ICANN Registry Interfaces
draft-lozano-icann-registry-interfaces-17

Abstract

   This document describes the technical details of the interfaces
   provided by the Internet Corporation for Assigned Names and Numbers
   (ICANN) to its contracted parties to fulfill reporting requirements.
   The interfaces provided by ICANN to Data Escrow Agents and Registry
   Operators to fulfill the requirements of Specifications 2 and 3 of
   the gTLD Base Registry Agreement are described in this document.
   Additionally, interfaces for retrieving the IP addresses of the probe
   nodes used in the SLA Monitoring System (SLAM) and interfaces for
   supporting maintenance window objects are described in this document.

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 September 21, 2022.

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

   This document describes the technical details of the interfaces
   provided by the Internet Corporation for Assigned Names and Numbers
   (ICANN) to other contracted parties to fulfill reporting
   requirements. The interface provided by ICANN to Registry Operators
   and Data Escrow Agents to fulfill the requirements of Specifications
   2 and 3 of the gTLD Base Registry Agreement [ICANN-GTLD-RA-20170731]
   are described in this document. Additionally, interfaces for
   retrieving the IP addresses of the probe nodes used in the SLA
   Monitoring System (SLAM) and interfaces for supporting maintenance
   window objects are described in this document.

   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.

   The provisioning of credentials and authentication methods used in
   the interfaces are outside of this document's scope.

1.1. 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.

XML is case sensitive. Unless stated otherwise, XML specifications
and examples provided in this document MUST be interpreted in the
character case presented to develop a conforming implementation.

1.2. Date and Time

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

1.3. Common elements used in this specification

Common elements used in this specification are explained in this
section.

o <base-url>: The base URL used in the reporting interfaces examples
   must be replaced with the URL indicated by ICANN.

1.4. Object Description

This section describes the base objects supported by this
specification.

1.4.1. <iirdea:result> object

The ICANN interfaces for registries and data escrow agents (IIRDEA)
<iirdea:result> object is used to provide information on the result
of a verification process when interacting with the interfaces. The
XML Schema for the <iirdea:result> object can be found in
Section 7.1.

The <iirdea:result> object contains the following attributes and
child elements:

o A "code" attribute whose value is a four-digit decimal number that
   identifies the result of a process. Available result code values
   MUST be defined for the corresponding process.

o An OPTIONAL "domainCount" attribute indicates the number of domain
   names related to the reported result.

o A <msg> element contains a human-readable description of the
   result code.

   o An OPTIONAL <description> element that includes additional details
      on the result conditions.

   An example of a <iirdea:result> object is presented below:

   <result code="2001">
       <msg>The structure of the report is invalid.</msg>
       <description>
          'XX' could not be parsed as a number (line: 2 column:3)
       </description>
   </result>

1.4.2. <rdeReport:report> object

   The contents of a data escrow deposit are described using a
   <rdeReport:report> object. The XML Schema for the <rdeReport:report>
   object can be found in Section 7.2.

   The <rdeReport:report> object contains the following child elements:

   o An <id> element that contains the identifier assigned to this
      report. The report identifier MUST be the same as the "id"
      attribute from the <deposit>. If the data escrow deposit does not
      include a unique identifier, the Data Escrow Agent MUST generate a
      unique identifier to reference the data escrow deposit and use it
      in the <id> element.

   o A <version> element contains the version of the specification
      used. This value MUST be 1.

   o A <rydeSpecEscrow> element contains the version of the Data Escrow
      Specification (e.g., RFC8909) used to create the deposit. After
      the specification is published as an RFC, the value MUST be the
      RFC number assigned by IANA.

   o An OPTIONAL <rydeSpecMapping> element contains the version of the
      Domain Name Registration Data (DNRD) Objects Mapping (e.g.,
      RFC9022) used to create the deposit. After the specification is
      published as an RFC, the value MUST be the RFC number assigned by
      IANA. The <rydeSpecMapping> element MUST be included if the
      deposit was created using any version of the DNRD objects mapping
      specification (see [RFC9022]).

   o A <resend> element contains the value of the "resend" attribute of
      the <deposit>.

   o A <crDate> element contains the date and time that the Registry
      Operator created the deposit.

o A <kind> element is used to identify the kind of deposit: FULL,
   INCR (Incremental), or DIFF (Differential).

o A <watermark> element contains the date and time corresponding to
   the Timeline Watermark (<watermark> element) of the <deposit>.

o A <rdeHeader:header> element contains the header of the <deposit>
   as defined in [RFC9022].

An example of a <rdeReport:report> object is available in
Section 2.1.

1.4.3. <rdeNotification:notification> object

The <rdeNotification:notification> object is used by Data Escrow
Agents to document the result of the data escrow deposit verification
process. The XML Schema for the <rdeNotification:notification>
object can be found in Section 7.3.

The <rdeNotification:notification> object contains the following
child elements:

o A <deaName> element contains the name of the Data Escrow Agent.

o A <version> element contains the version of the specification
   used. This value MUST be 1.

o A <repDate> element contains the reported date. In the case of a
   DVPN or DVFN notification, this value MUST be the date of the
   <watermark> element of the <deposit>. In the case of a DRFN
   deposit notification, this value MUST be the date for which no
   deposit was received from the Registrar or Registry Operator.

o A <status> element is used to specify the status of <repDate>.
   The possible values of status are DVPN, DVFN, and DRFN. The three
   types of notices determine the value for the <status> element:

   * Deposit Receipt Failure Notice (DRFN): generated by the Data
      Escrow Agent when no deposit is received according to the data
      escrow deposit schedule.

   * Deposit Verification Failure Notice (DVFN): generated by the
      Data Escrow Agent when a deposit is received, but the final
      result of the verification process is a failure.

   * Deposit Verification Pass Notice (DVPN): generated by the Data
      Escrow Agent when a deposit is received, and the final result
      of the verification process is a success.

o An OPTIONAL <results> element contains the errors detected during
   the data escrow deposit verification process performed by the Data
   Escrow Agent. The <results> element includes one or more
   <iirdea:result> elements as defined in Section 1.4.1. In the case
   of a DRFN or DVPN deposit notification, the <results> element MUST
   NOT be present.

o An OPTIONAL <reDate> element contains the date and time that the
   Data Escrow Agent successfully received the deposit. In the case
   of a DRFN deposit notification, this element MUST NOT be present.

o An OPTIONAL <vaDate> element contains the date and time that the
   Data Escrow Agent processed the deposit for validation. In the
   case of a DRFN deposit notification, this element MUST NOT be
   present.

o An OPTIONAL <lastFullDate> element contains the date of the
   Timeline Watermark (<watermark> element) of the most recent FULL
   deposit that the Data Escrow Agent successfully validated. This
   element MUST NOT be present if a successfully validated full
   deposit has never been deposited.

o An OPTIONAL <rdeReport:report> element is used by the Data Escrow
   Agent to provide extended information about the deposit. In the
   case of a DRFN deposit notification, this element MUST NOT be
   present. In the case of a DVPN or DVFN deposit notification, this
   element MUST be present. When this element is present, the
   <rdeHeader:header> element MUST be generated by the Data Escrow
   Agent for the Timeline Watermark (<watermark> element) of the
   deposit being processed. If the deposit being processed is a
   differential or incremental deposit, the Data Escrow Agent MUST
   create a dataset by following Section 5.2 of [RFC8909] to generate
   the <rdeHeader:header> element.

o Note: In the case of a DVPN or DVFN deposit notification, the <id>
   is used as a unique identifier.

An example of a <rdeNotification:notification> object is available in
Section 2.2.

