J. Mitchell
AusRegistry Pty Ltd
September 2010

Key-Value Mapping for the Extensible Provisioning Protocol (EPP)

Abstract

This document describes an Extensible Provisioning Protocol (EPP) extension mapping for the transport of arbitrary data associated with domain names stored in a shared central repository.

Status of This Document

This document specifies an extension to the EPP protocol first implemented in AusRegistry's Domain Name Registry EPP service. Please refer to AusRegistry for more information on the status of this document. Distribution of this document and use of the protocol extensions defined within is unrestricted and unlimited.

Copyright Notice

Table of Contents

1. Introduction
   1.1. Conventions Used In This Document
2. Key-Value Pairs
3. EPP Command Mapping
   3.1. EPP Query Commands
      3.1.1. EPP <check> Command
      3.1.2. EPP <info> Command
      3.1.3. EPP <transfer> Command
   3.2. EPP Transform Commands
      3.2.1. EPP <create> Command
      3.2.2. EPP <delete> Command
      3.2.3. EPP <renew> Command
      3.2.4. EPP <transfer> Command
      3.2.5. EPP <update> Command
4. Formal Syntax
5. Internationalization Considerations
6. Security Considerations
7. References
   7.1. Normative References
   7.2. Informative References
§ Author's Address

TOC

1. Introduction

This document describes an domain mapping for version 1.0 of the Extensible Provisioning Protocol (EPP) that allows for the transport of arbitrary data associated with a domain name. This mapping, an extension of the domain name mapping described in **[RFC5731]**, is specified using the Extensible Markup Language (XML) 1.0 **[W3C.REC-xml-20040204]** and XML Schema notation (**[W3C.REC-xmlschema-1-20041028]** **[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 and specification of relevant extensions is necessary to understand the mapping described in this document.

TOC

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 **[RFC2119]**.

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.

TOC

2. Key-Value Pairs

This document describes the use of association lists, <kvlist>, to represent logical groups of data. Each list contains one or more items that associate a value with a key. Multiple lists are uniquely identified by their name, which SHOULD be unique to the server.

The <kvlist> element contains the following child elements:

- One or more <item> elements that contain the value of an item in the list. A MANDATORY "key" attribute MUST be present to indicate the key for the specified value.

The key is represented by an token (XML Datatypes) that identifies the data value. Servers SHOULD describe the set of allowed keys for each list, the behavior when duplicate keys are specified, and the lexical representation of each key.

The value is an unbounded string. Data elements that are not native strings must be represented as a string. Servers SHOULD identify the lexical forms of values pertaining to each key.

TOC

3. 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 generic domain extensions via EPP.

TOC

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.

TOC

3.1.1. EPP <check> Command

This extension does not add any elements to the EPP <check> command or <check> response described in the EPP domain mapping **[RFC5731]**.

TOC

3.1.2. EPP <info> Command

The EPP <info> command is used to retrieve information associated with a domain object. The response to this command MAY vary depending on the identity of the querying client, use of authorisation information, and server policy towards unauthorised clients. If the querying client is the sponsoring client, all available information MUST be returned. If the querying client is not the sponsoring client but the client provides valid authorisation information, all available information SHOULD be returned. If the querying client is not the sponsoring client and the client does not provide valid authorisation information, server policy determines which items are returned.

This extension does not add any elements 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 an <infData> element identifying the extension namespace. The <infData> element contains the following child elements:

- One or more <kvlist> elements that contain a list of key-value items that are associated with the specified domain. A MANDATORY "name" attribute MUST be present to identify the key-value list. Child elements of the <kvlist> element are described in **Section 2**.

