Extensible Provisioning Protocol (EPP) Domain Name Mapping Extension for Internationalized Domain Name (IDN) Variants

draft-kong-epp-idn-variants-mapping-00

Abstract

This document describes an extension of Extensible Provisioning Protocol (EPP) domain name mapping for the provisioning and management of Internationalized Domain Name (IDN) Variants. Specified in XML, this mapping extends the EPP domain name mapping to provide additional features required for the provisioning of IDN variants.

Status of this Memo

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

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

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

This Internet-Draft will expire on September 6, 2012.

Copyright Notice

Copyright (c) 2012 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
# Table of Contents

1. Introduction .............................. 4  
2. Terminology .............................. 4  
3. Definitions .............................. 5  
4. Overview ................................. 5  
5. Object Attributes ....................... 6  
   5.1. OIDN ................................ 6  
   5.2. PIDN ................................ 6  
   5.3. VIDN ................................ 7  
6. EPP Command Mapping ..................... 7  
   6.1. EPP Query Commands ................... 7  
      6.1.1. EPP <check> Command .............. 7  
      6.1.2. EPP <info> Command ............... 7  
      6.1.3. EPP <transfer> Query Command ... 10  
   6.2. EPP Transform Commands ............... 11  
      6.2.1. EPP <create> Command .......... 12  
      6.2.2. EPP <delete> Command .......... 14  
      6.2.3. EPP <renew> Command .......... 15  
      6.2.4. EPP <transfer> Command ... 17  
      6.2.5. EPP <update> Command ... 18  
7. Formal Syntax ................................ 20  
8. Internationalization Considerations ........... 23  
9. IANA Considerations ........................ 23  
10. Security Considerations .................... 24  
11. Acknowledgements .......................... 24  
12. Change History ............................ 24  
   12.1. draft-kong-epp-idn-variants-mapping: Version 00 ... 24  
13. References ................................ 25  
   13.1. Normative References .................. 25  
   13.2. Informative References .............. 26  
Authors’ Addresses ........................... 26
1. Introduction

As stated in "The IDN Variant Issues Project Report [Final.Integrated.Issues.Report]", variant has been used variably to refer to, for example, a particular relationship between specific characters or code points in a particular script, or a set of alternate labels where some linkage relationship is articulated, or a desired procedure whereby names are registered in multiples, or a desired functionality causing shared behavior by some set of identifiers. Since the complexity of IDN variants, consideration is needed to be given to the registration policy on whether the variant IDN should be associated with the same registrant. Some registries adopt the policy that variant IDNs which are identified as equivalent are allocated or delegated to the same registrant. For example, the specified registration policy of Chinese Domain Name (CDN) is that a registrant can apply an original CDN in any forms: Simplified Chinese (SC) form, Traditional Chinese (TC) form, or other variant forms, then the corresponding variant CDN in SC form and that in TC form will also be delegated to the same registrant. All the other forms for the CDN are reserved and forbidden to be applied by other registrants. Moreover, any reserved variant CDN can be validated by the same registrant later. Based on such registration policy of CDN, all the variant CDNs contain same attributes.

In order to meet above requirements of the IDN variants registration, this document describes an extension of the Extensible Provisioning Protocol (EPP) domain name mapping [RFC5731] for the provisioning and management of IDN variants. This document is specified using the Extensible Markup Language (XML) 1.0 as described in [W3C.REC-xml-20040204] and XML Schema notation as described in [W3C.REC-xmlschema-1-20041028] and [W3C.REC-xmlschema-2-20041028].

The EPP core protocol specification [RFC5730] provides a complete description of EPP command and response structures. A thorough understanding of the base protocol specification is necessary to understand the extension of mapping described in this document.

This document uses lots of the concepts of the IDN, so a thorough understanding of the IDNs for Application (IDNA, described in [RFC5890], [RFC5891], and [RFC5892]) and a thorough understanding of variant approach discussed in [RFC4290] are both required.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].
"idnv-1.0" in this document is used as an abbreviation for urn:ietf:params:xml:ns:idnv-1.0.