1.4.4. <rriReporting:summary> Object

Interfaces that support monitoring the reporting status for a
specific repository provide a <rriReporting:summary> object as
defined by the schema in Section 7.4 in the HTTP Entity-body when a
HTTP/200 code is sent by the interface.

The <rriReporting:summary> element includes the following child
elements:

o A choice of one of the elements as defined in the
   "rdeHeader:repositoryTypeGroup" (see [RFC9022]) that indicates the
   unique identifier for the repository being escrowed.

o A <creationDate> element with the date and time in which the
   queried repository was created in the system.

o A <depositSchedule> OPTIONAL element indicating the current Data
   Escrow Deposit schedule for the queried repository. Possible
   values are "None", "Weekly", and "Daily".

o An OPTIONAL <lastFullDate> element indicating the date of the
   Timeline Watermark (<watermark> element) of the most recent FULL
   deposit that was successfully validated for the queried repository
   as notified by the Data Escrow Agent.

o A <statusReports> element with a <statusReport> element for each
   report type for the queried repository. Each <statusReport>
   element includes the following child elements:

   * <type>: a string value indicating the report type to which the
      information provided pertains.

   * <enabled>: a boolean value indicating if the report type is
      enabled for the repository.

   * <status>: a string value indicating the reporting status. A
      value of "ok" indicates no reporting issues in the
      corresponding report type, otherwise the value of
      "unsatisfactory" is shown.

   * An OPTIONAL <issues> element included only when the <status>
      element has a value of "unsatisfactory", and includes an empty
      <issue> element for each date with a reporting problem found in
      the corresponding report type. Each <issue> element includes a
      REQUIRED "date" attribute in "YYYY-MM-DD" format and a REQUIRED
      "description" attribute to describe the issue. The possible
      values to describe each reporting issue are:

      + "Missing_Deposit_Full": If the latest notification received
         from the Data Escrow Agent for the date indicates that a
         scheduled "Full" deposit was not submitted by the repository
         owner.

+ "Missing_Deposit_Diff": If the latest notification received
  from the Data Escrow Agent for the date indicates that a
  scheduled "Differential" deposit was not submitted by the
  repository owner.

+ "Invalid_Deposit_Full": If the latest notification received
  from the Data Escrow Agent for the date indicates that a
  "Full" deposit was received by the Data Escrow Agent, but
  failed the verification process.

+ "Invalid_Deposit_Diff": If the latest notification received
  from the Data Escrow Agent for the date indicates that a
  "Differential" deposit was received by the Data Escrow
  Agent, but failed the verification process.

+ "No_Report_Received" If no report has been received for the
  date.

o A <timestamp> element to indicate the date and time the reporting
   status response was created.

1.4.5. <rdeReports:reports> Object

Interfaces that support monitoring and retrieving Data Escrow Reports
received, provide a <rdeReports:reports> object as defined by the
schema in Section 7.6 in the HTTP Entity-body when an HTTP/200 code
is sent by the interface.

The <rdeReports:reports> element includes a list of
<rdeReports:receivedReport> objects, one for each <rdeReport:report>
successfully received by ICANN.

Each <rdeReports:receivedReport> object includes the following child
elements:

o A <received> element to indicate the date and time in which ICANN
   received the report.

o A <rdeReport:report> element as defined in Section 1.4.2 as
   received by ICANN.

1.4.6. <rdeNotifications:notifications> Object

Interfaces that support monitoring and retrieving Data Escrow
Notifications received from Data Escrow Agents, provide a
<rdeNotifications:notifications> object as defined by the schema in
Section 7.5 in the HTTP Entity-body when an HTTP/200 code is sent by
the interface.

The <rdeNotifications:notifications> element includes a list of
<rdeNotifications:receivedNotification> objects, one for each
<rdeNotification:notification> successfully received by ICANN.

Each <rdeNotifications:receivedNotification> object includes the
following child elements:

o A <received> element to indicate the date and time in which ICANN
   received the notification.

o A <rdeNotification:notification> element as defined in
   Section 1.4.3 as received by ICANN.

1.4.7. <schedule:schedule> object

The <schedule:schedule> object is used to provide details of an
instance of a maintenance window. The maintenance window is
identified by a Universally Unique IDentifier version 4 (see
[RFC4122]) captured in the attribute "scheduleId". An additional
attribute "enabled" defined as a boolean can be used to enable (value
of true) or disable (value of false) the maintenance window. The XML
Schema for the <schedule:schedule> object can be found in
Section 7.7.

The <schedule:schedule> object contains the following child elements:

o A <name> element contains a descriptive name of the maintenance
   window.

o A <description> element contains a description of the maintenance
   window.

o A <version> element contains the version of the specification
   used. This value MUST be 1.

o A <startTime> element contains the date and time when the
   maintenance window starts being active.

o A <endTime> element contains the date and time when the
   maintenance window ends being active.

An example of a <schedule:schedule> object is available in Section 4.

1.4.8. <schedule:schedules> object

The <schedule:schedules> object is used to provide the list of
maintenance window objects. The XML Schema for the
<schedule:schedules> object can be found in Section 7.7.

The <schedule:schedules> object contains the following child
elements:

o Zero or more <schedule:schedule> elements. See, Section 1.4.7 for
   the definition of <schedule:schedule>.

An example of a <schedule:schedules> object is available in
Section 4.

1.4.9. <probeNode:probeNodes> object

The <probeNode:probeNodes> object is used to provide the list of
probe nodes used by the SLA Monitoring System. The XML Schema for
the <probeNode:probeNodes> object can be found in Section 7.8.

The <probeNode:probeNodes> object contains the following child
elements:

o One or more <probeNode> elements. Each <probeNode> element
   contains the following child elements:

   * A <city> element contains the name of the city where the node
      is located.

   * One or more <addr> element contains the IP addresses used in
      the node to originate test connections. An "ip" attribute
      contains the string "v4" if the IP address is an IPv4 address
      or "v6" if the IP address is an IPv6 address.

o A <updateTime> element contains the date and time of the last
   update of the list.

o A <version> element contains the version of the specification
   used. This value MUST be 1.

An example of a <probeNode:probeNodes> object is available in
Section 5.

2. Interfaces for Specification 2 - Data Escrow Reporting

This section describes the interfaces provided by ICANN to Registry
Operators and Data Escrow Agents to fulfill the reporting
requirements detailed in Specification 2 of the gTLD Base Registry
Agreement [ICANN-GTLD-RA-20170731].

2.1. Registry Operator Reporting

The gTLD Base Registry Agreement [ICANN-GTLD-RA-20170731],
Specification 2, Part A, Section 7 requires Registry Operators to
provide ICANN with a written statement that includes a copy of the
report generated upon creation of a deposit and a statement that the
deposit has been inspected by the Registry Operator and is complete
and accurate.

In order to satisfy this requirement, the Registry Operator sends to
ICANN a <rdeReport:report> object as defined in Section 1.4.2 for
each deposit successfully sent to the Data Escrow Agent, using the
PUT HTTP verb in the interface provided by ICANN at:

    <base-url>/report/registry-escrow-report/<tld>/<id>

    Where:

    *   <tld> MUST be substituted by the TLD for which the report is
        being provided. In the case of an IDN TLD, the A-label (see
        [RFC5890]) MUST be used.

    *   <id> MUST be substituted by the identifier assigned to this
        report, which MUST be the same as the "id" attribute from the
        <deposit>.

    Note: the interface supports overwriting the information of a
    particular report <id> to support asynchronous interfaces between
    Registry Operators and Data Escrow Agents.

