Variant Protocol Extension for the Extensible Provisioning Protocol (EPP)

Author

J. Khallouf

I. Treadwell

Status of This Document

Draft - AusRegistry Pty Ltd

Copyright Notice

Copyright (C) 2010 AusRegistry Pty Ltd

Abstract

This document describes application layer client-server protocol extensions for the Extensible Provisioning Protocol (EPP) as needed for the management of domain objects stored in a shared central repository. Specified in XML, the protocol defines specific object management operations and an extensible framework that maps the protocol extensions to the relevant objects.

Table of Contents

[1. Introduction 3](#_Toc237238099)

[1.1 Conventions used in this document 3](#_Toc237238100)

[1.2 Additional Services 3](#_Toc237238101)

[1.3 Enabling Extended Services 3](#_Toc237238102)

[2 Protocol Extension Description 5](#_Toc237238103)

[2.1 Command Format 5](#_Toc237238104)

[2.2 Response Format 5](#_Toc237238105)

[2.3 Extension Protocol Commands 6](#_Toc237238106)

[2.3.1 Query Commands 6](#_Toc237238107)

[3. Formal Syntax 7](#_Toc237238108)

[4. Internationalization Considerations 8](#_Toc237238109)

[5. Security Considerations 9](#_Toc237238110)

[6. References 9](#_Toc237238111)

[6.1. Normative References 9](#_Toc237238112)

[6.2. Informative References 9](#_Toc237238113)

[Appendix A. Object Mapping Template 10](#_Toc237238114)

[1. Introduction 10](#_Toc237238115)

[2. Object Attributes 10](#_Toc237238116)

[3. EPP Command Mapping 10](#_Toc237238117)

[3.1 EPP Query Commands 10](#_Toc237238118)

# Introduction

This document describes the AusRegistry extensions to the Extensible Provisioning Protocol (EPP) and the additional object management services supported by these extensions. These extensions are specified using the Extensible Markup Language (XML) 1.0, as described in [3], and XML Schema notation, as described in [4] and [5].

The AusRegistry extension commands and responses defined within this document mimic those defined in the EPP core protocol specification [1]. The EPP core protocol specification provides a complete description of core EPP command and response structures, and a thorough understanding of the base protocol is necessary to understand the extensions defined 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 [2].

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.

## Additional Services

The following additional commands [1 Section 2.7.1] are defined:

* <variantInfo>

## Enabling Extended Services

In accordance with the service extension usage mechanism described in the EPP specification [1], server support for AusRegistry EPP extensions is published in a <greeting> service element via the <svcExtension> element. A server supporting the AusRegistry EPP extensions MUST inform the client of its capability via an <extURI> child element whose value is the AusRegistry EPP extension URI (urn:X-ar:params:xml:ns:viext-1.0). A client wishing to use such services MUST present the AusRegistry EPP extension URI as the value of an <extURI> element in the EPP <login> command which initiates the session in which the services will be used.

Example <greeting> publishing support for viext extension:

S:<?xml version="1.0" encoding="UTF-8" standalone="no"?>

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

S: <greeting>

S: <svID>AusRegistry EPP Server</svID>

S: <svDate>2006-02-09T15:42:29.0Z</svDate>

S: <svcMenu>

S: <version>1.0</version>

S: <lang>en</lang>

S: <objURI>urn:ietf:params:xml:ns:domain-1.0</objURI>

S: <objURI>urn:ietf:params:xml:ns:contact-1.0</objURI>

S: <objURI>urn:ietf:params:xml:ns:host-1.0</objURI>

S: <svcExtension>

S: <extURI>urn:X-ar:params:xml:ns:viext-1.0</extURI>

S: </svcExtension>

S: </svcMenu>

S: <dcp>

S: <access><all/></access>

S: <statement>

S: <purpose><admin/><prov/></purpose>

S: <recipient><ours/><public/></recipient>

S: <retention><stated/></retention>

S: </statement>

S: </dcp>

S: </greeting>

S:</epp>

Example <login> requesting viext extension services:

C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>

C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">

C: <command>

C: <login>

C: <clID>REGISTRAR</clID>

C: <pw>p4ssw0rd!</pw>

C: <options>

C: <version>1.0</version>

C: <lang>en</lang>

C: </options>

C: <svcs>

C: <objURI>urn:ietf:params:xml:ns:contact-1.0</objURI>

C: <objURI>urn:ietf:params:xml:ns:domain-1.0</objURI>

C: <objURI>urn:ietf:params:xml:ns:host-1.0</objURI>

C: <svcExtension>

C: <extURI>urn:X-ar:params:xml:ns:viext-1.0</extURI>

C: </svcExtension>

C: </svcs>

C: </login>

C: </command>

C:</epp>

# Protocol Extension Description

The AusRegistry Extension to EPP provides two basic service elements: commands and responses. These are layered on top of the protocol extension framework of EPP. The EPP server state machine is unaffected by these extension elements; they are equivalent in the state machine to core command and response elements.