In examples, "C:" represents lines sent by a protocol client and "S:" represents lines returned by a protocol server. Indentation and white space in examples are provided only to illustrate element relationships and are not a REQUIRED feature of this specification.

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

3. Definitions

The following definitions are used in this document:

- Original Internationalized Domain Name (OIDN), represents the valid IDN that users submitted for registration by the first time.

- Preferred Internationalized Domain Name (PIDN), represents the IDN that are commonly used and recommended according to the corresponding IDN table.

- Variant Internationalized Domain Name (VIDN), represents the IDN which is made up of the combinations of variant characters or code points, or a set of alternate domain labels according to the corresponding IDN table. In addition, OIDN and PIDN are excluded.

- Internationalized Domain Name Bundle (IDN Bundle), represents a package of IDNs which may be OIDN, corresponding PIDN or corresponding VIDNs. All of the IDNs within a same bundle MUST contain same attributes.

4. Overview

Domain registries have traditionally adopted a registration model whereby metadata relating to a domain name, such as its expiration date and sponsoring registrar, are stored as properties of the domain object. The domain object is then considered an atomic unit of registration, on which operations such as update, renewal and deletion may be performed.

IDNs, particularly variants, brought about the need for multiple domain names to be registered and managed as a single package, or registration bundle. In this model, the registry typically accepts a domain registration request (i.e. EPP domain <create> command)
containing the domain name to be registered. This domain name is referred to as the OIDN in this document. As part of the processing of the registration request, the registry generates a set of zero or more variants that are related to the OIDN, either programmatically or with the guidance of registration policies, and place them in the registration bundle together with the OIDN.

Registration policies may dictate the need for identifying a PIDN out of the set of related variants. The document uses the term VIDN to refer to the remaining variants that are neither the OIDN nor PIDN. Together, the set of OIDN, PIDN and VIDNs make up an IDN Bundle.

All of IDNs with the same bundle have the same properties, such as expiration date and sponsoring registrar, by sharing one domain object. So when users update any properties within a bundle, all of the domains’ properties in the bundle will be changed.

5. Object Attributes

This extension defines following additional elements to the EPP domain name mapping [RFC5731]. All of these additional elements can be got from <domain:info> command.

5.1. OIDN

The OIDN is an IDN with the A-label [RFC5890] form which is converted from the corresponding OIDN. In this document, its corresponding element is <idn:oidn>. An attribute "uLabel" associated with <idn:oidn> is used to represent the U-label [RFC5890] form. An optional boolean "activated" attribute, with a default true value, is used to indicate the presence of the label in the zone file.

For example: <idn:oidn uLabel="U+5B9E"U+4f8b"U+4E2D"U+56FD"> xn--fsq270a.xn--fiqs8s</idn:oidn>

5.2. PIDN

The PIDN is an IDN with the A-label [RFC5890] form which is converted from the corresponding PIDN. In this document, its corresponding element is <idn:pidn>. An attribute "uLabel" associated with <idn:pidn> is used to represent the U-label [RFC5890] form. An optional boolean "activated" attribute, with a default true value, is used to indicate the presence of the label in the zone file.

For example: <idn:pidn uLabel="U+5BE6"U+4f8b"U+4E2D"U+570B"> xn--fsqz41a.xn--fiqz9s</idn:pidn>
5.3. VIDN

The VIDN is an IDN with the A-label [RFC5890] form which is converted from the corresponding VIDN. In this document, its corresponding element is `<idn:vidn>`. An attribute "uLabel" associated with `<idn:vidn>` is used to represent the U-label [RFC5890] form. An optional boolean "activated" attribute, with a default true value, is used to indicate the presence of the label in the zone file.

For example: `<idn:vidn uLabel="U+5B9F""U+4F8b"."U+4E2D""U+570B"> xn--fsq470a.xn--fizq9s</idn:vidn>`

6. EPP Command Mapping

A detailed description of the EPP syntax and semantics can be found in the EPP core protocol specification [RFC5730]. The command mappings described here are specifically for use in provisioning and managing IDN variants via EPP.