Example of a <rdeReport:report> object for a data escrow deposit
corresponding to a TLD Registry repository:

```
<?xml version="1.0" encoding="UTF-8"?>
<rdeReport:report
  xmlns:rdeReport="urn:ietf:params:xml:ns:rdeReport-1.0"
  xmlns:rdeHeader="urn:ietf:params:xml:ns:rdeHeader-1.0">
  <rdeReport:id>20101017001</rdeReport:id>
  <rdeReport:version>1</rdeReport:version>
  <rdeReport:rydeSpecEscrow>
    RFC8909
  </rdeReport:rydeSpecEscrow>
  <rdeReport:rydeSpecMapping>
    RFC9022
  </rdeReport:rydeSpecMapping>
  <rdeReport:resend>0</rdeReport:resend>
  <rdeReport:crDate>2010-10-17T00:15:00.0Z</rdeReport:crDate>
  <rdeReport:kind>FULL</rdeReport:kind>
  <rdeReport:watermark>2010-10-17T00:00:00Z</rdeReport:watermark>
  <rdeHeader:header>
    <rdeHeader:tld>test</rdeHeader:tld>
    <rdeHeader:count
      uri="urn:ietf:params:xml:ns:rdeDomain-1.0">2</rdeHeader:count>
    <rdeHeader:count
      uri="urn:ietf:params:xml:ns:rdeHost-1.0">1</rdeHeader:count>
    <rdeHeader:count
      uri="urn:ietf:params:xml:ns:rdeContact-1.0">1</rdeHeader:count>
    <rdeHeader:count
      uri="urn:ietf:params:xml:ns:rdeRegistrar-1.0">1
    </rdeHeader:count>
    <rdeHeader:count
      uri="urn:ietf:params:xml:ns:rdeIDN-1.0">1</rdeHeader:count>
    <rdeHeader:count
      uri="urn:ietf:params:xml:ns:rdeNNDN-1.0">1</rdeHeader:count>
    <rdeHeader:count
      uri="urn:ietf:params:xml:ns:rdeEppParams-1.0">1
    </rdeHeader:count>
  </rdeHeader:header>
</rdeReport:report>
```

2.2. Data Escrow Agent Reporting

The gTLD Base Registry Agreement [ICANN-GTLD-RA-20170731],
Specification 2, Part B, Section 7 requires Data Escrow Agents to
deliver ICANN with a notification object every time a successfully
processed deposit is received from the Registry Operator regardless
of the final status of the verification process.

To satisfy this requirement, the Data Escrow Agent sends to ICANN a
<rdeNotification:notification> object as defined in Section 1.4.3,
using the POST HTTP verb in the interface provided by ICANN at:

```
<base-url>/report/escrow-agent-notification/<tld>
```

Where:

*   <tld> MUST be substituted by the TLD for which the notification
    is being provided. In the case of an IDN TLD, the A-label (see
    [RFC5890]) MUST be used.

If by 23:59:59 UTC, a deposit has not been successfully processed
regardless of the final status of the verification process, a
<rdeNotification:notification> object with DRFN status MUST be sent
to ICANN.

Example of a <rdeNotification:notification> object of a Data Escrow
Agent notification corresponding to a Registry repository Data Escrow
Deposit:

```
<?xml version="1.0" encoding="UTF-8"?>
<rdeNotification:notification
  xmlns:rdeNotification="urn:ietf:params:xml:ns:rdeNotification-1.0"
  xmlns:rdeReport="urn:ietf:params:xml:ns:rdeReport-1.0"
  xmlns:rdeHeader="urn:ietf:params:xml:ns:rdeHeader-1.0">
  <rdeNotification:deaName>Escrow Agent Inc.</rdeNotification:deaName>
  <rdeNotification:version>1</rdeNotification:version>
  <rdeNotification:repDate>2010-10-17</rdeNotification:repDate>
  <rdeNotification:status>DVPN</rdeNotification:status>
  <rdeNotification:reDate>
    2010-10-17T03:15:00.0Z
  </rdeNotification:reDate>
  <rdeNotification:vaDate>
    2010-10-17T05:15:00.0Z
  </rdeNotification:vaDate>
  <rdeNotification:lastFullDate>
    2010-10-14
  </rdeNotification:lastFullDate>
  <rdeReport:report>
    <rdeReport:id>20101017001</rdeReport:id>
    <rdeReport:version>1</rdeReport:version>
    <rdeReport:rydeSpecEscrow>
      RFC8909
    </rdeReport:rydeSpecEscrow>
    <rdeReport:rydeSpecMapping>
      RFC9022
    </rdeReport:rydeSpecMapping>
    <rdeReport:resend>0</rdeReport:resend>
    <rdeReport:crDate>2010-10-17T00:15:00.0Z</rdeReport:crDate>
    <rdeReport:kind>FULL</rdeReport:kind>
    <rdeReport:watermark>2010-10-17T00:00:00Z</rdeReport:watermark>
```

```
      <rdeHeader:header>
        <rdeHeader:tld>test</rdeHeader:tld>
        <rdeHeader:count
      uri="urn:ietf:params:xml:ns:rdeDomain-1.0">1</rdeHeader:count>
        <rdeHeader:count
      uri="urn:ietf:params:xml:ns:rdeHost-1.0">3</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">3</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">10</rdeHeader:count>
        <rdeHeader:count
      uri="urn:ietf:params:xml:ns:rdeEppParams-1.0">0</rdeHeader:count>
      </rdeHeader:header>
    </rdeReport:report>
  </rdeNotification:notification>
```

2.3. Monitoring the reporting status

Registries MAY monitor the status of the reports described in
Specification 2 [ICANN-GTLD-RA-20170731] using the following
interfaces that support the HEAD HTTP verb:

2.3.1. Monitoring the status of Data Escrow Reports

Registries MAY monitor the status of Data Escrow Reports using the
following interface:

    <base-url>/info/report/registry-escrow-report/<tld>/<date>

    Where:

    *   <tld> MUST be substituted by the TLD being queried. In the
        case of an IDN TLD, the A-label (see [RFC5890]) MUST be used.

    *   <date> MUST be substituted by the day being queried. For
        example, 2013-03-02.

    Possible results are:

    *   The interface provides an HTTP/200 status code if a
        syntactically valid data escrow report was received for the
        queried date.

* The interface provides an HTTP/404 status code if a syntactically valid data escrow report has not been received for the queried date.

2.3.2. Monitoring the status of Data Escrow Notifications

Registries and Data Escrow Agents MAY monitor the status of Data Escrow Notifications using the following interface:

<base-url>/info/report/escrow-agent-notification/<tld>/<date>

Where:

* <tld> MUST be substituted by the TLD being queried. In the case of an IDN TLD, the A-label (see [RFC5890]) MUST be used.

* <date> MUST be substituted by the day being queried. For example, 2013-03-02.

Possible results are:

* The interface provides an HTTP/200 status code if a syntactically valid data escrow notification was received for the queried date.

* The interface provides an HTTP/404 status code if a syntactically valid data escrow notification has not been received for the queried date.

3. Interfaces of Specification 3 - Registry Operator Monthly Reporting

Specification 3 of the gTLD Base Registry Agreement [ICANN-GTLD-RA-20170731] requires Registry Operators to provide a set of monthly reports per gTLD. Two types of reports are required to be sent by Registries: Per-Registrar Transactions Report and Registry Functions Activity Report. This section specifies the interfaces provided by ICANN to automate the upload of these reports by Registry Operators.

The cut-off date for the reception of the reports specified in specification 3 is defined in the gTLD Base Registry Agreement [ICANN-GTLD-RA-20170731]. Before the cut-off date, the Registry Operator MAY replace a successfully validated report as many times as needed.

