Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: June 19, 2021

G. Lozano
ICANN
J. Gould
C. Thippeswamy
VeriSign

Dec 16, 2020

Domain Name Registration Data (DNRD) Objects Mapping
draft-ietf-regext-dnrd-objects-mapping-11

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 June 19, 2021.

Copyright Notice

Copyright (c) 2020 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 5
  2.1. XML Model 5
  2.2. CSV Model 6
3. Terminology 6
  3.1. Glossary 6
4. Conventions Used in This Document 7
  4.1. Date and Time 7
  4.2. Country names 8
  4.3. Telephone numbers 8
  4.4. CSV Integrity Check 8
  4.5. IP addresses 8
  4.6. Conventions applicable to the CSV Model 8
5. Object Description 17
  5.1. Domain Name Object 17
  5.2. Host Object 36
  5.3. Contact Object 46
  5.4. Registrar Object 64
  5.5. IDN Table Reference Object 72
  5.6. NNDN Object 75
  5.7. EPP Parameters Object 80
  5.8. Policy Object 82
  5.9. Header Object 82
  5.10. DNRD Common Objects Collection 85
6. RDE IDN Variants handling 85
7. Profile 86
8. Data escrow agent extended verification process 86
9. Formal Syntax 87
  9.1. RDE CSV Schema 87
  9.2. RDE Domain Object 97
  9.3. CSV Domain Object 100
  9.4. RDE Host Object 103
  9.5. CSV Host Object 105
  9.6. RDE Contact Object 107
  9.7. CSV Contact Object 110
  9.8. RDE Registrar Object 116
  9.9. CSV Registrar Object 119
  9.10. RDE IDN Table Reference Objects 122
  9.11. CSV IDN Language Object 123
  9.12. EPP Parameters Object 124
  9.13. NNDN Object 125
  9.14. CSV NNDN Object 127
  9.15. Policy Object 129
  9.16. Header Object 130
  9.17. DNRD Common Objects 132
10. Internationalization Considerations 132

11. IANA Considerations 132
12. Implementation Status 140
12.1. Implementation in the gTLD space 141
13. Security Considerations 141
14. Privacy Considerations 142
15. Acknowledgments 142
16. Change History 143
16.1. Changes from draft-arias-noguchi-registry-data-escrow-02 to -dnrd-objects-mapping-00 143
16.2. Changes from 00 to 01 143
16.3. Changes from 01 to 02 144
16.4. Changes from 02 to 03 144
16.5. Changes from 03 to 04 144
16.6. Changes from 04 to 05 145
16.7. Changes from 05 to 06 146
16.8. Changes from 06 to 07 146
16.9. Changes from 07 to 08 147
16.10. Changes from 08 to 09 147
16.11. Changes from 09 to 10 147
16.12. Changes from 10 to REGEXT 00 147
16.13. Changes REGEXT 00 to REGEXT 01 147
16.14. Changes REGEXT 01 to REGEXT 02 147
16.15. Changes REGEXT 02 to REGEXT 03 149
16.16. Changes REGEXT 03 to REGEXT 04 149
16.17. Changes REGEXT 04 to REGEXT 05 150
16.18. Changes REGEXT 05 to REGEXT 06 150
16.19. Changes REGEXT 06 to REGEXT 07 150
16.20. Changes REGEXT 07 to REGEXT 08 150
16.21. Changes REGEXT 08 to REGEXT 09 151
16.22. Changes REGEXT 09 to REGEXT 10 151
16.23. Changes REGEXT 10 to REGEXT 11 151
17. Example of a Full Deposit using the XML model 151
18. Example of Differential Deposit using the XML model 157
19. Example of a Full Deposit using the CSV model 158
20. Example of Differential Deposit using the CSV model 168
21. References 178
21.1. Normative References 178
21.2. Informative References 181
Authors' Addresses 182

1. Introduction

Registry Data Escrow (RDE) is the process by which a 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 cannot 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 also the users of services relying on the registry data.

In the context of domain name registries, registration data escrow is a requirement for generic top-level domains (e.g., Specification 2 of the ICANN Base Registry Agreement, see [ICANN-GTLD-RA-20170731]) 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 defines the standard set of objects for a Domain Name Registry that uses the Registry Data Escrow Specification described in [I-D.ietf-regext-data-escrow] for escrow. The set of objects include:

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

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

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

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

- NNDN (NNDN's not domain name): Domain Name Registries may maintain domain names without being persisted as domain objects in the registry system, for example, a list of reserved names not available for registration. The NNDN is a lightweight domain-like object that is used to escrow domain names not maintained as domain name objects.

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: Contains the 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.

Extensible Markup Language (XML) 1.0 as described in [W3C.REC-xml-20081126] and XML Schema notation as described in [W3C.REC-xmlschema-1-20041028] and [W3C.REC-xmlschema-2-20041028] are used in this specification.

2. Models

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

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

This document does not suggest the use of a particular model, and both are equivalent. A Domain Name Registry may choose the model that is more appropriate for the peculiarities of its systems. For example, a registry may use the CSV-export functionality of the Relational Database Management System (RDBMS) for escrow; therefore, the CSV model may be more appropriate. Another registry may use the code developed for EPP to implement escrow.

2.1. XML Model

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).

2.2. CSV Model

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).

3. Terminology

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.

3.1. Glossary

In the following section, the most common terms are briefly explained:

- 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.

- Comma-Separated Values (CSV), see [RFC4180].

- Domain name: see definition of Domain name in [RFC8499].

- Extensible Provisioning Protocol (EPP), see definition of the Extensible Provisioning Protocol in [RFC8499].

- Fully-Qualified Domain Name (FQDN), see definition of FQDN in [RFC8499].

- Internationalized Domain Name (IDN), see definition of Internationalized Domain Name in [RFC8499].

- Label: see definition of Label in [RFC8499].

- Registrant: see definition of Registrant in [RFC8499].

- Registrar: see definition of Registrar in [RFC8499].

- o Registry: see definition of Registry in [RFC8499].

- o 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.

- o Registry Data Escrow (RDE): registry data escrow is the process by which a 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 cannot function and is unable or unwilling to facilitate an orderly transfer of service.

- o 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; responding to queries for contact and other information concerning DNS registrations in the RCDN; and 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.

- o SRS: Shared Registration System, see also [ICANN-GTLD-AGB-20120604].

- o Top-Level Domain Name (TLD), see definition of Top-Level Domain in [RFC8499].

- o UTC: Coordinated Universal Time, as maintained by the Bureau International des Poids et Mesures (BIPM); see also [RFC3339].

4. Conventions Used in This Document

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 0x2B), followed by a country code defined in [ITU-E164], followed by a dot (".", ASCII value 0x2E), followed by a sequence of digits representing the telephone number.

4.4. CSV Integrity Check

A checksum MAY be used to verify the integrity of the CSV files, for example, if another layer (i.e., encryption of an archive containing the deposit files) does not provide integrity. By default the CRC32 algorithm (see, 8.1.1.6.2 of [V42]) is used. A stronger algorithm, such as SHA-256 (see, [RFC6234]) MAY be used for enhanced security if required.

4.5. IP addresses

The syntax of IP addresses MUST conform to the text representation of either Internet Protocol Version 4 [RFC0791] or Internet Protocol Version 6 [RFC5952].

4.6. Conventions applicable to the CSV Model

4.6.1. 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 (see [I-D.ietf-regext-data-escrow]).