6.1. EPP Query Commands

EPP provides three commands to retrieve domain information: `<check>` to determine if a domain object can be provisioned within a repository, `<info>` to retrieve detailed information associated with a domain object, and `<transfer>` to retrieve domain-object transfer status information.

6.1.1. EPP `<check>` Command

This extension does not add any element to the EPP `<check>` command or `<check>` response described in the EPP domain name mapping [RFC5731]. When a domain name has not been registered, but the domain which the user submitted for check is a registered domain name’s variant, `<check>` response SHOULD contain explanation in the reason field to tell the user that this domain name is an IDN variant of a registered domain name according to the corresponding IDN table, and can be validated by the registrant using `<update>` command according to the server’s policies.

6.1.2. EPP `<info>` Command

This extension does not add any element to the EPP `<info>` command described in the EPP domain mapping [RFC5731]. However, additional elements are defined for the `<info>` response.

When an `<info>` command has been processed successfully, the EPP `<resData>` element MUST contain child elements as described in the EPP
domain mapping [RFC5731]. In addition, the EPP <extension> element SHOULD contain a child <idn:infData> element that identifies the extension namespace if the domain object has data associated with this extension and based on its service policy. The <idn:infData> element contains the <idn:bundle> which has the following child elements:

- An <idn:oidn> element that contains the OIDN, along with the attributes described below.
- An OPTIONAL <idn:pidn> element that contains the PIDN, along with the attributes described below.
- Zero or more <idn:vidn> elements that contain the VIDNs, along with the attributes described below.

The above elements contain the following attributes:

- A "uLabel" attribute represents the U-label of the element.
- An optional "activated" attribute that defaults to true, indicating the presence of the variant in the zone file.
Example <info> Response for an authorized client:

S: <?xml version="1.0" encoding="UTF-8" standalone="no"?>
S: <epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S:   <response>
S:     <result code="1000">
S:       <msg>Command completed successfully</msg>
S:     </result>
S:     <resData>
S:       <domain:infData
S:         xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
S:         <domain:name>xn--fsq270a.xn--fiqs8s</domain:name>
S:         <domain:roid>58812678-domain</domain:roid>
S:         <domain:status s="ok"/>
S:         <domain:registrant>123</domain:registrant>
S:         <domain:contact type="admin">123</domain:contact>
S:         <domain:contact type="tech">123</domain:contact>
S:         <domain:ns>
S:           <domain:hostObj>ns1.example.cn</domain:hostObj>
S:         </domain:ns>
S:         <domain:clID>ClientX</domain:clID>
S:         <domain:crID>ClientY</domain:crID>
S:         <domain:crDate>2011-04-03T22:00:00.0Z</domain:crDate>
S:         <domain:exDate>2012-04-03T22:00:00.0Z</domain:exDate>
S:         <domain:authInfo>
S:           <domain:pw>2fooBAR</domain:pw>
S:         </domain:authInfo>
S:       </domain:infData>
S:       <extension>
S:         <idn:infData
S:           xmlns:idn="urn:ietf:params:xml:ns:idnv-1.0">
S:           <idn:bundle>
S:             <idn:oidn uLabel="U+5B9E"U+4f8b"U+4E2D"U+56FD"
S:                activated="true">xn--fsq270a.xn--fiqs8s</idn:oidn>
S:             <idn:pidn uLabel="U+5B9E"U+4f8b"U+4E2D"U+570B"
S:                activated="true">xn--fsqz41a.xn--fiqz9s</idn:pidn>
S:             <idn:vidn uLabel="U+5B9F"U+4f8b"U+4E2D"U+570B"
S:                activated="true">xn--fsq470a.xn--fiqz9s</idn:vidn>
S:             </idn:bundle>
S:         </idn:infData>
S:       </extension>
S:     </resData>
S:   </response>
S: </epp>
<info> Response for the unauthorized client has not been changed, see [RFC5731] for detail.