3.1. Per-Registrar Transactions Report

The Per-Registrar Transactions Report is a CSV report encoded in
UTF-8 described in Section 1 of Specification 3 of the gTLD Base
Registry Agreement [ICANN-GTLD-RA-20170731].

To satisfy this requirement, the Registry Operator sends a CSV report
monthly as described in the gTLD Base Registry Agreement
[ICANN-GTLD-RA-20170731], using the PUT HTTP verb in the interface
provided by ICANN at:

    <base-url>/report/registrar-transactions/<tld>/<date>

    Where:

    *   <tld> MUST be substituted by the TLD for which the report is
        being provided. In the case of an IDN TLD, the A-label (see
        [RFC5890]) MUST be used.

    *   <date> MUST be substituted by the month for which the report is
        being provided in the form of YYYY-MM. Where 'YYYY' is the
        year, and 'MM' is the two-digit month number. For example,
        2013-03.

3.2. Registry Functions Activity Report

The Registry Functions Activity Report is a CSV report encoded in
UTF-8 described in Section 2 of Specification 3 of the gTLD Base
Registry Agreement [ICANN-GTLD-RA-20170731].

To satisfy this requirement, the Registry Operator sends a CSV report
monthly as described in the gTLD Base Registry Agreement
[ICANN-GTLD-RA-20170731], using the PUT HTTP verb in the interface
provided by ICANN at:

    <base-url>/report/registry-functions-activity/<tld>/<date>

    Where:

    *   <tld> MUST be substituted by the TLD for which the report is
        being provided. In the case of an IDN TLD, the A-label (see
        [RFC5890]) MUST be used.

    *   <date> MUST be substituted by the month for which the report is
        being provided in the form of YYYY-MM. Where 'YYYY' is the
        year, and 'MM' is the two-digit month number. For example,
        2013-03.

3.3. Monitoring the reporting status

Registries MAY monitor the status of the reports described in
Specification 3 of the gTLD Base Registry Agreement
[ICANN-GTLD-RA-20170731] using the following interfaces that support
the HEAD HTTP verb:

3.3.1. Monitoring the status of Registry Functions Activity Report

Registries MAY monitor the status of Registry Functions Activity
Report using the following interface:

    <base-url>/info/report/registry-functions-activity/<tld>/<date>

    Where:

    *   <tld> MUST be substituted by the TLD being queried. In the
        case of an IDN TLD, the A-label (see [RFC5890]) MUST be used.

    *   <date> MUST be substituted by the month being queried. For
        example, 2013-03.

    Possible results are:

    *   The interface provides an HTTP/200 status code if a
        syntactically valid registry functions activity report was
        received for the queried month.

    *   The interface provides an HTTP/404 status code if a
        syntactically valid registry functions activity report has not
        been received for the queried month.

3.3.2. Monitoring the status of the Per-Registrar Transactions Report

Registries MAY monitor the status of Per-Registrar Transactions
Report using the following interface:

    <base-url>/info/report/registrar-transactions/<tld>/<date>

    Where:

    *   <tld> MUST be substituted by the TLD being queried. In the
        case of an IDN TLD, the A-label (see [RFC5890]) MUST be used.

    *   <date> MUST be substituted by the month being queried. For
        example, 2013-03.

    Possible results are:

* The interface provides an HTTP/200 status code if a
  syntactically valid per-registrar transactions report was
  received for the queried month.

* The interface provides an HTTP/404 status code if a
  syntactically valid per-registrar transactions report has not
  been received for the queried month.

4. Maintenance Windows

Specification 10 of the gTLD Base Registry Agreement
[ICANN-GTLD-RA-20170731] allows Registry Operators to provide notice
to ICANN about plans of maintenance. This section specifies the
interfaces provided by ICANN to programmatically manage maintenance
windows by Registry Operators.

4.1. Creating and updating a maintenance window

To create or update a previously created maintenance window, the
Registry Operator sends a <schedule:schedule> object (see
Section 1.4.7) using the PUT HTTP verb in the interface provided by
ICANN at:

    <base-url>/maintenance-window/<tld>/<service>/<scheduleId>

    Where:

* <tld> MUST be substituted by the TLD for which the report is
  being provided. In the case of an IDN TLD, the A-label (see
  [RFC5890]) MUST be used.

* <service> MUST be substituted by a service for which a
  maintenance window could be created (e.g., RDDS).

* <scheduleId> MUST be substituted by a Universally Unique
  IDentifier version 4 (see [RFC4122]).

4.2. Deleting a maintenance window

To delete a previously created maintenance window, the Registry
Operator uses the DELETE HTTP verb in the interface provided by ICANN
at:

    <base-url>/maintenance-window/<tld>/<service>/<scheduleId>

4.3. Getting the persisted data for a maintenance window

   To get the persisted data for a maintenance window, the Registry
   Operator uses the GET HTTP verb in the interface provided by ICANN
   at:

      <base-url>/maintenance-window/<tld>/<service>/<scheduleId>

   The interface provides an HTTP/200 status code and sets the HTTP
   header Content-type: text/xml after a successful request, and
   includes a <schedule:schedule> object as defined in Section 1.4.7.

   Example of a <schedule:schedule> object of a Maintenance Window:

```
 <?xml version="1.0" encoding="UTF-8"?>
 <schedule:schedule xmlns:schedule="urn:ietf:params:xml:ns:schedule-1.0"
 enabled="true" scheduleId="6257b2b9-1ff7-4159-bc0d-397fec035d38">
     <schedule:name>Maintenance window for RDDS semester II-2017
     </schedule:name>
     <schedule:description>Pre-planned maintenance window for RDDS
     </schedule:description>
     <schedule:startTime>2010-10-17T00:15:00.0Z</schedule:startTime>
     <schedule:endTime>2010-10-17T02:15:00.0Z</schedule:endTime>
     <schedule:version>1</schedule:version>
 </schedule:schedule>
```

4.4. Getting a list of maintenance windows

   To get the list of all the maintenance windows that have not ended
   yet (see Section 4), the Registry Operator uses the GET HTTP verb in
   the interface provided by ICANN at:

      <base-url>maintenance-window/<tld>/<service>

   The interface provides an HTTP/200 status code and sets the HTTP
   header Content-type: text/xml after a successful request, and
   includes a <schedule:schedules> object as defined in Section 1.4.8.

   Example of a <schedule:schedules>:

```
<?xml version="1.0" encoding="UTF-8"?>
<schedule:schedules
    xmlns:schedule="urn:ietf:params:xml:ns:schedule-1.0">
    <schedule:schedule enabled="true"
    scheduleId="6257b2b9-1ff7-4159-bc0d-397fec035d38">
        <schedule:name>Maintenance window for example</schedule:name>
        <schedule:description>Pre-planned maintenance window 1
        </schedule:description>
        <schedule:startTime>2010-10-17T00:15:00.0Z</schedule:startTime>
        <schedule:endTime>2010-10-17T02:15:00.0Z</schedule:endTime>
        <schedule:version>1</schedule:version>
    </schedule:schedule>
    <schedule:schedule enabled="true"
    scheduleId="3da01e98-1945-4cc9-9334-bd3ad5b36371">
        <schedule:name>Maintenance window for example</schedule:name>
        <schedule:description>Pre-planned maintenance window 2
        </schedule:description>
        <schedule:startTime>2010-10-19T00:15:00.0Z</schedule:startTime>
        <schedule:endTime>2010-10-19T02:15:00.0Z</schedule:endTime>
        <schedule:version>1</schedule:version>
    </schedule:schedule>
</schedule:schedules>
```

5. Interfaces for the SLAM Probe Node List

The current list of probe nodes used by the SLA Monitoring System may
be retrieved by using the GET HTTP verb in the interface provided by
ICANN at:

    <base-url>/slam-probe-nodes/list

The interface provides an HTTP/200 status code and sets the HTTP
header Content-type: text/xml after a successful request, and
includes a <probeNode:probeNodes> object as defined in Section 1.4.9.

