Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 10, 2012

F. Arias
ICANN
S. Noguchi
JPRS
March 9, 2012

Registry Data Escrow Specification
draft-arias-noguchi-registry-data-escrow-03

Abstract

This document specifies the format and contents of Data Escrow deposits targeted primarly for Domain Name Registries. However, the specification was designed to be independent of the underlying objects that are being escrowed, therefore it could be used for other than Domain Name Registries.

Status of this Memo

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

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

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

This Internet-Draft will expire on September 10, 2012.

Copyright Notice

Table of Contents

1. Introduction

Registry Data Escrow is the process by which an Internet 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 could not function and was 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 registration organization are not just those registering information there, but all relying parties that need to identify the owners of objects.

In the context of domain name registries, registration data escrow is a requirement for the current generic top-level domains and it is expected to be for new registries. Some country code top-level domain managers are also currently escrowing data. There is also a similar requirement for ICANN's generic top-level domain accredited registrars.

This document specifies a format for Data Escrow deposits independent of the objects being escrowed. An specific profile extending this specification is required for each type of registry/set of objects that is expected to be escrowed.

2. Terminology

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

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

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

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

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

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

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

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

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

3. Problem Scope

Starting a few years ago, the issue of Registry continuity has been carefully considered in the gTLD and ccTLD space. Various organizations have carried out a risk analysis and developed Business Continuity Plans to deal with those risks, should they materialize.

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

So far, almost every Registry that uses Registry Data Escrow has its own specification. It is anticipated that more Registries will be implementing Escrow especially with the advent of new TLDs, adding

complexity to this issue.

The main motivation for deveoloping this solution is rooted on the domain name registry industry. However, the specification has been designed to be as general as possible to allow other type of registries to use the base specification and develop their own profiles covering the objects used by other registration organizations.

Therefore, it would seem beneficial to have a standardized specification for Registry Data Escrow that can be used by any Registry to submit its Deposits.

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

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

Given the confidentiality and importance of some of the information that would be handled in order to offer the registration services, the solution SHALL define confidentiality and integrity mechanisms when handling the registration data.

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

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

Non-technical issues around Data Escrow and the overall question of the use of Registry Data Escrow are outside of scope of this document.

4. General Conventions

4.1. Date and Time

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

from the zero meridian.

5. Protocol Description

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

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

5.1. Root element <deposit>

The container or root element for a Registry Data Escrow deposits is <deposit>. This element contains the following child elements: watermark, deletes, contents, and extension. The latter is explained in Section 7. This element also contains the following attributes:

- A "type" attribute that MUST be used to identify the kind of deposit: FULL, INCR (Incremental) or DIFF (Differential).
- An "id" attribute that MUST be used to uniquely identify the escrow deposit. Each registry is responsible for maintaining its own escrow deposits identifier space to ensure uniqueness, e.g., using identifiers as described in Section 2.8 of [RFC5730].
- An OPTIONAL "prevId" attribute that can be used to identify the previous incremental, differential or full escrow deposit. This attribute MUST be used in Differential Deposits ("DIFF" type).
- An OPTIONAL "resend" attribute that is used to identify resend attempts in case of previous failure. The first time a deposit is attempted to be sent, the attribute MUST be zero; The second attempt to send (first resend attempt) the attribute MUST be set to one; and so on. This would be used when for example, the previous deposit was not received complete, it failed verification at the receiving party, etc.

Example of root element object:

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

5.2. Child <watermark> element

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

Example of <watermark> element object:

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

5.3. Child <rdeMenu> element

This element ...

The <rdeMenu> element contains the following child elements:

- A <version> element that identify the RDE protocol version.
- One or more <objURI> elements that contain namespace URIs representing the <contetns> and <deletes> element objects.
- An OPTIONAL <rdeExtension> element that contains one or more <extURI> elements that contain namespace URIs representing object extensions.

Example of <rdeMenu> element object:

```
<?xml version="1.0" encoding="UTF-8"?>
<rde:deposit
    xmlns:rde="urn:ietf:params:xml:ns:rde-1.0"
    ...
    <rde:rdeMenu>
        <rde:version>1.0</rde:version>
        <rde:objURI>urn:ietf:params:xml:ns:rdeContact-1.0</rde:objURI>
        <rde:objURI>urn:ietf:params:xml:ns:rdeHost-1.0</rde:objURI>
        <rde:objURI>urn:ietf:params:xml:ns:rdeDomain-1.0</rde:objURI>
        <rde:objURI>urn:ietf:params:xml:ns:rdeRegistrar-1.0</rde:objURI>
        <rde:objURI>urn:ietf:params:xml:ns:rdeIDN-1.0</rde:objURI>
        <rde:objURI>urn:ietf:params:xml:ns:rdeEppParams-1.0</rde:objURI>
    </rde:rdeMenu>
    ...
</rde:deposit>
```

5.4. Child <deletes> element

This element SHOULD only be present in deposits of type Incremental or Differential. It contains the list of objects that were deleted since the base previous deposit. Each object in this section SHALL contain an ID for the object deleted.

