType>

  <!--
    End of schema.
  -->
</schema>
END
```

5. IANA Considerations

5.1. XML Namespace

This document uses URNs to describe XML namespaces and XML schemas conforming to a registry mechanism described in [RFC3688]. The following URI assignment is requested of IANA:

Registration request for the changePoll namespace:

URI: urn:ietf:params:xml:ns:changePoll-1.0
Registrant Contact: IESG
XML: None. Namespace URIs do not represent an XML specification.

Registration request for the changePoll XML schema:

URI: urn:ietf:params:xml:ns:changePoll-1.0
Registrant Contact: IESG
XML: See the "Formal Syntax" section of this document.

5.2. EPP Extension Registry

The EPP extension described in this document should be registered by the IANA in the EPP Extension Registry described in [RFC7451]. The details of the registration are as follows:

Name of Extension: "Change Poll Extension for the Extensible Provisioning Protocol (EPP)"

Document status: Standards Track

Reference: (insert reference to RFC version of this document)

Registrant Name and Email Address: IESG, <iesg@ietf.org>

TLDs: Any

IPR Disclosure: None

Status: Active

Notes: None

6. Implementation Status

Note to RFC Editor: Please remove this section and the reference to RFC 7942 [RFC7942] before publication.

This section records the status of known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft, and is based on a proposal described in RFC 7942 [RFC7942]. The description of implementations in this section is intended to assist the IETF in its decision processes in progressing drafts to RFCs. Please note that the listing of any individual implementation here does not imply endorsement by the IETF. Furthermore, no effort has been spent to verify the information presented here that was supplied by IETF contributors. This is not intended as, and must not be construed to be, a catalog of available implementations or their features. Readers are advised to note that other implementations may exist.

According to RFC 7942 [RFC7942], "this will allow reviewers and working groups to assign due consideration to documents that have the benefit of running code, which may serve as evidence of valuable

experimentation and feedback that have made the implemented protocols more mature. It is up to the individual working groups to use this information as they see fit".

6.1. Verisign EPP SDK

Organization: Verisign Inc.

Name: Verisign EPP SDK

Description: The Verisign EPP SDK includes both a full client implementation and a full server stub implementation of draft-ietf-regext-change-poll.

Level of maturity: Production

Coverage: All aspects of the protocol are implemented.

Licensing: GNU Lesser General Public License

Contact: jgould@verisign.com

URL: https://www.verisign.com/en_US/channel-resources/domain-registry-products/epp-sdks

6.2. Verisign Consolidated Top Level Domain (CTLD) SRS

Organization: Verisign Inc.

Name: Verisign Consolidated Top Level Domain (CTLD) Shared Registry System (SRS)

Description: The Verisign Consolidated Top Level Domain (CTLD) Shared Registry System (SRS) implements the server-side of draft-ietf-regext-change-poll for a variety of Top Level Domains (TLD's).

Level of maturity: Production

Coverage: The "after" state poll message for an "update" transform operation of a domain name due to server policy.

Licensing: Proprietary

Contact: jgould@verisign.com

6.3. Verisign .COM / .NET SRS

Organization: Verisign Inc.

Name: Verisign .COM / .NET Shared Registry System (SRS)

Description: The Verisign Shared Registry System (SRS) for .COM and .NET implements the server-side of draft-ietf-regext-change-poll.

Level of maturity: Production

Coverage: The "after" state poll message for an "update" transform operation of a domain name due to server policy.

Licensing: Proprietary

Contact: jgould@verisign.com

6.4. Neustar EPP SDK

Organisation: Neustar Inc.

Name: Neustar EPP SDK

Description: The Neustar EPP SDK includes a full client implementation of draft-ietf-regext-change-poll.

Level of maturity: Production

Coverage: All client side aspects of the protocol are implemented.

Licensing: GNU Lesser General Public License

Contact: quoc-anh.np@team.neustar

7. Security Considerations

The mapping extensions described in this document do not provide any security services beyond those described by EPP [RFC5730] and protocol layers used by EPP. The security considerations described in these other specifications apply to this specification as well.

8. Acknowledgements

The authors wish to acknowledge the original concept for this draft and the efforts in the initial versions of this draft by Trung Tran and Sharon Wodjenski.

Special suggestions that have been incorporated into this document were provided by Scott Hollenbeck, Michael Holloway, and Patrick Mevzek.

9. References

9.1. Normative References

- [RFC0020] Cerf, V., "ASCII format for network interchange", STD 80, RFC 20, DOI 10.17487/RFC0020, October 1969, <<https://www.rfc-editor.org/info/rfc20>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC3915] Hollenbeck, S., "Domain Registry Grace Period Mapping for the Extensible Provisioning Protocol (EPP)", RFC 3915, DOI 10.17487/RFC3915, September 2004, <<https://www.rfc-editor.org/info/rfc3915>>.
- [RFC5730] Hollenbeck, S., "Extensible Provisioning Protocol (EPP)", STD 69, RFC 5730, DOI 10.17487/RFC5730, August 2009, <<https://www.rfc-editor.org/info/rfc5730>>.
- [RFC5731] Hollenbeck, S., "Extensible Provisioning Protocol (EPP) Domain Name Mapping", STD 69, RFC 5731, DOI 10.17487/RFC5731, August 2009, <<https://www.rfc-editor.org/info/rfc5731>>.
- [RFC7942] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", BCP 205, RFC 7942, DOI 10.17487/RFC7942, July 2016, <<https://www.rfc-editor.org/info/rfc7942>>.
- [W3C.REC-xmlschema-2-20041028] Biron, P. and A. Malhotra, "XML Schema Part 2: Datatypes Second Edition", World Wide Web Consortium Recommendation REC-xmlschema-2-20041028, October 2004, <<http://www.w3.org/TR/2004/REC-xmlschema-2-20041028>>.

9.2. Informative References

- [RFC7451] Hollenbeck, S., "Extension Registry for the Extensible Provisioning Protocol", RFC 7451, DOI 10.17487/RFC7451, February 2015, <<https://www.rfc-editor.org/info/rfc7451>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

Appendix A. Change History

A.1. Change from 00 to 01

1. Added an optional caseId element that defines the case identifier from UDRP, URS, or custom case, based on feedback from Michael Holloway.

A.2. Change from 01 to 02

1. Amended XML Namespace section of IANA Considerations, added EPP Extension Registry section.
2. Moved Change History to the back section as an Appendix.

A.3. Change from 02 to 03

1. Fixed "before" state example to use the "before" state value based on feedback from Patrick Mevzek.

A.4. Change from 03 to 04

1. Updated the authors for the draft.

A.5. Change from 04 to 05

1. Ping update.

A.6. Change from 05 to REGEXT 00

1. Changed to regext working group draft by changing draft-gould-change-poll to draft-ietf-regext-change-poll.

A.7. Change from REGEXT 00 to REGEXT 01

1. Ping update.

A.8. Change from REGEXT 01 to REGEXT 02

1. Added the Implementation Status section.

A.9. Change from REGEXT 02 to REGEXT 03

1. Changed Neustar author to Kal Feher.

A.10. Change from REGEXT 03 to REGEXT 04

1. Added Neustar implementation to the Implementation Status section.

A.11. Change from REGEXT 04 to REGEXT 05

1. Updates based on feedback from Patrick Mevzek, that include:
 1. Added a missing comma to "Using this extension, clients" in the Introduction section.
 2. Modified the description of the "transfer", "restore", and "custom" operations to include "MUST set the "op" attribute" language.
 3. Rephrased the first sentence of the Who section.
 4. Added references to the <changePoll:who> element in the Who section.
 5. Revise the sentence that describes how the extension extends the info response in the EPP <info> Command section.
 6. Refer to EPP Object Mapping as EPP object mapping throughout the document.
 7. Add a Dates and Times section to the Object Attributes section.

A.12. Change from REGEXT 05 to REGEXT 06

1. Added the "State" sub-section to the "Object Attributes" section to describe the expected behavior for the "before" and "after" states, based on feedback from Patrick Mevzek.
2. Added a colon suffix to each hangText entry to provide better separation.

A.13. Change from REGEXT 06 to REGEXT 07

1. Updates based on feedback from Scott Hollenbeck, that include:
 1. Changed MAY to may in the Abstract.
 2. Revised the "IANA Considerations" section to include the registration of the XML schema.

3. Revised the description of the <changePoll:caseId> "name" attribute and the "changePoll:operation" "op" attribute as containing 7-bit US-ASCII identifiers for the case type or the operation type, respectively.

A.14. Change from REGEXT 07 to REGEXT 08

1. Updated obsoleted RFC 6982 to RFC 7942.
2. Moved RFC 7451 to an informational reference based on a check done by the Idnits Tool.
3. Changed Kal Feher's contact e-mail address.
4. Changed Neustar's Implementation Status contact e-mail address.

A.15. Change from REGEXT 08 to REGEXT 09

1. Fixed Section 1.1 (Conventions) to contain the updated language (e.g. "NOT RECOMMENDED", RFC 8174, BCP 14), based on feedback from the Document Shepherd.

A.16. Change from REGEXT 09 to REGEXT 10

1. Updates based on the AD review by Adam Roach, that include:
 1. Fix the "purge" and "autoPurge" examples to use the normative "before" state instead of the default "after" state.
 2. Added the sentences "The extension only extends the EPP <poll> response in [RFC5730] and does not extend the EPP <poll> command. Please refer to [RFC5730] for information and examples of the EPP <poll> command." in the "Introduction" to clarify what is extended and reference [RFC5730] for the EPP <poll> command.
 3. Added missing hyphens to "client-sponsored" and "court-directed".
 4. Removed "changePoll-1.0" is used as an abbreviation for "urn:ietf:params:xml:ns:changePoll-1.0" and replaced the paragraph based on what was done in draft-ietf-regext-allocation-token.
 5. Changed normative "SHOULD" to non-normative "should" in "An operation consists of any transform operation that impacts objects that the client sponsors and should be notified of."
 6. Added normative reference to [RFC0020] to define "7-bit US-ASCII".
 7. Added the sentence "The custom operations supported is up to server policy." to the description of the "custom" operation.

8. Broke up the "This extension adds operation detail..." sentence into two separate sentences to address the "does" and the "is" separately.
9. Removed the commas from "Any transform operation to an object..." sentence.
10. Changed to use an IPv6 address from the documentation-only prefix "2001:DB8::/32" in RFC 3849. The IPv6 address 2001:db8:0:0:1:0:0:1 was used.

A.17. Change from REGEXT 10 to REGEXT 11

1. Updates based on the review by Benjamin Kaduk, that include:
 1. Change references of "The enumerated list ... include:" to "The enumerated list ... is:".
 2. In section 2.2, explicitly state what the message is inserted into, with the change of "... MUST be inserted prior to ..." to "... MUST be inserted into the message queue prior to ...".

A.18. Change from REGEXT 11 to REGEXT 12

1. Added clarification for the <changePoll:who> element based on the feedback from Benjamin Kaduk.

Authors' Addresses

James Gould
VeriSign, Inc.
12061 Bluemont Way
Reston, VA 20190
US

Email: jgould@verisign.com
URI: <http://www.verisign.com>

Kal Feher
Neustar
lvl 8/10 Queens Road
Melbourne, VIC 3004
AU

Email: ietf@feherfamily.org
URI: <http://www.neustar.biz>

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: June 3, 2019

L. Zhou
CNNIC
N. Kong
Consultant
G. Zhou
J. Yao
CNNIC
J. Gould
Verisign, Inc.
November 30, 2018

Extensible Provisioning Protocol (EPP) Organization Mapping
draft-ietf-regext-org-12

Abstract

This document describes an Extensible Provisioning Protocol (EPP) mapping for provisioning and management of organization objects stored in a shared central repository. Specified in Extensible Markup Language (XML), this extended mapping is applied to provide additional features required for the provisioning of organizations.

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 <https://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 June 3, 2019.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must 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	3
2. Conventions Used in This Document	3
3. Object Attributes	3
3.1. Organization Identifier	4
3.2. Organization Roles	4
3.2.1. Role Type	4
3.2.2. Role Status	4
3.2.3. Role Identifier	4
3.3. Contact and Client Identifiers	5
3.4. Organization Status Values	5
3.5. Role Status Values	6
3.6. Parent Identifier	7
3.7. URL	7
3.8. Dates and Times	7
4. EPP Command Mapping	8
4.1. EPP Query Commands	8
4.1.1. EPP <check> Command	8
4.1.2. EPP <info> Command	10
4.1.3. EPP <transfer> Query Command	16
4.2. EPP Transform Commands	16
4.2.1. EPP <create> Command	16
4.2.2. EPP <delete> Command	20
4.2.3. EPP <renew> Command	21
4.2.4. EPP <transfer> Command	21
4.2.5. EPP <update> Command	22
4.3. Offline Review of Requested Actions	26
5. Formal Syntax	28
6. Internationalization Considerations	37
7. IANA Considerations	37
7.1. XML Namespace	37
7.2. EPP Extension Registry	38
7.3. Role Type Values Registry	38
7.3.1. Registration Template	38
7.3.2. Initial Registry Contents	38
8. Implementation Status	39
8.1. Verisign EPP SDK	40
8.2. CNNIC Implementation	40
9. Security Considerations	41
10. Acknowledgment	41

11. References	41
11.1. Normative References	41
11.2. Informative References	42
Appendix A. Change Log	43
Authors' Addresses	46

1. Introduction

There are many entities, such as registrars, resellers, DNS service operators, or privacy proxies involved in the domain registration business. These kind of entities have not been formally defined as having an object in Extensible Provisioning Protocol (EPP). This document provides a way to specify them as "organization" entities.

This document describes an organization object mapping for version 1.0 of the EPP [RFC5730]. This mapping is specified using the XML 1.0 as described in [W3C.REC-xml-20040204] and XML Schema notation as described in [W3C.REC-xmlschema-1-20041028] and [W3C.REC-xmlschema-2-20041028].

2. Conventions Used in This Document

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

In examples, "C:" represents lines sent by a protocol client and "S:" represents lines returned by a protocol server. Indentation and white space in examples are provided only to illustrate element relationships and are not a required feature of this specification.

XML is case sensitive. Unless stated otherwise, XML specifications and examples provided in this document MUST be interpreted in the character case presented.

The XML namespace prefix "org" is used for the namespace "urn:ietf:params:xml:ns:epp:org-1.0", but implementations MUST NOT depend on it and instead employ a proper namespace-aware XML parser and serializer to interpret and output the XML documents.

3. Object Attributes

An EPP organization object has attributes and associated values that can be viewed and modified by the sponsoring client or the server. This section describes each attribute type in detail. The formal syntax for the attribute values described here can be found in the

"Formal Syntax" section of this document and in the appropriate normative references.

3.1. Organization Identifier

All EPP organizations are identified by a server-unique identifier. Organization identifiers are character strings with a specified minimum length, a specified maximum length, and a specified format. Organization identifiers use the "clIDType" client identifier syntax described in [RFC5730]. Its corresponding element is <org:id>.

3.2. Organization Roles

The organization roles are used to represent the relationship an organization could have. Its corresponding element is <org:role>. An organization object MUST always have at least one associated role. Roles can be set only by the client that sponsors an organization object. A client can change the role of an organization object using the EPP <update> command.

3.2.1. Role Type

An organization role MUST have a type field. This may have any of the values listed in Section 7.3. An organization could have multiple roles with different role types. Its corresponding element is <org:type>.

3.2.2. Role Status

A role of an organization object MAY have its own statuses. Its corresponding element is <org:status>. The values of the role status are defined in Section 3.5.

3.2.3. Role Identifier

A role MAY have a third-party-assigned identifier such as the IANA ID for registrars. Its corresponding element is <org:roleID>.

Example of organization role identifier:

```
<org:role>
  <org:type>registrar</org:type>
  <org:status>ok</org:status>
  <org:status>linked</org:status>
  <org:roleID>1362</org:roleID>
</org:role>
```

3.3. Contact and Client Identifiers

All EPP contacts are identified by server-unique identifiers. Contact identifiers are character strings with a specified minimum length, a specified maximum length, and a specified format. Contact identifiers use the "clIDType" client identifier syntax described in [RFC5730].

3.4. Organization Status Values

An organization object MUST always have at least one associated status value. Status values can be set only by the client that sponsors an organization object and by the server on which the object resides. A client can change the status of an organization object using the EPP <update> command. Each status value MAY be accompanied by a string of human-readable text that describes the rationale for the status applied to the object.

A client MUST NOT alter server status values set by the server. A server MAY alter or override status values set by a client, subject to local server policies. The status of an object MAY change as a result of either a client-initiated transform command or an action performed by a server operator.

Status values that can be added or removed by a client are prefixed with "client". Corresponding server status values that can be added or removed by a server are prefixed with "server". The "hold" and "terminated" status values are server-managed when the organization has no parent identifier [Section 3.6] and otherwise MAY be client-managed based on server policy. Other status values that do not begin with either "client" or "server" are server-managed.

Status Value Descriptions:

- o ok: This is the normal status value for an object that has no operations pending or active prohibitions. This value is set and removed by the server as other status values are added or removed.
- o hold: Organization transform commands and new links MUST be rejected.
- o terminated: The organization which has been terminated MUST NOT be linked. Organization transform commands and new links MUST be rejected.
- o linked: The organization object has at least one active association with another object. The "linked" status is not

explicitly set by the client. Servers should provide services to determine existing object associations.

- o `clientLinkProhibited`, `serverLinkProhibited`: Requests to add new links to the organization MUST be rejected.
- o `clientUpdateProhibited`, `serverUpdateProhibited`: Requests to update the object (other than to remove this status) MUST be rejected.
- o `clientDeleteProhibited`, `serverDeleteProhibited`: Requests to delete the object MUST be rejected.
- o `pendingCreate`, `pendingUpdate`, `pendingDelete`: A transform command has been processed for the object, but the action has not been completed by the server. Server operators can delay action completion for a variety of reasons, such as to allow for human review or third-party action. A transform command that is processed, but whose requested action is pending, is noted with response code 1001.

"`pendingCreate`", "`ok`", "`hold`", and "`terminated`" are mutually exclusive statuses. Organization MUST have exactly one of these statuses set.

"`ok`" status MAY only be combined with "`linked`" status.

A client or server MAY combine "`linked`" with either "`clientLinkProhibited`" or "`serverLinkProhibited`" if new links must be prohibited.

"`pendingDelete`" status MUST NOT be combined with either "`clientDeleteProhibited`" or "`serverDeleteProhibited`" status.

The `pendingCreate`, `pendingDelete`, and `pendingUpdate` status values MUST NOT be combined with each other.

If "`clientUpdateProhibited`" or "`serverUpdateProhibited`" is set, the client will not be able to update the object. For "`clientUpdateProhibited`", the client will first need to remove "`clientUpdateProhibited`" prior to attempting to update the object. The server can modify the object at any time.

3.5. Role Status Values

A role SHOULD have at least one associated status value. Valid values include "`ok`", "`linked`", "`clientLinkProhibited`", and "`serverLinkProhibited`".

Status Value Descriptions:

- o ok: This is the normal status value for a role that has no operations pending or active prohibitions. This value is set and removed by the server as other status values are added or removed.
- o linked: The role of an organization object has at least one active association with another object. The "linked" status is not explicitly set by the client. Servers SHOULD provide services to determine existing object associations.
- o clientLinkProhibited, serverLinkProhibited: Requests to add new links to the role MUST be rejected.

3.6. Parent Identifier

There can be more than one layer of organizations, such as a reseller. The parent identifier, as defined with the <org:parentId> element, represents the parent organization identifier in a child organization.

The case of reseller organizations provides an example. The parent identifier is not defined for the top level reseller, namely the registrar of the registry. An N-tier reseller has a parent reseller and at least one child reseller. A reseller customer has a parent reseller and no child resellers.

Loops MUST be prohibited. For example: if organization A has B as its parent identifier, organization B cannot have organization A as its parent identifier. The same is true for larger loops involving three or more organizations.

3.7. URL

The URL represents the organization web home page, as defined with the <org:url> element.

3.8. Dates and Times

Date and time attribute values MUST be represented in Universal Coordinated Time (UTC) using the Gregorian calendar. The extended date-time form using upper case "T" and "Z" characters defined in [W3C.REC-xmlschema-2-20041028] MUST be used to represent date-time values, as XML Schema does not support truncated date-time forms or lower case "t" and "z" characters.

4. EPP Command Mapping

A detailed description of the EPP syntax and semantics can be found in the EPP core protocol specification [RFC5730]. The command mappings described here are specifically for use in provisioning and managing organization information via EPP.

4.1. EPP Query Commands

EPP provides two commands to retrieve organization information: `<check>` to determine if an organization object can be provisioned within a repository, and `<info>` to retrieve detailed information associated with an organization object. This document does not define a mapping for the EPP `<transfer>` command to retrieve organization-object transfer status information.

4.1.1. EPP `<check>` Command

The EPP `<check>` command is used to determine if an object can be provisioned within a repository. It provides a hint that allows a client to anticipate the success or failure of provisioning an object using the `<create>` command, as object-provisioning requirements are ultimately a matter of server policy.

In addition to the standard EPP command elements, the `<check>` command MUST contain an `<org:check>` element. This element or its ancestor element MUST identify the organization namespace "urn:ietf:params:xml:ns:epp:org-1.0". The `<org:check>` element contains the following child elements:

- o One or more `<org:id>` elements that contain the server-unique identifier of the organization objects to be queried.

Example `<check>` command:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <check>
C:      <org:check
C:        xmlns:org="urn:ietf:params:xml:ns:epp:org-1.0">
C:          <org:id>res1523</org:id>
C:          <org:id>re1523</org:id>
C:          <org:id>1523res</org:id>
C:        </org:check>
C:      </check>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

When a `<check>` command has been processed successfully, the EPP `<resData>` element MUST contain a child `<org:chkData>` element. This element or its ancestor element MUST identify the organization namespace "urn:ietf:params:xml:ns:epp:org-1.0". The `<org:chkData>` element contains one or more `<org:cd>` elements that contain the following child elements:

- o An `<org:id>` element that identifies the queried object. This element MUST contain an "avail" attribute whose value indicates object availability (can it be provisioned or not) at the moment the `<check>` command was completed. A value of "1" or "true" means that the object can be provisioned. A value of "0" or "false" means that the object cannot be provisioned.
- o An OPTIONAL `<org:reason>` element that may be provided when an object cannot be provisioned. If present, this element contains server-specific text to help explain why the object cannot be provisioned. This text MUST be represented in the response language previously negotiated with the client; an OPTIONAL "lang" attribute as defined in [RFC5646] may be present to identify the language if the negotiated value is something other than the default value of "en" (English).

Example `<check>` response:

```

S:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S:  <response>
S:    <result code="1000">
S:      <msg lang="en">Command completed successfully</msg>
S:    </result>
S:    <resData>
S:      <org:chkData
S:        xmlns:org="urn:ietf:params:xml:ns:epp:org-1.0">
S:        <org:cd>
S:          <org:id avail="1">res1523</org:id>
S:        </org:cd>
S:        <org:cd>
S:          <org:id avail="0">re1523</org:id>
S:          <org:reason lang="en">In use</org:reason>
S:        </org:cd>
S:        <org:cd>
S:          <org:id avail="1">1523res</org:id>
S:        </org:cd>
S:      </org:chkData>
S:    </resData>
S:    <trID>
S:      <clTRID>ABC-12345</clTRID>
S:      <svTRID>54322-XYZ</svTRID>
S:    </trID>
S:  </response>
S:</epp>

```

An EPP error response MUST be returned if a <check> command cannot be processed for any reason.

4.1.2. EPP <info> Command

The EPP <info> command is used to retrieve information associated with an organization object. In addition to the standard EPP command elements, the <info> command MUST contain a <org:info> element. This element or its ancestor element MUST identify the organization namespace "urn:ietf:params:xml:ns:epp:org-1.0". The <org:info> element contains the following child elements:

- o An <org:id> element that contains the server-unique identifier of the organization object to be queried.

Example <info> command:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <info>
C:      <org:info
C:        xmlns:org="urn:ietf:params:xml:ns:epp:org-1.0">
C:          <org:id>res1523</org:id>
C:        </org:info>
C:      </info>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

When an `<info>` command has been processed successfully, the EPP `<resData>` element MUST contain a child `<org:infData>` element. This element or its ancestor element MUST identify the organization namespace "urn:ietf:params:xml:ns:epp:org-1.0". The `<org:infData>` element contains the following child elements:

- o An `<org:id>` element that contains the server-unique identifier of the organization object, as defined in Section 3.1.
- o An `<org:roid>` element that contains the Repository Object Identifier assigned to the organization object when the object was created.
- o One or more `<org:role>` elements that contain the role type, role statuses and optional role id of the organization.
 - * An `<org:type>` element that contains the type of the organization, as defined in Section 3.2.
 - * One or more `<org:status>` elements that contain the role statuses. The values of the role status are defined in Section 3.5.
 - * An OPTIONAL `<org:roleID>` element that contains a third-party-assigned identifier, such as IANA ID for registrars, as defined in Section 3.2.3.
- o One or more `<org:status>` elements that contain the operational status of the organization, as defined in Section 3.4.
- o An OPTIONAL `<org:parentId>` element that contains the identifier of the parent object, as defined in Section 3.6.
- o Zero to two `<org: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 Unicode in the range U+0020 - U+007E. If a localized form (type="loc") is provided, element content MAY be represented in unrestricted UTF-8. The <org:postalInfo> element contains the following child elements:

- * An <org:name> element that contains the name of the organization.
- * An OPTIONAL <org:addr> element that contains address information associated with the organization. A <org:addr> element contains the following child elements:
 - + One, two, or three <org:street> elements that contain the organization's street address.
 - + An <org:city> element that contains the organization's city.
 - + An OPTIONAL <org:sp> element that contains the organization's state or province.
 - + An OPTIONAL <org:pc> element that contains the organization's postal code.
 - + An <org:cc> element that contains the alpha-2 organization's country code. The detailed format of this element is described in section 2.4.3 of [RFC5733].
- o An OPTIONAL <org:voice> element that contains the organization's voice telephone number. The detailed format of this element is described in Section 2.5 of [RFC5733].
- o An OPTIONAL <org:fax> element that contains the organization's facsimile telephone number.
- o An OPTIONAL <org:email> element that contains the organization's email address. The detailed format of this element is described in section 2.6 of [RFC5733].
- o An OPTIONAL <org:url> element that contains the URL to the website of the organization. The detailed format of this element is described in [RFC3986].
- o Zero or more <org:contact> elements that contain identifiers for the contact objects to be associated with the organization object.

Contact object identifiers MUST be known to the server before the contact object can be associated with the organization object. The required "type" is used to represent contact types. The type values include "admin", "tech", "billing", "abuse", and "custom". The OPTIONAL "typeName" attribute is used to define the name of a "custom" type.

- o An OPTIONAL <org:clID> element that contains the organization identifier of the sponsoring client. There is no <org:clID> element if the organization is managed by the registry.
- o An <org:crID> element that contains the identifier of the client that created the organization object.
- o An <org:crDate> element that contains the date and time of organization object creation.
- o An <org:upID> element that contains the identifier of the client that last updated the organization object. This element MUST NOT be present if the organization has never been modified.
- o An <org:upDate> element that contains the date and time of the most recent organization object modification. This element MUST NOT be present if the organization object has never been modified.

Example <info> response for "Example Registrar Inc." organization organization object with identifier "registrar1362":

```
S:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S:  <response>
S:    <result code="1000">
S:      <msg lang="en">Command completed successfully</msg>
S:    </result>
S:    <resData>
S:      <org:infData
S:        xmlns:org="urn:ietf:params:xml:ns:epp:org-1.0">
S:        <org:id>registrar1362</org:id>
S:        <org:roid>registrar1362-REP</org:roid>
S:        <org:role>
S:          <org:type>registrar</org:type>
S:          <org:status>ok</org:status>
S:          <org:status>linked</org:status>
S:          <org:roleID>1362</org:roleID>
S:        </org:role>
S:        <org:status>ok</org:status>
S:        <org:postalInfo type="int">
```

```
S:      <org:name>Example Registrar Inc.</org:name>
S:      <org:addr>
S:          <org:street>123 Example Dr.</org:street>
S:          <org:street>Suite 100</org:street>
S:          <org:city>Dulles</org:city>
S:          <org:sp>VA</org:sp>
S:          <org:pc>20166-6503</org:pc>
S:          <org:cc>US</org:cc>
S:      </org:addr>
S:      </org:postalInfo>
S:      <org:voice x="1234">+1.7035555555</org:voice>
S:      <org:fax>+1.7035555556</org:fax>
S:      <org:email>contact@organization.example</org:email>
S:      <org:url>https://organization.example</org:url>
S:      <org:contact type="admin">sh8013</org:contact>
S:      <org:contact type="billing">sh8013</org:contact>
S:      <org:contact type="custom"
S:          typeName="legal">sh8013</org:contact>
S:      <org:crID>ClientX</org:crID>
S:      <org:crDate>1999-04-03T22:00:00.0Z</org:crDate>
S:      <org:upID>ClientX</org:upID>
S:      <org:upDate>1999-12-03T09:00:00.0Z</org:upDate>
S:      </org:infData>
S:  </resData>
S:  <trID>
S:      <clTRID>ABC-12345</clTRID>
S:      <svTRID>54322-XYZ</svTRID>
S:  </trID>
S: </response>
S:</epp>
```

Example <info> response for "Example Reseller Inc." organization object of reseller type managed by identifier "registrar1362":