Example of the response upon successful retrieval of the SLAM Probe
Node List:

```
<?xml version="1.0" encoding="UTF-8"?>
<probeNode:probeNodes
    xmlns:probeNode="urn:ietf:params:xml:ns:probeNode-1.0">
    <probeNode:probeNode>
        <probeNode:city>Mumbai</probeNode:city>
        <probeNode:addr ip="v4">192.0.2.1</probeNode:addr>
        <probeNode:addr ip="v6">2001:db8:1::1</probeNode:addr>
    </probeNode:probeNode>
    <probeNode:probeNode>
        <probeNode:city>Seoul</probeNode:city>
        <probeNode:addr ip="v4">198.51.100.1</probeNode:addr>
    </probeNode:probeNode>
    <probeNode:updateTime>2020-11-18T02:23:56-08:00
    </probeNode:updateTime>
    <probeNode:version>1</probeNode:version>
</probeNode:probeNodes>
```

6. Technical details of the interfaces

This section describes the technical details of the interfaces
described in this document.

6.1. General technical details

This section describes the technical details that apply to all the
interfaces described in this document.

The following HTTP status codes are returned:

o The interface provides an HTTP/401 status code and sets the HTTP
   header Content-type: text/plain, if the credentials provided do
   not authorize the user to perform the action requested for that
   <tld>.

o The interface provides an HTTP/403 status code and sets the HTTP
   header Content-type: text/plain, if the user attempts to access a
   resource that permission is not granted for, or the connecting IP
   address is not authorized for the <tld>.

o The interface provides an HTTP/404 status code if the object
   referenced in the URL is not found.

o The interface provides an HTTP/405 status code if the interface
   does not support the request method.

o The interface provides an HTTP/500 status code and sets the HTTP
   header Content-type: text/plain, if the system is experiencing a

general failure. The sending party is responsible for sending the
input again.

o The interface provides an HTTP/501 status code and sets the HTTP
   header Content-type: text/plain, if the functionality has not yet
   been implemented.

After sending the response, the interfaces close the TCP connection.

The interfaces support HTTP streams only (HTTP multi-part forms are
not supported).

6.2. Specific technical details

This section describes the technical details that apply to the
following interfaces:

o Registry Operator Reporting (see Section 2.1)

o Data Escrow Agent Reporting (see Section 2.2)

o Per-Registrar Transactions Report (see Section 3.1)

o Registry Functions Activity Report (see Section 3.2)

o Creating and updating a maintenance window (see Section 4.1)

o Deleting a maintenance window (see Section 4.2)

The following additional HTTP status codes are returned:

o The interface provides an HTTP/200 status code and sets the HTTP
   header Content-type: text/xml, if the interface was able to
   receive the input successfully, the client MUST review the
   response object (see Section 6.2.1) to verify the result code
   after processing the input.

o The interface provides an HTTP/400 status code and sets the HTTP
   header Content-type: text/xml, if the input is incorrect or the
   syntax of the input is invalid. A response object (see
   Section 6.2.1) is included with the input validation failure
   details.

Content-type value in the HTTP header:

o The client MUST set "text/xml" in the HTTP header Content-type
   when using the interfaces described in Section 2.1, Section 2.2,
   Section 4.1, and Section 4.2.

   o The client MUST set "text/csv" in the HTTP header Content-type
      when using the interfaces described in Section 3.1 and
      Section 3.2.

6.2.1. Response Object

   After processing the input provided to the interfaces described in
   this section, a response object defined in Section 1.4.1 is sent in
   the HTTP Entity-body when an HTTP/200 or HTTP/400 status code is sent
   by the interface.

   An example of a response object upon successful input receipt is
   presented below:

   HTTP/1.1 200 OK
   Content-Type: text/xml
   Content-Length: 248

   <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
   <response
     xmlns="urn:ietf:params:xml:ns:iirdea-1.0">
     <result code="1000">
       <msg>No ERRORs were found and the report has been
             accepted by ICANN.</msg>
     </result>
   </response>

   An example of a response object in the event of input error is
   presented below:

   HTTP/1.1 400 Bad Request
   Content-Type: text/xml
   Content-Length: 279

   <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
   <response
     xmlns="urn:ietf:params:xml:ns:iirdea-1.0">
     <result code="2001">
       <msg>The structure of the report is invalid.</msg>
       <description>
             'XX' could not be parsed as a number (line: 2 column:3)
         </description>
     </result>
   </response>

   The following sections provide the IIRDEA Result Codes per interface:

6.2.1.1. **Registry Operator Reporting**

   The following table lists the result codes of the Registry Operator
   Reporting interface (see Section 2.1):

Result Code	Message
1000	No ERRORs were found, and the report has been accepted by ICANN.
2001	The request did not validate against the schema.
2004	Report for a date in the future. The \<crDate> and \<watermark> date should not be in the future.
2005	Version is not supported.
2006	The \<id> in the \<report> element and the \<id> in the URL path do not match.
2007	Interface is disabled for this TLD.
2008	The \<crDate> and \<watermark> date should not be before the creation date of the TLD in the system.
2202	The \<tld> in the \<header> and the TLD in the URL path do not match.
2205	Report regarding a differential deposit received when a full deposit was expected (\<watermark>).
2206	csvDomain and rdeDomain count provided in the \<header>.
2209	Missing required \<tld> element in the \<header>.
2210	The value of the "rcdn" attribute in the \<count> element does not match the same or lower level names in the \<tld> in the URL path.
2211	Multiple count elements with the same "uri", "rcdn", and "registrarId" attribute values provided in the \<header>.
2212	An invalid NR-LDH label or A-label was found or the domain name syntax is invalid in the "rcdn" attribute.

                     Data Escrow Reporting Result Codes

6.2.1.2. **Data Escrow Agent Reporting**

   The following table lists the result codes of the Data Escrow Agent
   Reporting interface (see Section 2.2):

```
+--------+------------------------------------------------------+
| Result | Message                                              |
| Code   |                                                      |
+--------+------------------------------------------------------+
| 1000   | No ERRORs were found, and the notification has been  |
|        | accepted by ICANN.                                   |
| 2001   | The request did not validate against the schema.     |
| 2002   | A DVPN notification exists for that date (<repDate>). |
| 2004   | Notification for a date in the future. The <crDate> and |
|        | <watermark> and <repDate> date should not be in the  |
|        | future.                                              |
| 2005   | Version is not supported.                            |
| 2007   | Interface is disabled for this TLD.                  |
| 2008   | The <crDate> and <watermark> and <repDate> date should |
|        | not be before the creation date of the TLD in the   |
|        | system.                                              |
| 2201   | The <repDate> and <watermark> in the notification do not |
|        | match.                                               |
| 2202   | The <tld> in the <header> and the TLD in the URL path do |
|        | not match.                                           |
| 2203   | A Deposit Verification Pass Notice (DVPN) notification |
|        | was received, but the Domain Name count is missing in |
|        | the <header>.                                        |
| 2204   | The notification for the report "id" already exists. |
| 2205   | Notification regarding a differential deposit received |
|        | when a full deposit was expected (<repDate>).        |
| 2206   | csvDomain and rdeDomain count provided in the <header>. |
| 2207   | A DVPN or DVFN was received, but the <report> element is |
|        | missing in the notification.                         |
| 2208   | A DRFN was received, but an <report> element exists in |
|        | the notification.                                    |
| 2209   | Missing required <tld> element in the <header>.      |
| 2210   | The value of the "rcdn" attribute in the <count> element |
|        | does not match the same or lower level names in the  |
|        | <tld> in the URL path.                               |
| 2211   | Multiple count elements with the same "uri", "rcdn", and |
|        | "registrarId" attribute values provided in the <header>. |
| 2212   | An invalid NR-LDH label or A-label was found or the  |
|        | domain name syntax is invalid in the "rcdn" attribute. |
+--------+------------------------------------------------------+
```