An EPP error response MUST be returned if an <info> command cannot be processed for any reason.

6.1.3. EPP <transfer> Query Command

This extension does not add any element to the EPP <transfer> command described in the EPP domain mapping [RFC5731]. However, additional elements are defined for the <transfer> response.

When a <transfer> command has been processed successfully, the EPP <trnData> element MUST contain child elements as described in the EPP domain mapping [RFC5731]. In addition, the EPP <extension> element SHOULD contain a child <idn:trnData> element that identifies the extension namespace if the domain object has data associated with this extension and based on its service policy. The <idn:trnData> element contains the <idn:bundle> which has the following child elements:

- An <idn:oidn> element that contains the OIDN, along with the attributes described below.

- An OPTIONAL <idn:pidn> element that contains the PIDN, along with the attributes described below.

- Zero or more <idn:vidn> elements that contain the VIDNs, along with the attributes described below.

The above elements contain the following attributes:

- A "uLabel" attribute represents the U-label of the element.

- An optional "activated" attribute that defaults to true, indicating the presence of the variant in the zone file.
Example <transfer> Response for an authorized client:

S: <?xml version="1.0" encoding="UTF-8" standalone="no"?>
S: <epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S:  <response>
S:    <result code="1000">
S:      <msg>Command completed successfully</msg>
S:    </result>
S:    <resData>
S:      <domain:trnData
S:       xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
S:        <domain:name>xn--fsq270a.xn--fiqs8s</domain:name>
S:        <domain:trStatus>pending</domain:trStatus>
S:        <domain:reID>ClientX</domain:reID>
S:        <domain:reDate>2010-06-06T22:00:00.0Z</domain:reDate>
S:        <domain:acID>ClientY</domain:acID>
S:        <domain:acDate>2011-06-11T22:00:00.0Z</domain:acDate>
S:        <domain:exDate>2012-09-08T22:00:00.0Z</domain:exDate>
S:      </domain:trnData>
S:    </resData>
S:    <extension>
S:      <idn:trnData
S:       xmlns:idn="urn:ietf:params:xml:ns:idnv-1.0">
S:        <idn:bundle>
S:          <idn:oidn uLabel="U+5B9E""U+4f8b"."U+4E2D""U+56FD" activated="true">xn--fsq270a.xn--fiqs8s</idn:oidn>
S:          <idn:pidn uLabel="U+5BE6""U+4f8b"."U+4E2D""U+570B" activated="true">xn--fsqz41a.xn--fiqz9s</idn:pidn>
S:          <idn:vidn uLabel="U+5B9F""U+4f8b"."U+4E2D""U+570B" activated="true">xn--fsq470a.xn--fiqz9s</idn:vidn>
S:        </idn:bundle>
S:      </idn:trnData>
S:    </extension>
S:  </trID>
S:  </epp>

An EPP error response MUST be returned if a <transfer> command cannot be processed for any reason.

6.2. EPP Transform Commands

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

6.2.1. EPP <create> Command

This extension defines additional elements to extend the EPP <create> command described in the EPP domain name mapping [RFC5731] for IDN variants registration.

In addition to the EPP command elements described in the EPP domain mapping [RFC5731], the <create> command SHALL contain an <extension> element. The <extension> element SHOULD contain a child <idn:create> element that identifies the IDN namespace and the location of the IDN schema. The <idn:create> element SHOULD contain one or more <idn:vidn>: 

Example <create> command:

C: <?xml version="1.0" encoding="UTF-8" standalone="no"?>
C: <epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <create>
C:      <domain:create
C:       xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
C:        <domain:name>xn--fsq270a.xn--fiqs8s</domain:name>
C:        <domain:period unit="y">2</domain:period>
C:        <domain:registrant>123</domain:registrant>
C:        <domain:contact type="admin">123</domain:contact>
C:        <domain:contact type="tech">123</domain:contact>
C:        <domain:authInfo>
C:          <domain:pw>2fooBAR</domain:pw>
C:        </domain:authInfo>
C:      </domain:create>
C:    </create>
C:    <extension:
C:      <idn:create
C:       xmlns:idn="urn:ietf:params:xml:ns:idnv-1.0">
C:          <idn:vidn uLabel="U+5B9F""U+4f8b"."U+4E2D""U+570B">
C:               xn--fsq470a.xn--fiqz9s</idn:vidn>
C:      </idn:create>
C:    </extension>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>

