Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: October 4, 2015

J. Gould
VeriSign, Inc.
F. Obispo
L. Munoz
Uniregistry Corp.

April 2, 2015

Extensible Provisioning Protocol (EPP) Internationalized Domain Name (IDN) Table Mapping
draft-gould-idn-table-02

Abstract

This document describes an Extensible Provisioning Protocol (EPP) mapping for getting Internationalized Domain Name (IDN) Table information for the registration of IDNs, using the EPP domain name mapping, and optionally with the IDN mapping extension. An IDN Table defines the valid set of characters (code points) that can be used in a domain name. Code points may overlap across IDN Tables and the IDN Tables supported by the servers are up to server policy.

The IDN Table information can be used to validate an IDN prior to registration, can be cached by the client for pre-validation, can be used to select the best IDN Table for the IDN, and can be used to know if and what IDN Table Identifier to pass in a domain create.

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 October 4, 2015.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction 3
  1.1. Conventions Used in This Document 3
2. Object Attributes 4
  2.1. IDN Table Identifier 4
  2.2. Domain Name 4
3. EPP Command Mapping 4
  3.1. EPP Query Commands 5
    3.1.1. EPP <check> Command 5
      3.1.1.1. Domain Check Form 5
      3.1.1.2. Table Check Form 7
    3.1.2. EPP <info> Command 9
      3.1.2.1. Domain Info Form 9
      3.1.2.2. Table Info Form 13
      3.1.2.3. List Info Form 17
    3.1.3. EPP <transfer> Command 19
  3.2. EPP Transform Commands 20
    3.2.1. EPP <create> Command 20
    3.2.2. EPP <delete> Command 20
    3.2.3. EPP <renew> Command 20
    3.2.4. EPP <transfer> Command 20
    3.2.5. EPP <update> Command 20
4. Formal Syntax 20
  4.1. IDN Table Mapping Schema 21
5. IANA Considerations 25
  5.1. XML Namespace 26
  5.2. EPP Extension Registry 26
6. Security Considerations 26
7. Normative References 26
Appendix A. Change History 27
  A.1. Change from 00 to 01 27
  A.2. Change from 01 to 02 27

Authors' Addresses 27

1. Introduction

This document describes an extension mapping for version 1.0 of the Extensible Provisioning Protocol (EPP) [RFC5730]. This EPP mapping provides Internationalized Domain Name (IDN) Table information for the registration of IDNs, using the EPP domain name mapping [RFC5731], and optionally with the IDN mapping extension [I-D.ietf-eppext-idnmap]. An IDN Table defines the valid set of characters (code points) that can be used in a domain name. Code points may overlap across IDN Tables and the IDN Tables supported by the servers are up to server policy. This mapping provides the information clients need to register IDNs across a variety of servers with differing IDN policies. The IDN Table Mapping can be used for the following:

"Validate IDN Domain Name" Validate that an IDN meets the server IDN policy. The validation can be done prior to submitting a domain create, per [RFC5731].

"Get IDN Tables Matching IDN Domain Name Along with Meta-Data" Since IDN Table code points may overlap, the mapping can be used to identify the matching set of IDN Tables (language or script), along with the IDN Table meta-data.

"Cache IDN Table Code Points" Clients can query for the complete list of IDN Tables and can get the IDN Table meta-data, based on server policy, to support pre-validation in the client.

"Get the IDN Table Identifier to Pass with a Domain Create" Each IDN Table includes a server unique IDN Table Identifier that may be used as the value of the <idn:table> element in the IDN mapping extension [I-D.ietf-eppext-idnmap]. A flag indicates whether the IDN mapping extension [I-D.ietf-eppext-idnmap] is needed for the domain name.

1.1. Conventions Used in This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

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

In examples, "C:" represents lines sent by a protocol client and "S:" represents lines returned by a protocol server. Indentation and

white space in examples are provided only to illustrate element relationships and are not a REQUIRED feature of this protocol.

"idnTable-1.0" is used as an abbreviation for "urn:ietf:params:xml:ns:idnTable-1.0". The XML namespace prefix "idnTable" is used, but implementations MUST NOT depend on it and instead employ a proper namespace-aware XML parser and serializer to interpret and output the XML documents.

2. Object Attributes

An EPP IDN Table object has attributes and associated values that can help in the registration of IDNs. This section describes each type in detail. The formal syntax for the attribute values described here can be found in the "Formal Syntax" section of this document and in the appropriate normative references.