The following is an example of the CSV file definitions, using the element <rdeCsv:csv> (see Section 4.6.2.1), 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>
```

4.6.2. CSV elements

4.6.2.1. <rdeCsv:csv> element

To support the CSV model, 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 are 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 4.6.1). 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. For example, setting this attribute to "gzip" signals that the CSV file is compressed using the GZIP file format (see, [RFC1952]). The supported compression formats are negotiated out-of-band.

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

cksum Defines the checksum of the CSV file, as described in Section 4.4, using the algorithm defined by the "cksumAlg" attribute. If the "cksumAlg" attribute is not present, the checksum is calculated using "CRC32".

cksumAlg Defines the checksum algorithm used to calculate the "cksum" attribute, with the default value of "CRC32". If the value "SHA256" is specified, the SHA-256 algorithm (see, [RFC6234]) MUST be used to calculate the "cksum" attribute. Parties receiving and processing data escrow deposits MUST support CRC32 and SHA-256. If this attribute is present, the "cksum" attribute MUST also be present. Additional checksum algorithms are negotiated out-of-band.

The <rdeCsv:csv> element 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 need for quoting/escaping of the CSV data could be avoided by choosing a separator character that is not in the data set of the CSV files.

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.example,Ddomain2-TEST,,,registrantid,registrarX,registrarX,
clientY,2009-04-03T22:00:00.0Z,registrarX,clientY,
2009-12-03T09:05:00.0Z,2025-04-03T22:00:00.0Z
```

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.example
domain2.example
domainN.example
```

4.6.2.2. CSV common field elements

The <rdeCsv:fields> element defined in the <rdeCsv:csv> element (Section 4.6.2.1) 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 the 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 the client that created the object with type="eppcom:clIDType".

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

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

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

<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".

4.6.3. Internationalized and Localized Elements

Some elements MAY be provided in either internationalized form ("int") or 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 internalized 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 isRequired="false"/>
      <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. 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. <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], Registry Grace Period (RGP) status from [RFC3915], and data from the EPP <secDns:create> command, see Section 5.2.1. of [RFC5910].

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

The <domain> element contains the following child elements:

- A <name> element that contains the fully-qualified name of the domain name object. For IDNs the A-Label is used (see [RFC5891], Section 4.4).

- A <roid> element that contains the repository object identifier assigned to the domain name object when it was created.

- An OPTIONAL <uName> element that contains the fully-qualified domain name in Unicode character set. It MUST be provided if available.

- An OPTIONAL <idnTableId> element that references the IDN Table used for the IDN. This corresponds to the "id" attribute of the <idnTableRef> element. This element MUST be present if the domain name is an IDN.

- An OPTIONAL <originalName> element is used to indicate that the domain name is an IDN variant. This element contains the domain name used to generate the IDN variant.

- One or more <status> elements that contain the current status descriptors associated with the domain name.

- Zero or more OPTIONAL <rgpStatus> elements to represent "pendingDelete" sub-statuses, including "redemptionPeriod",

"pendingRestore", and "pendingDelete", that a domain name can be in as a result of grace period processing as specified in [RFC3915].

- An OPTIONAL <registrant> element that contains the identifier for the human or organizational social information object associated as the holder of the domain name object.

- Zero or more OPTIONAL <contact> elements that contain identifiers for the human or organizational social information objects associated with the domain name object.

- An OPTIONAL <ns> element that contains the fully-qualified names of the delegated host objects or host attributes (name servers) associated with the domain name object. See Section 1.1 of [RFC5731] for a description of the elements used to specify host objects or host attributes.

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

- 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.

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

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

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

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

- 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 name 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 name object:

```
...
<rdeDomain:domain>
  <rdeDomain:name>xn--exampl-gva.example</rdeDomain:name>
  <rdeDomain:roid>Dexample1-TEST</rdeDomain:roid>
  <rdeDomain:idnTableId>pt-BR</rdeDomain:idnTableId>
  <rdeDomain:originalName>example.example</rdeDomain:originalName>
  <rdeDomain:status s="ok"/>
  <rdeDomain:registrant>jd1234</rdeDomain:registrant>
  <rdeDomain:contact type="admin">sh8013</rdeDomain:contact>
  <rdeDomain:contact type="tech">sh8013</rdeDomain:contact>
  <rdeDomain:ns>
    <domain:hostObj>ns1.example.com</domain:hostObj>
    <domain:hostObj>ns1.example1.example</domain:hostObj>
  </rdeDomain:ns>
  <rdeDomain:clID>RegistrarX</rdeDomain:clID>
  <rdeDomain:crRr client="jdoe">RegistrarX</rdeDomain:crRr>
  <rdeDomain:crDate>1999-04-03T22:00:00.0Z</rdeDomain:crDate>
  <rdeDomain:exDate>2025-04-03T22:00:00.0Z</rdeDomain:exDate>
</rdeDomain: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.example</rdeDomain:name>
    <rdeDomain:name>bar.example</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" and "csvRegistrar" fields, 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> or <csvRegistrar:fGurid> A choice of:

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

<csvRegistrar:fGurid> Contains the Globally Unique Registrar Identifier (GURID) assigned by ICANN with type="positiveInteger" and isRequired="true".

The following "rdeCsv" fields, defined in section CSV common field elements (Section 4.6.2.2), 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 domain name object.

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

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

<rdeCsv:fUpID> Identifier of the client that last updated the domain name 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:fCrDate> Created date and time of the domain name object.

<rdeCsv:fUpDate> Date and time of the last update to the domain name object. This field MUST NOT be set if the domain name object has never been modified.

<rdeCsv:fExDate> Expiration date and time for the domain name object.

<rdeCsv:fTrDate> Date and time of the last transfer for the domain name object. This field MUST NOT be set if the domain name object has never been transferred.

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.example,Ddomain1-TEST,,,registrantid,registrarX,registrarX,
clientY,2009-04-03T22:00:00.0Z,registrarX,clientY,
2009-12-03T09:05:00.0Z,2025-04-03T22:00:00.0Z
domain2.example,Ddomain2-TEST,,,registrantid,registrarX,registrarX,
clientY,1999-04-03T22:00:00.0Z,registrarX,clientY,
2009-12-03T09:05:00.0Z,2025-04-03T22:00:00.0Z
xn--bc123-3ve.example,Dxnabc123-TEST,LANG-1,,registrantid,registrarX,
registrarX,clientY,2009-04-03T22:00:00.0Z,registrarX,clientY,
2009-12-03T09:05:00.0Z,2025-04-03T22:00:00.0Z
xn--bc321-3ve.example,Dxnabc321-TEST,LANG-1,xn--bc123-3ve.example,
registrantid,registrarX,registrarX,clientY,2009-04-03T22:00:00.0Z,
registrarX,clientY,2009-12-03T09:05:00.0Z,2025-04-03T22:00:00.0Z
```

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, 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".

<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 "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/>
    <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.example, domain2.example, xn--bc123-3ve.example and xn--bc321-3ve.example.

```
domain1.example,domain1admin,admin
domain1.example,domain1tech,tech
domain1.example,domain1billing,billing
domain2.example,domain2admin,admin
domain2.example,domain2tech,tech
domain2.example,domain2billing,billing
xn--bc123-3ve.example,xnabc123admin,admin
xn--bc123-3ve.example,xnabc123tech,tech
xn--bc123-3ve.example,xnabc123billing,billing
xn--bc321-3ve.example,xnabc123admin,admin
xn--bc321-3ve.example,xnabc123tech,tech
xn--bc321-3ve.example,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" 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".

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

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

The following "rdeCsv" fields, defined in section CSV common field elements (Section 4.6.2.2), MAY be used in the "domainStatuses" <rdeCsv:csv> <rdeCsv:fields> element:

<rdeCsv:fStatusDescription> Domain name 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.example, domain2.example, xn--bc123-3ve.example and xn--bc321-3ve.example.

```
domain1.example,clientUpdateProhibited,"Disallow update",
en,
domain1.example,clientDeleteProhibited,"Disallow delete",
en,
domain2.example,ok,,,
xn--bc123-3ve.example,ok,,,
xn--bc321-3ve.example,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 isRequired="true".

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

<csvHost:fName> or <rdeCsv:fRoid> A choice of:

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

- <rdeCsv:fRoid> Host object Registry Object IDentifier (ROID) assigned to the host object 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/>
  </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.example, domain2.example, xn--bc123-3ve.example and xn--bc321-3ve.example referenced via the <rdeCsv:fRoid> field element.