This section of the deposit SHOULD NOT be present in Full deposits. When rebuilding a registry it SHOULD be ignored if present in a Full deposit.

The specification for each object to be escrowed MUST declare the identificator to be used to reference the object to be deleted.

Example of <deletes> element object:

```
<?xml version="1.0" encoding="UTF-8"?>
<rde:deposit
    xmlns:rde="urn:ietf:params:xml:ns:rde-1.0"
    ...
    <rde:deletes>
        <rdeObj1:delete>
            <rdeObj1:name>foo.test</rdeObj1:name>
            <rdeObj1:name>bar.test</rdeObj1:name>
        </rdeObj1:delete>
        <rdeObj2:delete>
            <rdeObj2:id>sh8013-TEST</rdeObj2:id>
            <rdeObj2:id>co8013-TEST</rdeObj2:id>
        </rdeObj2:delete>
    </rde:deletes>
    ...
</rde:deposit>
```

5.5. Child <contents> element

This element of the deposit contains the objects in the deposit. It MUST be present in all type of deposits. It contains the data for the objects to be escrowed. The actual objects have to be specified individually. This element MAY also contain an extension element allowing extending the format.

In the case of Incremental or Differential deposits, the objects indicate whether the object was added or modified after the base previous deposit. In order to distinguish between one and the other, it will be sufficient to check existence of the referenced object in the base previous deposit.

When applying Incremental or Differential deposits, i.e., when rebuilding the registry from data escrow deposits, the order of the <deletes> and <contents> elements is important. First, all the deletes MUST be applied and then the adds and updates, i.e., first apply what is in <deletes> and later what is in <contents>.

Example of <contents> element object:

```
<?xml version="1.0" encoding="UTF-8"?>
<rde:deposit
    xmlns:rde="urn:ietf:params:xml:ns:rde-1.0"
    ...
    <rde:contents>
        ...
        <rdeObj1:contents>
            <rdeObj1:element1>
                <rdeObj1:child1>Object1 specific.</rdeObj1:child1>
                ...
            </rdeObj1:element1>
            <rdeObj2:element2>
                <rdeObj2:field1>Object2 specific.</rdeObj2:field1>
                ...
            </rdeObj2:element2>
        </rdeObj1:contents>
        ...
    </rde:contents>
    ...
</rde:deposit>
```

6. Formal Syntax

6.1. RDE Schema

BEGIN

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

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

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

  <!--
  Root element
  -->
  <element name="deposit" type="rde:escrowDepositType"/>

  <!--
  RDE types
  -->
  <complexType name="escrowDepositType">
    <sequence>
      <element name="watermark" type="dateTime"/>
      <element name="rdeMenu" type="rde:rdeMenuType"/>
      <element name="deletes" type="rde:rdeDeletesType"
       minOccurs="0"/>
      <element name="contents" type="rde:rdeContentsType"/>
    </sequence>
    <attribute name="type" type="rde:depositType"
     use="required"/>
    <attribute name="id" type="rde:depositIdType"
     use="required"/>
    <attribute name="prevId" type="rde:depositIdType"
     use="optional"/>
    <attribute name="resend" type="unsignedShort"
     default="0"/>
```

```
    </complexType>

    <complexType name="rdeContentsType">
      <sequence
       minOccurs="0" maxOccurs="unbounded">
        <element ref="rde:contents"/>
      </sequence>
    </complexType>

    <element name="contents" type="rde:contentsType" abstract="true" />
    <complexType name="contentsType">
      <sequence
       minOccurs="0" maxOccurs="unbounded">
        <element ref="rde:content"/>
        <element name="extension" type="rde:extAnyType"
         minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
    </complexType>

    <element name="content" type="rde:contentType" abstract="true" />
    <complexType name="contentType">
      <sequence/>
    </complexType>

    <complexType name="rdeDeletesType">
      <sequence
       minOccurs="0" maxOccurs="unbounded">
        <element ref="rde:delete"/>
        <element name="extension" type="rde:extAnyType"
          minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
    </complexType>