2.1. IDN Table Identifier

The IDN Table Identifier is a server-defined unique value for an IDN Table object. It is represented using an <idnTable:table> element or an <idnTable:name> element. The IDN Table Identifier is used in the <check> and <info> commands and responses. The IDN Table Identifier MAY also be used as the value for the IDN mapping extension [I-D.ietf-eppext-idnmap] <idn:table> element with a domain create of an IDN object.

2.2. Domain Name

A Domain Name, as represented by an <idnTable:domain> element, is used to enable validating the code points against the server IDN Tables and IDN policies and for retrieving IDN Table information associated with the domain name. The Domain Name MUST be represented as either a U-label or A-label as defined in [RFC5890].

An OPTIONAL attribute "form" MAY be used to specify the representation. When present, the "form" attribute MUST be set to either "aLabel" for A-label or "uLabel" for U-label, depending on the chosen representation for the domain name. The default "form" attribute is "aLabel" for A-label.

3. EPP Command Mapping

A detailed description of the EPP syntax and semantics can be found in the EPP core protocol specification [RFC5730].

3.1. EPP Query Commands

EPP provides three commands to retrieve object information: <check> to determine if an object is known to the server, <info> to retrieve detailed information associated with an object, and <transfer> to retrieve object transfer status information.

3.1.1. EPP <check> Command

There are two forms of the EPP <check> command: the Domain Check Form (Section 3.1.1.1) and the Table Check Form (Section 3.1.1.2).

3.1.1.1. Domain Check Form

The Domain Check Form is used to check the validity of the domain name against the server IDN Tables and IDN policies, return whether the IDN mapping extension [I-D.ietf-eppext-idnmap] is needed with a domain <create> command, and provide the matching list of IDN Table Identifiers. This mapping is not intended to determine the availability of the domain name.

In addition to the standard EPP command elements, the <check> command MUST contain an <idnTable:check> element that identifies the idnTable namespace. The <idnTable:check> element in the Domain Check Form contains the following child elements:

<idnTable:domain> One or more <idnTable:domain> elements that contain the fully qualified names of the domain objects, as defined in Section 2.2, to validate.

Example Domain Check Form <check> command with three IDNs:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <check>
C:      <idnTable:check
C:       xmlns:idnTable="urn:ietf:params:xml:ns:idnTable-1.0">
C:        <idnTable:domain
C:         form="uLabel">idn1.example</idnTable:domain>
C:        <idnTable:domain
C:         form="aLabel">idn2.example</idnTable:domain>
C:        <idnTable:domain>idn3.example</idnTable:domain>
C:      </idnTable:check>
C:    </check>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

When a <check> command has been processed successfully, the EPP <resData> element MUST contain a child <idnTable:chkData> element that identifies the idnTable namespace. The <idnTable:chkData> element in the Domain Check Form contain one or more <idnTable:domain> elements that contain the following child elements:

<idnTable:name> The fully qualified name of the domain object, as defined in Section 2.2. The element MUST contain a "valid" attribute whose value indicates whether the domain name is valid according to the server IDN Tables and IDN policies. A value of "1" or "true" means that the domain name is valid according to the server IDN Tables and policies. A value of "0" or "false" means that the domain name is not valid according to the server IDN Tables and policies. The element MAY contain an "idnmap" attribute value that indicates whether the server requires the use of the IDN mapping extension [I-D.ietf-eppext-idnmap] with a domain create of the domain name. A value of "1" or "true" means that the IDN mapping extension [I-D.ietf-eppext-idnmap] is required using one of the <idnTable:table> values. A value of "0" or "false" means that the IDN mapping extension [I-D.ietf-eppext-idnmap] is not required.

<idnTable:reason> OPTIONAL reason that the domain name is not valid. If present, this element contains server-specific text to help explain why the domain name is not valid. This text MUST be represented in the response language previously negotiated with the client; an OPTIONAL "lang" attribute MAY be present to identify the language if the negotiated value is something other than the default value of "en" (English).

<idnTable:table> Zero or more OPTIONAL <idnTable:table> elements that contain the server defined IDN Table Identifier, as defined in Section 2.1, that matches the code points of the <idnTable:name> element. The <idnTable:table> MAY be used as the value of the <idn:table> element in the IDN mapping extension [I-D.ietf-eppext-idnmap] for creating the IDN object or MAY be used as the value of the <idnTable:table> element of the Table Info Form (Section 3.1.2.2) <info> command, as described in Section 3.1.2, to retrieve more information about the IDN Table.