```
domain1.example,Hns1_domain1_test-TEST
domain1.example,Hns2_domain1_test-TEST
domain2.example,Hns1_domain2_test-TEST
domain2.example,Hns2_domain2_test-TEST
xn--bc123-3ve.example,Hns1_example_test-TEST
xn--bc123-3ve.example,Hns2_example_test-TEST
xn--bc321-3ve.example,Hns1_example_test-TEST
xn--bc321-3ve.example,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 domain host 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.example, domain2.example, xn--bc123-3ve.example and xn--bc321-3ve.example.

```
domain1.example,ns1.domain1.example,192.0.2.1,v4
domain1.example,ns2.domain1.example,2001:DB8::1,v6
domain2.example,ns1.example.net,,
domain2.example,ns2.example.net,,
xn--bc123-3ve.example,ns1.example.net,,
xn--bc123-3ve.example,ns2.example.net,,
xn--bc321-3ve.example,ns1.example.net,,
xn--bc321-3ve.example,ns2.example.net,,
```

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 name 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.example.

```
domain1.example,604800,30730,8,2,91C9B176EB////F1C46F6A55
domain1.example,604800,61882,8,2,9F8FEAC94B////1272AF09F3
```

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 parent="true"/>
    <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.example.

```
domain1.example,604800,257,3,8,AwEAAZD1+z////G1jqviK8c=
domain1.example,604800,257,3,8,AwEAAbntWP////vwDitt940=
```

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 4.6.2.2), 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 4.6.2.2), 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 the client that requested the transfer.

<rdeCsv:fAcID> Identifier of the 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 name 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.example,pending,registrarX,clientY,
2011-03-08T19:38:00.0Z,registrarY,,2011-03-13T23:59:59.0Z,
2025-04-03T22:00:00.0Z
```

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.example
domain2.example
```

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:

- A <name> element that contains the fully-qualified name of the host object.

- A <roid> element that contains the repository object identifier assigned to the host object when the object was created.

- One or more <status> elements that describe the status of the host object.

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

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

- An OPTIONAL <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.

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

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

- 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 transfered.

Example of <host> object:

```
...
<rdeHost:host>
  <rdeHost:name>ns1.example1.example</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">2001:DB8:1::1</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.example</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 4.6.2.2), 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" and "csvRegistrar" fields, MAY be used in the "host" <rdeCsv:csv> <rdeCsv:fields> element:

<rdeCsv:fClID> or <csvRegistrar:fGurid> A choice of:

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

<csvRegistrar:fGurid> Contains the Globally Unique Registrar Identifier (GURID) assigned by ICANN with type="positiveInteger" and isRequired="true".

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

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

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

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

<rdeCsv:fCrDate> Date and time that the host object was created.

<rdeCsv:fUpDate> Date and time that the host object was last updated. This field MUST NOT be set if the domain name object has never been modified.

<rdeCsv:fTrDate> Date and time that the host object was last transferred. This field MUST NOT be set if the domain name object has never been 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.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.domain1.example,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.example,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.example,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.net,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.net,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 4.6.2.2), 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 4.6.2.2), 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.

<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.example, ns2.domain1.example, ns1.domain2.example, ns2.domain2.example, ns1.example.net and ns2.example.net.

```
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 4.6.2.2), MUST be used in the "hostAddresses" <rdeCsv:csv> <rdeCsv:fields> element:

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

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 hostAddresses-YYYYMMDD.csv file. The file contains the IP addresses for the host names ns1.domain1.example, ns2.domain1.example, ns1.domain2.example and ns2.domain2.example.

```
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 4.6.2.2), 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 An OPTIONAL <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 An OPTIONAL <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: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.example</rdeContact:email>
  <rdeContact:clID>RegistrarX</rdeContact:clID>
  <rdeContact:crRr client="jdoe">RegistrarX</rdeContact:crRr>
  <rdeContact:crDate>2009-09-13T08:01:00.0Z</rdeContact:crDate>
  <rdeContact:upRr client="jdoe">RegistrarX</rdeContact:upRr>
  <rdeContact:upDate>2009-11-26T09:10:00.0Z</rdeContact:upDate>
  <rdeContact:trDate>2009-12-03T09:05:00.0Z</rdeContact:trDate>
  <rdeContact:trnData>
    <rdeContact:trStatus>pending</rdeContact:trStatus>
    <rdeContact:reRr client="jstiles">clientW</rdeContact:reRr>
    <rdeContact:reDate>2011-03-08T19:38:00.0Z</rdeContact:reDate>
    <rdeContact:acRr client="rmiles">RegistrarX</rdeContact:acRr>
    <rdeContact:acDate>2011-03-13T23:59:59.0Z</rdeContact:acDate>
  </rdeContact:trnData>
  <rdeContact:disclose flag="0">
    <contact:voice/>
    <contact:email/>
  </rdeContact:disclose>
</rdeContact:contact>
...
```

5.3.1.2. <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:e164StringType".

<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:e164StringType".

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

The following "rdeCsv" and "csvRegistrar" fields, 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> or <csvRegistrar:fGurid> A choice of:

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

<csvRegistrar:fGurid> Contains the Globally Unique Registrar Identifier (GURID) assigned by ICANN with type="positiveInteger" and isRequired="true".

The following "rdeCsv" fields, defined in section CSV common field elements (Section 4.6.2.2), MAY be used in the "contact" <rdeCsv:csv> <rdeCsv:fields> element:

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

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

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

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

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

<rdeCsv:fUpDate> Date and time of the last update to the contact object. This field MUST NOT be set if the domain name object has never been modified.

<rdeCsv:fTrDate> Date and time of the last transfer for the contact object. This field MUST NOT be set if the domain name object has never been transferred.

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.

```
domain1admin,Cdomain1admin-TEST,+1.7035555555,1234,
+1.7035555556,,jdoe@example.example,registrarX,registarX,
clientY,2009-09-13T08:01:00.0Z,registarX,clientY,
2009-11-26T09:10:00.0Z
domain1tech,Cdomain1tech-TEST,+1.7035555555,1234,
+1.7035555556,,jdoe@example.example,registrarX,registarX,
clientY,2009-09-13T08:01:00.0Z,registarX,clientY,
2009-11-26T09:10:00.0Z
domain1billing,Cdomain1billing-TEST,+1.7035555555,1234,
+1.7035555556,,jdoe@example.example,registrarX,registarX,
clientY,2009-09-13T08:01:00.0Z,registarX,clientY,
2009-11-26T09:10:00.0Z
domain2admin,Cdomain2admin-TEST,+1.7035555555,1234,
+1.7035555556,,jdoe@example.example,registrarX,registarX,
clientY,2009-09-13T08:01:00.0Z,registarX,clientY,
2009-11-26T09:10:00.0Z
domain2tech,Cdomain2tech-TEST,+1.7035555555,1234,
+1.7035555556,,jdoe@example.example,registrarX,registarX,
clientY,2009-09-13T08:01:00.0Z,registarX,clientY,
2009-11-26T09:10:00.0Z
domain2billing,Cdomain2billing-TEST,+1.7035555555,1234,
+1.7035555556,,jdoe@example.example,registrarX,registarX,
clientY,2009-09-13T08:01:00.0Z,registarX,clientY,
2009-11-26T09:10:00.0Z
xnabc123admin,Cxnabc123admin-TEST,+1.7035555555,1234,
+1.7035555556,,jdoe@example.example,registrarX,registarX,
clientY,2009-09-13T08:01:00.0Z,registarX,clientY,
2009-11-26T09:10:00.0Z
xnabc123tech,Cxnabc123tech-TEST,+1.7035555555,1234,
+1.7035555556,,jdoe@example.example,registrarX,registarX,
clientY,2009-09-13T08:01:00.0Z,registarX,clientY,
2009-11-26T09:10:00.0Z
xnabc123billing,Cxnabc123billing-TEST,+1.7035555555,1234,
+1.7035555556,,jdoe@example.example,registrarX,registarX,
clientY,2009-09-13T08:01:00.0Z,registarX,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" and parent="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 4.6.2.2), 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 4.6.3, 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 is used to indicate the localized or internationalized form as defined in Section 4.6.3.

<csvContact:fStreet> Contains the 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 incrementing for each line up to index "2" for the third line. An OPTIONAL "isLoc" attribute is used to indicate the localized or internationalized form as defined in Section 4.6.3.

<csvContact:fCity> Contains the contact's city with type="contact:postalLineType" and isRequired="true". An OPTIONAL "isLoc" attribute is used to indicate the localized or internationalized form as defined in Section 4.6.3.

<csvContact:fCc> Contains the contact's country code with type="contact:ccType" and isRequired="true". An OPTIONAL "isLoc" attribute is used to indicate the localized or internationalized form as defined in Section 4.6.3.

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 is used to indicate the localized or internationalized form as defined in Section 4.6.3.

<csvContact:fSp> Contains the contact's state or province with type="contact:optPostalLineType". An OPTIONAL "isLoc" attribute is used to indicate the localized or internationalized form as defined in Section 4.6.3.

<csvContact:fPc> Contains the contact's postal code with type="contact:pcType". An OPTIONAL "isLoc" attribute is used to indicate the localized or internationalized form as defined in Section 4.6.3.

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" and parent="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.

