



EXTENSIBLE PROVISIONING PROTOCOL EXTENSION MAPPING: <CLIENT OBJECT ATTRIBUTE>

Version 1.0

LEGAL DISCLAIMER

COPYRIGHT NOTIFICATION

Copyright © 2012 VeriSign, Inc. All rights reserved. VERISIGN; the Verisign logo; and other trademarks, service marks and Verisign designs are registered or unregistered trademarks of VeriSign Inc. and its subsidiaries in the United States and foreign countries. Copyright laws and international treaties protect this document, and any Verisign product to which it relates.

VERISIGN PROPRIETARY INFORMATION

This document is the property of VeriSign, Inc. and its subsidiaries (together "Verisign") It may be used by recipient only for the purpose for which it was transmitted and must be returned upon request or when no longer needed by recipient. It may not be copied or communicated without the prior written consent of Verisign.

DISCLAIMER AND LIMITATION OF LIABILITY

Verisign has made efforts to ensure the accuracy and completeness of the information in this document. However, Verisign makes no warranties of any kind (whether express, implied or statutory) with respect to the information contained herein. Verisign assumes no liability to any party for any loss or damage (whether direct or indirect) caused by any errors, omissions or statements of any kind contained in this document. Further, Verisign assumes no liability arising from the application or use of the product or service described herein and specifically disclaims any representation that the products or services described herein do not infringe upon any existing or future intellectual property rights. Nothing herein grants the reader any license to make, use or sell equipment or products constructed in accordance with this document. Finally, all rights and privileges related to any intellectual property right described herein are vested in the patent, trademark or service mark owner and no other person may exercise such rights without express permission, authority or license secured from the patent, trademark or service mark owner. Verisign reserves the right to make changes to any information herein without further notice.

Any statements contained within this document concerning Verisign's future prospects are "forward looking statements" under the Federal Securities laws. There can be no assurance that future results will be achieved and actual results could differ materially from forecasts, estimates, and summary information contained in the document. Important factors that could cause actual results to differ materially include but are not limited to factors discussed in Verisign's SEC filings.

NOTICE AND CAUTION

Concerning U.S. Patent or Trademark Rights

Verisign and other trademarks, service marks and logos are registered or unregistered trademarks of Verisign and its subsidiaries in the United States and in foreign countries. The inclusion in this document, the associated on-line file or the associated software of any information covered by any other patent, trademark or service mark rights does not constitute nor imply a grant of or authority to exercise, any right or privilege protected by such patent, trademark or service mark. All such rights and privileges are vested in the patent, trademark or service mark owner and no other person may exercise such rights without express permission, authority or license secured from the patent, trademark or service mark owner.

Table of Contents

1	INTRODUCTION.....	1
2	OBJECT ATTRIBUTES.....	1
2.1	Client Object Attribute.....	1
2.2	Key.....	1
3	EPP COMMAND MAPPING.....	2
3.1	EPP Query Commands	2
3.1.1	EPP <check> Command	3
3.1.2	EPP <info> Command	4
3.1.3	EPP <transfer> Command	5
3.2	EPP Transform Commands.....	6
3.2.1	EPP <create> Command	6
3.2.2	EPP <delete> Command	8
3.2.3	EPP <renew> Command.....	9
3.2.4	EPP <transfer> Command	10
3.2.5	EPP <update> Command.....	11
4	FORMAL SYNTAX	14
5	REFERENCES.....	16

1 Introduction

This document describes a Client Object Attribute mapping for version 1.0 of the Extensible Provisioning Protocol (EPP). This mapping, an extension of the domain name mapping described in [EPP-D], of the host mapping described in [EPP-H], and of the contact mapping described in [EPP-C], is specified using the Extensible Markup Language (XML) 1.0 as described in [XML] and XML Schema notation as described in [XMLS-1] and [XMLS-2]. The domain name mapping [EPP-D] is used as a reference mapping throughout the document.

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

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.

