IDNA Domain Extension Mapping for the Extensible Provisioning Protocol (EPP)

Author

I. Treadwell

J. Khallouf

Status of This Document

Draft - AusRegistry Pty Ltd

Copyright Notice

Copyright (C) 2010 AusRegistry Pty Ltd

Abstract

This document describes an Extensible Provisioning Protocol (EPP) extension mapping for the provisioning and management of Internationalised Domain Names in Applications (IDNA) domain names stored in a shared central repository. Specified in XML, this mapping extends the EPP domain name mapping to provide additional features required for the provisioning of IDNA domains.

Table of Contents

[1. Introduction 3](#_Toc237241275)

[1.1 Conventions used in this document 3](#_Toc237241276)

[2 Object Attributes 3](#_Toc237241277)

[2.1 IDNA Domain Names 3](#_Toc237241278)

[3 EPP Command Mapping 3](#_Toc237241279)

[3.1 EPP Query Commands 3](#_Toc237241280)

[3.1.1 EPP <check> Command 4](#_Toc237241281)

[3.1.2 EPP <info> Command 4](#_Toc237241282)

[3.2 EPP Transform Commands 5](#_Toc237241283)

[3.2.1 EPP <create> Command 5](#_Toc237241284)

[3.2.2 EPP <delete> Command 6](#_Toc237241285)

[3.2.3 EPP <renew> Command 6](#_Toc237241286)

[3.2.4 EPP <transfer> Command 7](#_Toc237241287)

[3.2.5 EPP <update> Command 7](#_Toc237241288)

[4. Formal Syntax 7](#_Toc237241289)

[5. Internationalization Considerations 9](#_Toc237241290)

[6. Security Considerations 9](#_Toc237241291)

[7. References 9](#_Toc237241292)

[7.1 Normative References 9](#_Toc237241293)

[7.2 Informative References 9](#_Toc237241294)

# Introduction

This document describes an IDNA domain mapping for version 1.0 of the Extensible Provisioning Protocol (EPP). This mapping, an extension of the domain name mapping described in [1], is specified using the Extensible Markup Language (XML) 1.0, as described in [4], and XML Schema notation, as described in [5] and [6].

The EPP core protocol specification [2] provides a complete description of EPP command and response structures. A thorough understanding of the base protocol and specification relevant extensions is necessary to understand the mapping described in this document.

## 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 [3].

In examples, "C:" and "S:" indicate lines sent by the client and server respectively. Indentation and white space in examples is provided only to illustrate element relationships and is not a mandatory feature of this protocol.

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.

# Object Attributes

This extension adds elements to the EPP domain name mapping [1]. Only new element descriptions are described here.

## IDNA Domain Names

Internationalizing Domain Names in Applications (IDNA) is a mechanism for handling Internationalised Domain Names (IDN) in a standard fashion. To provide interoperability with legacy systems, IDNA defines the ASCII Compatible Encoding (ACE) for non-ASCII labels. This extension makes the distinction between the two representations of domain names; the ACE-encoded form and native presentation form, denoted by the elements dnsForm and userForm respectively.

# EPP Command Mapping

A detailed description of the EPP syntax and semantics can be found in the EPP core protocol specification [2]. The command mappings described here are specifically for use in implementing IDNA domain provisioning processes via EPP.

## 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.

### EPP <check> Command

This extension does not add any elements to the EPP <check> command or <check> response described in the EPP domain mapping [1].

### EPP <info> Command

This extension adds extensions to the EPP <info> response. The EPP <info> response is described in the EPP domain mapping [1]. The response to this command MAY vary depending on the identity of the querying client, use of authorization information, and server policy towards unauthorized clients.

When an <info> command has been successfully processed, the EPP <resData> element in the <info> response MUST contain child elements as described in the EPP domain mapping [1]. In addition, the <info> response MUST contain an EPP <extension> element, which MUST contain an <infData> element that identifies the idnadomain namespace. The <infData> element contains the following child elements:

* A <userForm> element that contains the fully qualified name of the registered domain in presentation form. A MANDATORY "language" attribute indicates the language used in registration.
* A <canonicalForm> element that contains the canonical form of the registered domain.