```
domain1admin,int,"John Doe","Example Inc.",
"123 Example Dr.","Suite 100",,Reston,VA,20190,US
domain1tech,int,"John Doe","Example Inc.",
"123 Example Dr.","Suite 100",,Reston,VA,20190,US
domain1billing,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 4.6.2.2), 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 4.6.2.2), MAY be used in the "contactTransfer" <rdeCsv:csv> <rdeCsv:fields> element:

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

<rdeCsv:fAcID> Identifier of the 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,0,0,0,0,0,0,0,1,1,1
```

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>
...
</csvContact: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, and is typically referred to as the sponsoring registrar. 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:

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

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

- An OPTIONAL <gurid> element that contains the Globally Unique Registrar Identifier (GURID) assigned by ICANN.

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

- o One 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 registrar-object modification. This element MUST NOT be present if the registrar-object has never been modified.

Example of a <registrar> object:

```
...
<rdeRegistrar:registrar>
  <rdeRegistrar:id>RegistrarX</rdeRegistrar:id>
  <rdeRegistrar:name>Registrar X</rdeRegistrar:name>
  <rdeRegistrar:gurid>8</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.example</rdeRegistrar:email>
  <rdeRegistrar:url>http://www.example.example</rdeRegistrar:url>
  <rdeRegistrar:whoisInfo>
    <rdeRegistrar:name>whois.example.example</rdeRegistrar:name>
    <rdeRegistrar:url>http://whois.example.example</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> or <csvRegistrar:fGurid> A choice of:

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

<csvRegistrar:fGurid> Contains the Globally Unique Registrar Identifier (GURID) assigned by ICANN with type="positiveInteger" 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". This field is included in this section in addition to the section above to support optionally providing the <csvRegistrar:fGurid> field when the <csvRegistrar:fId> field is used.

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

The following "rdeCsv" fields, defined in section CSV common field elements (Section 4.6.2.2), 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. This field MUST NOT be set if the domain name object has never been modified.

<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 4.6.3.

<csvContact:fCity> Registrar city with an OPTIONAL "isLoc" attribute that is used to indicate the localized or internationalized form, as defined in Section 4.6.3.

<csvContact:fCc> Registrar country code with an OPTIONAL "isLoc" attribute that is used to indicate the localized or internationalized form, as defined in Section 4.6.3.

<csvContact:fEmail> Registrar email address. The attribute "isRequired" MUST equal "false".

<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 4.6.3.

<csvContact:fPc> Registrar postal code with an OPTIONAL "isLoc" attribute that is used to indicate the localized or internationalized form, as defined in Section 4.6.3.

<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 isRequired="false"/>
      <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 one registrar record.

```
registrarX,"Example Inc.",8,ok,"123 Example Dr.",
"Suite 100",,Dulles,VA,20166-6503,US,+1.7035555555,1234,
+1.7035555556,,jdoe@example.example,http://www.example.example,
http://whois.example.example,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> or <csvRegistrar:fGurid> A choice of:

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

<csvRegistrar:fGurid> Contains the Globally Unique Registrar Identifier (GURID) assigned by ICANN with type="positiveInteger". The attribute "isRequired" MUST equal "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.

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

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

Example of <idnTableRef> object:

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

5.5.2. 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 4.6.2.2, 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 <csvDomain:fName> field in the "domain" CSV File Definition Section 5.1.2.1.1 files. The attribute "isRequired" MUST equal "true".

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

```
...
<csvIDN:contents>
...
  <rdeCsv:csv name="idnLanguage" 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

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

Domain Name Registries may maintain domain names without their being persisted as domain objects in the registry system, for example, a list of reserved names not available for registration. The NNDN is a lightweight domain-like object that is used to escrow domain names not maintained as domain name objects.

A domain name can only exist as a domain name object or an 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 an 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:

- An <aName> element that contains the fully-qualified qualified name of the NNDN. For IDNs the A-Label is used (see [RFC5891], Section 4.4).

- An OPTIONAL <uName> element that contains the fully-qualified name of the NNDN in Unicode character set. It MUST be provided if available.

- An OPTIONAL <idnTableId> element that references the IDN Table used for the NNDN. This corresponds to the "id" attribute of the <idnTableRef> element. This element MUST be present if the NNDN is an IDN.

- An OPTIONAL <originalName> element is used to indicate that the NNDN is used for an IDN variant. This element contains the domain name used to generate the IDN variant.

- A <nameState> element that indicates the state of the NNDN: blocked, withheld or mirrored.

  * If an NNDN is considered undesirable for registration (i.e., unavailable for allocation to anyone), then the NNDN will be tagged as "blocked".

  * If an NNDN is considered a potential registration of a domain name 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 an NNDN is considered a mirrored IDN variant of a domain name object, then the NNDN will be tagged as "mirrored". A mirroringNS attribute is used to specify if the mirrored IDN variant uses the NS mirror mechanism, meaning that the activated variant domain name (i.e., NNDN) is delegated in the DNS using the same NS records as in the <originalName>. The default value of mirroringNS is true. If another mechanism

such as DNAME is used, the value of mirroringNS attribute MUST be false.

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

Example of an <rdeNNDN:NNDN> object:

```
...
<rdeNNDN:NNDN>
  <rdeNNDN:aName>xn--exampl-gva.example</rdeNNDN:aName>
  <rdeNNDN:idnTableId>pt-BR</rdeNNDN:idnTableId>
  <rdeNNDN:originalName>example.example</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 NNDN that was deleted, i.e., the <aName>.

Example of an <rdeNNDN::delete> object:

```
...
<rde:deletes>
  ...
  <rdeNNDN:delete>
    <rdeNNDN:aName>xn--pingino-q2a.example</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:fAName> Fully-qualified name of the NNDN with type="eppcom:labelType" and isRequired="true". For IDNs the A-Label is used (see [RFC5891], Section 4.4).

<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 4.6.2.2), MAY be used in the "NNDN" <rdeCsv:csv> <rdeCsv:fields> element:

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

<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 an "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--bc456-3ve.example,LANG-1,xn--bc123-3ve.example,
blocked,,2005-04-23T11:49:00.0Z
xn--bc789-3ve.example,LANG-1,xn--bc123-3ve.example,
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> Fully-qualified name of the NNDN with type="eppcom:labelType" and isRequired="true".

Example of an "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--bc456-3ve.example
```

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. A maximum of 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:

- One or more <version> elements that indicate the EPP versions supported by the registry.

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

- 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 4.6.2.1, 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:

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

- <scope> that defines the XPath (see, [W3C.REC-xpath-31-20170321]) of the element referenced by <element>.

Example of <rdePolicy:policy> object:

```
...
<rdePolicy:policy scope="//rde:deposit/rde:contents/rdeDomain:domain"
  element="rdeDomain: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 Deposit. The Header Object may also be used to provide additional information on the contents of the deposit. 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:

- 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 RCDN being escrowed in the case of a Registry data escrow deposit. For IDNs the A-Label is used (see [RFC5891], Section 4.4).

* 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.

* A <rdeHeader:reseller> element that defines the provider ID corresponding to a Reseller data escrow deposit.