Example Domain Check Form <check> response with three IDNs:

```
S:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S:  <response>
S:    <result code="1000">
S:      <msg>Command completed successfully</msg>
S:    </result>
S:    <resData>
S:      <idnTable:chkData
S:       xmlns:idnTable="urn:ietf:params:xml:ns:idnTable-1.0">
S:          <idnTable:domain>
S:             <idnTable:name valid="true">
S:             idn1.example</idnTable:name>
S:             <idnTable:table>CHI</idnTable:table>
S:          </idnTable:domain>
S:          <idnTable:domain>
S:             <idnTable:name valid="true" idnmap="true">
S:             idn2.example</idnTable:name>
S:             <idnTable:table>CHI</idnTable:table>
S:             <idnTable:table>JPN</idnTable:table>
S:          </idnTable:domain>
S:          <idnTable:domain>
S:             <idnTable:name valid="false">
S:             idn3.example</idnTable:name>
S:             <idnTable:reason>Commingled scripts
S:             </idnTable:reason>
S:          </idnTable:domain>
S:      </idnTable:chkData>
S:    </resData>
S:    <trID>
S:      <clTRID>ABC-12345</clTRID>
S:      <svTRID>54321-XYZ</svTRID>
S:    </trID>
S:  </response>
S:</epp>
```

3.1.1.2. Table Check Form

The Table Check Form is used to check the existence of an IDN Table using the IDN Table Identifier represented by the <idnTable:table> element.

In addition to the standard EPP command elements, the <check> command MUST contain an <idnTable:check> element that identifies the idnTable namespace. The <idnTable:check> element in the Table Check Form contains the following child elements:

<idnTable:table> One or more <idnTable:table> elements that contain the IDN Table Identifier, as defined in Section 2.1, to check for existence.

Example Table Check Form <check> command with three IDN Table Identifiers:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <check>
C:      <idnTable:check
C:       xmlns:idnTable="urn:ietf:params:xml:ns:idnTable-1.0">
C:        <idnTable:table>CHI</idnTable:table>
C:        <idnTable:table>JPN</idnTable:table>
C:        <idnTable:table>INVALID</idnTable:table>
C:      </idnTable:check>
C:    </check>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

When a <check> command has been processed successfully, the EPP <resData> element MUST contain a child <idnTable:chkData> element that identifies the idnTable namespace. The <idnTable:chkData> element in the Table Check Form contains the following child elements:

<idnTable:table> One or more <idnTable:table> elements that contain the IDN Table Identifier, as defined in Section 2.1. The element MUST contain an "exists" attribute whose value indicates the existence of the IDN Table Identifier. A value of "1" or "true" means that the IDN Table Identifier exists. A value of "0" or "false" means that the IDN Table Identifier does not exist.

Example Table Check Form <check> response with three IDN Table Identifiers:

```
S:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S:  <response>
S:    <result code="1000">
S:      <msg>Command completed successfully</msg>
S:    </result>
S:    <resData>
S:      <idnTable:chkData
S:       xmlns:idnTable="urn:ietf:params:xml:ns:idnTable-1.0">
S:          <idnTable:table exists="true">CHI
S:          </idnTable:table>
S:          <idnTable:table exists="true">JPN
S:          </idnTable:table>
S:          <idnTable:table exists="false">INVALID
S:          </idnTable:table>
S:      </idnTable:chkData>
S:    </resData>
S:    <trID>
S:      <clTRID>ABC-12345</clTRID>
S:      <svTRID>54321-XYZ</svTRID>
S:    </trID>
S:  </response>
S:</epp>
```

3.1.2. EPP <info> Command

There are three forms of the EPP <info> command: the Domain Info Form (Section 3.1.2.1), the Table Info Form (Section 3.1.2.2), and the List Info Form (Section 3.1.2.3).

3.1.2.1. Domain Info Form

The Domain Info Form is used to validate the domain name code points against the IDN Tables and IDN policies, and to return the matching IDN Table meta-data. The domain name, as defined in Section 2.2 can be provided as either a U-label or A-label.

In addition to the standard EPP command elements, the <info> command MUST contain an <idnTable:info> element that identifies the idnTable namespace. The <idnTable:info> element in the Domain Info Form contains the following child elements:

<idnTable:domain> The domain name, as defined in Section 2.2, to validate against the IDN Tables and IDN policies, and to retrieve the matching IDN Table meta-data.