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>domain.example</name>
S:        <roid>D0001337-AR</roid>
S:        <status s="ok" />
S:        <registrant>Registrant</registrant>
S:        <contact type="tech">TECH-1</contact>
S:        <clID>Client1</clID>
S:        <crID>Client2</crID>
S:        <crDate>2010-09-02T04:26:42.0Z</crDate>
S:        <exDate>2012-09-02T04:26:42.0Z</exDate>
S:        <authInfo>
S:          <pw>123paSSword</pw>
S:        </authInfo>
S:      </infData>
S:    </resData>
S:    <extension>
S:      <infData xmlns="urn:X-ar:params:xml:ns:kv-1.0">
S:        <kvlist name="ext">
S:          <item key="key1">value1</item>
S:          <item key="key2">value2</item>
S:        </kvlist>
S:      </infData>
S:    </extension>
S:    <trID>
S:      <clTRID>ABC12345</clTRID>
S:      <svTRID>XYZ54321</svTRID>
S:    </trID>
S:  </response>
S:</epp>
```

TOC

3.1.3. EPP <transfer> Command

This extension does not add any elements to the EPP <transfer> command or <transfer> response described in the EPP domain mapping **[RFC5731]**.

TOC

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.

TOC

3.2.1. EPP <create> Command

This extension defines additional elements for the EPP <create> command described in the EPP domain mapping **[RFC5731]**. No additional elements are defined for the EPP

<create> response.

The EPP <create> command provides a transform operation that allows a client to create a domain object. In addition to the EPP command elements described in the EPP domain mapping **[RFC5731]**, the <create> command MUST contain an <extension> element, and the <extension> MUST contain a <create> element that identifies the extension namespace. The <create> element contains the following child elements:

- One or more <kvlist> elements that contain the list of key-value items to be associated with the specified domain. A MANDATORY "name" attribute MUST be present to identify the name of the key-value list. Child elements of the <kvlist> element are described in **Section 2**.

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>domain.example</name>
C:        <registrant>Registrant</registrant>
C:        <contact type="tech">Tech1</contact>
C:        <authInfo>
C:          <pw>SjweDcB84E</pw>
C:        </authInfo>
C:      </create>
C:    </create>
C:    <extension>
C:      <create xmlns="urn:X-ar:params:xml:ns:kv-1.0">
C:        <kvlist name="ext">
C:          <item key="key1">value1</item>
C:          <item key="key2">value2</item>
C:        </kvlist>
C:      </create>
C:    </extension>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

When an extended <create> command has been processed successfully, the EPP response is as described in the EPP domain mapping **[RFC5731]**.

TOC

3.2.2. EPP <delete> Command

This extension does not add any elements to the EPP <delete> command or <delete> response described in the EPP domain mapping **[RFC5731]**.

TOC

3.2.3. EPP <renew> Command

This extension does not add any elements to the EPP <renew> command or <renew> response described in the EPP domain mapping **[RFC5731]**.

TOC

3.2.4. EPP <transfer> Command

This extension does not add any elements to the EPP <transfer> command or <transfer> response described in the EPP domain mapping **[RFC5731]**. No additional elements are defined for the EPP <transfer> response.

TOC

3.2.5. 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 MUST contain an <extension> element, and the <extension> element MUST contain a child <update> element that identifies the extension namespace if the client wants to update the domain object with data defined in this extension. The <update> element contains the following child elements:

- One or more <kvlist> elements that contain the list of key-value items to be associated with the specified domain. A MANDATORY "name" attribute MUST be present to identify the name of the key-value list. Child elements of the <kvlist> element are described in **Section 2**.

The server MUST treat the list as a replacement of an existing list and any server-defined mandatory items MUST be provided with each update.

Example <update> 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:    <update>
C:      <update xmlns="urn:ietf:params:xml:ns:domain-1.0">
C:        <name>domain.example</name>
C:      </update>
C:    </update>
C:    <extension>
C:      <update xmlns="urn:X-ar:params:xml:ns:kv-1.0">
C:        <kvlist name="ext">
C:          <item key="key1">value1</item>
C:          <item key="key2">value2</item>
C:        </kvlist>
C:      </update>
C:    </extension>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

When an extended <update> command has been processed successfully, the EPP response is as described in the EPP domain mapping **[RFC5731]**.

4. Formal Syntax [TOC]

An EPP object 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:kv-1.0"
   xmlns="http://www.w3.org/2001/XMLSchema"
   xmlns:kv="urn:X-ar:params:xml:ns:kv-1.0"
   elementFormDefault="qualified">

   <!--
      Definition of kvlist to be reused in other DTDs
   -->
   <group name="kvlist">
      <sequence>
         <element name="kvlist" type="kv:listType" />
      </sequence>
   </group>

   <!--
      Command extension elements
    -->
   <element name="create" type="kv:createType" />
   <element name="update" type="kv:updateType" />

   <complexType name="createType">
      <group ref="kv:kvlist" maxOccurs="unbounded" />
   </complexType>

   <complexType name="updateType">
      <group ref="kv:kvlist" maxOccurs="unbounded" />
   </complexType>

   <complexType name="listType">
      <sequence maxOccurs="unbounded">
         <element name="item" type="kv:itemType" />
      </sequence>
      <attribute name="name" type="token" use="required" />
   </complexType>

   <complexType name="itemType">
      <simpleContent>
         <extension base="string">
            <attribute name="key" type="token" use="required" />
         </extension>
      </simpleContent>
   </complexType>

   <!--
      Response extension types
    -->
   <element name="infData" type="kv:infDataType" />

   <complexType name="infDataType">
      <group ref="kv:kvlist" maxOccurs="unbounded" />
   </complexType>

</schema>
```

```
END
```