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 RCDN of the objects included in the <count> element. For IDNs the A-Label is used [RFC5891], Section 4.4. 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.

- o An OPTIONAL <contentTag> element that contains a tag that defines the expected content in the deposit. The producer and consumer of the deposits will coordinate the set of possible <contentTag> element values.

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>
...
```

5.10. DNRD Common Objects Collection

The DNRD Common Objects Collection contains data structures referenced by two or more of the main objects in the XML model.

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 an NNDN object, but not both.

If using domain objects to represent IDN variants, the normal behavior during restoration of an 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.

An 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 for defining 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 some 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

A Data Escrow Agent SHOULD perform an extended verification process that starts by creating a dataset to be tested by following section 5.2 in [I-D.ietf-regext-data-escrow].

The following are the minimum suggested tests on the dataset:

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 No domain name exists as both a domain name and an NNDN.

- 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

This standard is specified in XML Schema notation. The formal syntax presented here is a complete schema representation suitable for automated validation.

The <CODE BEGINS> and <CODE ENDS> tags are not part of the schema; they are used to note the beginning and ending of the schema for URI registration purposes.

9.1. RDE CSV Schema

```
<CODE BEGINS>
<?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" />
  <import namespace="urn:ietf:params:xml:ns:rde-1.0" />
  <annotation>
    <documentation>
```

```
      Registry Data Escrow Comma-Separated 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" />
        <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 />
          <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 the 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:fClIDType"
             substitutionGroup="rdeCsv:field" />
    <!-- Identifier of the 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 the 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:fClIDRequiredType"
              substitutionGroup="rdeCsv:field" />
     <!-- Identifier of the 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 the 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>
     <!-- dateTime fields (fCrDate, fUpDate, fExDate) -->
     <element name="fCrDate"
              type="rdeCsv:fDateTimeType"
              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>
    <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>
      <!-- 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" />
            <attribute name="cksumAlg"
                       type="token"
                       default="CRC32" />
          </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>
      <CODE ENDS>
```

9.2. RDE Domain Object

```
<CODE BEGINS>
<?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:rdeDnrdCommon="urn:ietf:params:xml:ns:rdeDnrdCommon-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" />
  <import namespace="urn:ietf:params:xml:ns:rdeDnrdCommon-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" />
            <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="rdeDnrdCommon:rrType"
                     minOccurs="0" />
            <element name="crDate"
                     type="dateTime"
                     minOccurs="0" />
            <element name="exDate"
                     type="dateTime"
                     minOccurs="0" />
            <element name="upRr"
                     type="rdeDnrdCommon: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="rdeDnrdCommon:rrType" />
        <element name="reDate"
                 type="dateTime" />
        <element name="acRr"
                 type="rdeDnrdCommon: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>
  <CODE ENDS>
```

9.3. CSV Domain Object

```
<CODE BEGINS>
<?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" />
  <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:rdeCsv-1.0" />
  <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>
  <!--
  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 />
           <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" />
    <!-- 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>
  <CODE ENDS>
```

9.4. RDE Host Object

```
<CODE BEGINS>
<?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:rdeDnrdCommon="urn:ietf:params:xml:ns:rdeDnrdCommon-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" />
  <import namespace="urn:ietf:params:xml:ns:rdeDnrdCommon-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="rdeDnrdCommon:rrType"
                     minOccurs="0" />
            <element name="crDate"
                     type="dateTime"
                     minOccurs="0" />
            <element name="upRr"
                     type="rdeDnrdCommon: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>
  </schema>
  <CODE ENDS>
```

9.5. CSV Host Object

```
   <CODE BEGINS>
   <?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" />
     <import namespace="urn:ietf:params:xml:ns:host-1.0" />
     <import namespace="urn:ietf:params:xml:ns:rde-1.0" />
     <import namespace="urn:ietf:params:xml:ns:rdeCsv-1.0" />
     <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>
<!-- 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>
    <!--
    End of schema.
    -->
  </schema>
  <CODE ENDS>
```

9.6. RDE Contact Object

```
<CODE BEGINS>
<?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:rdeDnrdCommon="urn:ietf:params:xml:ns:rdeDnrdCommon-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" />
  <import namespace="urn:ietf:params:xml:ns:rdeDnrdCommon-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:e164Type"
                     minOccurs="0" />
            <element name="fax"
                     type="contact:e164Type"
                     minOccurs="0" />
            <element name="email"
                     type="eppcom:minTokenType" />
            <element name="clID"
                     type="eppcom:clIDType" />
            <element name="crRr"
                     type="rdeDnrdCommon:rrType"
                     minOccurs="0" />
            <element name="crDate"
                     type="dateTime"
                     minOccurs="0" />
            <element name="upRr"
                     type="rdeDnrdCommon: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="rdeDnrdCommon:rrType" />
        <element name="reDate"
                 type="dateTime" />
        <element name="acRr"
                 type="rdeDnrdCommon: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>
  <CODE ENDS>
```

9.7. CSV Contact Object

```
<CODE BEGINS>
<?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" />
  <import namespace="urn:ietf:params:xml:ns:contact-1.0" />
  <import namespace="urn:ietf:params:xml:ns:rde-1.0" />
  <import namespace="urn:ietf:params:xml:ns:rdeCsv-1.0" />
  <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 />
           <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>
  <CODE ENDS>
```

9.8. RDE Registrar Object

```
  <CODE BEGINS>
  <?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:e164Type"
                     minOccurs="0" />
            <element name="fax"
                     type="contact:e164Type"
                     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>
  <CODE ENDS>
```

9.9. CSV Registrar Object

```
  <CODE BEGINS>
  <?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" />
      <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" />
      <import namespace="urn:ietf:params:xml:ns:rdeCsv-1.0" />
      <annotation>
        <documentation>
        Registar 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:fieldOptionalType">
           <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>
  <CODE ENDS>
```

9.10. RDE IDN Table Reference Objects

```
  <CODE BEGINS>
  <?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>
    <!-- Simple Types -->
    <simpleType name="idType">
      <restriction base="token">
        <minLength value="1" />
        <maxLength value="64" />
      </restriction>
    </simpleType>
  </schema>
  <CODE ENDS>
```

9.11. CSV IDN Language Object

```
  <CODE BEGINS>
  <?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" />
    <import namespace="urn:ietf:params:xml:ns:rdeCsv-1.0" />
    <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>
  <CODE ENDS>
```

9.12. EPP Parameters Object

```
  <CODE BEGINS>
  <?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>
  <CODE ENDS>
```

9.13. NNDN Object

```
  <CODE BEGINS>
  <?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"
                       minOccurs="0" />
            </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"
                     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>
  <CODE ENDS>
```

9.14. CSV NNDN Object

```
  <CODE BEGINS>
  <?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" />
    <import namespace="urn:ietf:params:xml:ns:rdeCsv-1.0" />
```

```
<import namespace="urn:ietf:params:xml:ns:rdeNNDN-1.0" />
<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>
<!-- A-Label format 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" />
<!-- 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>
   <CODE ENDS>
```

9.15. Policy Object

```
<CODE BEGINS>
<?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>
<CODE ENDS>
```

9.16. Header Object

```
<CODE BEGINS>
<?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" />
            <element name="contentTag"
                     type="token"
                     minOccurs="0" />
          </sequence>
        </extension>
      </complexContent>
    </complexType>
    <group name="repositoryTypeGroup">
      <choice>
        <element name="tld"
                 type="eppcom:labelType" />
        <element name="registrar"
                 type="positiveInteger" />
        <element name="ppsp"
                 type="token" />
        <element name="reseller"
                 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>
  <CODE ENDS>
```

9.17. DNRD Common Objects