Example Domain Info Form <info> command using a U-label Domain Name:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <info>
C:      <idnTable:info
C:       xmlns:idnTable="urn:ietf:params:xml:ns:idnTable-1.0">
C:        <idnTable:domain>idn1.example</idnTable:domain>
C:      </idnTable:info>
C:    </info>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

Example Domain Info Form <info> command using an A-label Domain Name:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <info>
C:      <idnTable:info
C:       xmlns:idnTable="urn:ietf:params:xml:ns:idnTable-1.0">
C:        <idnTable:domain>xn--idn1.example</idnTable:domain>
C:      </idnTable:info>
C:    </info>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

When an <info> command has been processed successfully, the EPP <resData> element MUST contain a child <idnTable:infData> element that identifies the idnTable namespace. The <idnTable:infData> element in the Domain Info Form contains the <idnTable:domain> element that contains the following child elements:

<idnTable:name> The fully qualified name of the domain object, as defined in Section 2.2. The element MUST contain a "valid" attribute whose value indicates whether the domain name is valid according to the server IDN Tables and IDN policies. A value of "1" or "true" means that the domain name is valid according to the server IDN Tables and policies. A value of "0" or "false" means that the domain name is not valid according to the server IDN Tables and policies. The element MAY contain an "idnmap"

attribute value that indicates whether the server requires the use of the IDN mapping extension [I-D.ietf-eppext-idnmap] with a domain create of the domain name. A value of "1" or "true" means that the IDN mapping extension [I-D.ietf-eppext-idnmap] is required using one of the <idnTable:table> <idnTable:name> values. A value of "0" or "false" means that the IDN mapping extension [I-D.ietf-eppext-idnmap] is not required.

<idnTable:uname> or <idnTable:aname> OPTIONAL U-label or A-label form of the domain name value of the <idnTable:name> element in the opposite form, as defined in Section 2.2.

<idnTable:table> Zero or more OPTIONAL <idnTable:table> elements that provide the IDN Table meta-data information. The <idnTable:table> element contains the following child elements:

<idnTable:name> Server defined IDN Table Identifier, as defined in Section 2.1.

<idnTable:type> The type of the IDN Table with the possible values of "language", to reflect a Language IDN Table, and "script", to reflect a Script IDN Table.

<idnTable:description> Server defined description of the IDN Table. This text MUST be represented in the response language previously negotiated with the client; an OPTIONAL "lang" attribute MAY be present to identify the language if the negotiated value is something other than the default value of "en" (English).

<idnTable:variantGen> OPTIONAL boolean flag indicating that domains created using the IDN Table will have IDN variants generated. The management of variants is up to server policy.

Example Domain Info Form <info> response for a U-label Domain Name:

```
S:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S:  <response>
S:    <result code="1000">
S:      <msg>Command completed successfully</msg>
S:    </result>
S:    <resData>
S:      <idnTable:infData
S:       xmlns:idnTable="urn:ietf:params:xml:ns:idnTable-1.0">
S:        <idnTable:domain>
S:          <idnTable:name valid="true">idn1.example
S:          </idnTable:name>
S:          <idnTable:aname>xn--idn1.example</idnTable:aname>
S:          <idnTable:table>
S:            <idnTable:name>THAI</idnTable:name>
S:            <idnTable:type>script</idnTable:type>
S:            <idnTable:description lang="en">Thai
S:            </idnTable:description>
S:            <idnTable:variantGen>false</idnTable:variantGen>
S:          </idnTable:table>
S:        </idnTable:domain>
S:      </idnTable:infData>
S:    </resData>
S:    <trID>
S:      <clTRID>ABC-12345</clTRID>
S:      <svTRID>54321-XYZ</svTRID>
S:    </trID>
S:  </response>
S:</epp>
```

Example Domain Info Form <info> response for an A-label Domain Name:

```
S:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S:  <response>
S:    <result code="1000">
S:      <msg>Command completed successfully</msg>
S:    </result>
S:    <resData>
S:      <idnTable:infData
S:       xmlns:idnTable="urn:ietf:params:xml:ns:idnTable-1.0">
S:        <idnTable:domain>
S:          <idnTable:name valid="true" idnmap="true">
S:          xn--idn1.example
S:          </idnTable:name>
S:          <idnTable:uname>idn1.example</idnTable:uname>
S:          <idnTable:table>
S:            <idnTable:name>CHI</idnTable:name>
S:            <idnTable:type>language</idnTable:type>
S:            <idnTable:description>Chinese (CHI)
S:            </idnTable:description>
S:            <idnTable:variantGen>true</idnTable:variantGen>
S:          </idnTable:table>
S:          <idnTable:table>
S:            <idnTable:name>JPN</idnTable:name>
S:            <idnTable:type>language</idnTable:type>
S:            <idnTable:description>Japanese (JPN)
S:            </idnTable:description>
S:            <idnTable:variantGen>false</idnTable:variantGen>
S:          </idnTable:table>
S:        </idnTable:domain>
S:      </idnTable:infData>
S:    </resData>
S:    <trID>
S:      <clTRID>ABC-12345</clTRID>
S:      <svTRID>54321-XYZ</svTRID>
S:    </trID>
S:  </response>
S:</epp>
```