```
S:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S:  <response>
S:    <result code="1000">
S:      <msg lang="en">Command completed successfully</msg>
S:    </result>
S:    <resData>
S:      <org:infData
S:        xmlns:org="urn:ietf:params:xml:ns:epp:org-1.0">
S:        <org:id>reseller1523</org:id>
S:        <org:roid>reseller1523-REP</org:roid>
S:        <org:role>
S:          <org:type>reseller</org:type>
S:          <org:status>ok</org:status>
S:          <org:status>linked</org:status>
S:        </org:role>
S:        <org:status>ok</org:status>
S:        <org:parentId>registrar1362</org:parentId>
S:        <org:postalInfo type="int">
S:          <org:name>Example Reseller Inc.</org:name>
S:          <org:addr>
S:            <org:street>123 Example Dr.</org:street>
S:            <org:street>Suite 100</org:street>
S:            <org:city>Dulles</org:city>
S:            <org:sp>VA</org:sp>
S:            <org:pc>20166-6503</org:pc>
S:            <org:cc>US</org:cc>
S:          </org:addr>
S:        </org:postalInfo>
S:        <org:fax>+1.7035555556</org:fax>
S:        <org:url>https://organization.example</org:url>
S:        <org:contact type="admin">sh8013</org:contact>
S:        <org:clID>1362</org:clID>
S:        <org:crID>ClientX</org:crID>
S:        <org:crDate>1999-04-03T22:00:00.0Z</org:crDate>
S:        <org:upID>ClientX</org:upID>
S:        <org:upDate>1999-12-03T09:00:00.0Z</org:upDate>
S:      </org:infData>
S:    </resData>
S:    <trID>
S:      <clTRID>ABC-12345</clTRID>
S:      <svTRID>54322-XYZ</svTRID>
S:    </trID>
S:  </response>
S:</epp>
```

An EPP error response MUST be returned if an <info> command cannot be processed for any reason.

4.1.3. EPP <transfer> Query Command

The transfer semantics does not apply to organization object. No EPP <transfer> query command is defined in this document.

4.2. EPP Transform Commands

This document provides three commands to transform organization object information: <create> to create an instance of an organization object, <delete> to delete an instance of an organization object, and <update> to change information associated with an organization object. This document does not define a mapping for the EPP <transfer> and <renew> command.

Transform commands are typically processed and completed in real time. Server operators MAY receive and process transform commands but defer completing the requested action if human or third-party review is required before the requested action can be completed. In such situations, the server MUST return a 1001 response code to the client to note that the command has been received and processed but that the requested action is pending. The server MUST also manage the status of the object that is the subject of the command to reflect the initiation and completion of the requested action. Once the action has been completed, the client MUST be notified using a service message that the action has been completed and that the status of the object has changed. Other notification methods MAY be used in addition to the required service message.

4.2.1. EPP <create> Command

The EPP <create> command provides a transform operation that allows a client to create an organization object. In addition to the standard EPP command elements, the <create> command MUST contain a <org:create> element. This element or its ancestor element MUST identify the organization namespace "urn:ietf:params:xml:ns:epp:org-1.0". The <org:create> element contains the following child elements:

- o An <org:id> element that contains the desired server-unique identifier for the organization to be created, as defined in Section 3.1.
- o One or more <org:role> elements that contain the role type, role statuses and optional role id of the organization.
- * An <org:type> element that contains the type of the organization, as defined in Section 3.2.

- * Zero or more <org:status> elements that contain the role statuses. The values of the role status are defined in Section 3.5.
- * An OPTIONAL <org:roleID> element that contains a third-party-assigned identifier, such as IANA ID for registrars, as defined in Section 3.2.3.
- o Zero or more <org:status> elements that contain the operational status of the organization, as defined in Section 3.4.
- o An OPTIONAL <org:parentId> element that contains the identifier of the parent object, as defined in Section 3.6.
- o Zero to two <org: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 Unicode in the range U+0020 - U+007E. If a localized form (type="loc") is provided, element content MAY be represented in unrestricted UTF-8. The <org:postalInfo> element contains the following child elements:
 - * An <org:name> element that contains the name of the organization.
 - * An OPTIONAL <org:addr> element that contains address information associated with the organization. A <org:addr> element contains the following child elements:
 - + One, two, or three <org:street> elements that contain the organization's street address.
 - + An <org:city> element that contains the organization's city.
 - + An OPTIONAL <org:sp> element that contains the organization's state or province.
 - + An OPTIONAL <org:pc> element that contains the organization's postal code.
 - + An <org:cc> element that contains the alpha-2 organization's country code. The detailed format of this element is described in section 2.4.3 of [RFC5733].

- o An OPTIONAL <org:voice> element that contains the organization's voice telephone number. The detailed format of this element is described in Section 2.5 of [RFC5733]
- o An OPTIONAL <org:fax> element that contains the organization's facsimile telephone number.
- o An OPTIONAL <org:email> element that contains the organization's email address. The detailed format of this element is described in section 2.6 of [RFC5733].
- o An OPTIONAL <org:url> element that contains the URL to the website of the organization. The detailed format of this element is described in [RFC3986].
- o Zero or more <org:contact> elements that contain identifiers for the contact objects associated with the organization object.

Example <create> command:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <create>
C:      <org:create
C:        xmlns:org="urn:ietf:params:xml:ns:epp:org-1.0">
C:          <org:id>res1523</org:id>
C:          <org:role>
C:            <org:type>reseller</org:type>
C:          </org:role>
C:          <org:parentId>1523res</org:parentId>
C:          <org:postalInfo type="int">
C:            <org:name>Example Organization Inc.</org:name>
C:            <org:addr>
C:              <org:street>123 Example Dr.</org:street>
C:              <org:street>Suite 100</org:street>
C:              <org:city>Dulles</org:city>
C:              <org:sp>VA</org:sp>
C:              <org:pc>20166-6503</org:pc>
C:              <org:cc>US</org:cc>
C:            </org:addr>
C:          </org:postalInfo>
C:          <org:voice x="1234">+1.7035555555</org:voice>
C:          <org:fax>+1.7035555556</org:fax>
C:          <org:email>contact@organization.example</org:email>
C:          <org:url>https://organization.example</org:url>
C:          <org:contact type="admin">sh8013</org:contact>
C:          <org:contact type="billing">sh8013</org:contact>
C:        </org:create>
C:      </create>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

When a <create> command has been processed successfully, the EPP <resData> element MUST contain a child <org:creData> element. This element or its ancestor element MUST identify the organization namespace "urn:ietf:params:xml:ns:epp:org-1.0". The <org:creData> element contains the following child elements:

- o An <org:id> element that contains the server-unique identifier for the created organization, as defined in Section 3.1.
- o An <org:crDate> element that contains the date and time of organization-object creation.

Example <create> response:


```
S:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S:  <response>
S:    <result code="1000">
S:      <msg lang="en">Command completed successfully</msg>
S:    </result>
S:    <resData>
S:      <org:creData
S:        xmlns:org="urn:ietf:params:xml:ns:epp:org-1.0">
S:        <org:id>res1523</org:id>
S:        <org:crDate>1999-04-03T22:00:00.0Z</org:crDate>
S:      </org:creData>
S:    </resData>
S:    <trID>
S:      <clTRID>ABC-12345</clTRID>
S:      <svTRID>54321-XYZ</svTRID>
S:    </trID>
S:  </response>
S:</epp>
```

An EPP error response MUST be returned if a <create> command cannot be processed for any reason.

4.2.2. EPP <delete> Command

The EPP <delete> command provides a transform operation that allows a client to delete an organization object. In addition to the standard EPP command elements, the <delete> command MUST contain an <org:delete> element. This element or its ancestor element MUST identify the organization namespace "urn:ietf:params:xml:ns:epp:org-1.0". The <org:delete> element MUST contain the following child element:

- o An <org:id> element that contains the server-unique identifier of the organization object to be deleted, as defined in Section 3.1.

An organization object MUST NOT be deleted if it is associated with other known objects. An associated organization MUST NOT be deleted until associations with other known objects have been broken. A server MUST notify clients that object relationships exist by sending a 2305 error response code when a <delete> command is attempted and fails due to existing object relationships.

Example <delete> command:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <delete>
C:      <org:delete
C:        xmlns:org="urn:ietf:params:xml:ns:epp:org-1.0">
C:          <org:id>res1523</org:id>
C:        </org:delete>
C:      </delete>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

When a <delete> command has been processed successfully, a server MUST respond with an EPP response with no <resData> element.

Example <delete> response:

```
S:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S:  <response>
S:    <result code="1000">
S:      <msg lang="en">Command completed successfully</msg>
S:    </result>
S:    <trID>
S:      <clTRID>ABC-12345</clTRID>
S:      <svTRID>54321-XYZ</svTRID>
S:    </trID>
S:  </response>
S:</epp>
```

An EPP error response MUST be returned if a <delete> command cannot be processed for any reason.

4.2.3. EPP <renew> Command

Renewal semantics do not apply to organization objects, so there is no mapping defined for the EPP <renew> command.

4.2.4. EPP <transfer> Command

Transfer semantics do not apply to organization objects, so there is no mapping defined for the EPP <transfer> command.

4.2.5. EPP <update> Command

The EPP <update> command provides a transform operation that allows a client to modify the attributes of an organization object. In addition to the standard EPP command elements, the <update> command MUST contain a <org:update> element. This element or its ancestor element MUST identify the organization namespace "urn:ietf:params:xml:ns:epp:org-1.0". The <org:update> element contains the following child elements:

- o An <org:id> element that contains the server-unique identifier of the organization object to be updated, as defined in Section 3.1.
- o An OPTIONAL <org:add> element that contains attribute values to be added to the object.
- o An OPTIONAL <org:rem> element that contains attribute values to be removed from the object.
- o An OPTIONAL <org:chg> element that contains attribute values to be changed.

At least one <org:add>, <org:rem> or <org:chg> element MUST be provided if the command is not being extended. All of these elements MAY be omitted if an <update> extension is present. The OPTIONAL <org:add> and <org:rem> elements contain the following child elements:

- o Zero or more <org:contact> elements that contain the identifiers for contact objects to be associated with or removed from the organization object. Contact object identifiers MUST be known to the server before the contact object can be associated with the organization object.
- o Zero or more <org:role> elements that contain the role type, role statuses and optional role id of the organization.
 - * An <org:type> element that contains the role type of the organization, as defined in Section 3.2. The role type uniquely identifies the role to update.
 - * Zero or more <org:status> elements that contain the role statuses. The values of the role status are defined in Section 3.5.
 - * An OPTIONAL <org:roleID> element that contains a third-party-assigned identifier, such as IANA ID for registrars, as defined in Section 3.2.3.

- o Zero or more <org:status> elements that contain the operational status of the organization.

An OPTIONAL <org:chg> element contains the following child elements, where at least one child element MUST be present:

- o An OPTIONAL <org:parentId> element that contains the identifier of the parent object.
- o Zero to two <org: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 Unicode in the range U+0020 - U+007E. If a localized form (type="loc") is provided, element content MAY be represented in unrestricted UTF-8. The change of the postal info is defined as a replacement of that postal info element with the contents of the sub-elements included in the update command. An empty <org:postalInfo> element is supported to allow a type of postal info to be removed. The <org:postalInfo> element contains the following child elements:
 - * An <org:name> element that contains the name of the organization.
 - * An OPTIONAL <org:addr> element that contains address information associated with the organization. A <org:addr> element contains the following child elements:
 - + One, two, or three <org:street> elements that contain the organization's street address.
 - + An <org:city> element that contains the organization's city.
 - + An OPTIONAL <org:sp> element that contains the organization's state or province.
 - + An OPTIONAL <org:pc> element that contains the organization's postal code.
 - + An <org:cc> element that contains the alpha-2 organization's country code. The detailed format of this element is described in section 2.4.3 of [RFC5733].
- o An OPTIONAL <org:voice> element that contains the organization's voice telephone number. The detailed format of this element is described in Section 2.5 of [RFC5733]

- o An OPTIONAL <org:fax> element that contains the organization's facsimile telephone number.
- o An OPTIONAL <org:email> element that contains the organization's email address. The detailed format of this element is described in section 2.6 of [RFC5733].
- o An OPTIONAL <org:url> element that contains the URL to the website of the organization. The detailed format of this element is described in [RFC3986]

Example <update> command:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <update>
C:      <org:update
C:        xmlns:org="urn:ietf:params:xml:ns:epp:org-1.0">
C:          <org:id>res1523</org:id>
C:          <org:add>
C:            <org:contact type="tech">sh8013</org:contact>
C:            <org:role>
C:              <org:type>privacyproxy</org:type>
C:              <org:status>clientLinkProhibited</org:status>
C:            </org:role>
C:            <org:status>clientLinkProhibited</org:status>
C:          </org:add>
C:          <org:rem>
C:            <org:contact type="billing">sh8014</org:contact>
C:            <org:role>
C:              <org:type>reseller</org:type>
C:            </org:role>
C:          </org:rem>
C:          <org:chg>
C:            <org:postalInfo type="int">
C:              <org:addr>
C:                <org:street>124 Example Dr.</org:street>
C:                <org:street>Suite 200</org:street>
C:                <org:city>Dulles</org:city>
C:                <org:sp>VA</org:sp>
C:                <org:pc>20166-6503</org:pc>
C:                <org:cc>US</org:cc>
C:              </org:addr>
C:            </org:postalInfo>
C:            <org:voice>+1.7034444444</org:voice>
C:            <org:fax/>
C:          </org:chg>
C:        </org:update>
C:      </update>
C:      <clTRID>ABC-12345</clTRID>
C:    </command>
C:</epp>
```

When an <update> command has been processed successfully, a server MUST respond with an EPP response with no <resData> element.