```
<CODE BEGINS>
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="urn:ietf:params:xml:ns:rdeDnrdCommon-1.0"
        xmlns:rdeDnrdCommon="urn:ietf:params:xml:ns:rdeDnrdCommon-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" />
  <annotation>
    <documentation>
      Data Escrow Deposit Common Objects schema
    </documentation>
  </annotation>
  <complexType name="rrType">
    <simpleContent>
      <extension base="eppcom:clIDType">
        <attribute name="client"
                   type="eppcom:clIDType" />
      </extension>
    </simpleContent>
  </complexType>
</schema>
<CODE ENDS>
```

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]. The following URI assignments is requested of IANA.

Registration request for the RDE CSV namespace:

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

Registrant Contact: IESG <regext@ietf.org>

Note to RFC Editor: Please remove the email address from the RFC after IANA records it.

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: IESG <regext@ietf.org>

Note to RFC Editor: Please remove the email address from the RFC after IANA records it.

See Section 9.1 of this document.

Registration request for the RDE domain namespace:

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

Registrant Contact: IESG <regext@ietf.org>

Note to RFC Editor: Please remove the email address from the RFC after IANA records it.

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: IESG <regext@ietf.org>

Note to RFC Editor: Please remove the email address from the RFC after IANA records it.

See Section 9.2 of this document.

Registration request for the CSV domain namespace:

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

Registrant Contact: IESG <regext@ietf.org>

Note to RFC Editor: Please remove the email address from the RFC after IANA records it.

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: IESG <regext@ietf.org>

Note to RFC Editor: Please remove the email address from the RFC after IANA records it.

See Section 9.3 of this document.

Registration request for the RDE host namespace:

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

Registrant Contact: IESG <regext@ietf.org>

Note to RFC Editor: Please remove the email address from the RFC after IANA records it.

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: IESG <regext@ietf.org>

Note to RFC Editor: Please remove the email address from the RFC after IANA records it.

See Section 9.4 of this document.

Registration request for the CSV host namespace:

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

Registrant Contact: IESG <regext@ietf.org>

Note to RFC Editor: Please remove the email address from the RFC after IANA records it.

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: IESG <regext@ietf.org>

Note to RFC Editor: Please remove the email address from the RFC after IANA records it.

See Section 9.5 of this document.

Registration request for the RDE contact namespace:

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

Registrant Contact: IESG <regext@ietf.org>

Note to RFC Editor: Please remove the email address from the RFC after IANA records it.

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: IESG <regext@ietf.org>

Note to RFC Editor: Please remove the email address from the RFC after IANA records it.

See Section 9.6 of this document.

Registration request for the CSV contact namespace:

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

Registrant Contact: IESG <regext@ietf.org>

Note to RFC Editor: Please remove the email address from the RFC after IANA records it.

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: IESG <regext@ietf.org>

Note to RFC Editor: Please remove the email address from the RFC after IANA records it.

See Section 9.7 of this document.

Registration request for the RDE registrar namespace:

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

Registrant Contact: IESG <regext@ietf.org>

Note to RFC Editor: Please remove the email address from the RFC after IANA records it.

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: IESG <regext@ietf.org>

Note to RFC Editor: Please remove the email address from the RFC after IANA records it.

See Section 9.8 of this document.

Registration request for the CSV registrar namespace:

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

Registrant Contact: IESG <regext@ietf.org>

Note to RFC Editor: Please remove the email address from the RFC after IANA records it.

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: IESG <regext@ietf.org>

Note to RFC Editor: Please remove the email address from the RFC after IANA records it.

See Section 9.9 of this document.

Registration request for the RDE IDN namespace:

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

Registrant Contact: IESG <regext@ietf.org>

Note to RFC Editor: Please remove the email address from the RFC after IANA records it.

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: IESG <regext@ietf.org>

Note to RFC Editor: Please remove the email address from the RFC after IANA records it.

See Section 9.10 of this document.

Registration request for the CSV IDN namespace:

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

Registrant Contact: IESG <regext@ietf.org>

Note to RFC Editor: Please remove the email address from the RFC after IANA records it.

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: IESG <regext@ietf.org>

Note to RFC Editor: Please remove the email address from the RFC after IANA records it.

See Section 9.11 of this document.

Registration request for the RDE EPP parameters namespace:

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

Registrant Contact: IESG <regext@ietf.org>

Note to RFC Editor: Please remove the email address from the RFC after IANA records it.

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: IESG <regext@ietf.org>

Note to RFC Editor: Please remove the email address from the RFC after IANA records it.

See Section 9.12 of this document.

Registration request for the RDE NNDN namespace:

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

Registrant Contact: IESG <regext@ietf.org>

Note to RFC Editor: Please remove the email address from the RFC after IANA records it.

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: IESG <regext@ietf.org>

Note to RFC Editor: Please remove the email address from the RFC after IANA records it.

See Section 9.13 of this document.

Registration request for the CSV NNDN namespace:

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

Registrant Contact: IESG <regext@ietf.org>

Note to RFC Editor: Please remove the email address from the RFC after IANA records it.

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: IESG <regext@ietf.org>

Note to RFC Editor: Please remove the email address from the RFC after IANA records it.

See Section 9.14 of this document.

Registration request for the RDE Policy namespace:

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

Registrant Contact: IESG <regext@ietf.org>

Note to RFC Editor: Please remove the email address from the RFC after IANA records it.

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

Registration request for the RDE Policy XML schema:

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

Registrant Contact: IESG <regext@ietf.org>

Note to RFC Editor: Please remove the email address from the RFC after IANA records it.

See Section 9.15 of this document.

Registration request for the RDE Header namespace:

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

Registrant Contact: IESG <regext@ietf.org>

Note to RFC Editor: Please remove the email address from the RFC after IANA records it.

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

Registration request for the RDE Header XML schema:

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

Registrant Contact: IESG <regext@ietf.org>

Note to RFC Editor: Please remove the email address from the RFC after IANA records it.

See Section 9.16 of this document.

Registration request for the RDE Common Objects namespace:

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

Registrant Contact: IESG <regext@ietf.org>

Note to RFC Editor: Please remove the email address from the RFC after IANA records it.

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

Registration request for the RDE Common Objects XML schema:

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

Registrant Contact: IESG <regext@ietf.org>

Note to RFC Editor: Please remove the email address from the RFC after IANA records it.

See Section 9.17 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 - 09 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 parties SHOULD take all the necessary precautions such as encrypting the data at rest and in transit to avoid inadvertent disclosure of private data. Regardless of the precautions taken by the parties regarding

data at rest and in transit, authentication credentials MUST NOT be escrowed.

Authentication of the parties passing data escrow deposit files is also of the utmost importance. The escrow agent MUST properly authenticate the registry's identity before accepting data escrow deposits. The registry MUST authenticate the escrow agent's identity before submitting any data, and the data escrow agent MUST authenticate the identity of the party receiving the data escrow deposits for the purposes deemed appropriate.

Additionally, the registry and the escrow agent MUST use integrity checking mechanisms to ensure the data transmitted is what the source intended. Validation of the contents by the parties is RECOMMENDED to ensure that the file was transmitted correctly from the registry or escrow agent and that the contents are "meaningful".

A few elements in this specification contain URLs, the use of HTTP over TLS (Transport Layer Security), [RFC2818] is RECOMMENDED on the URLs.

The various data structures in the document include a few places that have internal redundancy, and if the values become inconsistent there can be harmful consequences, such as different entities using different fields as their reference.

Note: if Transport Layer Security (TLS) is used when providing an escrow services, the recommendations in [BCP195] MUST be implemented.

14. Privacy Considerations

This specification defines a format that may be used to escrow personal data. The process of data escrow is governed by a legal document agreed by the parties, and such legal document must ensure that privacy-sensitive and/or personal data receives the required protection.

15. Acknowledgments

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

Special suggestions that have been incorporated into this document were provided by 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, Bhadresh Modi, Alexander Mayrhofer and Benjamin Kaduk.

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

16. Change History

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

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

1. Added definition for child elements under the <domain> element.

2. Added definition for child elements under the <host> element.

3. Added definition for child elements under the <contact> element.

4. Rewrote the IDN Variants Handling section to use the variant states as described in ICANN's Study of Issues Related to the Management of IDN Variant TLDs.