When an <create> command has been processed successfully, the EPP <creData> element MUST contain child elements as described in the EPP domain mapping [RFC5731]. In addition, the EPP <extension> element...
SHOULD contain a child <idn:creData> element that identifies the extension namespace if the domain object has data associated with this extension and based on its service policy. The <idn:creData> element contains the <idn:bundle> which has the following child elements:

- An <idn:oidn> element that contains the OIDN, along with the attributes described below.

- An OPTIONAL <idn:pidn> element that contains the PIDN, along with the attributes described below. If the PIDN is allowed to be presented to the registrant by the registration policy, the PIDN SHALL be generated by the server during an <create> command. After the <create> command has been processed successfully, the response of the command SHOULD contain an <idn:pidn> element.

- Zero or more <idn:vidn> elements that contain the VIDNs, along with the attributes described below.

The above elements contain the following attributes:

- A "uLabel" attribute represents the U-label of the element.

- An optional "activated" attribute that defaults to true, indicating the presence of the variant in the zone file.
Example <create> Response for an authorized client:

```
S: <?xml version="1.0" encoding="UTF-8" standalone="no"?>
S: <epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S:  <response>
S:    <result code="1000">
S:      <msg>Command completed successfully</msg>
S:    </result>
S:    <resData>
S:      <domain:creData
S:       xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
S:        <domain:name>xn--fsq270a.xn--fiqs8s</domain:name>
S:        <domain:crDate>1999-04-03T22:00:00.0Z</domain:crDate>
S:        <domain:exDate>2001-04-03T22:00:00.0Z</domain:exDate>
S:      </domain:creData>
S:    </resData>
S:    <extension>
S:      <idn:creData
S:       xmlns:idn="urn:ietf:params:xml:ns:idnv-1.0">
S:        <idn:bundle>
S:          <idn:oidn uLabel="U+5B9E""U+4f8b""U+4E2D""U+56FD" activated="true">xn--fsq270a.xn--fiqs8s</idn:oidn>
S:          <idn:pidn uLabel="U+5BE6""U+4f8b""U+4E2D""U+570B" activated="true">xn--fsqz41a.xn--fiqz9s</idn:pidn>
S:          <idn:vidn uLabel="U+5B9F""U+4f8b""U+4E2D""U+570B" activated="true">xn--fsq470a.xn--fiqz9s</idn:vidn>
S:        </idn:bundle>
S:      </idn:creData>
S:    </extension>
S:    <trID>
S:      <clTRID>ABC-12345</clTRID>
S:      <svTRID>54322-XYZ</svTRID>
S:    </trID>
S:  </response>
S:</epp>
```

<create> Response for the unauthorized client has not been changed, see [RFC5731] for detail.

An EPP error response MUST be returned if an <create> command cannot be processed for any reason.

6.2.2. EPP <delete> Command

This extension does not add any element to the EPP <delete> command described in the EPP domain mapping [RFC5731]. However, additional elements are defined for the <delete> response.
When a <delete> command has been processed successfully, the EPP <delData> element MUST contain child elements as described in the EPP domain mapping [RFC5731]. In addition, the EPP <extension> element SHOULD contain a child <idn:delData> element that identifies the extension namespace if the domain object has data associated with this extension and based on its service policy. The <idn:delData> element SHOULD contain the <idn:bundle> which has the following child elements:

- An <idn:oidn> element that contains the OIDN.
- An OPTIONAL <idn:pidn> element that contains the PIDN.
- Zero or more <idn:vidn> elements that contain the VIDNs.