                 Data Escrow Reporting Result Codes

6.2.1.3. Per-Registrar Transactions Report

The following table lists the result codes of the Per-Registrar
Transactions Report interface (see Section 3.1):

```
+----------+-------------------------------------------------------+
| Result   | Message                                               |
| Code     |                                                       |
+----------+-------------------------------------------------------+
| 1000     | No ERRORs were found, and the report has been accepted|
|          | by ICANN.                                             |
| 2001     | The structure of the report is invalid.               |
| 2002     | A report for that month already exists, the cut-off   |
|          | date already passed.                                  |
| 2003     | Negative numeric value present in the report.         |
| 2004     | Report for a month in the future.                     |
| 2007     | Interface is disabled for this TLD.                   |
| 2008     | Reported month before the creation date of the TLD in |
|          | the system.                                           |
| 2101     | Incorrect totals present in the report.               |
| 2102     | A non-ICANN-accredited registrar is present in the    |
|          | report.                                               |
| 2103     | Values found in the second field of the totals line.  |
| 2105     | The report is not encoded in UTF-8. Note: reports     |
|          | encoded in US-ASCII are accepted.                     |
+----------+-------------------------------------------------------+
```

Per-Registrar Transactions Report Result Codes

6.2.1.4. Registry Functions Activity Report

The following table lists the result codes of the Registry Functions
Activity Report interface (see Section 3.2):

```
+----------+-------------------------------------------------------+
| Result   | Message                                               |
| Code     |                                                       |
+----------+-------------------------------------------------------+
| 1000     | No ERRORs were found, and the report has been accepted|
|          | by ICANN.                                             |
| 2001     | The structure of the report is invalid.               |
| 2002     | A report for that month already exists, the cut-off   |
|          | date already passed.                                  |
| 2003     | Negative numeric value present in the report.         |
| 2004     | Report for a month in the future.                     |
| 2007     | Interface is disabled for this TLD.                   |
| 2008     | Reported month before the creation date of the TLD in |
|          | the system.                                           |
| 2105     | The report is not encoded in UTF-8. Note: reports     |
|          | encoded in US-ASCII are accepted.                     |
+----------+-------------------------------------------------------+
```

Registry Functions Activity Report Result Codes

6.2.1.5. **Maintenance Window Management**

   The following table lists the result codes of the Maintenance Window
   Management interface (see Section 4):

Result Code	PUT	DELETE	GET	Message
1000	*	*		No ERRORs were found, and the report has been accepted by ICANN.
2001	*			The request did not validate against the schema.
2005	*			Version is not supported.
2006	*			The scheduleId in the schedule object and the scheduleId in the URL PATH do not match.
2007	*	*	*	Interface is disabled for this TLD.
2401	*	*	*	The syntax of the UUID in the PATH is incorrect.
2402	*			The maintenance window start date and time is not 24 hours ahead of the current date and time.
2403	*			The period specified by start and end date and time is greater than the monthly SLR for the service.
2404	*			The period specified in the maintenance window collides with a previously scheduled maintenance window for the service.
2405		*		The maintenance window that you are trying to delete already started.
2406	*			The endTime is in the past, before or equal to the startTime.
2407	*			The maintenance window that you are trying to update already ended, updates are not allowed.
2408	*			The maintenance window that you are trying to update already started, only enabled and endTime fields can be modified.

                 Maintenance Window Management Result Codes

7. Formal Syntax

The schema of the IIRDEA Result, Report, Notification, RRI Reporting, Notifications, and Reports objects described in Section 1.4 are presented here.

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.

7.1. RRI IIRDEA Result Schema

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

  <annotation>
    <documentation>
      ICANN interfaces for registries and data escrow agents
    </documentation>
  </annotation>
  <element name="response" type="iirdea:responseType"/>
  <element name="result" type="iirdea:resultType"/>
  <complexType name="responseType">
    <sequence>
      <element ref="iirdea:result" />
    </sequence>
  </complexType>
  <complexType name="resultType">
    <sequence>
      <element name="msg" type="token"/>
      <element name="description" type="string"
       minOccurs="0"/>
    </sequence>
    <attribute name="code" type="iirdea:codeType"
     use="required"/>
    <attribute name="domainCount" type="unsignedInt"/>
  </complexType>
  <simpleType name="codeType">
    <restriction base="unsignedShort">
      <minInclusive value="1000"/>
      <maxInclusive value="9999"/>
    </restriction>
  </simpleType>
</schema>
<CODE ENDS>
```

7.2. RRI Report Object

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

  <import namespace="urn:ietf:params:xml:ns:rde-1.0" />
  <import namespace="urn:ietf:params:xml:ns:rdeHeader-1.0" />
  <annotation>
    <documentation>
      Data Escrow Report schema
    </documentation>
  </annotation>
  <element name="report" type="rdeReport:reportType"/>
  <complexType name="reportType">
    <sequence>
      <element name="id" type="rde:depositIdType"/>
      <element name="version" type="unsignedShort"/>
      <element name="rydeSpecEscrow" type="token"/>
      <element name="rydeSpecMapping" type="token" minOccurs="0"/>
      <element name="resend" type="unsignedShort"/>
      <element name="crDate" type="dateTime"/>
      <element name="kind" type="rde:depositTypeType"/>
      <element name="watermark" type="dateTime"/>
      <element ref="rdeHeader:header"/>
    </sequence>
  </complexType>
</schema>
<CODE ENDS>
```

7.3. RRI Notification Object

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

  <import namespace="urn:ietf:params:xml:ns:rdeReport-1.0"/>
  <import namespace="urn:ietf:params:xml:ns:iirdea-1.0"/>
  <annotation>
    <documentation>
```

```
      Data Escrow Notification schema
    </documentation>
  </annotation>
  <element name="notification"
  type="rdeNotification:notificationType"/>
  <complexType name="notificationType">
    <sequence>
      <element name="deaName" type="rdeNotification:nameType"/>
      <element name="version" type="unsignedShort"/>
      <element name="repDate" type="date"/>
      <element name="status" type="rdeNotification:statusType"/>
      <element name="results" type="rdeNotification:resultsType"
        minOccurs="0" />
      <element name="reDate" type="dateTime" minOccurs="0"/>
      <element name="vaDate" type="dateTime" minOccurs="0"/>
      <element name="lastFullDate" type="date" minOccurs="0"/>
      <element ref="rdeReport:report" minOccurs="0"/>
    </sequence>
  </complexType>
  <simpleType name="nameType">
    <restriction base="normalizedString">
      <minLength value="1" />
      <maxLength value="255" />
    </restriction>
  </simpleType>
  <complexType name="resultsType">
    <sequence>
      <element ref="iirdea:result" maxOccurs="unbounded" />
    </sequence>
  </complexType>
  <simpleType name="statusType">
    <restriction base="token">
      <enumeration value="DVPN"/>
      <enumeration value="DVFN"/>
      <enumeration value="DRFN"/>
    </restriction>
  </simpleType>
</schema>
<CODE ENDS>
```

7.4. RRI Reporting Summary Object