5. Renamed <icannID> to <gurid> in the <rdeRegistrar>.

6. Renamed <dnssec> to <secDNS> in the <domain> element.

7. Renamed <transfData> to <trnData> in the <domain> element.

8. Added <whoisInfo> element under <rdeRegistrar> element.

9. Fixed some typographical errors and omissions.

16.2. Changes from 00 to 01

1. Specify OPTIONAL elements in the draft.

2. Added NNDN object to support list of reserved names and different IDN variants models.

3. Removed subordinated host element from the domain object.

4. Added eppParams object.

5. Added variantGenerator element to the domain object.

6. Added lgr to the IDN table object.

16.3. Changes from 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.

16.4. Changes from 02 to 03

1. Remove authinfo from the XML Schema.

2. Resend attribute is now an element

3. Scope attribute added to policy object.

16.5. Changes from 03 to 04

1. Merged draft-gould-thippeswamy-dnrd-csv-mapping-03 into draft-arias-noguchi-dnrd-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.

16.6. Changes from 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.

16.7. Changes from 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.

16.8. Changes from 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

16.9. Changes from 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.

16.10. Changes from 08 to 09

1. Implementation Status section was added

16.11. Changes from 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.

16.12. Changes from 10 to REGEXT 00

1. Internet Draft (I-D) adopted by the REGEXT WG.

16.13. Changes REGEXT 00 to REGEXT 01

1. Added the <rdeHeader:reseller> element to the "repositoryTypeGroup" group element in the rdeHeader.

2. Privacy consideration section was added

3. Updates on section 8

16.14. Changes REGEXT 01 to REGEXT 02

1. Added a choice between the use of the <rdeCsv:fClID> or <csvRegistrar:fGurid> fields in the CSV "domain", "host", and "contact" definitions.

2. Added a choice between the use of the <rdeCsv:fRoid> or <csvHost:fName> fields in the CSV "domainNameServers" definition.

3. Changed "of client" to "of the client" throughout the document.

4. Modified all references of 'The attribute isRequired MUST equal "true".' to 'The attribute "isRequired" MUST equal "true".'

5. Combined the <csvDomain:fName> and <csvDomain:fContactType> fields in a single required list for the CSV "domainContacts" definition.

6. Combined the <csvDomain:fName>, <csvDomain:fStatus>, and <csvDomain:fRgpStatus> fields in a single required list for the CSV "domainStatuses" definition.

7. Moved the <rdeCsv:fCrRr> the <rdeCsv:fUpRr> fields to the MAY list for the CSV "domain", "host", and "contact" definitions.

8. Made the order of the <rdeCsv:fCrRr>, <rdeCsv:crID>, <rdeCsv:UpRr>, and <rdeCsv:UpID> fields more consistent in the CSV lists.

9. Fixed an error in the order of the <contact> object example.

10. Changed <rdeCsv:fCrDate> to be optional to match <crDate> being optional in the XML model, by having it use type rdeCsv:fDateTimeType instead of rdeCsv:fRequiredDateTimeType and ensuring that <rdeCsv:fCrDate> is included in the MAY field lists and not the MUST field lists.

11. Made <rdeCsv:fExDate> optional for the "domain" CSV definition to be consistent with the XML model, by removing the sentence 'The attribute "isRequired" MUST equal "true".' from the description and moving the field to the MAY field list.

12. Made <rdeCsv:fUpDate> optional for the "domain" and "contact" CSV definitions to be consistent with the XML model, by moving the field to the MAY field list.

13. Made <rdeCsv:fCrRr> optional to be consistent with the XML model, by having it use type rdeCsv:fClIDType instead of rdeCsv:fClIDRequiredType.

14. Made <rdeCsv:fReRr> required to be consistent with the XML model, by having it use type rdeCsv:fClIDRequiredType instead of rdeCsv:fClIDType.

15. Made the <csvRegistrar:fGurid> field in the "host", "contact", and "registrar" CSV definitions required explicitly by removing 'and isRequired="true"' and adding the sentence 'The attribute isRequired MUST equal "true".', when it is chosen as the primary field.

16. Removed extra '/>.' at the end of the <csvHost:fStatus> field description in the "hostStatuses" CSV definition.

17. Made the <csvRegistrar:fStatus> field optional to be consistent with the XML model, by having csvRegistrar:fStatusType extend rdeCsv:fieldOptionalType instead of rdeCsv:fRequiredType.

18. Made the <csvContact:fEmail> field for the "registrar" CSV definition explicitly optional to be consistent with the XML model, by adding the sentence 'The attribute isRequired MUST equal "false".' to the field description and including the definition of isRequired="false" in the "registrar" CSV definition examples.

19. Added the choice between the use of the <csvRegistrar:fId> and <csvRegistrar:fGurid> fields in the deletes "registrar" CSV definition to be consistent with the "registrar" CSV definition.

20. Made the <crRr> and <crDate> elements optional for the host and contact objects in the XML model to be consistent with the domain object.

16.15. Changes REGEXT 02 to REGEXT 03

1. Added the optional element contentTag in the header object.

2. Editorial updates.

16.16. Changes REGEXT 03 to REGEXT 04

1. Note: Updates from version REGEXT 03 to REGEXT 04 attend the feedback provided during the document shepherd review.

2. Editorial updates.

3. Examples now use domain names from the .example TLD.

4. The introduction was enhanced by explaining the need for data escrow and the proposed solution.

5. Explanation regarding NNDN was improved.

6. Explanation regarding the CSV and XML model was improved.

7. Section 4.5 updated to make the text clearer.

8. draft-arias-noguchi-registry-data-escrow is now referenced from the I-D repository.

9. The XML prefix "rdeDomain" is now consistently used.

10. The prevID attribute was removed from the examples of full deposits.

11. The examples were updated to use present dates.

16.17. Changes REGEXT 04 to REGEXT 05

1. draft-ietf-regext-data-escrow (version 04) is now referenced from the I-D repository.

2. The example in idnLanguage CSV file definition updated to use the sep attribute.

3. The reference in the example in hostAddresses CSV file definition was updated.

4. Moved [RFC0791] and [RFC5952] to the Normative References section.

16.18. Changes REGEXT 05 to REGEXT 06

1. Changes based on the feedback provided here: https://mailarchive.ietf.org/arch/msg/regext/nA8eTYIrXJ44_6ullQlRLW6T74s

16.19. Changes REGEXT 06 to REGEXT 07

1. Changes based on the feedback provided here: https://mailarchive.ietf.org/arch/msg/regext/hDLz2ym4oR-ukA4Fm-QJ8FzaxxE

2. Changes based on the feedback provided here: https://mailarchive.ietf.org/arch/msg/regext/780Xw-z1RMRb79nmZ6ABmRTo1fU

16.20. Changes REGEXT 07 to REGEXT 08

1. Changes based on the feedback provided here: https://mailarchive.ietf.org/arch/msg/regext/UaMNvl1xh60ldjpqHHYc3TNsfhg

2. Changes based on the feedback provided here: https://mailarchive.ietf.org/arch/msg/regext/B3QTxUCWUE4R_QharAQlA3041j0

16.21. Changes REGEXT 08 to REGEXT 09

1. Changes based on the feedback provided here: https://mailarchive.ietf.org/arch/msg/regext/EmKW32exlPgLbBUIbS8OjdYUJWc

16.22. Changes REGEXT 09 to REGEXT 10

1. Changes based on the feedback provided here: https://mailarchive.ietf.org/arch/msg/regext/tmoKLAV6jhh2zp4JczjeWdr_jJE

2. Changes based on the feedback provided here: https://mailarchive.ietf.org/arch/msg/regext/m7gyDTjHuRqIQCuKMHF-OLSS99k

3. Changes based on the feedback provided here: https://mailarchive.ietf.org/arch/msg/regext/3Acx5KHfeUdxZbx6A7zgoZHxIto

4. Changes based on the feedback provided here: https://mailarchive.ietf.org/arch/msg/regext/3Acx5KHfeUdxZbx6A7zgoZHxIto

5. Changes based on the feedback provided here: https://mailarchive.ietf.org/arch/msg/regext/7JiP2fzOr8KCnzI2rwoP-_KlxZY

6. Changes based on the feedback provided here: https://mailarchive.ietf.org/arch/msg/regext/dbuyW5YTYj4VcFHUQYC-D8OMv_g

7. Changes based on the feedback provided here: https://mailarchive.ietf.org/arch/msg/regext/ExUZenwC81ZQe9x24-8IKT_FWm8

16.23. Changes REGEXT 10 to REGEXT 11

1. Changes based on the feedback provided here: https://mailarchive.ietf.org/arch/msg/regext/ghEr55r7CVdwUSvkvMGpol4aSh0

17. 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="20191017001"
```

```
     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:rdeDomain="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>2019-10-17T00:00:00Z</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>

        <!-- Domain: example1.example -->
        <rdeDomain:domain>
          <rdeDomain:name>example1.example</rdeDomain:name>
          <rdeDomain:roid>Dexample1-TEST</rdeDomain:roid>
          <rdeDomain:status s="ok"/>
          <rdeDomain:registrant>jd1234</rdeDomain:registrant>
          <rdeDomain:contact type="admin">sh8013</rdeDomain:contact>
          <rdeDomain:contact type="tech">sh8013</rdeDomain:contact>
          <rdeDomain:ns>
            <domain:hostObj>ns1.example.com</domain:hostObj>
            <domain:hostObj>ns1.example1.example</domain:hostObj>
          </rdeDomain:ns>
          <rdeDomain:clID>RegistrarX</rdeDomain:clID>
          <rdeDomain:crRr client="jdoe">RegistrarX</rdeDomain:crRr>
          <rdeDomain:crDate>1999-04-03T22:00:00.0Z</rdeDomain:crDate>
          <rdeDomain:exDate>2025-04-03T22:00:00.0Z</rdeDomain:exDate>
        </rdeDomain:domain>

        <!-- Domain: example2.example -->
        <rdeDomain:domain>
          <rdeDomain:name>example2.example</rdeDomain:name>
          <rdeDomain:roid>Dexample2-TEST</rdeDomain:roid>
          <rdeDomain:status s="ok"/>
          <rdeDomain:status s="clientUpdateProhibited"/>
          <rdeDomain:registrant>jd1234</rdeDomain:registrant>
          <rdeDomain:contact type="admin">sh8013</rdeDomain:contact>
          <rdeDomain:contact type="tech">sh8013</rdeDomain:contact>
          <rdeDomain:clID>RegistrarX</rdeDomain:clID>
          <rdeDomain:crRr>RegistrarX</rdeDomain:crRr>
          <rdeDomain:crDate>1999-04-03T22:00:00.0Z</rdeDomain:crDate>
          <rdeDomain:exDate>2025-04-03T22:00:00.0Z</rdeDomain:exDate>
        </rdeDomain:domain>
```

```
<!-- Host: ns1.example.example -->
<rdeHost:host>
  <rdeHost:name>ns1.example1.example</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">2001:DB8:1::1</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.example
  </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>8</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.example
          </rdeRegistrar:email>
          <rdeRegistrar:url>http://www.example.example
          </rdeRegistrar:url>
          <rdeRegistrar:whoisInfo>
            <rdeRegistrar:name>whois.example.example
            </rdeRegistrar:name>
            <rdeRegistrar:url>http://whois.example.example
            </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.example -->
      <rdeNNDN:NNDN>
        <rdeNNDN:aName>xn--exampl-gva.example</rdeNNDN:aName>
        <rdeNNDN:idnTableId>pt-BR</rdeNNDN:idnTableId>
        <rdeNNDN:originalName>example1.example</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="rdeDomain:registrant" />
    </rde:contents>
  </rde:deposit>
```