## Command Format

An EPP client may interact with an EPP server by sending an extension command and receiving an EPP response as defined in this document, in the same manner as described in [1]. In addition to the standard EPP elements, an extension command contains the following elements:

* A <command> element that contains the following child elements:
  + A command element whose tag corresponds to one of the valid extension commands described in this document. The command element contains object-specified child elements.
  + An OPTIONAL <extension> element that MAY be used for server-defined command extensions.
  + An OPTIONAL <clTRID> (client transaction identifier) element that MAY be used to uniquely identify the command to the client. Clients are responsible for maintaining their own transaction identifier space to ensure uniqueness.

Example command with object-specified child elements:

C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>

C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">

C: <extension>

C: <command xmlns="urn:X-ar:params:xml:ns:viext-1.0">

C: <variantInfo>

C: <variantInfo xmlns="urn:X-ar:params:xml:ns:obj">

C: <name>example</name>

C: </variantInfo>

C: </variantInfo>

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

C: </command>

C: </extension>

C:</epp>

## Response Format

An EPP server responds to an EPP client extension command by returning an EPP response to the client as described in [1].

Example response with <resData>:

S:<?xml version="1.0" encoding="UTF-8" standalone="no"?>

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

S: <response xmlns="urn:ietf:params:xml:ns:epp-1.0">

S: <result code="1000">

S: <msg>Command completed successfully</msg>

S: </result>

S: <resData>

S: <varInfData xmlns="urn:X-ar:params:xml:ns:obj">

S: <name>example</name>

S: </varInfData>

S: </resData>

S: <trID>

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

S: <svTRID>54321-XYZ</svTRID>

S: </trID>

S: </response>

S:</epp>

## Extension Protocol Commands

The AusRegistry extension to EPP provides commands to perform transformation operations on objects. This describes each extension command, including examples with representative server responses.

### Query Commands

The AusRegistry extensions to EPP provide one command to query objects: <variantInfo>, to determine which objects would be provisioned in a Registry if a specified object was created.

#### EPP <variantInfo> Command

The EPP <variantInfo> command is used to determine which objects would be provisioned in a Registry if a specified object was created. The behaviour and syntax of this command is unaffected by whether the specified object has already been created or not.

The elements needed to identify an object are object-specific, so the child elements of the <variantInfo> command are specified in an object-specific mapping using the EPP extension framework. In addition to the <command> elements, the <variantInfo> command contains the following child elements:

* An object-specific <variantInfo> element that identifies the object to be queried.

Example <variantInfo> command:

C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>

C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">

C: <extension>

C: <command xmlns="urn:X-ar:params:xml:ns:viext-1.0">

C: <variantInfo>

C: <variantInfo xmlns="urn:org:params:xml:ns:obj-1.0">

C: <!-- Object-specific elements. -->

C: </variantInfo>

C: </variantInfo>

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

C: </command>

C: </extension>

C:</epp>

When a <variantInfo> command has been processed successfully, a server MAY respond with a <resData> element that MUST contain a child element that identifies the object namespace. The child elements of the <resData> element are object-specific.

Example <variantInfo> response with <resData>:

S:<?xml version="1.0" encoding="UTF-8" standalone="no"?>

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

S: <response xmlns="urn:ietf:params:xml:ns:epp-1.0">

S: <result code="1000">

S: <msg>Command completed successfully</msg>

S: </result>

S: <resData>

S: <varInfData xmlns="urn:org:params:xml:ns:obj-1.0">

S: <!-- Object-specific elements. -->

S: </varInfData>

S: </resData>

S: <trID>

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

S: <svTRID>54321-XYZ</svTRID>

S: </trID>

S: </response>

S:</epp>

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

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

<!--

Import epp element types.

-->

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

<annotation>

<documentation> Internationalised Domain Name Extensions to the Extensible Provisioning Protocol v1.0 schema.

Variant-specific commands.</documentation>

</annotation>

<element name="command" type="viext:commandType" />

<complexType name="commandType">

<sequence>

<element name="variantInfo" type="viext:readWriteType" />

<element name="extension" type="viext:extAnyType" minOccurs="0" />

<element name="clTRID" type="epp:trIDStringType" minOccurs="0" />

</sequence>

</complexType>

<complexType name="readWriteType">

<sequence>

<any namespace="##other" />

</sequence>

</complexType>

<!--

Extension framework types.

-->

<complexType name="extAnyType">

<sequence>

<any namespace="##other" maxOccurs="unbounded" />

</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 [6]. Conformant XML processors recognize both UTF-8 and UTF-16 [7]. 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 protocol [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 extensions described in this document do not provide any security services beyond those described by EPP [1].

# References

## Normative References

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

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

## Informative References

[3] 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>>.

[4] 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>>.

[5] 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>>.

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

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

# Appendix A. Object Mapping Template

This appendix describes a recommended outline for documenting the EPP mapping of an object to the service extensions described within this document. Documents that describe object mappings SHOULD describe the mapping in a format similar to the one used here.

## Introduction

Provide an introduction that describes the object and an overview of the mapping to EPP.

## Object Attributes

Describe the attributes associated with the object, including references to syntax specifications as appropriate. Examples of object attributes include a name or identifier and dates associated with modification events.

## EPP Command Mapping

### EPP Query Commands

#### EPP <variantInfo> Command

Describe the object-specific mappings required to implement the <variantInfo> command. Include both sample commands and sample responses.