Example <update> response:

```
S:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S:  <response>
S:    <result code="1000">
S:      <msg lang="en">Command completed successfully</msg>
S:    </result>
S:    <trID>
S:      <clTRID>ABC-12345</clTRID>
S:      <svTRID>54321-XYZ</svTRID>
S:    </trID>
S:  </response>
S:</epp>
```

An EPP error response MUST be returned if an <update> command cannot be processed for any reason.

4.3. Offline Review of Requested Actions

Commands are processed by a server in the order they are received from a client. Though an immediate response confirming receipt and processing of the command is produced by the server, a server operator MAY perform an offline review of requested transform commands before completing the requested action. In such situations, the response from the server MUST clearly note that the transform command has been received and processed, but the requested action is pending. The status in the response of the corresponding object MUST clearly reflect processing of the pending action. The server MUST notify the client when offline processing of the action has been completed.

Examples describing a <create> command that requires offline review are included here. Note the result code and message returned in response to the <create> command.

```
S:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S:  <response>
S:    <result code="1001">
S:      <msg lang="en">Command completed successfully;
S:        action pending</msg>
S:    </result>
S:    <resData>
S:      <org:creData
S:        xmlns:org="urn:ietf:params:xml:ns:epp:org-1.0">
S:        <org:id>res1523</org:id>
S:        <org:crDate>1999-04-03T22:00:00.0Z</org:crDate>
S:      </org:creData>
S:    </resData>
S:    <trID>
S:      <clTRID>ABC-12345</clTRID>
S:      <svTRID>54321-XYZ</svTRID>
S:    </trID>
S:  </response>
S:</epp>
```

The status of the organization object after returning this response MUST include "pendingCreate". The server operator reviews the request offline, and informs the client of the outcome of the review by queuing a service message for retrieval via the <poll> command; it MAY additionally use an out-of-band mechanism to inform the client of the outcome.

The service message MUST contain text that describes the notification in the child <msg> element of the response <msgQ> element. In addition, the EPP <resData> element MUST contain a child <org:panData> element. This element or its ancestor element MUST identify the organization namespace "urn:ietf:params:xml:ns:epp:org-1.0". The <org:panData> element contains the following child elements:

- o An <org:id> element that contains the server-unique identifier of the organization object. The <org:id> element contains a REQUIRED "paResult" attribute. A positive boolean value indicates that the request has been approved and completed. A negative boolean value indicates that the request has been denied and the requested action has not been taken.
- o An <org:paTRID> element that contains the client transaction identifier and server transaction identifier returned with the original response to process the command. The client transaction

identifier is OPTIONAL and will only be returned if the client provided an identifier with the original <create> command.

- o An <org:paDate> element that contains the date and time describing when review of the requested action was completed.

Example "review completed" service message:

```
S:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S:  <response>
S:    <result code="1301">
S:      <msg lang="en">Command completed successfully;
S:        ack to dequeue</msg>
S:    </result>
S:    <msgQ count="5" id="12345">
S:      <qDate>1999-04-04T22:01:00.0Z</qDate>
S:      <msg>Pending action completed successfully.</msg>
S:    </msgQ>
S:    <resData>
S:      <org:panData
S:        xmlns:org="urn:ietf:params:xml:ns:epp:org-1.0">
S:        <org:id paResult="1">res1523</org:id>
S:        <org:paTRID>
S:          <clTRID>ABC-12345</clTRID>
S:          <svTRID>54321-XYZ</svTRID>
S:        </org:paTRID>
S:        <org:paDate>1999-04-04T22:00:00.0Z</org:paDate>
S:      </org:panData>
S:    </resData>
S:    <trID>
S:      <clTRID>BCD-23456</clTRID>
S:      <svTRID>65432-WXY</svTRID>
S:    </trID>
S:  </response>
S:</epp>
```

5. Formal Syntax

An EPP object mapping is specified in XML Schema notation. The formal syntax presented here is a complete schema representation of the object mapping suitable for automated validation of EPP XML instances. The BEGIN and END tags are not part of the schema; they are used to note the beginning and ending of the schema for URI registration purposes.

```
BEGIN
<?xml version="1.0" encoding="UTF-8"?>

<schema targetNamespace="urn:ietf:params:xml:ns:epp:org-1.0"
  xmlns:org="urn:ietf:params:xml:ns:epp:org-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 common element types.
  -->
  <import namespace="urn:ietf:params:xml:ns:eppcom-1.0"/>
  <import namespace="urn:ietf:params:xml:ns:epp-1.0"/>

  <annotation>
    <documentation>
      Extensible Provisioning Protocol v1.0
      organization provisioning schema.
    </documentation>
  </annotation>

  <!--
  Child elements found in EPP commands.
  -->
  <element name="create" type="org:createType"/>
  <element name="delete" type="org:sIDType"/>
  <element name="update" type="org:updateType"/>
  <element name="check" type="org:mIDType"/>
  <element name="info" type="org:infoType"/>
  <element name="panData" type="org:panDataType"/>

  <!--
  Utility types.
  -->
  <simpleType name="statusType">
    <restriction base="token">
      <enumeration value="ok"/>
      <enumeration value="hold"/>
      <enumeration value="terminated"/>
      <enumeration value="clientDeleteProhibited"/>
      <enumeration value="clientUpdateProhibited"/>
      <enumeration value="clientLinkProhibited"/>
      <enumeration value="linked"/>
      <enumeration value="pendingCreate"/>
      <enumeration value="pendingUpdate"/>
      <enumeration value="pendingDelete"/>
    </restriction>
  </simpleType>
</schema>
```

```
        <enumeration value="serverDeleteProhibited"/>
        <enumeration value="serverUpdateProhibited"/>
        <enumeration value="serverLinkProhibited"/>
    </restriction>
</simpleType>

<simpleType name="roleStatusType">
    <restriction base="token">
        <enumeration value="ok"/>
        <enumeration value="clientLinkProhibited"/>
        <enumeration value="linked"/>
        <enumeration value="serverLinkProhibited"/>
    </restriction>
</simpleType>

<complexType name="roleType">
    <sequence>
        <element name="type" type="token"/>
        <element name="status" type="org:roleStatusType"
            minOccurs="0" maxOccurs="3"/>
        <element name="roleID" type="token" minOccurs="0"/>
    </sequence>
</complexType>

<complexType name="postalInfoType">
    <sequence>
        <element name="name"
            type="org:postalLineType"/>
        <element name="addr"
            type="org:addrType" minOccurs="0"/>
    </sequence>
    <attribute name="type"
        type="org:postalInfoEnumType"
        use="required"/>
</complexType>

<complexType name="contactType">
    <simpleContent>
        <extension base="eppcom:clIDType">
            <attribute name="type" type="org:contactAttrType"
                use="required"/>
            <attribute name="typeName" type="token"/>
        </extension>
    </simpleContent>
</complexType>

<simpleType name="contactAttrType">
    <restriction base="token">
```

```
        <enumeration value="admin"/>
        <enumeration value="billing"/>
        <enumeration value="tech"/>
        <enumeration value="abuse"/>
        <enumeration value="custom"/>
    </restriction>
</simpleType>

<complexType name="e164Type">
    <simpleContent>
        <extension base="org:e164StringType">
            <attribute name="x" type="token" />
        </extension>
    </simpleContent>
</complexType>

<simpleType name="e164StringType">
    <restriction base="token">
        <pattern value="(\+[0-9]{1,3}\.[0-9]{1,14})?" />
        <maxLength value="17" />
    </restriction>
</simpleType>

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

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

<!--
Child element of commands that require only an identifier.
-->
<complexType name="sIDType">
  <sequence>
    <element name="id" type="eppcom:clIDType"/>
  </sequence>
</complexType>

<!--
Child element of commands that accept multiple identifiers.
-->
<complexType name="mIDType">
  <sequence>
    <element name="id"
      type="eppcom:clIDType" maxOccurs="unbounded"/>
  </sequence>
</complexType>

<!--
Pending action notification response elements.
-->
<complexType name="panDataType">
  <sequence>
    <element name="id" type="org:paCLIDType"/>
    <element name="paTRID" type="epp:trIDType"/>
    <element name="paDate" type="dateTime"/>
  </sequence>
```

```
</complexType>

<complexType name="paCLIDType">
  <simpleContent>
    <extension base="eppcom:clIDType">
      <attribute name="paResult" type="boolean"
        use="required"/>
    </extension>
  </simpleContent>
</complexType>

<!--
Child elements of the <info> commands.
-->
<complexType name="infoType">
  <sequence>
    <element name="id"
      type="eppcom:clIDType"/>
  </sequence>
</complexType>

<!--
Child elements of the <create> command.
-->
<complexType name="createType">
  <sequence>
    <element name="id"
      type="eppcom:clIDType"/>
    <element name="role"
      type="org:roleType" maxOccurs="unbounded"/>
    <element name="status"
      type="org:statusType" minOccurs="0" maxOccurs="4"/>
    <element name="parentId"
      type="eppcom:clIDType" minOccurs="0"/>
    <element name="postalInfo"
      type="org:postalInfoType" minOccurs="0" maxOccurs="2"/>
    <element name="voice"
      type="org:e164Type" minOccurs="0"/>
    <element name="fax"
      type="org:e164Type" minOccurs="0"/>
    <element name="email"
      type="eppcom:minTokenType" minOccurs="0"/>
    <element name="url"
      type="anyURI" minOccurs="0"/>
    <element name="contact"
      type="org:contactType"
      minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
```

```
</complexType>

<!--
Child elements of the <update> command.
-->
<complexType name="updateType">
  <sequence>
    <element name="id"
      type="eppcom:clIDType"/>
    <element name="add"
      type="org:addRemType" minOccurs="0"/>
    <element name="rem"
      type="org:addRemType" minOccurs="0"/>
    <element name="chg"
      type="org:chgType" minOccurs="0"/>
  </sequence>
</complexType>

<!--
Data elements that can be added or removed.
-->
<complexType name="addRemType">
  <sequence>
    <element name="contact"
      type="org:contactType" minOccurs="0" maxOccurs="unbounded"/>
    <element name="role" type="org:roleType"
      minOccurs="0" maxOccurs="unbounded"/>
    <element name="status" type="org:statusType"
      minOccurs="0" maxOccurs="9"/>
  </sequence>
</complexType>

<!--
Data elements that can be changed.
-->
<complexType name="chgType">
  <sequence>
    <element name="parentId"
      type="eppcom:clIDType" minOccurs="0"/>
    <element name="postalInfo"
      type="org:chgPostalInfoType"
      minOccurs="0" maxOccurs="2"/>
    <element name="voice"
      type="org:e164Type" minOccurs="0"/>
    <element name="fax"
      type="org:e164Type" minOccurs="0"/>
    <element name="email"
      type="eppcom:minTokenType" minOccurs="0"/>
  </sequence>
</complexType>
```

```
        <element name="url"
            type="anyURI" minOccurs="0"/>
    </sequence>
</complexType>

<complexType name="chgPostalInfoType">
    <sequence>
        <element name="name"
            type="org:postalLineType" minOccurs="0"/>
        <element name="addr"
            type="org:addrType" minOccurs="0"/>
    </sequence>
    <attribute name="type"
        type="org:postalInfoEnumType" use="required"/>
</complexType>

<!--
Child response elements.
-->
<element name="chkData" type="org:chkDataType"/>
<element name="creData" type="org:creDataType"/>
<element name="infData" type="org:infDataType"/>

<!--
<check> response elements.
-->
<complexType name="chkDataType">
    <sequence>
        <element name="cd" type="org:checkType"
            maxOccurs="unbounded" />
    </sequence>
</complexType>

<complexType name="checkType">
    <sequence>
        <element name="id" type="org:checkIDType" />
        <element name="reason" type="eppcom:reasonType"
            minOccurs="0" />
    </sequence>
</complexType>

<complexType name="checkIDType">
    <simpleContent>
        <extension base="eppcom:clIDType">
            <attribute name="avail" type="boolean"
                use="required" />
        </extension>
    </simpleContent>
</complexType>
```



```
</complexType>

<!--
<info> response elements.
-->
<complexType name="infDataType">
  <sequence>
    <element name="id"
      type="eppcom:clIDType"/>
    <element name="roid"
      type="eppcom:roidType"/>
    <element name="role"
      type="org:roleType" maxOccurs="unbounded"/>
    <element name="status"
      type="org:statusType" maxOccurs="9"/>
    <element name="parentId"
      type="eppcom:clIDType" minOccurs="0"/>
    <element name="postalInfo"
      type="org:postalInfoType" minOccurs="0" maxOccurs="2"/>
    <element name="voice"
      type="org:e164Type" minOccurs="0"/>
    <element name="fax"
      type="org:e164Type" minOccurs="0"/>
    <element name="email"
      type="eppcom:minTokenType" minOccurs="0"/>
    <element name="url"
      type="anyURI" minOccurs="0"/>
    <element name="contact"
      type="org:contactType" minOccurs="0" maxOccurs="unbounded"/>
    <element name="clID"
      type="eppcom:clIDType" minOccurs="0"/>
    <element name="crID"
      type="eppcom:clIDType"/>
    <element name="crDate"
      type="dateTime"/>
    <element name="upID"
      type="eppcom:clIDType" minOccurs="0"/>
    <element name="upDate"
      type="dateTime" minOccurs="0"/>
  </sequence>
</complexType>

<!--
<create> response elements.
-->
<complexType name="creDataType">
  <sequence>
    <element name="id" type="eppcom:clIDType" />
    <element name="crDate" type="dateTime" />
  </sequence>
</complexType>
```

```
        </sequence>
      </complexType>

      <!--
      End of schema.
      -->
    </schema>
  END
```

6. Internationalization Considerations

EPP is 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 [RFC3629] and UTF-16 [RFC2781]. 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.

As an extension of the EPP organization object mapping, the elements and element content described in this document MUST inherit the internationalization conventions used to represent higher-layer domain and core protocol structures present in an XML instance that includes this extension.

7. IANA Considerations

7.1. XML Namespace

This document uses URNs to describe XML namespaces and XML schemas conforming to a registry mechanism described in [RFC3688]. IANA is requested to assign the following URI.

Registration request for the organization namespace:

URI: urn:ietf:params:xml:ns:epp:org-1.0

Registrant Contact: IESG

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

Registration request for the organization XML schema:

URI: urn:ietf:params:xml:schema:epp:org-1.0

Registrant Contact: IESG

XML: See the "Formal Syntax" section of this document.