Example <delete> response:

```
S: <?xml version="1.0" encoding="UTF-8" standalone="no"?>
S: <epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S:  <response>
S:    <result code="1000">
S:      <msg>Command completed successfully</msg>
S:    </result>
S:    <extension>
S:      <idn:delData xmlns:idn="urn:ietf:params:xml:ns:idnv-1.0">
S:        <idn:bundle>
S:          <idn:oidn>xn--fsq270a.xn--fiqs8s</idn:oidn>
S:          <idn:pidn>xn--fsqz41a.xn--fiqz9s</idn:pidn>
S:              <idn:vidn>xn--fsq470a.xn--fiqz9s</idn:vidn>
S:        </idn:bundle>
S:      </idn:delData>
S:    </extension>
S:    <trID>
S:      <clTRID>ABC-12345</clTRID>
S:      <svTRID>54321-XYZ</svTRID>
S:    </trID>
S:  </response>
S: </epp>
```

An EPP error response MUST be returned if a <delete> command cannot be processed for any reason.

6.2.3. EPP <renew> Command

This extension does not add any element to the EPP <renew> command described in the EPP domain mapping [RFC5731]. However, additional
elements are defined for the <renew> response.

When a <renew> command has been processed successfully, the EPP <renData> element MUST contain child elements as described in the EPP domain mapping [RFC5731]. In addition, the EPP <extension> element SHOULD contain a child <idn:renData> element that identifies the extension namespace if the domain object has data associated with this extension and based on its service policy. The <idn:renData> element SHOULD contain the <idn:bundle> which has the following child elements:

- An <idn:oidn> element that contains the OIDN.
- An OPTIONAL <idn:pidn> element that contains the PIDN.
- Zero or more <idn:vidn> elements that contain the VIDNs.

Example <renew> response:

```xml
S: <?xml version="1.0" encoding="UTF-8" standalone="no"?>
S: <epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S:  <response>
S:    <result code="1000">
S:      <msg>Command completed successfully</msg>
S:    </result>
S:    <resData>
S:      <domain:renData
S:       xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
S:        <domain:name>example.com</domain:name>
S:        <domain:exDate>2012-04-03T22:00:00.0Z</domain:exDate>
S:      </domain:renData>
S:    </resData>
S:    <extension>
S:      <idn:renData
S:       xmlns:idn="urn:ietf:params:xml:ns:idnv-1.0">
S:        <idn:bundle>
S:          <idn:oidn>xn--fsq270a.xn--fiqs8s</idn:oidn>
S:          <idn:pidn>xn--fsqz41a.xn--fiqz9s</idn:pidn>
S:        </idn:bundle>
S:      </idn:renData>
S:    </extension>
S:    <trID>
S:      <clTRID>ABC-12345</clTRID>
S:      <svTRID>54322-XYZ</svTRID>
S:    </trID>
S:  </response>
```
An EPP error response MUST be returned if a <renew> command cannot be processed for any reason.

6.2.4. EPP <transfer> Command

This extension does not add any element to the EPP <transfer> command described in the EPP domain mapping [RFC5731]. However, additional elements are defined for the <transfer> response.