    <element name="delete" type="rde:deleteType" abstract="true" />
    <complexType name="deleteType">
      <sequence/>
    </complexType>

    <!--
    Type of deposit
    -->
    <simpleType name="depositType">
      <restriction base="token">
        <enumeration value="FULL"/>
        <enumeration value="INCR"/>
        <enumeration value="DIFF"/>
      </restriction>
    </simpleType>
```

```
<!--
Deposit identifier type
-->
<simpleType name="depositIdType">
  <restriction base="token">
    <pattern value="\w{1,13}"/>
  </restriction>
</simpleType>

<!--
  Identifies available object services.
-->
<complexType name="rdeMenuType">
  <sequence>
    <element name="version" type="rde:versionType"
      maxOccurs="unbounded"/>
    <element name="objURI" type="anyURI"
      maxOccurs="unbounded"/>
    <element name="svcExtension" type="rde:extURIType"
      minOccurs="0"/>
  </sequence>
</complexType>

<!--
Extension framework type
-->
<complexType name="extAnyType">
  <sequence>
    <any namespace="##other"
     maxOccurs="unbounded"/>
  </sequence>
</complexType>

<complexType name="extURIType">
  <sequence>
    <element name="extURI" type="anyURI"
      maxOccurs="unbounded"/>
  </sequence>
</complexType>

<!--
  A RDE version number is a dotted pair of decimal numbers.
-->
<simpleType name="versionType">
  <restriction base="token">
    <pattern value="[1-9]+\.[0-9]+"/>
    <enumeration value="1.0"/>
  </restriction>
```

```
    </simpleType>

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

7. Extension Guidelines

TBD

8. 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.

9. IANA Considerations

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

Registration request for the RDE namespace:

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

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

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

Registration request for the RDE XML schema:

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

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

See the "Formal Syntax" section of this document.

10. Security Considerations

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

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

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

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

11. Acknowledgments

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

TBD

12. Change History

12.1. Changes from version 00 to 01

1. Included DNSSEC elements as part of the basic <domain> element as defined in RFC 5910.

2. Included RGP elements as part of the basic <domain> element as defined in RFC 3915.

3. Added support for IDNs and IDN variants.

4. Eliminated the <summary> element and all its subordinate objects, except <watermarkDate>.

5. Renamed <watermarkDate> to <watermark> and included it directly under root element.

6. Renamed root element to <deposit>.

7. Added <authinfo> element under <registrar> element.

8. Added <roid> element under <registrar> element.

9. Reversed the order of the <deletes> and <contents> elements.

10. Removed <rdeDomain:status> minOccurs="0".

11. Added <extension> element under root element.

12. Added <extension> element under <contact> element.

13. Removed <period> element from <domain> element.

14. Populated the "Security Considerations" section.

15. Populated the "Internationalization Considerations" section.

16. Populated the "Extension Example" section.

17. Added <deDate> element under <domain> element.

18. Added <icannID> element under <registrar> element.

19. Added <eppParams> element under root element.

20. Fixed some typographical errors and omissions.

12.2. Changes from version 01 to 02

1. Added definition for "canonical" in the "IDN variants Handling" section.

2. Clarified that "blocked" and "reserved" IDN variants are optional.

3. Made <rdeRegistrar:authInfo> optional.

4. Introduced substitutionGroup as the mechanism for extending the protocol.

5. Moved <eppParams> element to be child of <contents>

6. Text improvements in the Introduction, Terminology, and Problem Scope per Jay's suggestion.

7. Removed <trDate> from <rdeDomain> and added <trnData> instead, which include all the data from the last (pending/processed) transfer request

8. Removed <trDate> from <rdeContact> and added <trnData> instead, which include all the data from the last (pending/processed) transfer request

9. Fixed some typographical errors and omissions.

12.3. Changes from version 02 to 03

1. Separated domain name objects from protocol.

2. Moved <extension> elements to be child of <deletes> and <contents>, additionally removed <extension> element from <rdeDomain>,<rdeHost>, <rdeContact>,<rdeRegistrar> and <rdeIDN> elements.

3. Modified the definition of <rde:id> and <rde:prevId>.

4. Added <rdeMenu> element under <deposit> element.

5. Fixed some typographical errors and omissions.

13. References

13.1. Normative References

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

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

13.2. Informative References

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

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

Authors' Addresses

Francisco Arias
Internet Corporation for Assigned Names and Numbers
4676 Admiralty Way, Suite 330
Marina del Rey 90292
United States of America

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

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

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