3.1.2.2. Table Info Form

The Table Info Form is used to retrieve information associated with an IDN Table object. The information provided meta-data about the IDN Table object.

In addition to the standard EPP command elements, the <info> command MUST contain an <idnTable:info> element that identifies the

idnTable namespace. The <idnTable:info> element in the Table Info Form contains the following child elements:

<idnTable:table> Contains the IDN Table Identifier, as defined in Section 2.1, of the IDN Table object to be queried.

Example Table Info Form <info> command for the "CHI" IDN Table Identifier, which represents a Language IDN Table:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <info>
C:      <idnTable:info
C:       xmlns:idnTable="urn:ietf:params:xml:ns:idnTable-1.0">
C:        <idnTable:table>CHI</idnTable:table>
C:      </idnTable:info>
C:    </info>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

Example Table Info Form <info> command for the "THAI" IDN Table Identifier, which represents a Script IDN Table:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <info>
C:      <idnTable:info
C:       xmlns:idnTable="urn:ietf:params:xml:ns:idnTable-1.0">
C:        <idnTable:table>THAI</idnTable:table>
C:      </idnTable:info>
C:    </info>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

When an <info> command has been processed successfully, the EPP <resData> element MUST contain a child <idnTable:infData> element that identifies the idnTable namespace. The <idnTable:infData> element in the Table Info Form contains the <idnTable:table> element that contains the following child elements:

<idnTable:name> Server defined IDN Table Identifier, as defined in Section 2.1.

<idnTable:type> The type of the IDN Table with the possible values of "language", to reflect a Language IDN Table, and "script", to reflect a Script IDN Table.

<idnTable:description> Server defined description of the IDN Table. This text MUST be represented in the response language previously negotiated with the client; an OPTIONAL "lang" attribute MAY be present to identify the language if the negotiated value is something other than the default value of "en" (English).

<idnTable:upDate> Contains the date and time that the IDN Table was created or last updated.

<idnTable:version> OPTIONAL server defined version number of the IDN Table.

<idnTable:effectiveDate> OPTIONAL effective date for the IDN Table.

<idnTable:variantGen> OPTIONAL boolean flag indicating that domains created using the IDN Table will have IDN variants generated. The management of variants is up to server policy.

<idnTable:url> OPTIONAL URL for downloading the IDN Table with the applicable set of code points and rules.

Example Table Info Form <info> response for the "CHI" IDN Table Identifier, which represents a Language IDN Table:

```
S:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S:  <response>
S:    <result code="1000">
S:      <msg>Command completed successfully</msg>
S:    </result>
S:    <resData>
S:      <idnTable:infData
S:       xmlns:idnTable="urn:ietf:params:xml:ns:idnTable-1.0">
S:        <idnTable:table>
S:           <idnTable:name>CHI</idnTable:name>
S:           <idnTable:type>language</idnTable:type>
S:           <idnTable:description lang="en">Chinese (CHI)
S:           </idnTable:description>
S:           <idnTable:upDate>2015-02-04T09:30:00.0Z
S:           </idnTable:upDate>
S:           <idnTable:version>1.0</idnTable:version>
S:           <idnTable:effectiveDate>2014-11-24
S:           </idnTable:effectiveDate>
S:           <idnTable:variantGen>true</idnTable:variantGen>
S:           <idnTable:url>
S:   https://www.iana.org/domains/idn-tables/tables/tld_chi_1.0.txt
S:           </idnTable:url>
S:         </idnTable:table>
S:      </idnTable:infData>
S:    </resData>
S:    <trID>
S:      <clTRID>ABC-12345</clTRID>
S:      <svTRID>54321-XYZ</svTRID>
S:    </trID>
S:  </response>
S:</epp>
```