7.2. EPP Extension Registry

The EPP extension described in this document should be registered by the IANA in the EPP Extension Registry described in [RFC7451]. The details of the registration are as follows:

Name of Extension: Extensible Provisioning Protocol (EPP)
Organization Mapping

Document status: Standards Track

Reference: RFCXXXX (please replace "XXXX" with the RFC number for this document after a number is assigned by the RFC Editor)

Registrant Name and Email Address: IESG, iesg@ietf.org

TLDs: Any

IPR Disclosure: None

Status: Active

Notes: None

7.3. Role Type Values Registry

IANA has created a new category of protocol registry for values of the organization roles. The name of this registry is "EPP Organization Role Values". The registration policy for this registry is "Expert Review" [RFC8126].

7.3.1. Registration Template

Value: the string value being registered.

Description: Brief description of the organization role values.

Registrant Name: For IETF RFCs, state "IESG". For others, give the name of the responsible party.

Registrant Contact Information: an email address, postal address, or some other information to be used to contact the registrant.

7.3.2. Initial Registry Contents

Followings are the initial registry contents:

Value: registrar

Description: The entity object instance represents the authority responsible for the registration in the registry.

Registrant Name: IESG

Registrant Contact Information: iesg@ietf.org

Value: reseller

Description: The entity object instance represents a third party through which the registration was conducted (i.e., not the registry or registrar).

Registrant Name: IESG

Registrant Contact Information: iesg@ietf.org

Value: privacyproxy

Description: The entity object instance represents a third-party who could help to register a domain without exposing the registrants' private information.

Registrant Name: IESG

Registrant Contact Information: iesg@ietf.org

Value: dns-operator

Description: The entity object instance represents a third-party DNS operator that maintains the name servers and zone data on behalf of a registrant.

Registrant Name: IESG

Registrant Contact Information: iesg@ietf.org

8. Implementation Status

Note to RFC Editor: Please remove this section and the reference to [RFC7942] before publication. This section records the status of known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft, and is based on a proposal described in [RFC7942]. The description of implementations in this section is intended to assist the IETF in its decision processes in progressing drafts to RFCs. Please note that the listing of any individual implementation here does not imply endorsement by the IETF. Furthermore, no effort has been spent to

verify the information presented here that was supplied by IETF contributors. This is not intended as, and must not be construed to be, a catalog of available implementations or their features. Readers are advised to note that other implementations may exist.

According to [RFC7942], "this will allow reviewers and working groups to assign due consideration to documents that have the benefit of running code, which may serve as evidence of valuable experimentation and feedback that have made the implemented protocols more mature. It is up to the individual working groups to use this information as they see fit".

8.1. Verisign EPP SDK

Organization: Verisign Inc.

Name: Verisign EPP SDK

Description: The Verisign EPP SDK includes both a full client implementation and a full server stub implementation of draft-ietf-regext-org.

Level of maturity: Development

Coverage: All aspects of the protocol are implemented.

Licensing: GNU Lesser General Public License

Contact: jgould@verisign.com

URL: https://www.verisign.com/en_US/channel-resources/domain-registry-products/epp-sdks

8.2. CNNIC Implementation

Organization: CNNIC

Name: EPP Organization Mapping

Description: CNNIC is trying to update EPP organization mapping from previous reseller mapping according to this document.

Level of maturity: Development

Coverage: EPP organization mapping

Contact: zhouguiqing@cnnic.cn

9. Security Considerations

The organization object may have personally identifiable information, such as <org:contact>. This information is not a required element in this document which can be provided on a voluntary basis. If it is provided, both client and server MUST ensure that authorization information is stored and exchanged with high-grade encryption mechanisms to provide privacy services, which is specified in [RFC5733]. The security considerations described in [RFC5730] or those caused by the protocol layers used by EPP will apply to this specification as well.

10. Acknowledgment

The authors would like to thank Rik Ribbers, Marc Groeneweg, Patrick Mevzek, Antoin Verschuren and Scott Hollenbeck for their careful review and valuable comments.

11. References

11.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, DOI 10.17487/RFC3629, November 2003, <<https://www.rfc-editor.org/info/rfc3629>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <<https://www.rfc-editor.org/info/rfc3986>>.
- [RFC5646] Phillips, A., Ed. and M. Davis, Ed., "Tags for Identifying Languages", BCP 47, RFC 5646, DOI 10.17487/RFC5646, September 2009, <<https://www.rfc-editor.org/info/rfc5646>>.
- [RFC5730] Hollenbeck, S., "Extensible Provisioning Protocol (EPP)", STD 69, RFC 5730, DOI 10.17487/RFC5730, August 2009, <<https://www.rfc-editor.org/info/rfc5730>>.

- [RFC5733] Hollenbeck, S., "Extensible Provisioning Protocol (EPP) Contact Mapping", STD 69, RFC 5733, DOI 10.17487/RFC5733, August 2009, <<https://www.rfc-editor.org/info/rfc5733>>.
- [RFC7942] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", BCP 205, RFC 7942, DOI 10.17487/RFC7942, July 2016, <<https://www.rfc-editor.org/info/rfc7942>>.
- [RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <<https://www.rfc-editor.org/info/rfc8126>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [W3C.REC-xml-20040204]
Bray, T., Paoli, J., Sperberg-McQueen, C., Maler, E., and F. Yergeau, "Extensible Markup Language (XML) 1.0 (Third Edition)", World Wide Web Consortium First Edition REC-xml-20040204, February 2004, <<http://www.w3.org/TR/2004/REC-xml-20040204>>.
- [W3C.REC-xmlschema-1-20041028]
Thompson, H., Beech, D., Maloney, M., and N. Mendelsohn, "XML Schema Part 1: Structures Second Edition", World Wide Web Consortium Recommendation REC-xmlschema-1-20041028, October 2004, <<http://www.w3.org/TR/2004/REC-xmlschema-1-20041028>>.
- [W3C.REC-xmlschema-2-20041028]
Biron, P. and A. Malhotra, "XML Schema Part 2: Datatypes Second Edition", World Wide Web Consortium Recommendation REC-xmlschema-2-20041028, October 2004, <<http://www.w3.org/TR/2004/REC-xmlschema-2-20041028>>.

11.2. Informative References

- [RFC2781] Hoffman, P. and F. Yergeau, "UTF-16, an encoding of ISO 10646", RFC 2781, DOI 10.17487/RFC2781, February 2000, <<https://www.rfc-editor.org/info/rfc2781>>.
- [RFC7451] Hollenbeck, S., "Extension Registry for the Extensible Provisioning Protocol", RFC 7451, DOI 10.17487/RFC7451, February 2015, <<https://www.rfc-editor.org/info/rfc7451>>.

Appendix A. Change Log

Initial -00: Individual document submitted.

-01:

- * Updated abstract text.
- * Added sentences to avoid loop of parent identifiers in section 3.4.
- * Revised typos in section 3.6.
- * Added explanation of contact type attribute in section 4.1.2.
- * Updated <info> responses.
- * Deleted description of <transfer> command in section 4.1 and 4.2.
- * Deleted whoisInfo disclose type in XML schema.
- * Deleted maxOccurs of addRemType.
- * Deleted extra "OPTIONAL" in section 4.2.5.
- * Updated typos in <update> response.

-02:

- * Changed author information.
- * Updated url definition.
- * Updated XML schema.

-03:

- * Changed author information.
- * Updated section 3.1.
- * Refactoried the XSD file. Added <chgPostalInfoType> element.
- * Added acknowledgment.

WG document-00: WG document submitted

WG document-01: Keep document alive for further discussion.
Reseller object or entity object with multiple roles?

Organization WG document-00: Change to a generic organization object mapping.

Organization WG document-01: Added "Implementation Status" section.

Organization WG document-02: Accepted some of the feedbacks on the mailing list.

Organization WG document-03:

- * Updated section 3.2, changed the structure of organization role.
- * Updated section 4.2.5 for the "add", "rem" and "chg" example.
- * Updated section 5 of formal syntax.
- * Updated section 7.2 for the registration template and initial values.
- * Updated section 8 of implementation status.

Organization WG document-04:

- * Updated section 3.2, changed the structure of organization role.
- * Updated references.
- * Updated section 8 of implementation status.

Organization WG document-05:

- * Updated the description of <org:status> of a role.
- * Removed the third paragraph of "Implementation Status".
- * Remove the Informative Reference to draft-ietf-regext-reseller from the draft.

Organization WG document-06:

- * Updated typos.
- * Added "Query" for "<Transfer> Query Command".

- * Change "Registrant Contact" to IESG in section 7.1.
- * Modified section 7.2.

Organization WG document-07:

- * Updated typos.
- * Added dns-operator in section 7.1.
- * Added "OPTIONAL" for <org:addr>

Organization WG document-08:

- * Updated "Offline Review of Requested Actions".

Organization WG document-09:

- * Updated "This element or its ancestor element MUST identify the organization namespace." in section 4.1.1 and other parts of this document.
- * Updated text in section 2 match RFC 8174.
- * Modified "roleid" to "roleID".
- * Updated text about loops in section 3.6.
- * Referred section 2.5 of RFC5733 for voice format.
- * Updated XML schema for the maxOccurs value of "reason" element.
- * Updated section 7.3.
- * Replaced "http" with "https" in the examples.
- * Updated writing typos.
- * Modified XML namespace and schema.

Organization WG document-10:

- * Modified XML namespace and schema.
- * Removed the maxOccurs value of "reason" element.

Organization WG document-11:

- * Typo of RFC2781 and moved this reference in "Informative References".
- * "Loops MUST be prohibited." in section 3.6.

Organization WG document-12:

- * Removed "OPTIONAL" when "zero or more" or "zero to two" appears.
- * Updated the "Organization Status Values" text.
- * Updated the full xml namespace.
- * Updated the text in "Offline review".
- * Updated the text in "Security Considerations".
- * Added "Document satus" and "Reference" in section "EPP Extension Registry".
- * Added references of RFC3688, RFC3986 and RFC5646.

Authors' Addresses

Linlin Zhou
CNNIC
4 South 4th Street, Zhongguancun, Haidian District
Beijing, Beijing 100190
China

Email: zhoulinlin@cnnic.cn

Ning Kong
Consultant

Email: ietfing@gmail.com

Guiqing Zhou
CNNIC
4 South 4th Street, Zhongguancun, Haidian District
Beijing, Beijing 100190
China

Email: zhouguiqing@cnnic.cn

Jiankang Yao
CNNIC
4 South 4th Street, Zhongguancun, Haidian District
Beijing, Beijing 100190
China

Email: yaojk@cnnic.cn

James Gould
Verisign, Inc.
12061 Bluemont Way
Reston, VA 20190
US

Email: jgould@verisign.com

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: June 8, 2019

L. Zhou
CNNIC
N. Kong
Consultant
J. Wei
J. Yao
CNNIC
J. Gould
Verisign, Inc.
December 5, 2018

Organization Extension for the Extensible Provisioning Protocol (EPP)
draft-ietf-regext-org-ext-11

Abstract

This document describes an extension to Extensible Provisioning Protocol (EPP) object mappings, which is designed to support assigning an organization to any existing object (domain, host, contact) as well as any future 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 <https://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 June 8, 2019.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must 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	2
2. Conventions Used in This Document	3
3. Object Attributes	4
3.1. Organization Identifier	4
4. EPP Command Mapping	4
4.1. EPP Query Commands	4
4.1.1. EPP <check> Command	4
4.1.2. EPP <info> Command	4
4.1.3. EPP <transfer> Query Command	7
4.2. EPP Transform Commands	8
4.2.1. EPP <create> Command	8
4.2.2. EPP <delete> Command	10
4.2.3. EPP <renew> Command	10
4.2.4. EPP <transfer> Command	11
4.2.5. EPP <update> Command	11
5. Formal Syntax	15
6. Internationalization Considerations	18
7. IANA Considerations	18
7.1. XML Namespace	18
7.2. EPP Extension Registry	18
8. Implementation Status	19
8.1. Verisign EPP SDK	19
8.2. CNNIC Implementation	20
9. Security Considerations	20
10. Acknowledgment	20
11. References	20
11.1. Normative References	20
11.2. Informative References	22
Appendix A. Change Log	22
Authors' Addresses	25

1. Introduction

In the business model of domain registration, we usually have three roles of entities: a registrant, a registrar and a registry, as defined in section 9 of [ID.draft-ietf-dnsop-terminology-bis]. There may be other roles of entities involved in the domain registration process, such as resellers, DNS operators in section 9 of [ID.draft-ietf-dnsop-terminology-bis], privacy proxies, etc.

A domain reseller is an individual or a company that acts as an agent for accredited registrars. DNS operator is defined in section 9 of [ID.draft-ietf-dnsop-terminology-bis]. A privacy proxy is an entity used for domain registrations to protect the private information of the individuals and organizations. These kind of entities are defined as "organizations" with different role types in this document.

In order to facilitate provisioning and management of organization information in a shared central repository, this document proposes an organization extension mapping for any Extensible Provisioning Protocol (EPP) object like domain names in [RFC5731], hosts in [RFC5732] and contacts in [RFC5733]. The examples provided in this document are used for the domain object for illustration purpose. The host and contact object could be extended in the same way with the domain object.

Organization object identifiers defined in [ID.draft-ietf-regext-org] MUST be known to the server before the organization object can be associated with the EPP object.

This document is specified using the XML 1.0 as described in [W3C.REC-xml-20040204] and XML Schema notation as described in [W3C.REC-xmlschema-1-20041028] and [W3C.REC-xmlschema-2-20041028].

2. Conventions Used in This Document

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

In examples, "C:" represents lines sent by a protocol client and "S:" represents lines returned by a protocol server. Indentation and white space in examples are provided only to illustrate element relationships and are not a required feature of this specification.

XML is case sensitive. Unless stated otherwise, XML specifications and examples provided in this document MUST be interpreted in the character case.

The XML namespace prefix "orgext" is used for the namespace "urn:ietf:params:xml:ns:epp:orgext-1.0", but implementations MUST NOT depend on it and instead employ a proper namespace-aware XML parser and serializer to interpret and output the XML documents.

3. Object Attributes

This extension adds additional elements to EPP object mappings like the EPP domain name mapping [RFC5731]. Only the new elements are described here.

3.1. Organization Identifier

Organization identifier provides the ID of an organization. Its corresponding element is `<orgext:id>` which refers to the `<org:id>` element defined in [ID.draft-ietf-regext-org]. All organization objects are identified by a server-unique identifier. A "role" attribute is used to represent the relationship that the organization has to the EPP object. Any given object MUST have at most one associated organization ID for any given role value.

4. EPP Command Mapping

A detailed description of the EPP syntax and semantics can be found in the EPP core protocol specification [RFC5730]. The command mappings described here are specifically for assigning organizations to EPP objects.