Example <info> 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 lang="en">Command completed successfully</msg>

S: </result>

S: <resData>

S: <infData xmlns="urn:ietf:params:xml:ns:domain-1.0">

S: <name>xn--kxaefcf8b.idn</name>

S: <roid>DA25CB8C51B42CA2A84B6B4876AFC512B-AR</roid>

S: <status s="ok"/>

S: <registrant>CON-1</registrant>

S: <clID>EPP</clID>

S: <crID>EPP</crID>

S: <crDate>2008-09-22T02:40:12.0Z</crDate>

S: <exDate>2010-09-22T02:40:12.0Z</exDate>

S: <authInfo>

S: <pw>2fooBAR</pw>

S: </authInfo>

S: </infData>

S: </resData>

S: <extension>

S: <infData xmlns="urn:X-ar:params:xml:ns:idnadomain-1.0">

S: <userForm language="gr">δείγμα.idn</userForm>

S: <canonicalForm>δείγμα.idn</canonicalForm>

S: </infData>

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 an extended <info> command cannot be processed for any reason.

## 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.

### EPP <create> Command

This extension adds elements to the EPP <create> command as well as the EPP <create> response, both described in the EPP domain mapping [1].

In addition to the EPP command elements described in the EPP domain mapping [1], the <create> command MAY contain an <extension> element, which MAY contain a <create> element identifying the idnadomain namespace. The <create> element contains the following child elements:

* One <userForm> element that contains the presentation form of the fully-qualified name of the domain to be queried. A "language" attribute MUST be present to indicate the language of the name.

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: <create xmlns="urn:ietf:params:xml:ns:domain-1.0">

C: <name>xn--kxaefcf8b.idn</name>

C: <registrant>CON-1</registrant>

C: <authInfo>

C: <pw>SjweDcB84E</pw>

C: </authInfo>

C: </create>

C: </create>

C: <extension>

C: <create xmlns="urn:X-ar:params:xml:ns:idnadomain-1.0">

C: <userForm language="gr">δείγμα.idn</userForm>

C: </create>

C: </extension>

C: <clTRID>ABC-12345</clTRID>

C: </command>

C: </epp>

When a <create> command has been successfully processed, the EPP <resData> element in the <create> response MUST contain child elements as described in the EPP domain mapping [1]. In addition, the <create> response MUST contain an EPP <extension> element, which MUST contain an <creData> element identifying the idnadomain namespace. The <creData> contains the following child elements:

* A <userForm> element that contains the fully qualified name of the registered domain in native presentation form. A MANDATORY "language" attribute indicates the language used in registration.
* A <canonicalForm> element that contains the canonical form of the registered domain.

Example <create> 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 lang="en">Command completed successfully</msg>

S: </result>

S: <resData>

S: <creData xmlns="urn:ietf:params:xml:ns:domain-1.0">

S: <name>xn--kxaefcf8b.idn</name>

S: <crDate>2008-10-06T23:49:30.0Z</crDate>

S: <exDate>2010-10-06T23:49:30.0Z</exDate>

S: </creData>

S: </resData>

S: <extension>

S: <creData xmlns="urn:X-ar:params:xml:ns:idnadomain-1.0">

S: <userForm language="gr">δείγμα.idn</userForm>

S: <canonicalForm>δεiγμα.idn</canonicalForm>

S: </creData>

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 an extended <create> command cannot be processed for any reason.

### EPP <delete> Command

This extension does not add any elements to the EPP <delete> command or <delete> response described in the EPP domain mapping [1].

### EPP <renew> Command

This extension does not add any elements to the EPP <renew> command or <renew> response described in the EPP domain mapping [1].

### EPP <transfer> Command

This extension does not add any elements to the EPP <transfer> command or <transfer> response described in the EPP domain mapping [1].