Example Table Info Form <info> response for the "THAI" IDN Table Identifier, which represents a Script IDN Table:

```
S:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S:  <response>
S:    <result code="1000">
S:      <msg>Command completed successfully</msg>
S:    </result>
S:    <resData>
S:      <idnTable:infData
S:       xmlns:idnTable="urn:ietf:params:xml:ns:idnTable-1.0">
S:        <idnTable:table>
S:           <idnTable:name>THAI</idnTable:name>
S:           <idnTable:type>script</idnTable:type>
S:           <idnTable:description>Thai</idnTable:description>
S:           <idnTable:upDate>2014-08-16T09:20:00.0Z
S:           </idnTable:upDate>
S:           <idnTable:version>1.0</idnTable:version>
S:           <idnTable:effectiveDate>2014-11-24
S:           </idnTable:effectiveDate>
S:           <idnTable:variantGen>false</idnTable:variantGen>
S:           <idnTable:url>
S:  https://www.iana.org/domains/idn-tables/tables/tld_thai_1.0.txt
S:           </idnTable:url>
S:         </idnTable:table>
S:      </idnTable:infData>
S:    </resData>
S:    <trID>
S:      <clTRID>ABC-12345</clTRID>
S:      <svTRID>54321-XYZ</svTRID>
S:    </trID>
S:  </response>
S:</epp>
```

3.1.2.3. List Info Form

The List Info Form is used to retrieve the list of IDN Tables supported by the server. The list of IDN Table Identifiers MAY be used to query for the IDN Table information using the Table Info Form (Section 3.1.2.2).

In addition to the standard EPP command elements, the <info> command MUST contain an <idnTable:info> element that identifies the idnTable namespace. The <idnTable:info> element in the List Info Form contains the following child elements:

<idnTable:list> Empty element used as a marker to the server of the List Info Form to retrieve the list of IDN Tables.

Example List Info Form <info> command:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <info>
C:      <idnTable:info
C:       xmlns:idnTable="urn:ietf:params:xml:ns:idnTable-1.0">
C:        <idnTable:list/>
C:      </idnTable:info>
C:    </info>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

When an <info> command has been processed successfully, the EPP <resData> element MUST contain a child <idnTable:infData> element that identifies the idnTable namespace. The <idnTable:infData> element in the List Info Form contains the <idnTable:list> element that contains the following child elements:

<idnTable:table> Zero or more <idnTable:table> elements that contain the following child elements:

<idnTable:name> Server defined IDN Table Identifier, as defined in Section 2.1, that is supported by the server. The <idnTable:table> element value MAY be used as the value of the <idnTable:table> element value in the Table Info Form (Section 3.1.2.2) <info> command to retrieve the information on the IDN Table.

<idnTable:upDate> Contains the date and time that the IDN Table was created or last updated. This element can be used to determine whether a client-side cache needs to be refreshed for the IDN Table using the Table Info Form (Section 3.1.2.2).

Example List Info Form <info> response that contains three IDN Table Identifiers:

```
S:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S:  <response>
S:    <result code="1000">
S:      <msg>Command completed successfully</msg>
S:    </result>
S:    <resData>
S:      <idnTable:infData
S:       xmlns:idnTable="urn:ietf:params:xml:ns:idnTable-1.0">
S:        <idnTable:list>
S:           <idnTable:table>
S:              <idnTable:name>CHI</idnTable:name>
S:              <idnTable:upDate>2015-02-04T09:30:00.0Z
S:              </idnTable:upDate>
S:           </idnTable:table>
S:           <idnTable:table>
S:              <idnTable:name>JPN</idnTable:name>
S:              <idnTable:upDate>2015-01-01T09:40:00.0Z
S:              </idnTable:upDate>
S:           </idnTable:table>
S:           <idnTable:table>
S:              <idnTable:name>THAI</idnTable:name>
S:              <idnTable:upDate>2014-08-16T09:20:00.0Z
S:              </idnTable:upDate>
S:           </idnTable:table>
S:         </idnTable:list>
S:      </idnTable:infData>
S:    </resData>
S:    <trID>
S:      <clTRID>ABC-12345</clTRID>
S:      <svTRID>54321-XYZ</svTRID>
S:    </trID>
S:  </response>
S:</epp>
```

3.1.3. EPP <transfer> Command

Transfer semantics do not apply to IDN table objects, so there is no mapping defined for the EPP <transfer> command.

3.2. EPP Transform Commands

EPP provides five commands to transform objects: <create> to create an instance of an object, <delete> to delete an instance of an object, <renew> to extend the validity period of an object, <transfer> to manage object sponsorship changes, and <update> to change information associated with an object.