4.1. EPP Query Commands

EPP provides three commands to retrieve EPP object information: `<check>` to determine if an object can be provisioned within a repository, `<info>` to retrieve detailed information associated with an object, and `<transfer>` to retrieve object transfer status information.

4.1.1. EPP `<check>` Command

This extension does not add any elements to the EPP `<check>` command or `<check>` response described in the EPP object mapping.

4.1.2. EPP `<info>` Command

This extension does not add any elements to the EPP `<info>` command described in the EPP object mapping. However, additional elements are defined for the `<info>` response in the EPP object mapping.

When an `<info>` command has been processed successfully, the EPP `<resData>` element MUST contain child elements as described in the EPP object extensions. In addition, the EPP `<extension>` element SHOULD contain a child `<orgext:infData>` element. This element or its ancestor element MUST identify the extension namespace "urn:ietf:params:xml:ns:epp:orgext-1.0" if the object has data

associated with this extension and based on server policy. The <orgext:infData> element contains the following child elements:

- o Zero or more <orgext:id> elements are allowed that contain the identifier of the organization, as defined in Section 3.1. The "role" attribute is used to represent the relationship that the organization has to the object. See Section 7.3 in [ID.draft-ietf-regext-org] for a list of values.

Example <info> response for an authorized client with multiple organizations:

```
S:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S:  <response>
S:    <result code="1000">
S:      <msg lang="en-US">Command completed successfully</msg>
S:    </result>
S:    <resData>
S:      <domain:infData
S:        xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
S:        <domain:name>example.com</domain:name>
S:        <domain:roid>EXAMPLE1-REP</domain:roid>
S:        <domain:status s="ok"/>
S:        <domain:registrant>jdl1234</domain:registrant>
S:        <domain:contact type="admin">sh8013</domain:contact>
S:        <domain:contact type="billing">sh8013</domain:contact>
S:        <domain:contact type="tech">sh8013</domain:contact>
S:        <domain:ns>
S:          <domain:hostObj>ns1.example.com</domain:hostObj>
S:        </domain:ns>
S:        <domain:clID>ClientX</domain:clID>
S:        <domain:crID>ClientY</domain:crID>
S:        <domain:crDate>2015-02-06T04:01:21.0Z</domain:crDate>
S:        <domain:exDate>2018-02-06T04:01:21.0Z</domain:exDate>
S:        <domain:authInfo>
S:          <domain:pw>2fooBAR</domain:pw>
S:        </domain:authInfo>
S:      </domain:infData>
S:    </resData>
S:    <extension>
S:      <orgext:infData
S:        xmlns:orgext="urn:ietf:params:xml:ns:epp:orgext-1.0">
S:        <orgext:id role="reseller">reseller1523</orgext:id>
S:        <orgext:id role="privacyproxy">proxy2935</orgext:id>
S:      </orgext:infData>
S:    </extension>
S:    <trID>
S:      <clTRID>ngcl-IvJjzMZc</clTRID>
S:      <svTRID>test142AWQONJZ</svTRID>
S:    </trID>
S:  </response>
S:</epp>
```

Example <info> response for an authorized client with no organization:

```

S:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S:  <response>
S:    <result code="1000">
S:      <msg lang="en-US">Command completed successfully</msg>
S:    </result>
S:    <resData>
S:      <domain:infData
S:        xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
S:        <domain:name>example.com</domain:name>
S:        <domain:roid>EXAMPLE1-REP</domain:roid>
S:        <domain:status s="ok"/>
S:        <domain:registrar>jd1234</domain:registrar>
S:        <domain:contact type="admin">sh8013</domain:contact>
S:        <domain:contact type="billing">sh8013</domain:contact>
S:        <domain:contact type="tech">sh8013</domain:contact>
S:        <domain:ns>
S:          <domain:hostObj>ns1.example.com</domain:hostObj>
S:        </domain:ns>
S:        <domain:clID>ClientX</domain:clID>
S:        <domain:crID>ClientY</domain:crID>
S:        <domain:crDate>2015-02-06T04:01:21.0Z</domain:crDate>
S:        <domain:exDate>2018-02-06T04:01:21.0Z</domain:exDate>
S:        <domain:authInfo>
S:          <domain:pw>2fooBAR</domain:pw>
S:        </domain:authInfo>
S:      </domain:infData>
S:    </resData>
S:    <extension>
S:      <orgext:infData
S:        xmlns:orgext="urn:ietf:params:xml:ns:epp:orgext-1.0"/>
S:      </extension>
S:    <trID>
S:      <clTRID>ngcl-IvJjzMZc</clTRID>
S:      <svTRID>test142AWQONJZ</svTRID>
S:    </trID>
S:  </response>
S:</epp>

```

An EPP error response MUST be returned if an <info> command cannot be processed for any reason.

4.1.3. EPP <transfer> Query Command

This extension does not add any elements to the EPP <transfer> query command or <transfer> query response described in the EPP object mapping.

4.2. EPP Transform Commands

EPP provides five commands to transform EPP objects: `<create>` to create an instance of an object, `<delete>` to delete an instance of an object, `<renew>` to extend the validity period of an object, `<transfer>` to manage the object sponsorship changes, and `<update>` to change information associated with an object.

4.2.1. EPP `<create>` Command

This extension defines additional elements for the EPP `<create>` command described in the EPP object extensions. No additional elements are defined for the EPP `<create>` response.

The EPP `<create>` command provides a transform operation that allows a client to create an object. In addition to the EPP command elements described in the EPP object extensions, the command MUST contain an `<extension>` element, and the `<extension>` element MUST contain a child `<orgext:create>` element. This element or its ancestor element MUST identify the extension namespace "urn:ietf:params:xml:ns:epp:orgext-1.0" if the client wants to associate data defined in this extension to the object. The `<orgext:create>` element contains the following child elements:

- o One or more `<orgext:id>` elements that contain the identifier of the organization, as defined in Section 3.1. The "role" attribute is used to represent the relationship that the organization has to the object. See Section 7.3 in [ID.draft-ietf-regext-org] for a list of values.

Example `<create>` Command with only one organization:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <create>
C:      <domain:create
C:        xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
C:          <domain:name>example.com</domain:name>
C:          <domain:period unit="y">3</domain:period>
C:          <domain:ns>
C:            <domain:hostObj>ns1.example.com</domain:hostObj>
C:          </domain:ns>
C:          <domain:registrant>jd1234</domain:registrant>
C:          <domain:contact type="tech">sh8013</domain:contact>
C:          <domain:contact type="billing">sh8013</domain:contact>
C:          <domain:contact type="admin">sh8013</domain:contact>
C:          <domain:authInfo>
C:            <domain:pw>fooBAR</domain:pw>
C:          </domain:authInfo>
C:        </domain:create>
C:      </create>
C:    <extension>
C:      <orgext:create
C:        xmlns:orgext="urn:ietf:params:xml:ns:epp:orgext-1.0">
C:          <orgext:id role="reseller">reseller1523</orgext:id>
C:        </orgext:create>
C:      </extension>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

Example <create> Command with multiple organizations:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <create>
C:      <domain:create
C:        xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
C:          <domain:name>example.com</domain:name>
C:          <domain:period unit="y">3</domain:period>
C:          <domain:ns>
C:            <domain:hostObj>ns1.example.com</domain:hostObj>
C:          </domain:ns>
C:          <domain:registrant>jd1234</domain:registrant>
C:          <domain:contact type="tech">sh8013</domain:contact>
C:          <domain:contact type="billing">sh8013</domain:contact>
C:          <domain:contact type="admin">sh8013</domain:contact>
C:          <domain:authInfo>
C:            <domain:pw>fooBAR</domain:pw>
C:          </domain:authInfo>
C:        </domain:create>
C:      </create>
C:    <extension>
C:      <orgext:create
C:        xmlns:orgext="urn:ietf:params:xml:ns:epp:orgext-1.0">
C:          <orgext:id role="reseller">reseller1523</orgext:id>
C:          <orgext:id role="privacyproxy">proxy2935</orgext:id>
C:        </orgext:create>
C:      </extension>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

When a <create> command has been processed successfully, the EPP response is as described in the EPP object extension.

An EPP error response MUST be returned if a <create> command cannot be processed for any reason.

4.2.2. EPP <delete> Command

This extension does not add any elements to the EPP <delete> command or <delete> response described in the EPP object mapping.

4.2.3. EPP <renew> Command

This extension does not add any elements to the EPP <renew> command or <renew> response described in the EPP object mapping.

4.2.4. EPP <transfer> Command

This extension does not add any elements to the EPP <transfer> command or <transfer> response described in the EPP object mapping, but after a successful transfer of an object with an assigned organization, the handling of the assigned organization is dependent on the organization roles and server policy.

4.2.5. EPP <update> Command

This extension defines additional elements for the EPP <update> command described in the EPP domain mapping [RFC5731], host mapping [RFC5732] and contact mapping [RFC5733]. No additional elements are defined for the EPP <update> response.

The EPP <update> command provides a transform operation that allows a client to modify the attributes of an object. In addition to the EPP <update> command elements, the command MUST contain an <extension> element, and the <extension> element MUST contain a child <orgext:update> element. This element or its ancestor element MUST identify the extension namespace "urn:ietf:params:xml:ns:epp:orgext-1.0" if the client wants to update the object with data defined in this extension. The <orgext:update> element contains the following child elements:

- o An OPTIONAL <orgext:add> element that contains one or more <orgext:id> elements, as defined in Section 3.1, that add non-existent organization roles to the object. The <orgext:id> element MUST have a non-empty organization identifier value. The server SHOULD validate that the <orgext:id> element role does not exist.
- o An OPTIONAL <orgext:rem> element that contains one or more <orgext:id> elements, as defined in Section 3.1, that remove organization roles from the object. The <orgext:id> element MAY have an empty organization identifier value. The server SHOULD validate the existence of the <orgext:id> element role and the organization identifier if provided.
- o An OPTIONAL <orgext:chg> element that contains one or more <orgext:id> elements, as defined in Section 3.1, that change organization role identifiers for the object. The existing organization identifier value will be replaced for the defined role. The server SHOULD validate the existence of the <orgext:id> element role.

At least one <orgext:add>, <orgext:rem> or <orgext:chg> element MUST be provided.

Example <update> command, adding a reseller:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <update>
C:      <domain:update>
C:        xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
C:        <domain:name>example.com</domain:name>
C:      </domain:update>
C:    </update>
C:    <extension>
C:      <orgext:update>
C:        xmlns:orgext="urn:ietf:params:xml:ns:epp:orgext-1.0">
C:        <orgext:add>
C:          <orgext:id role="reseller">reseller1523</orgext:id>
C:        </orgext:add>
C:      </orgext:update>
C:    </extension>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

Example <update> command, adding multiple organizations:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <update>
C:      <domain:update>
C:        xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
C:        <domain:name>example.com</domain:name>
C:      </domain:update>
C:    </update>
C:    <extension>
C:      <orgext:update>
C:        xmlns:orgext="urn:ietf:params:xml:ns:epp:orgext-1.0">
C:        <orgext:add>
C:          <orgext:id role="reseller">reseller1523</orgext:id>
C:          <orgext:id role="privacyproxy">proxy2935</orgext:id>
C:        </orgext:add>
C:      </orgext:update>
C:    </extension>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```


Example <update> command, removing a reseller:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <update>
C:      <domain:update>
C:        xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
C:        <domain:name>example.com</domain:name>
C:      </domain:update>
C:    </update>
C:  <extension>
C:    <orgext:update>
C:      xmlns:orgext="urn:ietf:params:xml:ns:epp:orgext-1.0">
C:      <orgext:rem>
C:        <orgext:id role="reseller"/>
C:      </orgext:rem>
C:    </orgext:update>
C:  </extension>
C:  <clTRID>ABC-12345</clTRID>
C: </command>
C:</epp>
```

Example <update> command, removing multiple organizations:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <update>
C:      <domain:update>
C:        xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
C:        <domain:name>example.com</domain:name>
C:      </domain:update>
C:    </update>
C:  <extension>
C:    <orgext:update>
C:      xmlns:orgext="urn:ietf:params:xml:ns:epp:orgext-1.0">
C:      <orgext:rem>
C:        <orgext:id role="reseller"/>
C:        <orgext:id role="privacyproxy"/>
C:      </orgext:rem>
C:    </orgext:update>
C:  </extension>
C:  <clTRID>ABC-12345</clTRID>
C: </command>
C:</epp>
```

Example <update> command, updating reseller identifier:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <update>
C:      <domain:update>
C:        xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
C:          <domain:name>example.com</domain:name>
C:        </domain:update>
C:      </update>
C:    <extension>
C:      <orgext:update>
C:        xmlns:orgext="urn:ietf:params:xml:ns:epp:orgext-1.0">
C:          <orgext:chg>
C:            <orgext:id role="reseller">reseller1523</orgext:id>
C:          </orgext:chg>
C:        </orgext:update>
C:      </extension>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

Example <update> command, updating multiple organization identifiers:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <update>
C:      <domain:update>
C:        xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
C:          <domain:name>example.com</domain:name>
C:        </domain:update>
C:      </update>
C:    <extension>
C:      <orgext:update>
C:        xmlns:orgext="urn:ietf:params:xml:ns:epp:orgext-1.0">
C:          <orgext:chg>
C:            <orgext:id role="reseller">reseller1523</orgext:id>
C:            <orgext:id role="privacyproxy">proxy2935</orgext:id>
C:          </orgext:chg>
C:        </orgext:update>
C:      </extension>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

When an extended <update> command has been processed successfully, the EPP response is as described in the EPP object extension.

An EPP error response MUST be returned if an <update> command cannot be processed for any reason. An attempt to add one organization ID or multiple organization IDs with a particular role value when at least one of them already exists does not change the object at all. A server SHOULD notify clients that object relationships exist by sending a 2305 error response code. An attempt to remove an organization ID or multiple organization IDs with a particular role value when at least one of them does not exist does not change the object at all. A server SHOULD notify clients that object relationships does not exist by sending a 2305 error response code. An attempt to change an organization ID or multiple organization IDs with a particular role value when at least one of them does not exist does not change the object at all. A server SHOULD notify clients that object relationships does not exist by sending a 2305 error response code. Response format with error value elements is defined in Section 2.6 of [RFC5730].