18. 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="20191017002" prevId="20191017001"
  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:rdeDomain="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>2019-10-17T00:00:00Z</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>
      <rdeDomain:delete>
        <rdeDomain:name>example2.example</rdeDomain:name>
      </rdeDomain: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>
```

19. 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="20191017001">
      <rde:watermark>2019-10-18T00:00:00Z</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>
        </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 isRequired="false"/>
              <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. 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="20191017001" prevId="20191010001">
   <rde:watermark>2019-10-18T00:00:00Z</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>
          <rdeCsv:csv name="contact">
            <rdeCsv:fields>
              <csvContact:fId/>
            </rdeCsv:fields>
            <rdeCsv:files>
              <rdeCsv:file
                cksum="6F2B988F">
                contact-delete-YYYYMMDD.csv
              </rdeCsv:file>
            </rdeCsv:files>
          </rdeCsv:csv>
        </csvContact: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>
        </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: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 isRequired="false"/>
              <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>
```

21. References

21.1. Normative References

[BCP195] Sheffer, Y., Holz, R., and P. Saint-Andre, "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", BCP 195, RFC 7525, DOI 10.17487/RFC7525, May 2015, <https://www.rfc-editor.org/info/bcp195>.

[I-D.ietf-regext-data-escrow] Lozano, G., "Registry Data Escrow Specification", draft-ietf-regext-data-escrow-10 (work in progress), June 2020.

[ISO-3166-1]
3166, I. S., "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.

[RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <https://www.rfc-editor.org/info/rfc791>.

[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>.

[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>.

[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>.

[RFC5952] Kawamura, S. and M. Kawashima, "A Recommendation for IPv6 Address Text Representation", RFC 5952, DOI 10.17487/RFC5952, August 2010, <https://www.rfc-editor.org/info/rfc5952>.

[RFC6234] Eastlake 3rd, D. and T. Hansen, "US Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF)", RFC 6234, DOI 10.17487/RFC6234, May 2011, <https://www.rfc-editor.org/info/rfc6234>.

[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>.

[RFC8499] Hoffman, P., Sullivan, A., and K. Fujiwara, "DNS Terminology", BCP 219, RFC 8499, DOI 10.17487/RFC8499, January 2019, <https://www.rfc-editor.org/info/rfc8499>.

[V42] International Telecommunication Union, "V.42 : Error-correcting procedures for DCEs using asynchronous-to-synchronous conversion", March 2002, <https://www.itu.int/rec/T-REC-V.42/en>.

[W3C.REC-xml-20081126] Bray, T., Paoli, J., Sperberg-McQueen, C., Maler, E., and F. Yergeau, "Extensible Markup Language (XML) 1.0 (Fifth Edition) REC-xml-20081126", November 2008, <https://www.w3.org/TR/2008/REC-xml-20081126/>.

[W3C.REC-xmlschema-1-20041028] Thompson, H., Beech, D., Maloney, M., and N. Mendelsohn, "XML Schema Part 1: Structures Second Edition REC-xmlschema-1-20041028", October 2004, <https://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 REC-xmlschema-2-20041028", October 2004, <https://www.w3.org/TR/2004/REC-xmlschema-2-20041028/>.

[W3C.REC-xpath-31-20170321]
Robie, J., Dyck, M., and J. Spiegel, "XML Path Language (XPath) 3.1", March 2017, <https://www.w3.org/TR/2017/REC-xpath-31-20170321/>.

21.2. Informative References

[ICANN-GTLD-AGB-20120604]
ICANN, "gTLD Applicant Guidebook Version 2012-06-04", June 2012, <http://newgtlds.icann.org/en/applicants/agb/guidebook-full-04jun12-en.pdf>.

[ICANN-GTLD-RA-20170731]
ICANN, "Base Registry Agreement 2017-07-31", July 2017, <https://newgtlds.icann.org/sites/default/files/agreements/agreement-approved-31jul17-en.pdf>.

[RFC1952] Deutsch, P., "GZIP file format specification version 4.3", RFC 1952, DOI 10.17487/RFC1952, May 1996, <https://www.rfc-editor.org/info/rfc1952>.

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, DOI 10.17487/RFC2818, May 2000, <https://www.rfc-editor.org/info/rfc2818>.

[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>.

[RFC4180] Shafranovich, Y., "Common Format and MIME Type for Comma-Separated Values (CSV) Files", RFC 4180, DOI 10.17487/RFC4180, October 2005, <https://www.rfc-editor.org/info/rfc4180>.

[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>.

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