3.2.1. EPP <create> Command

Create semantics do not apply to IDN table objects, so there is no mapping defined for the EPP <create> command.

3.2.2. EPP <delete> Command

Delete semantics do not apply to IDN table objects, so there is no mapping defined for the EPP <delete> command.

3.2.3. EPP <renew> Command

Renewal semantics do not apply to IDN table objects, so there is no mapping defined for the EPP <renew> command.

3.2.4. EPP <transfer> Command

Transfer semantics do not apply to IDN table objects, so there is no mapping defined for the EPP <transfer> command.

3.2.5. EPP <update> Command

Update semantics do not apply to IDN table objects, so there is no mapping defined for the EPP <update> command.

4. Formal Syntax

One schema is presented here that is the EPP IDN Table Mapping schema.

The formal syntax presented here is a complete schema representation of the object mapping suitable for automated validation of EPP XML instances. The BEGIN and END tags are not part of the schema; they are used to note the beginning and ending of the schema for URI registration purposes.

4.1. IDN Table Mapping Schema

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

  <annotation>
    <documentation>
      Extensible Provisioning Protocol v1.0
      IDN Table Mapping.
    </documentation>
  </annotation>

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

  <!--
  Child elements found in EPP commands.
  -->
  <element name="check" type="idnTable:checkType"/>
  <element name="info" type="idnTable:infoType"/>

  <!--
   Form of the domain name.
   -->
  <simpleType name="domainFormType">
    <restriction base="token">
      <enumeration value="aLabel"/>
      <enumeration value="uLabel"/>
    </restriction>
  </simpleType>

  <!--
   Domain label element.
  -->
  <complexType name="domainLabelType">
    <simpleContent>
      <extension base="eppcom:labelType">
        <attribute name="form" type="idnTable:domainFormType"
          default="aLabel"/>
      </extension>
    </simpleContent>
  </complexType>
```

```
<!--
Child elements of check command.
-->
<complexType name="checkType">
  <sequence>
    <choice>
      <element name="table" type="eppcom:minTokenType"
        maxOccurs="unbounded"/>
      <element name="domain" type="idnTable:domainLabelType"
        maxOccurs="unbounded"/>
    </choice>
  </sequence>
</complexType>


<!--
Child elements of info command.
-->
<complexType name="infoType">
  <sequence>
    <choice>
      <element name="table" type="eppcom:minTokenType"/>
      <element name="domain" type="idnTable:domainLabelType"/>
      <element name="list"/>
    </choice>
  </sequence>
</complexType>


<!--
Child response elements.
-->
<element name="chkData" type="idnTable:chkDataType"/>
<element name="infData" type="idnTable:infDataType"/>

<!--
Child elements of check response.
-->
<complexType name="chkDataType">
  <sequence>
    <choice>
      <element name="table" type="idnTable:chkTableType"
        maxOccurs="unbounded"/>
      <element name="domain" type="idnTable:chkDomainType"
        maxOccurs="unbounded"/>
    </choice>
  </sequence>
</complexType>
```

```
<!--
Table check response element
-->
<complexType name="chkTableType">
  <simpleContent>
    <extension base="eppcom:minTokenType">
      <attribute name="exists" type="boolean" use="required"/>
    </extension>
  </simpleContent>
</complexType>


<!--
Domain name check response element
-->
<complexType name="domainNameType">
  <simpleContent>
    <extension base="eppcom:labelType">
      <attribute name="valid" type="boolean" use="required"/>
      <attribute name="idnmap" type="boolean" default="true"/>
    </extension>
  </simpleContent>
</complexType>

<!--
Domain check response element
-->
<complexType name="chkDomainType">
  <sequence>
    <element name="name" type="idnTable:domainNameType"/>
    <choice>
      <element name="reason" type="eppcom:reasonType"/>
      <element name="table" type="eppcom:minTokenType"
        maxOccurs="unbounded"/>
    </choice>
  </sequence>
</complexType>


<!--
Child elements of info response.
-->
<complexType name="infDataType">
  <sequence>
    <choice>
      <element name="table" type="idnTable:infTableType"/>
      <element name="domain" type="idnTable:infDomainType"/>
      <element name="list" type="idnTable:infListType"/>
```

```
        </choice>
      </sequence>
    </complexType>

    <!--
    Table types
    -->
    <simpleType name="tableTypeEnumType">
      <restriction base="token">
        <enumeration value="language"/>
        <enumeration value="script"/>
      </restriction>
    </simpleType>