2 Object Attributes

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

2.1 Client Object Attribute

The Client Object Attribute is a client specified name / value pair consisting of a and a 'value' text value of 1000 or less characters. It is represented by the <coa:attr> element, which in turn consists of the following child elements:

- A <coa:key> element that contains a text value of 50 or less characters.
- A <coa:value> element that contains a text value of 1000 or less characters.

An example <coa:attr> extension indicating a 'KEY1' attribute with a value of 'value1' is:

```
<coa:attr>
  <coa:key>KEY1</coa:key>
  <coa:value>value1</coa:value>
</coa:attr>
```

2.2 Key

The <coa:key> element can also be used outside of a <coa:attr> element in certain contexts where it is only necessary to identify the attribute and not specify its value.

3 EPP Command Mapping

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

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

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

3.1.2 EPP <info> Command

This extension does not add any elements to the EPP <info> command described in [EPP-D]. 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 [EPP-D]. In addition, the EPP <extension> element **SHOULD** contain a child <coa:infData> element that identifies the extension namespace if the object has one or more Client Object Attributes associated with it and based on server policy. The <coa:infData> element contains the following child elements:

- One or more <coa:attr> elements describing the Client Object Attributes currently associated with the object.

Example <info> response for a domain with an associated Client Object Attribute:

```
S:    <?xml version="1.0" encoding="UTF-8"?>
S:    <epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
S:        xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
S:        xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
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:                    xsi:schemaLocation="urn:ietf:params:xml:ns:domain-1.0 domain-1.0.xsd">
S:                    <domain:name>example.tv</domain:name>
S:                    <domain:roid>EXAMPLE1-VRSN</domain:roid>
S:                    <domain:status s="ok"/>
S:                    <domain:clID>ClientX</domain:clID>
S:                    <domain:crID>ClientY</domain:crID>
S:                    <domain:crDate>2011-02-04T15:44:37.0526Z</domain:crDate>
S:                    <domain:authInfo>
S:                        <domain:pw>2fooBAR</domain:pw>
S:                    </domain:authInfo>
S:                </domain:infData>
S:            </resData>
S:            <extension>
S:                <coa:infData xmlns:coa="urn:ietf:params:xml:ns:coa-1.0">
S:                    <coa:attr>
S:                        <coa:key>KEY1</coa:key>
S:                        <coa:value>value1</coa:value>
S:                    </coa:attr>
S:                </coa:infData>
S:            </extension>
S:            <trID>
S:                <clTRID>54321-CLI</clTRID>
S:                <svTRID>54321-SER</svTRID>
S:            </trID>
S:        </response>
S:    </epp>
```

3.1.3 EPP <transfer> Command

This extension does not add any elements to the EPP <transfer> query command or <transfer> query response described in [EPP-D].

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

This extension defines additional elements for the EPP <create> command described in [EPP-D].

The EPP <create> command provides a transform operation that allows a client to create an object. In addition to the EPP command elements described in [EPP-D], the command **MUST** contain a child <coa:create> element that identifies the extension namespace if the client wants to associate a Client Object Attribute to the object. The <coa:create> element contains the following child elements:

- One or more <coa:attr> elements. The <coa:attr> element is described in section 2.1.

Example <create> command to create a domain object with a COA:

```
C:    <?xml version="1.0" encoding="UTF-8"?>
C:    <epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
C:        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
C:        xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
C:        <command>
C:            <create>
C:                <domain:create
C:                    xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
C:                    xsi:schemaLocation="urn:ietf:params:xml:ns:domain-1.0 domain-1.0.xsd">
C:                    <domain:name>john.doe.name</domain:name>
C:                    <domain:authInfo>
C:                        <domain:pw>2fooBAR</domain:pw>
C:                    </domain:authInfo>
C:                </domain:create>
C:            </create>
C:            <extension>
C:                <coa:create xmlns:coa="urn:ietf:params:xml:ns:coa-1.0">
C:                    <coa:attr>
C:                        <coa:key>KEY1</coa:key>
C:                        <coa:value>value1</coa:value>
C:                    </coa:attr>
C:                </coa:create>
C:            </extension>
C:            <clTRID>ABC-12345</clTRID>
C:        </command>
C:    </epp>
```

When a <create> command has been processed successfully, the EPP

response is as described in [EPP-D].