5. Formal Syntax

An EPP object mapping is specified in XML Schema notation. The formal syntax presented here is a complete schema representation of the object mapping suitable for automated validation of EPP XML instances. The BEGIN and END tags are not part of the schema; they are used to note the beginning and ending of the schema for URI registration purposes.

```
BEGIN
<?xml version="1.0" encoding="UTF-8"?>

<schema
  targetNamespace="urn:ietf:params:xml:ns:epp:orgext-1.0"
  xmlns:orgext="urn:ietf:params:xml:ns:epp:orgext-1.0"
  xmlns="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified"
>

  <annotation>
    <documentation>
      Extensible Provisioning Protocol v1.0
      Organization Extension Schema v1.0
    </documentation>
  </annotation>

  <!-- Child elements found in EPP commands. -->
```

```
<element
  name="create"
  type="orgext:createType"/>
<element
  name="update"
  type="orgext:updateType"/>

<!--
  Organization identifier with required role
-->
<complexType name="orgIdType">
  <simpleContent>
    <extension base="token">
      <attribute
        name="role"
        type="token"
        use="required"/>
    </extension>
  </simpleContent>
</complexType>

<!--
  Child elements of the <orgext:create> command
  All elements must be present at time of creation
-->
<complexType name="createType">
  <sequence>
    <!-- agent identifier or the organization,
         e.g. registrar, reseller, privacy proxy, etc. -->
    <element
      name="id"
      type="orgext:orgIdType"
      maxOccurs="unbounded"/>
  </sequence>
</complexType>

<!--
  Child elements of <orgext:update> command
-->
<complexType name="updateType">
  <sequence>
    <element
      name="add"
      type="orgext:addRemChgType"
      minOccurs="0"
    />
    <element
      name="rem"
```

```
        type="orgext:addRemChgType"
        minOccurs="0"
    />
    <element
        name="chg"
        type="orgext:addRemChgType"
        minOccurs="0"
    />
</sequence>
</complexType>

<complexType name="addRemChgType">
    <sequence>
        <!-- agent identifier of the organization,
             e.g. registrar, reseller, privacy proxy, etc. -->
        <element
            name="id"
            type="orgext:orgIdType"
            maxOccurs="unbounded"/>
    </sequence>
</complexType>

<!-- Child response element -->
<element
    name="infData"
    type="orgext:infDataType"/>

<!-- <orgext:infData> response elements -->
<complexType name="infDataType">
    <sequence>
        <!-- agent identifier the organization,
             e.g. registrar, reseller, privacy proxy, etc. -->
        <element
            name="id"
            type="orgext:orgIdType"
            minOccurs="0"
            maxOccurs="unbounded"/>
    </sequence>
</complexType>

<!-- End of schema. -->
</schema>
END
```

6. Internationalization Considerations

EPP is 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.

As an extension of the EPP object mapping, the elements, element content described in this document MUST inherit the internationalization conventions used to represent higher-layer domain and core protocol structures present in an XML instance that includes this extension.

7. IANA Considerations

7.1. XML Namespace

This document uses URNs to describe XML namespaces and XML schemas conforming to a registry mechanism described in [RFC3688]. IANA is requested to assign the following URI.

Registration request for the organization extension namespace:

URI: urn:ietf:params:xml:ns:epp:orgext-1.0

Registrant Contact: IESG

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

Registration request for the organization XML schema:

URI: urn:ietf:params:xml:schema:epp:orgext-1.0

Registrant Contact: IESG

XML: See the "Formal Syntax" section of this document.

7.2. EPP Extension Registry

The EPP extension described in this document should be registered by the IANA in the EPP Extension Registry described in [RFC7451]. The details of the registration are as follows:

Name of Extension: Organization Extension for the Extensible Provisioning Protocol (EPP)

Document status: Standards Track

Reference: RFCXXXX (please replace "XXXX" with the RFC number for this document after a number is assigned by the RFC Editor)

Registrant Name and Email Address: IESG, iesg@ietf.org

TLDs: Any

IPR Disclosure: None

Status: Active

Notes: None

8. Implementation Status

Note to RFC Editor: Please remove this section and the reference to [RFC7942] before publication. This section records the status of known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft, and is based on a proposal described in [RFC7942]. The description of implementations in this section is intended to assist the IETF in its decision processes in progressing drafts to RFCs. Please note that the listing of any individual implementation here does not imply endorsement by the IETF. Furthermore, no effort has been spent to verify the information presented here that was supplied by IETF contributors. This is not intended as, and must not be construed to be, a catalog of available implementations or their features. Readers are advised to note that other implementations may exist.

According to [RFC7942], "this will allow reviewers and working groups to assign due consideration to documents that have the benefit of running code, which may serve as evidence of valuable experimentation and feedback that have made the implemented protocols more mature. It is up to the individual working groups to use this information as they see fit".

8.1. Verisign EPP SDK

Organization: Verisign Inc.

Name: Verisign EPP SDK

Description: The Verisign EPP SDK includes both a full client implementation and a full server stub implementation of draft-ietf-regext-org-ext.

Level of maturity: Development

Coverage: All aspects of the protocol are implemented.

Licensing: GNU Lesser General Public License

Contact: jgould@verisign.com

URL: https://www.verisign.com/en_US/channel-resources/domain-registry-products/epp-sdks

8.2. CNNIC Implementation

Organization: CNNIC

Name: Organization Extension for EPP

Description: CNNIC is trying to update organization extension from previous reseller extension according to this document.

Level of maturity: Development

Coverage: Organization extension for EPP

Contact: zhouguiqing@cnnic.cn

9. Security Considerations

The object mapping extension described in this document does not provide any other security services or introduce any additional considerations beyond those described by [RFC5730], [RFC5731], [RFC5732] and [RFC5733] or those caused by the protocol layers used by EPP.

10. Acknowledgment

The authors would like to thank Rik Ribbers, Marc Groeneweg, Patrick Mevzek, Antoin Verschuren and Scott Hollenbeck for their careful review and valuable comments.

11. References

11.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC5730] Hollenbeck, S., "Extensible Provisioning Protocol (EPP)", STD 69, RFC 5730, DOI 10.17487/RFC5730, August 2009, <<https://www.rfc-editor.org/info/rfc5730>>.
- [RFC5731] Hollenbeck, S., "Extensible Provisioning Protocol (EPP) Domain Name Mapping", STD 69, RFC 5731, DOI 10.17487/RFC5731, August 2009, <<https://www.rfc-editor.org/info/rfc5731>>.
- [RFC5732] Hollenbeck, S., "Extensible Provisioning Protocol (EPP) Host Mapping", STD 69, RFC 5732, DOI 10.17487/RFC5732, August 2009, <<https://www.rfc-editor.org/info/rfc5732>>.
- [RFC5733] Hollenbeck, S., "Extensible Provisioning Protocol (EPP) Contact Mapping", STD 69, RFC 5733, DOI 10.17487/RFC5733, August 2009, <<https://www.rfc-editor.org/info/rfc5733>>.
- [RFC7942] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", BCP 205, RFC 7942, DOI 10.17487/RFC7942, July 2016, <<https://www.rfc-editor.org/info/rfc7942>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [W3C.REC-xml-20040204]
Bray, T., Paoli, J., Sperberg-McQueen, C., Maler, E., and F. Yergeau, "Extensible Markup Language (XML) 1.0 (Third Edition)", World Wide Web Consortium FirstEdition REC-xml-20040204, February 2004, <<http://www.w3.org/TR/2004/REC-xml-20040204>>.
- [W3C.REC-xmlschema-1-20041028]
Thompson, H., Beech, D., Maloney, M., and N. Mendelsohn, "XML Schema Part 1: Structures Second Edition", World Wide Web Consortium Recommendation REC-xmlschema-1-20041028, October 2004, <<http://www.w3.org/TR/2004/REC-xmlschema-1-20041028>>.

[W3C.REC-xmlschema-2-20041028]

Biron, P. and A. Malhotra, "XML Schema Part 2: Datatypes Second Edition", World Wide Web Consortium Recommendation REC-xmlschema-2-20041028", October 2004, <<http://www.w3.org/TR/2004/REC-xmlschema-2-20041028>>.

11.2. Informative References

[ID.draft-ietf-dnsop-terminology-bis]

Hoffman, P., Sullivan, A., and K. Fujiwara, "DNS Terminology", September 2018, <<http://tools.ietf.org/html/draft-ietf-dnsop-terminology-bis>>.

[ID.draft-ietf-regext-org]

Zhou, L., Kong, N., Zhou, G., Yao, J., and J. Gould, "Extensible Provisioning Protocol (EPP) Reseller Mapping", November 2018, <<http://tools.ietf.org/html/draft-ietf-regext-org>>.

[RFC7451]

Hollenbeck, S., "Extension Registry for the Extensible Provisioning Protocol", RFC 7451, DOI 10.17487/RFC7451, February 2015, <<https://www.rfc-editor.org/info/rfc7451>>.

Appendix A. Change Log

Initial -00: Individual document submitted.

-01:

- * Updated abstract and introduction.
- * Revised typos in info response.
- * Added explanations on how to process reseller extension after successful transfer operation.
- * Modified <update> explanation.
- * Deleted reseller name element in <create> and <update> commands.
- * Removed some inaccurate comments from xml schema.
- * Modified the element name of reseller id and reseller name.

-02:

- * Changed author information.

- * Updated xml typos <reseller:infData> to <resellerext:infData> in <info> response.

-03:

- * Changed author information.
- * Updated section 3.1.
- * Removed reseller name element in <info> response.
- * Added acknowledgment.
- * Revised the typo "resellerr" to "resellerext".

WG document-00: WG document submitted

WG document-01: Keep document alive for further discussion. The requirement of reseller information is clear for both registrar and registry. What we should reach a consensus is whether the extension should support only a name or ID and name.

Organization WG document-00: Change to a generic organization object extension.

Organization WG document-01: Added "Implementation Status" section.

Organization WG document-02: Accepted some of the feedbacks on the mailing list. Modified the examples in the document.

Organization WG document-03:

- * Updated typos.
- * Changed some descriptions about <orgext:id> and role attribute.
- * Modified the example of "domain with no organization".
- * Updated section 8, adding implementation status of Verisign.

Organization WG document-04:

- * Updated typos.
- * Removed the example of <update> command, domain with no organization.
- * Updated references.

- * Updated section 8 of implementation status.

Organization WG document-05:

- * Removed the minOccurs="0" from the addRemChgType type of the XML schema
- * Removed the third paragraph of "Implementation Status".
- * Remove the Informative Reference to draft-ietf-regext-reseller-ext from the draft.

Organization WG document-06:

- * Updated "Abstraction".
- * Added "Query" for "<Transfer> Query Command".
- * Change "Registrant Contact" to IESG in section 7.1.
- * Modified section 7.2.

Organization WG document-07:

- * Updated "Abstraction".

Organization WG document-08:

- * Updated error codes of <update> response.
- * Modified XML namespace and schema.

Organization WG document-09:

- * Modified XML namespace and schema.
- * Changed "Exactly one" to "At least one" in section 4.2.5.

Organization WG document-10:

- * Updated the reseller id and dns proxy id in the document.
- * Updated the full xml namespace.
- * Updated the text of EPP <orgext:add>, <orgext:rem> and <orgext:chg>.

- * Added "Document satus" and "Reference" in section "EPP Extension Registry".

Organization WG document-11:

- * Added the reference of draft-ietf-dnsop-terminology-bis.

Authors' Addresses

Linlin Zhou
CNNIC
4 South 4th Street, Zhongguancun, Haidian District
Beijing, Beijing 100190
China

Email: zhoulinlin@cnnic.cn

Ning Kong
Consultant

Email: ietfing@gmail.com

Junkai Wei
CNNIC
4 South 4th Street, Zhongguancun, Haidian District
Beijing, Beijing 100190
China

Email: weijunkai@cnnic.cn

Jiankang Yao
CNNIC
4 South 4th Street, Zhongguancun, Haidian District
Beijing, Beijing 100190
China

Email: yaojk@cnnic.cn

James Gould
Verisign, Inc.
12061 Bluemont Way
Reston, VA 20190
US

Email: jgould@verisign.com

Registration Protocols Extensions
Internet-Draft
Updates: 7484 (if approved)
Intended status: Best Current Practice
Expires: February 4, 2019

S. Hollenbeck
Verisign Labs
A. Newton
ARIN
August 3, 2018

Registration Data Access Protocol (RDAP) Object Tagging
draft-ietf-regext-rdap-object-tag-05

Abstract

The Registration Data Access Protocol (RDAP) includes a method that can be used to identify the authoritative server for processing domain name, IP address, and autonomous system number queries. The method does not describe how to identify the authoritative server for processing other RDAP query types, such as entity queries. This limitation exists because the identifiers associated with these query types are typically unstructured. This document updates RFC 7484 by describing an operational practice that can be used to add structure to RDAP identifiers that makes it possible to identify the authoritative server for additional RDAP queries.

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 <https://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 February 4, 2019.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must 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	2
2. Object Naming Practice	3
3. Bootstrap Service Registry for Provider Object Tags	8
3.1. Registration Procedure	9
4. RDAP Conformance	10
5. IANA Considerations	10
5.1. Bootstrap Service Registry Structure	10
5.2. RDAP Extensions Registry	10
6. Implementation Status	11
6.1. Verisign Labs	11
6.2. OpenRDAP	11
7. Security Considerations	12
8. Acknowledgements	12
9. References	12
9.1. Normative References	12
9.2. Informative References	13
Appendix A. Change Log	13
Authors' Addresses	14

1. Introduction

The Registration Data Access Protocol (RDAP) includes a method ([RFC7484]) that can be used to identify the authoritative server for processing domain name, IP address, and autonomous system number (ASN) queries. This method works because each of these data elements is structured in a way that facilitates automated parsing of the element and association of the data element with a particular RDAP service provider. For example, domain names include labels (such as "com", "net", and "org") that are associated with specific service providers.

As noted in Section 9 of RFC 7484 [RFC7484], the method does not describe how to identify the authoritative server for processing entity queries, name server queries, help queries, or queries using certain search patterns. This limitation exists because the identifiers bound to these queries are typically not structured in a way that makes it easy to associate an identifier with a specific service provider. This document describes an operational practice that can be used to add structure to RDAP identifiers that makes it

possible to identify the authoritative server for additional RDAP queries.

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

2. Object Naming Practice

Tagging object identifiers with a service provider tag makes it possible to identify the authoritative server for processing an RDAP query using the method described in RFC 7484 [RFC7484]. A service provider tag is constructed by prepending the Unicode HYPHEN-MINUS character "-" (U+002D, described as an "unreserved" character in RFC 3986 [RFC3986]) to an IANA-registered value that represents the service provider. For example, a tag for a service provider identified by the string value "ARIN" is represented as "-ARIN".