TOC

5. 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 **[RFC3629]**. Conformant XML processors recognize both UTF-8 and UTF-16 **[RFC2781]**. 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 **[RFC5731]**, 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.

TOC

6. Security Considerations

The mapping extensions described in this document do not provide any security services beyond those described by EPP **[RFC5730]**, the EPP domain name mapping **[RFC5731]**, and protocol layers used by EPP. The security considerations described in these other specifications apply to this specification as well.

As with other domain object transforms, the EPP transform operations described in this document MUST be restricted to the sponsoring client as authenticated using the mechanisms described in Sections 2.9.1.1 and 7 of **[RFC5730]**. Any attempt to perform a transform operation on a domain object by any client other than the sponsoring client MUST be rejected with an appropriate EPP authorization error.

TOC

7. References

TOC

7.1. Normative References

[RFC2119]	**Bradner, S.**, "**Key words for use in RFCs to Indicate Requirement Levels**," BCP 14, RFC 2119, March 1997 (**TXT**, **HTML**, **XML**).
[RFC5730]	Hollenbeck, S., "**Extensible Provisioning Protocol (EPP)**," STD 69, RFC 5730, August 2009 (**TXT**).
[RFC5731]	Hollenbeck, S., "**Extensible Provisioning Protocol (EPP) Domain Name Mapping**," STD 69, RFC 573 August 2009 (**TXT**).
[W3C.REC-xml-20040204]	Bray, T., Maler, E., Paoli, J., Yergeau, F., and C. Sperberg-McQueen, "**Extensible Markup Language (XML) 1.0 (Third Edition)**," World Wide Web Consortium FirstEdition REC-xml-20040204, February 200 (**HTML**).
[W3C.REC-xmlschema-1-20041028]	Beech, D., Thompson, H., Mendelsohn, N., and M. Maloney, "**XML Schema Part 1: Structures Second Edition**," World Wide Web Consortium Recommendation REC-xmlschema-1-20041028, October 2004 (**HTML**).

[W3C.REC-xmlschema-2-20041028] Malhotra, A. and P. Biron, "**XML Schema Part 2: Datatypes Second Edition**," World Wide Web Consortium Recommendation REC-xmlschema-2-20041028, October 2004 (**HTML**).

TOC

7.2. Informative References

[RFC2781] **Hoffman, P.** and **F. Yergeau**, "**UTF-16, an encoding of ISO 10646**," RFC 2781, February 2000 (**TXT**).

[RFC3629] Yergeau, F., "**UTF-8, a transformation format of ISO 10646**," STD 63, RFC 3629, November 2003 (**TXT**).

TOC

Author's Address

James Mitchell
AusRegistry Pty Ltd
Email: support@ausregistry.com.au