```
<CODE BEGINS>
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="urn:ietf:params:xml:ns:rriReporting-1.0"
  xmlns:rriReporting="urn:ietf:params:xml:ns:rriReporting-1.0"
  xmlns:rdeHeader="urn:ietf:params:xml:ns:rdeHeader-1.0"
  xmlns="http://www.w3.org/2001/XMLSchema"
```

```
  elementFormDefault="qualified">

  <import namespace="urn:ietf:params:xml:ns:rdeHeader-1.0" />
  <element name="summary" type="rriReporting:summaryType"/>
  <complexType name="summaryType">
    <sequence>
      <group ref="rdeHeader:repositoryTypeGroup"/>
      <element name="creationDate" type="dateTime" />
      <element name="depositSchedule"
          type="rriReporting:depositScheduleType" />
      <element name="lastFullDate" type="date" minOccurs="0"/>
      <element name="statusReports"
          type="rriReporting:statusReportsType" />
      <element name="timestamp" type="dateTime" />
    </sequence>
  </complexType>
  <simpleType name="depositScheduleType">
    <restriction base="token">
      <enumeration value="None" />
      <enumeration value="Weekly" />
      <enumeration value="Daily" />
    </restriction>
  </simpleType>
  <complexType name="statusReportsType">
    <sequence>
      <element name="statusReport"
        type="rriReporting:statusReportType" maxOccurs="unbounded" />
    </sequence>
  </complexType>
  <complexType name="statusReportType">
    <sequence>
      <element name="type" type="rriReporting:statusReportTypeType" />
      <element name="enabled" type="boolean" />
      <element name="status" type="rriReporting:statusType" />
      <element name="issues" type="rriReporting:issuesType"
       minOccurs="0" />
    </sequence>
  </complexType>
  <simpleType name="statusReportTypeType">
    <restriction base="token">
      <enumeration value="DEA_Notification" />
      <enumeration value="Registrar_Escrow_Report" />
      <enumeration value="Registry_Escrow_Report" />
      <enumeration value="PPSP_Escrow_Report" />
      <enumeration value="Registry_Functions_Activity_Report" />
      <enumeration value="Registry_Per_Registrar_Transactions_Report" />
      <enumeration value="PPSP_Per_Registrar_Activity_Report" />
    </restriction>
```

```
    </simpleType>
    <simpleType name="statusType">
      <restriction base="token">
        <enumeration value="ok" />
        <enumeration value="unsatisfactory" />
      </restriction>
    </simpleType>
    <complexType name="issuesType">
      <sequence>
        <element name="issue" type="rriReporting:issueType"
          maxOccurs="unbounded" />
      </sequence>
    </complexType>
    <complexType name="issueType">
      <attribute name="date" type="rriReporting:issueDateType"
        use="required" />
      <attribute name="description" type="rriReporting:descriptionType"
        use="required" />
    </complexType>
    <simpleType name="issueDateType">
      <restriction base="token">
        <pattern
          value="\d{4}-(0[1-9]|1[012])(-(0[1-9]|[12][0-9]|3[01]))?" />
      </restriction>
    </simpleType>
    <simpleType name="descriptionType">
      <restriction base="token">
        <enumeration value="Missing_Deposit_Full" />
        <enumeration value="Missing_Deposit_Diff" />
        <enumeration value="Invalid_Deposit_Full" />
        <enumeration value="Invalid_Deposit_Diff" />
        <enumeration value="No_Report_Received" />
      </restriction>
    </simpleType>
</schema>
<CODE ENDS>
```

7.5. RRI Notifications Object

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

  <import namespace="urn:ietf:params:xml:ns:rdeNotification-1.0" />
  <element name="notifications"
    type="rdeNotifications:notificationsType"/>
  <complexType name="notificationsType">
    <sequence>
      <element name="receivedNotification" maxOccurs="unbounded"
       minOccurs="0">
        <complexType>
          <sequence>
            <element name="received" type="dateTime" />
            <element ref="rdeNotification:notification" />
          </sequence>
        </complexType>
      </element>
    </sequence>
  </complexType>
</schema>
<CODE ENDS>
```

7.6. RRI Reports Object

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

  <import namespace="urn:ietf:params:xml:ns:rdeReport-1.0" />
  <element name="reports" type="rdeReports:reportsType"/>
  <complexType name="reportsType">
    <sequence>
      <element name="receivedReport" maxOccurs="unbounded"
       minOccurs="0">
        <complexType>
          <sequence>
            <element name="received" type="dateTime" />
            <element ref="rdeReport:report" />
          </sequence>
        </complexType>
      </element>
    </sequence>
  </complexType>
</schema>
<CODE ENDS>
```

7.7. Maintenance Window

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

    <element name="schedule" type="schedule:scheduleType"/>
    <element name="schedules" type="schedule:schedulesType"/>
    <complexType name="scheduleType">
        <sequence>
          <element name="name" type="schedule:nameType"/>
          <element name="description" type="schedule:descriptionType"/>
          <element name="startTime" type="dateTime"/>
          <element name="endTime" type="dateTime"/>
          <element name="version" type="unsignedShort"/>
        </sequence>
         <attribute name="enabled" type="boolean"
         use="required"/>
         <attribute name="scheduleId" type="schedule:uuidType"/>
    </complexType>
    <complexType name="schedulesType">
        <sequence>
            <element name="schedule" type="schedule:scheduleType"
            maxOccurs="unbounded" minOccurs="0"/>
        </sequence>
    </complexType>
    <simpleType name="uuidType">
        <restriction base="token">
          <pattern
  value="[a-f0-9]{8}-[a-f0-9]{4}-[a-f0-9]{4}-[a-f0-9]{4}-[a-f0-9]{12}">
          </pattern>
        </restriction>
    </simpleType>
    <simpleType name="nameType">
        <restriction base="token">
            <minLength value="1"/>
        </restriction>
    </simpleType>
    <simpleType name="descriptionType">
        <restriction base="token">
            <minLength value="1"/>
        </restriction>
    </simpleType>
</schema>
<CODE ENDS>
```