In combination with the `rdapConformance` attribute described in Section 4, service provider tags are concatenated to the end of RDAP query object identifiers to unambiguously identify the authoritative server for processing an RDAP query. Building on the example from Section 3.1.5 of RFC 7482 [RFC7482], an RDAP entity handle can be constructed that allows an RDAP client to bootstrap an entity query. The following identifier is used to find information for the entity associated with handle "XXXX" at service provider "ARIN":

XXXX-ARIN

Clients that wish to bootstrap an entity query can parse this identifier into distinct handle and service provider identifier elements. Handles can themselves contain HYPHEN-MINUS characters; the service provider identifier is found following the last HYPHEN-MINUS character in the tagged identifier. The service provider identifier is used to retrieve a base RDAP URL from an IANA registry. The base URL and entity handle are then used to form a complete RDAP query path segment. For example, if the base RDAP URL "https://example.com/rdap/" is associated with service provider "YYYY" in an IANA registry, an RDAP client will parse a tagged entity identifier "XXXX-YYYY" into distinct handle ("XXXX") and service provider ("YYYY") identifiers. The service provider identifier "YYYY" is used to query an IANA registry to retrieve the base RDAP URL "https://example.com/rdap/". The RDAP query URL is formed using the base RDAP URL and entity path segment described in Section 3.1.5 of RFC 7482 [RFC7482], using "XXXX-YYY" as the value of the handle

identifier. The complete RDAP query URL becomes
"https://example.com/rdap/entity/XXXX-YYYY".

Implementation of this practice requires tagging of unstructured potential query identifiers in RDAP responses. Consider these elided examples ("..." is used to note elided response objects) from Section 5.3 of RFC 7483 [RFC7483] in which the handle identifiers have been tagged with service provider tags "RIR", "DNR", and "ABC" respectively:

```
{
  "objectClassName" : "domain",
  "handle" : "XXXX-RIR",
  "ldhName" : "0.2.192.in-addr.arpa",
  "nameservers" :
  [
    ...
  ],
  "secureDNS" :
  {
    ...
  },
  "remarks" :
  [
    ...
  ],
  "links" :
  [
    ...
  ],
  "events" :
  [
    ...
  ],
  "entities" :
  [
    {
      "objectClassName" : "entity",
      "handle" : "XXXX-RIR",
      "vcardArray":
      [
        ...
      ],
      "roles" : [ "registrant" ],
      "remarks" :
      [
        ...
      ],
    }
  ]
}
```

```

    "links" :
    [
        ...
    ],
    "events" :
    [
        ...
    ]
}
],
"network" :
{
    "objectClassName" : "ip network",
    "handle" : "XXXX-RIR",
    "startAddress" : "192.0.2.0",
    "endAddress" : "192.0.2.255",
    "ipVersion" : "v4",
    "name" : "NET-RTR-1",
    "type" : "DIRECT ALLOCATION",
    "country" : "AU",
    "parentHandle" : "YYYY-RIR",
    "status" : [ "active" ]
}
}

```

Figure 1

```

{
    "objectClassName" : "domain",
    "handle" : "XXXX-YYY-DNR",
    "ldhName" : "xn--fo-5ja.example",
    "unicodeName" : "foo.example",
    "variants" :
    [
        ...
    ],
    "status" : [ "locked", "transfer prohibited" ],
    "publicIds":
    [
        ...
    ],
    "nameservers" :
    [
        {
            "objectClassName" : "nameserver",
            "handle" : "XXXX-DNR",
            "ldhName" : "ns1.example.com",
            "status" : [ "active" ],

```

```
    "ipAddresses" :
    {
        ...
    },
    "remarks" :
    [
        ...
    ],
    "links" :
    [
        ...
    ],
    "events" :
    [
        ...
    ]
},
{
    "objectClassName" : "nameserver",
    "handle" : "XXXX-DNR",
    "ldhName" : "ns2.example.com",
    "status" : [ "active" ],
    "ipAddresses" :
    {
        ...
    },
    "remarks" :
    [
        ...
    ],
    "links" :
    [
        ...
    ],
    "events" :
    [
        ...
    ]
}
],
"secureDNS":
{
    ...
},
"remarks" :
[
    ...
],
```

```
"links" :
[
  ...
],
"port43" : "whois.example.net",
"events" :
[
  ...
],
"entities" :
[
  {
    "objectClassName" : "entity",
    "handle" : "XXXX-ABC",
    "vcardArray":
    [
      ...
    ],
    "status" : [ "validated", "locked" ],
    "roles" : [ "registrant" ],
    "remarks" :
    [
      ...
    ],
    "links" :
    [
      ...
    ],
    "events" :
    [
      ...
    ]
  }
]
```

Figure 2

As described in Section 5 of RFC 7483 [RFC7483], RDAP responses can contain "self" links. Service provider tags and self references SHOULD be consistent. If they are inconsistent, the service provider tag is processed with higher priority when using these values to identify a service provider.

There is a risk of unpredictable processing behavior if the HYPHEN-MINUS character is used for naturally occurring, non-separator purposes in an entity handle. This could lead to a client mistakenly assuming that a HYPHEN-MINUS character represents a separator and the

text that follows HYPHEN-MINUS is a service provider identifier. A client that queries the IANA registry for what they assume is a valid service provider will likely receive an unexpected, invalid result. As a consequence, use of the HYPHEN-MINUS character as a service provider tag separator MUST be noted by adding an `rdapConformance` value to query responses as described in Section 4.

The HYPHEN-MINUS character was chosen as a separator for two reasons: 1) it is a familiar separator character in operational use, and 2) it avoids collision with URI-reserved characters. The list of unreserved characters specified in Section 2.3 of RFC 3986 [RFC3986] provided multiple options for consideration:

unreserved = ALPHA / DIGIT / "-" / "." / "_" / "~"

ALPHA and DIGIT characters were excluded because they are commonly used in entity handles for non-separator purposes. HYPHEN-MINUS is commonly used as a separator and recognition of this practice will reduce implementation requirements and operational risk. The remaining characters were excluded because they are not broadly used as separators in entity handles.

3. Bootstrap Service Registry for Provider Object Tags

The bootstrap service registry for the RDAP service provider space is represented using the structure specified in Section 3 of RFC 7484 [RFC7484]. The JSON output of this registry contains contact information for the registered service provider identifiers, alphanumeric identifiers that identify RDAP service providers, and base RDAP service URLs as shown in this example.

```
{
  "version": "1.0",
  "publication": "YYYY-MM-DDTHH:MM:SSZ",
  "description": "RDAP bootstrap file for service provider object tags",
  "services": [
    [
      ["contact@example.com"],
      ["YYYY"],
      [
        "https://example.com/rdap/"
      ]
    ],
    [
      ["contact@example.org"],
      ["ZZ54"],
      [
        "http://rdap.example.org/"
      ]
    ],
    [
      ["contact@example.net"],
      ["1754"],
      [
        "https://example.net/rdap/",
        "http://example.net/rdap/"
      ]
    ]
  ]
}
```

Figure 3

Alphanumeric service provider identifiers conform to the suffix portion ("\\w{1,8}") of the "roidType" syntax specified in Section 4.2 of RFC 5730 [RFC5730].

3.1. Registration Procedure

The service provider registry is populated using the "First Come First Served" policy defined in RFC 8126 [RFC8126]. Provider identifier values can be derived and assigned by IANA on request. Registration requests include an email address to be associated with the registered service provider identifier, the requested service provider identifier (or an indication that IANA should assign an identifier), and one or more base RDAP URLs to be associated with the service provider identifier.

4. RDAP Conformance

RDAP responses that contain values described in this document MUST indicate conformance with this specification by including an `rdapConformance` ([RFC7483]) value of `"rdap_objectTag_level_0"`. The information needed to register this value in the RDAP Extensions Registry is described in Section 5.2.

Example `rdapConformance` structure with extension specified:

```
"rdapConformance" :  
[  
  "rdap_level_0",  
  "rdap_objectTag_level_0"  
]
```

Figure 4

5. IANA Considerations

IANA is requested to create the RDAP "Bootstrap Service Registry for Provider Object Tags" listed below and make it available as JSON objects. The contents of this registry is described in Section 3, with the formal syntax specified in Section 10 of RFC 7484 [RFC7484].

5.1. Bootstrap Service Registry Structure

Entries in this registry contain the following information:

- o An email address that identifies a contact associated with the registered RDAP service provider value.
- o An alphanumeric value that identifies the RDAP service provider being registered.
- o One or more URLs that provide the RDAP service regarding this registration. The URLs are expected to supply the same data, but they can differ in scheme or other components as required by the service operator.

5.2. RDAP Extensions Registry

IANA is requested to register the following value in the RDAP Extensions Registry:

```
Extension identifier: rdap_objectTag  
Registry operator: Any  
Published specification: This document.  
Contact: IESG <iesg@ietf.org>
```


Intended usage: This extension describes a best practice for structuring entity identifiers to enable query bootstrapping.

6. Implementation Status

NOTE: Please remove this section and the reference to RFC 7942 prior to publication as an RFC.

This section records the status of known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft, and is based on a proposal described in RFC 7942 [RFC7942]. The description of implementations in this section is intended to assist the IETF in its decision processes in progressing drafts to RFCs. Please note that the listing of any individual implementation here does not imply endorsement by the IETF. Furthermore, no effort has been spent to verify the information presented here that was supplied by IETF contributors. This is not intended as, and must not be construed to be, a catalog of available implementations or their features. Readers are advised to note that other implementations may exist.

According to RFC 7942, "this will allow reviewers and working groups to assign due consideration to documents that have the benefit of running code, which may serve as evidence of valuable experimentation and feedback that have made the implemented protocols more mature. It is up to the individual working groups to use this information as they see fit".

6.1. Verisign Labs

Responsible Organization: Verisign Labs
Location: <https://rdap.verisignlabs.com/>
Description: This implementation includes support for domain registry RDAP queries using live data from the .cc and .tv country code top-level domains. Client authentication is required to receive entity information in query responses.
Level of Maturity: This is a "proof of concept" research implementation.
Coverage: This implementation includes all of the features described in this specification.
Contact Information: Scott Hollenbeck, shollenbeck@verisign.com

6.2. OpenRDAP

Responsible Organization: OpenRDAP
Location: <https://www.openrdap.org>

Description: RDAP client implementing bootstrapping for entity handles with a service provider tag. A test Bootstrap Services Registry file is currently used in lieu of an official one.

Level of Maturity: Alpha

Coverage: Implements draft 04+, supports the HYPHEN-MINUS separator character only.

Contact Information: Tom Harwood, tfh@skip.org

7. Security Considerations

This practice uses IANA as a well-known, central trusted authority to allow users to get RDAP data from an authoritative source, reducing the risk of sending queries to non-authoritative sources and divulging query information to unintended parties. Using TLS [RFC5246] to protect the connection to IANA allows the server to authenticate itself as being operated by IANA and provides integrity protection for the resulting referral information, as well as providing privacy protection via data confidentiality. The subsequent RDAP connection is performed as usual, and retains the same security properties of the RDAP protocols themselves.

8. Acknowledgements

The author would like to acknowledge the following individuals for their contributions to the development of this document: Tom Harrison, Patrick Mevzek, and Marcos Sanz. In addition, the authors would like to recognize the Regional Internet Registry (RIR) operators (AFRINIC, APNIC, ARIN, LACNIC, and RIPE) that have been implementing and using the practice of tagging handle identifiers for several years. Their experience provided significant inspiration for the development of this document.

9. References

9.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC5730] Hollenbeck, S., "Extensible Provisioning Protocol (EPP)", STD 69, RFC 5730, DOI 10.17487/RFC5730, August 2009, <<https://www.rfc-editor.org/info/rfc5730>>.
- [RFC7484] Blanchet, M., "Finding the Authoritative Registration Data (RDAP) Service", RFC 7484, DOI 10.17487/RFC7484, March 2015, <<https://www.rfc-editor.org/info/rfc7484>>.

- [RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <<https://www.rfc-editor.org/info/rfc8126>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

9.2. Informative References

- [RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <<https://www.rfc-editor.org/info/rfc3986>>.
- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <<https://www.rfc-editor.org/info/rfc5246>>.
- [RFC7481] Hollenbeck, S. and N. Kong, "Security Services for the Registration Data Access Protocol (RDAP)", RFC 7481, DOI 10.17487/RFC7481, March 2015, <<https://www.rfc-editor.org/info/rfc7481>>.
- [RFC7482] Newton, A. and S. Hollenbeck, "Registration Data Access Protocol (RDAP) Query Format", RFC 7482, DOI 10.17487/RFC7482, March 2015, <<https://www.rfc-editor.org/info/rfc7482>>.
- [RFC7483] Newton, A. and S. Hollenbeck, "JSON Responses for the Registration Data Access Protocol (RDAP)", RFC 7483, DOI 10.17487/RFC7483, March 2015, <<https://www.rfc-editor.org/info/rfc7483>>.
- [RFC7942] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", BCP 205, RFC 7942, DOI 10.17487/RFC7942, July 2016, <<https://www.rfc-editor.org/info/rfc7942>>.

Appendix A. Change Log

- 00: Initial version.
- 01: Changed separator character from HYPHEN MINUS to COMMERCIAL AT. Added a recommendation to maintain consistency between service provider tags and "self" links (suggestion received from Tom Harrison). Fixed a spelling error, and corrected the network

- example in Section 2 (editorial erratum reported for RFC 7483 by Marcos Sanz). Added acknowledgements.
- 02: Changed separator character from COMMERCIAL AT to TILDE. Clarity updates and fixed an example handle. Added text to describe the risk of separator characters appearing naturally in entity handles and being misinterpreted as separator characters.
 - 03: Added Implementation Status section (Section 6).
 - 04: Keepalive refresh.
 - 05: Added OpenRDAP implementation information to Section 6.
 - 00: Initial working group version.
 - 01: Added text to describe why the TILDE character was chosen as the separator character.
 - 02: Nit fixes. Added rdapConformance text, switched back to HYPHEN MINUS, and added IANA registration instructions per working group last call discussion. Updated suffix syntax reference from the IANA EPP ROID registry to RFC 5730 (which is what the IANA registry references).
 - 03: Shepherd writeup review updates to explain examples in Section 2.
 - 04: AD review update to clarify query path construction.
 - 05: IESG review update: object naming practice, revised an example to include multiple separator HYPHEN-MINUS characters, revised security considerations, revised IANA considerations, revised IANA registry description and registration procedure to add email address contact information.

Authors' Addresses

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

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

Andrew Lee Newton
American Registry for Internet Numbers
PO Box 232290
Centreville, VA 20120
US

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