    <complexType name="descriptionType">
      <simpleContent>
        <extension base="token">
          <attribute name="lang" type="language"/>
        </extension>
      </simpleContent>
    </complexType>


    <!--
    Table info response information
    -->
    <complexType name="infTableType">
      <sequence>
        <element name="name" type="eppcom:minTokenType"/>
        <element name="type" type="idnTable:tableTypeEnumType"/>
        <element name="description" type="idnTable:descriptionType"/>
        <element name="upDate" type="dateTime"/>
        <element name="version" type="token" minOccurs="0"/>
        <element name="effectiveDate" type="date" minOccurs="0"/>
        <element name="variantGen" type="boolean" minOccurs="0"/>
        <element name="url" type="anyURI" minOccurs="0"/>
      </sequence>
    </complexType>

    <!--
    Domain info response information
    -->
    <complexType name="infDomainType">
      <sequence>
        <element name="name" type="idnTable:domainNameType"/>
        <choice minOccurs="0">
          <element name="uname" type="eppcom:labelType"/>
          <element name="aname" type="eppcom:labelType"/>
```

```
        </choice>
        <element name="table" type="idnTable:infDomainTableType"
          minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
    </complexType>

    <!--
    Domain table info response information
    -->
    <complexType name="infDomainTableType">
      <sequence>
        <element name="name" type="eppcom:minTokenType"/>
        <element name="type" type="idnTable:tableTypeEnumType"/>
        <element name="description" type="idnTable:descriptionType"/>
        <element name="variantGen" type="boolean" minOccurs="0"/>
      </sequence>
    </complexType>


    <!--
    Table list info response information.
    -->
    <complexType name="infListType">
      <sequence>
        <element name="table" type="idnTable:infListTableType"
          minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
    </complexType>

    <!--
    Table elements in list info form.
    -->
    <complexType name="infListTableType">
      <sequence>
        <element name="name" type="eppcom:minTokenType"/>
        <element name="upDate" type="dateTime"/>
      </sequence>
    </complexType>

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

5. IANA Considerations

5.1. XML Namespace

This document uses URNs to describe XML namespaces and XML schemas conforming to a registry mechanism described in [RFC3688]. The following URI assignment is requested of IANA:

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

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

XML: See the "Formal Syntax" section of this document.

5.2. EPP Extension Registry

The EPP extension described in this document should be registered by the IANA in the EPP Extension Registry described in [RFC7451]. The details of the registration are as follows:

Name of Extension: "Extensible Provisioning Protocol (EPP) Internationalized Domain Name (IDN) Table Mapping"

Document status: Standards Track

Reference: (insert reference to RFC version of this document)

Registrant Name and Email Address: IESG, <iesg@ietf.org>

TLDs: Any

IPR Disclosure: None

Status: Active

Notes: None

6. Security Considerations

The mapping extensions described in this document do not provide any security services beyond those described by EPP [RFC5730] and protocol layers used by EPP. The security considerations described in these other specifications apply to this specification as well.

7. Normative References

[I-D.ietf-eppext-idnmap]
Obispo, F., "Internationalized Domain Name Mapping Extension for the Extensible Provisioning Protocol (EPP)", draft-ietf-eppext-idnmap-02 (work in progress), January 2015.

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

[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.

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

[RFC5890] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework", RFC 5890, August 2010.

[RFC7451] Hollenbeck, S., "Extension Registry for the Extensible Provisioning Protocol", RFC 7451, February 2015.

Appendix A. Change History

A.1. Change from 00 to 01

1. Amended XML Namespace section of IANA Considerations, added EPP Extension Registry section.

A.2. Change from 01 to 02

1. Removed support for returning the idnTable:codePoint and idnTable:codeRange elements in the table info response based on the feedback at the Registration Operations Workshop (ROW), held on March 22, 2015, prior to IETF-92.
2. Fixed info command samples that referenced the <check> element instead of the <info> element.

Authors' Addresses

James Gould
VeriSign, Inc.
12061 Bluemont Way
Reston, VA 20190
US

Email: jgould@verisign.com
URI: http://www.verisigninc.com

Francisco Obispo
Uniregistry Corp.
3-110 Governors Square
Grand Cayman, Grand Cayman KY1-1108
KY

Email: fobispo@uniregistry.com
URI: http://www.uniregistry.com/

Luis Enrique Munoz
Uniregistry Corp.
3-110 Governors Square
Grand Cayman, Grand Cayman KY1-1108
KY

Email: fobispo@uniregistry.com
URI: http://www.uniregistry.com/