When a <transfer> command has been processed successfully, the EPP <trnData> element MUST contain child elements as described in the EPP domain mapping [RFC5731]. In addition, the EPP <extension> element SHOULD contain the same child <idn:trnData> element defined in the transfer query response in this mapping.
Example <transfer> Response for an authorized client:

```
S:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S:  <response>
S:    <result code="1000">
S:      <msg>Command completed successfully</msg>
S:    </result>
S:    <resData>
S:      <domain:trnData
S:       xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
S:        <domain:name>xn--fsq270a.xn--fiqs8s</domain:name>
S:        <domain:trStatus>pending</domain:trStatus>
S:        <domain:reID>ClientX</domain:reID>
S:        <domain:reDate>2010-06-06T22:00:00.0Z</domain:reDate>
S:        <domain:acID>ClientY</domain:acID>
S:        <domain:acDate>2011-06-11T22:00:00.0Z</domain:acDate>
S:        <domain:exDate>2012-09-08T22:00:00.0Z</domain:exDate>
S:      </domain:trnData>
S:    </resData>
S:    <extension>
S:      <idn:trnData
S:       xmlns:idn="urn:ietf:params:xml:ns:idnv-1.0">
S:        <idn:bundle>
S:          <idn:oidn uLabel="U+5B9E""U+4f8b"."U+4E2D""U+56FD" activa
S:          <idn:pidn uLabel="U+5BE6""U+4f8b"."U+4E2D""U+570B" activa
S:          <idn:vidn uLabel="U+5B9F""U+4f8b"."U+4E2D""U+570B" activa
S:        </idn:bundle>
S:      </IDN:trnData>
S:    </extension>
S:    <trID>
S:      <clTRID>ABC-12345</clTRID>
S:      <svTRID>54322-XYZ</svTRID>
S:    </trID>
S:  </response>
S:</epp>
```

An EPP error response MUST be returned if a <transfer> command cannot be processed for any reason.

6.2.5. EPP <update> Command

This extension defines additional elements for the EPP <update> command described in the EPP domain mapping [RFC5731]. No additional elements are defined for the EPP <update> response.
The EPP <update> command provides a transform operation that allows a client to modify the attributes of a domain object. In addition to the EPP command elements described in the EPP domain mapping [RFC5731], the command SHOULD contain an <extension> element, and the <extension> element MUST contain a child <idn:update> element that identifies the extension namespace if the client wants to update the domain object with data defined in this extension. The <idn:update> element SHOULD contain one or more of the following elements:

- An <idn:add> element with one or more <idn:vidn> child elements that represent VIDNs to be associated with the OIDN.

- An <idn:rem> element with one or more <idn:vidn> child elements that represent VIDNs to be disassociated from the OIDN.

- An <idn:activate> element with one or more <idn:vidn> child elements each identifying an already existing VIDN to be activated. A VIDN that is to be associated with the OIDN, as listed in the <idn:add> element of the same command, may also be specified to explicitly signal to the server that the VIDN to be added should be activated that the same time it is associated.

- An <idn:deactivate> element with one or more <idn:vidn> child elements each identifying an already existing VIDN to be deactivated. A VIDN that is to be associated with the OIDN, as listed in the <idn:add> element of the same command, may also be specified to explicitly signal to the server that the VIDN to be added should be in a deactivated state when it is associated.

It is an error to include the same VIDN in an <idn:activate> or <deactivate> element, and at the same time being included in the <idn:rem> element. By definition, a VIDN to be disassociated from the OIDN will be deactivated and it would be contradictory to attempt to activate it, and redundant to request for deactivation.