### EPP <update> Command

This extension does not add any elements to the EPP <update> command or <update> response described in the EPP domain mapping [1].

# Formal Syntax

An EPP protocol mapping 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"?>

<schema targetNamespace="urn:X-ar:params:xml:ns:idnadomain-1.0"

xmlns:idnadomain="urn:X-ar:params:xml:ns:idnadomain-1.0"

xmlns:eppcom="urn:ietf:params:xml:ns:eppcom-1.0"

xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

<!--

Import common element types.

-->

<import namespace="urn:ietf:params:xml:ns:eppcom-1.0"

schemaLocation="eppcom-1.0.xsd" />

<annotation>

<documentation>

Internationalised Domain Name Extensions to the Extensible

Provisioning Protocol v1.1 schema. Domain-specific types and

attributes.

</documentation>

</annotation>

<group name="namePair">

<annotation>

<documentation>

User and DNS presentation forms of a domain

</documentation>

</annotation>

<sequence>

<element name="userForm" type="eppcom:labelType">

</element>

<element name="dnsForm" type="eppcom:labelType" />

</sequence>

</group>

<complexType name="internationalisedLabelType">

<simpleContent>

<extension base="eppcom:labelType">

<attribute name="language" type="language" use="required">

<annotation>

<documentation>

Registration of language with IANA requires the definition

of a Script or Language Designator

(http://www.iana.org/procedures/idn-repository.html).

The linked document above notes that Language Designators

are defined in BCP 47

(http://www.rfc-editor.org/rfc/bcp/bcp47.txt), which

satisfies the requirements of the language datatype and

RFC3066 (BCP 47, Section 2.2.8).

</documentation>

</annotation>

</attribute>

</extension>

</simpleContent>

</complexType>

<!--

Custom extensions

-->

<element name="create" type="idnadomain:createType" />

<!--

Child elements of the <create> command extension.

-->

<complexType name="createType">

<sequence>

<element name="userForm"

type="idnadomain:internationalisedLabelType" />

</sequence>

</complexType>

<!--

Custom response

-->

<element name="infData" type="idnadomain:resDataType" />

<element name="creData" type="idnadomain:resDataType" />

<!--

Response extension elements.

-->

<complexType name="resDataType">

<sequence>

<element name="userForm"

type="idnadomain:internationalisedLabelType" />

<element name="canonicalForm" type="eppcom:labelType"

minOccurs="1" />

</sequence>

</complexType>

<!--

End of schema.

-->

</schema>

**END**

# 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 [7]. Conformant XML processors recognize both UTF-8 and UTF-16 [8]. 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 in environments where parser encoding support incompatibility exists.

As an extension of the EPP domain mapping [1], the elements, element content, attributes, and attribute values 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.

# Security Considerations

The mapping extensions described in this document do not provide any security services beyond those described by the EPP core protocol specification [2].

# References

## Normative References

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

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

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

## Informative References

[4] Maler, E., Yergeau, F., Paoli, J., Bray, T., and C. Sperberg-McQueen, "Extensible Markup Language (XML) 1.0 (Third Edition)", World Wide Web Consortium FirstEdition REC-xml-20040204, February 2004, <<http://www.w3.org/TR/2004/REC-xml-20040204>>.

[5] Thompson, H., Maloney, M., Mendelsohn, N., and D. Beech, "XML Schema Part 1: Structures Second Edition", World Wide Web Consortium Recommendation REC-xmlschema-1-20041028, October 2004, <<http://www.w3.org/TR/2004/REC-xmlschema-1-20041028>>.

[6] Biron, P. and A. Malhotra, "XML Schema Part 2: Datatypes Second Edition", World Wide Web Consortium Recommendation REC-xmlschema-2-20041028, October 2004, <<http://www.w3.org/TR/2004/REC-xmlschema-2-20041028>>.

[7] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, November 2003.

[8] Hoffman, P. and F. Yergeau, "UTF-16, an encoding of ISO 10646", RFC 2781, February 2000.