3.2.2 EPP <delete> Command

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

3.2.3 EPP <renew> Command

This extension does not add any elements to the EPP <renew> command described in [EPP-D].

3.2.4 EPP <transfer> Command

This extension does not add any elements to the EPP <transfer> command described in [EPP-D]. The successful completion of a transfer however **SHOULD** result in the clearing of any existing Client Object Attributes.

3.2.5 EPP <update> Command

This extension defines additional elements for the EPP <update> command described in [EPP-D].

The EPP <update> command provides a transform operation that allows a client to modify the attributes of an object. In addition to the EPP command elements described in the appropriate EPP object mapping, the command MUST contain an <extension> element, and the <extension> element MUST contain a child <coa:update> element that identifies the extension namespace if the client wants to update the object with a Client Object Attribute. The <coa:update> element contains a <coa:put> element to add or update one or more Client Object Attributes, or a <coa:rem> element to remove existing Client Object Attributes. At least one <coa:put> or <coa:rem> element MUST be provided.

The <coa:update> element contains the following child elements:

- An OPTIONAL <coa:rem> element that is used to remove Client Object Attributes from an existing set. The <coa:rem> element MUST contain one or more <coa:key> elements that are used to remove the Client Object Attributes identified by the values specified in the <coa:key> elements.
- An OPTIONAL <coa:put> element that is used to add Client Object Attributes to an existing set. The <coa:put> element MUST contain one or more <coa:attr> elements that are each used to identify the name and value of a Client Object Attribute being added to the object or having its value reset if it already exists.

Example <update> command to create a new Client Object Attribute 'KEY1' with the value 'value1':

```
C: <?xml version="1.0" encoding="UTF-8"?>
C:   <epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
C:     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
C:     xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
C:       <command>
C:         <update>
C:           <domain:update
C:             xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
C:             xsi:schemaLocation="urn:ietf:params:xml:ns:domain-1.0 domain-1.0.xsd">
C:             <domain:name>john.doe.name</domain:name>
C:             <domain:chg/>
C:           </domain:update>
C:         </update>
C:         <extension>
C:           <coa:update xmlns:coa="urn:ietf:params:xml:ns:coa-1.0">
C:             <coa:put>
C:               <coa:attr>
C:                 <coa:key>KEY1</coa:key>
C:                 <coa:value>value1</coa:value>
C:               </coa:attr>
C:             </coa:put>
C:           </coa:update>
C:         </extension>
C:       </command>
```

C: </epp>

Example <update> command to update the value of Client Object Attribute ‘KEY1’ to ‘value2’ :

```
C:     <?xml version="1.0" encoding="UTF-8"?>
C:     <epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
C:         xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
C:         xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
C:         <command>
C:             <update>
C:                 <domain:update
C:                     xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
C:                     xsi:schemaLocation="urn:ietf:params:xml:ns:domain-1.0 domain-1.0.xsd">
C:                         <domain:name>john.doe.name</domain:name>
C:                         <domain:chg/>
C:                     </domain:update>
C:             </update>
C:             <extension>
C:                 <coa:update xmlns:coa="urn:ietf:params:xml:ns:coa-1.0">
C:                     <coa:put>
C:                         <coa:attr>
C:                             <coa:key>KEY1</coa:key>
C:                             <coa:value>value2</coa:value>
C:                         </coa:attr>
C:                     </coa:put>
C:                 </coa:update>
C:             </extension>
C:         </command>
C:     </epp>
```