7.8. SLAM Probe Node List

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

  <element name="probeNode" type="probeNode:probeNodeType" />
  <element name="probeNodes" type="probeNode:probeNodesType" />
  <complexType name="probeNodeType">
    <sequence>
      <element name="city" type="token" />
      <element name="addr" type="probeNode:addrType"
            maxOccurs="unbounded" />
    </sequence>
  </complexType>
  <complexType name="probeNodesType">
    <sequence>
      <element name="probeNode" type="probeNode:probeNodeType"
            maxOccurs="unbounded" />
      <element name="updateTime" type="dateTime" />
      <element name="version" type="unsignedShort" />
    </sequence>
  </complexType>
  <complexType name="addrType">
    <simpleContent>
      <extension base="probeNode:addrStringType">
        <attribute name="ip" type="probeNode:ipType"
                use="required" />
      </extension>
    </simpleContent>
  </complexType>
  <simpleType name="addrStringType">
    <restriction base="token">
      <minLength value="3" />
      <maxLength value="45" />
    </restriction>
  </simpleType>
  <simpleType name="ipType">
    <restriction base="token">
      <enumeration value="v4" />
      <enumeration value="v6" />
    </restriction>
  </simpleType>
</schema>
<CODE ENDS>
```

8. Acknowledgements

Special suggestions that have been incorporated into this document were provided by David Kipling, James Gould, Gregory Zaltsman, Brett Carr, and Harel Efraim.

9. Change History

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

9.1. Version 00

Initial version.

9.2. Version 01

o   <rdeReport:report> and <rdeNotification:notification> moved from escrow drafts to this draft

o   Added <crDate> to <rdeReport:report>

o   <reDate> element of <rdeReport:report> is now OPTIONAL

o   Added <deaName> element to <rdeNotification:notification>

o   <rydeSpecEscrow> and <rydeSpecMapping> added to the draft

o   Several report elements are OPTIONAL to support async interfaces between Registry Operators and Data Escrow Agents

o   Added <TLD> and <id> to registry-escrow-report interface in order to make the interface idempotent and support async RyO-DEA interfaces

o   Added <TLD> to escrow-agent-notification interface

o   The escrow-agent-notification uses POST and not PUT, this has been fixed

o   Several clarifications

9.3. Version 02

o   Added and updated several result codes.

o   Added <version> element.

o   Added Content-type definition.

9.4. **Version 03**

   o Added several result codes.

   o unsignedShort is now used for result code in iirdea schema.

   o Enumeration was removed from the iirdea schema.

9.5. **Version 04**

   o Added result codes: 2207 and 2208.

   o Removed result codes: 2203.

   o Added clarification regarding the support of HTTP streams.

9.6. **Version 05**

   o Added result codes: 2007 and 2008.

9.7. **Version 06**

   o Added clarification of error code HTTP/403 in Section 6.

9.8. **Version 07**

   o Added Section 5: "Monitoring compliance with the New gTLD Base
      Agreement".

9.9. **Version 08**

   o Reorganized specification structure to allow easier references
      from new specifications expanding functionality in the ICANN
      Registry Interfaces.

   o Added Section 1.4 to document object definitions, previously
      defined in other sections.

   o Added <rriReporting>, <notifications>, and <reports> object
      descriptions to Section 1.4, and schema definitions to Section 7.

   o Renamed Section 5 title as "Monitoring the reporting status".

   o Updated element <rydeSpecMapping> as OPTIONAL in the <rdeReport>
      schema.

   o Added OPTIONAL attribute "domainCount" to the <iirdea:result>
      element.

o Added OPTIONAL element <results> to the <rdeNotification> schema.

o Added result codes: 2105, 2209, 2210 and 2211.

o Added "gTLD Base Registry Agreement" references.

o Added clarifications to Section 6.

9.10. Version 09

o Standardized XSD schema validation error message for notifications
   and reports.

o Element <lastFullDate> made optional in the <rriReporting> schema.

o Separated example RRI interface responses for successful and
   unsuccessful input.

9.11. Version 10

1. Ping update.

9.12. Version 11

1. Ping update.

9.13. Version 12

1. Ping update.

9.14. Version 13

1. IANA Considerations section added.

2. Implementation section added.

3. Internationalization Considerations status section added.

4. Security section added.

5. Editorial updates.

9.15. Version 14

1. Ping update.

1. Added description and technical details for the interfaces for
    management of maintenance windows and list of probe nodes.

2. I-D was restructured to make it easier to read.

3. Editorial updates.

1. Added interface description for supporting maintenance window
    objects and retrieving the IP addresses of the probe nodes used
    in the SLA Monitoring System (SLAM).

2. I-D was restructured to make it easier to read.

3. Editorial updates.

1. Ping update.

The interfaces described in this document use 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.

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

Registration request for the RRI IIRDEA Result namespace:

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

    Registrant Contact: ICANN <tech-services@icann.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 RRI IIRDEA Result XML schema:

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

Registrant Contact: ICANN <tech-services@icann.org>

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

See section Section 7.1 of this document.

Registration request for the RRI Report namespace:

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

Registrant Contact: ICANN <tech-services@icann.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 RRI Report schema:

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

Registrant Contact: ICANN <tech-services@icann.org>

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

See section Section 7.2 of this document.

Registration request for the RRI Notification namespace:

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

Registrant Contact: ICANN <tech-services@icann.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 RRI Notification XML schema:

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

Registrant Contact: ICANN <tech-services@icann.org>

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

See section [Section 7.3](#) of this document.

Registration request for the RRI Reporting Summary namespace:

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

Registrant Contact: ICANN <tech-services@icann.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 RRI Reporting Summary XML schema:

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

Registrant Contact: ICANN <tech-services@icann.org>

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

See section [Section 7.4](#) of this document.

Registration request for the RRI Notifications namespace:

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

Registrant Contact: ICANN <tech-services@icann.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 RRI Notifications XML schema:

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

Registrant Contact: ICANN <tech-services@icann.org>

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

See section [Section 7.5](#) of this document.

Registration request for the RRI Reports namespace:

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

Registrant Contact: ICANN <tech-services@icann.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 RRI Reports XML schema:

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

Registrant Contact: ICANN <tech-services@icann.org>

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

See section [Section 7.6](#) of this document.

Registration request for the Maintenance Window namespace:

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

Registrant Contact: ICANN <tech-services@icann.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 Maintenance Window XML schema:

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

Registrant Contact: ICANN <tech-services@icann.org>

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

See section [Section 7.7](#) of this document.

Registration request for the SLAM Probe Node List namespace:

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

   Registrant Contact: ICANN <tech-services@icann.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 SLAM Probe Node List XML schema:

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

   Registrant Contact: ICANN <tech-services@icann.org>

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

   See section Section 7.8 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 and Registries using this specification.

Level of maturity: production.

Coverage: all aspects of this specification are implemented.

Version compatibility: versions 00 - 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**

The interfaces described in this document MUST be provided using HTTPS. The recommendations in [RFC7525] MUST be implemented.

14. **References**

14.1. **Normative References**

[ICANN-GTLD-RA-20170731]
          ICANN, "Base Registry Agreement 2017-07-31", July 2017,
          <https://www.icann.org/en/registry-agreements/base-agreement>.

[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>.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688,
          DOI 10.17487/RFC3688, January 2004,
          <https://www.rfc-editor.org/info/rfc3688>.

[RFC4122]   Leach, P., Mealling, M., and R. Salz, "A Universally
            Unique IDentifier (UUID) URN Namespace", RFC 4122,
            DOI 10.17487/RFC4122, July 2005,
            <https://www.rfc-editor.org/info/rfc4122>.

[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>.

[RFC8909]   Lozano, G., "Registry Data Escrow Specification",
            RFC 8909, DOI 10.17487/RFC8909, November 2020,
            <https://www.rfc-editor.org/info/rfc8909>.

[RFC9022]   Lozano, G., Gould, J., and C. Thippeswamy, "Domain Name
            Registration Data (DNRD) Objects Mapping", RFC 9022,
            DOI 10.17487/RFC9022, May 2021,
            <https://www.rfc-editor.org/info/rfc9022>.

[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/>.

14.2. Informative References

[RFC5890]   Klensin, J., "Internationalized Domain Names for
            Applications (IDNA): Definitions and Document Framework",
            RFC 5890, DOI 10.17487/RFC5890, August 2010,
            <https://www.rfc-editor.org/info/rfc5890>.

[RFC7525]   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/rfc7525>.

   [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>.

Authors' Addresses

   Gustavo Lozano
   ICANN
   12025 Waterfront Drive, Suite 300
   Los Angeles 90292
   US

   Phone: +1.3103015800
   Email: gustavo.lozano@icann.org

   Eduardo Alvarez
   ICANN
   12025 Waterfront Drive, Suite 300
   Los Angeles 90292
   US

   Phone: +1.3103015800
   Email: eduardo.alvarez@icann.org