Example <update> Command:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <update>
      <domain:update xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>xn--fsq270a.xn--fiqs8s</domain:name>
        <domain:add>
          <domain:ns>
            <domain:hostObj>ns2.example.cn</domain:hostObj>
          </domain:ns>
        </domain:add>
      </domain:update>
    </update>
  </command>
</epp>
```
When an extended <update> command has been processed successfully, the EPP response is as described in the EPP domain name mapping [RFC5731].

7. Formal Syntax

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

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

<!--
   Import common element types.
   -->
<import namespace="urn:iana:xml:ns:eppcom-1.0"
schemaLocation="eppcom-1.0.xsd"/>
<import namespace="urn:iana:xml:ns:epp-1.0"
schemaLocation="epp-1.0.xsd"/>

<!--
   Child elements found in EPP commands.
   -->
<element name="create" type="idn:createDataType"/>
<element name="update" type="idn:updateType"/>

<!--
   Child elements of the <idn:create> command
   All elements must be present at time of creation
   -->
<complexType name="createDataType">
   <sequence>
     <element name="vidn" type="idn:oidnType"
      minOccurs="0" maxOccurs="unbounded" />
   </sequence>
</complexType>

<!----
Child elements of the <idn:update> command
All elements must be present at time of creation
--> 
<complexType name="updateType">
  <sequence>
    <element name="chg" type="idn:chgType" minOccurs="0" />
    <element name="add" type="idn:addRemType" minOccurs="0" />
    <element name="rem" type="idn:addRemType" minOccurs="0" />
    <element name="activate" type="idn:addRemType" minOccurs="0" />
    <element name="deactivate" type="idn:addRemType" minOccurs="0" />
  </sequence>
</complexType>

<complexType name="chgType">
  <sequence>
    <element name="pidn" type="idn:oidnType" minOccurs="0" />
  </sequence>
</complexType>

<complexType name="addRemType">
  <sequence>
    <element name="vidn" type="idn:oidnType" minOccurs="0" maxOccurs="unbounded" />
  </sequence>
</complexType>

<!--
Child elements found in EPP commands.
--> 
<element name="infData" type="idn:trnDataType" />
<element name="delData" type="idn:trnDataType" />
<element name="renData" type="idn:trnDataType" />
<element name="trnData" type="idn:trnDataType" />
<element name="creData" type="idn:trnDataType" />

<complexType name="trnDataType">
  <sequence>
    <element name="bundle" type="idn:bundleType" />
  </sequence>
</complexType>

<!--
<transfer> response elements.
All elements must be present at time of poll query
--> 
<complexType name="bundleType">
  <sequence>
    <element name="oidn" type="idn:oidnType" />
  </sequence>
</complexType>
<element name="pidn" type="idn:oidnType" minOccurs="0" />
<element name="vidn" type="idn:oidnType" minOccurs="0" />
</sequence>
</complexType>
<complexType name="oidnType">
  <simpleContent>
    <extension base="eppcom:labelType">
      <attribute name="uLabel" type="eppcom:labelType"/>
      <attribute name="activated" type="boolean"
        use="optional" default="true" />
    </extension>
  </simpleContent>
</complexType>

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

8. Internationalization Considerations

EPP is represented in XML, which provides native support for encoding information using the Unicode character set and its more compact representations including UTF-8. Conformant XML processors recognize both UTF-8 and UTF-16. Though XML includes provisions to identify and use other character encodings through use of an "encoding" attribute in an <?xml?> declaration, use of UTF-8 is RECOMMENDED.

As an extension of the EPP domain name mapping, the elements, element content described in this document MUST inherit the internationalization conventions used to represent higher-layer domain and core protocol structures present in an XML instance that includes this extension.

9. IANA Considerations

This document uses URNs to describe XML namespaces and XML schemas conforming to a registry mechanism described in [RFC3688]. IANA is requested to assignment the following two URIs.

Registration request for the IDN namespace:
10. Security Considerations

The object mapping extension described in this document does not provide any other security services or introduce any additional considerations beyond those described by [RFC5730] or those caused by the protocol layers used by EPP.

11. Acknowledgements

The authors especially thank the authors of [RFC5730] and [RFC5731] and the following ones of CNNIC: Weiping Yang, Chao Qi, Linlin Zhou.

Useful comments were made by John Klensin, Scott Hollenbeck and Edward Lewis.

12. Change History

RFC Editor: Please remove this section.

12.1. draft-kong-epp-idn-variants-mapping: Version 00

- First draft.

13. References
13.1. Normative References


13.2. Informative References


Authors’ Addresses

Ning Kong
CNNIC
4 South 4th Street, Zhongguancun, Haidian District
Beijing, Beijing 100190
China

Phone: +86 10 5881 3147
Email: nkong@cnnic.cn

Jiagui Xie
CNNIC
4 South 4th Street, Zhongguancun, Haidian District
Beijing, Beijing 100190
China

Phone: +86 10 5881 2639
Email: xiejiagui@cnnic.cn

Hongtao Li
CNNIC
4 South 4th Street, Zhongguancun, Haidian District
Beijing, Beijing 100190
China

Phone: +86 10 5881 3164
Email: lihongtao@cnnic.cn