Example <update> command to remove the Client Object Attribute ‘KEY1’:

```
C:     <?xml version="1.0" encoding="UTF-8"?>
C:     <epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
C:         xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
C:         xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
C:         <command>
C:             <update>
C:                 <domain:update
C:                     xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
C:                     xsi:schemaLocation="urn:ietf:params:xml:ns:domain-1.0 domain-1.0.xsd">
C:                         <domain:name>john.doe.name</domain:name>
C:                         <domain:chg/>
C:                     </domain:update>
C:             </update>
C:             <extension>
C:                 <coa:update xmlns:coa="urn:ietf:params:xml:ns:coa-1.0">
C:                     <coa:rem>
C:                         <coa:key>KEY1</coa:key>
C:                     </coa:rem>
C:                 </coa:update>
C:             </extension>
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

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.

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

<schema targetNamespace="urn:ietf:params:xml:ns:coa-1.0"
  xmlns:coa="urn:ietf:params:xml:ns:coa-1.0"
  xmlns="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">

  <annotation>
    <documentation>
      Extensible Provisioning Protocol v1.0 domain name
      extension schema for COA (Client Object Attributes) Tag.
    </documentation>
  </annotation>

  <element name="create" type="coa:mapType" />
  <element name="update" type="coa:updateType" />
  <element name="infData" type="coa:mapType" />

  <complexType name="updateType">
    <sequence>
      <element name="rem" type="coa:remType" minOccurs="0" />
      <element name="put" type="coa:mapType" minOccurs="0" />
    </sequence>
  </complexType>

  <!--
    A sequence of key/value pairs, for creating or updating a key/value
    pair in the server.
  -->
  <complexType name="mapType">
    <sequence>
      <element name="attr" type="coa:attrType" maxOccurs="unbounded" />
    </sequence>
  </complexType>

  <!--
    A single key/value pair
  -->
  <complexType name="attrType">
    <all>
      <element name="key" type="coa:keyType" />
      <element name="value" type="coa:valueType" />
    </all>
  </complexType>

  <!--
    A sequence of keys, for removing key/value pairs in the server.
  -->
```

```

<complexType name="remType">
  <sequence>
    <element name="key" type="coa:keyType" maxOccurs="unbounded" />
  </sequence>
</complexType>

<!--
  A key of a key/value pair.
-->
<simpleType name="keyType">
  <restriction base="token">
    <maxLength value="50" />
  </restriction>
</simpleType>

<!--
  A value of a key/value pair.
-->
<simpleType name="valueType">
  <restriction base="token">
    <maxLength value="1000" />
  </restriction>
</simpleType>

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

```

5 References

- [EPP] Hollenbeck, S., "Extensible Provisioning Protocol (EPP)", STD 69, RFC 5730, August 2009.
- [EPP-D] Hollenbeck, S., "Extensible Provisioning Protocol (EPP) Domain Name Mapping", STD 69, RFC 5731, August 2009.
- [EPP-H] Hollenbeck, S., "Extensible Provisioning Protocol (EPP) Host Mapping", STD 69, RFC 5732, August 2009.
- [EPP-C] Hollenbeck, S., "Extensible Provisioning Protocol (EPP) Contact Mapping", STD 69, RFC 5733, August 2009.
- [IETF-XML] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, January 2004.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [XML] Maler, E., Sperberg-McQueen, C., Bray, T., and J. Paoli, "Extensible Markup Language (XML) 1.0 (Second Edition)", World Wide Web Consortium FirstEdition REC-xml-20001006, October 2000, <<http://www.w3.org/TR/2000/REC-xml-20001006>>.
- [XMLS-1] Beech, D., Thompson, H., Mendelsohn, N., and M. Maloney, "XML Schema Part 1: Structures", World Wide Web Consortium FirstEdition REC-xmlschema-1-20010502, May 2001, <<http://www.w3.org/TR/2001/REC-xmlschema-1-20010502>>.
- [XMLS-2] Malhotra, A. and P. Biron, "XML Schema Part 2: Datatypes", World Wide Web Consortium FirstEdition REC-xmlschema-2-20010502, May 2001, <<http://www.w3.org/TR/2001/REC-xmlschema-2-20010502>>.