Internet Engineering Task Force
Internet-Draft
April 9, 2001

S. Hollenbeck
VeriSign, Inc.

Expires: October 9, 2001

Extensible Provisioning Protocol Host Mapping
<draft-hollenbeck-epp-host-01.txt>

Status of this Memo

Abstract

This document describes an Extensible Provisioning Protocol (EPP) mapping for the provisioning and management of Internet host names stored in a shared central repository. Specified in XML, the mapping defines EPP command syntax and semantics as applied to host names.

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 in examples is provided only to illustrate element relationships and is not a REQUIRED feature of this protocol.

XML protocol elements are case sensitive. Data carried in XML is case insensitive unless stated otherwise.

Table of Contents

1. Introduction

This document describes an internet host name mapping for version 1.0 of the Extensible Provisioning Protocol (EPP). This mapping 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].

[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.

This document assumes that host name objects have a subordinate relationship to a hierarchical domain name object. For example, host name "ns1.example.com" has a subordinate relationship to domain name "example.com". EPP actions (such as object transfers) that do not preserve this relationship MUST be explicitly disallowed.

It is important to note that XML is case sensitive. XML specifications and examples provided in this document MUST be interpreted in the exact character case presented to develop a conforming implementation.

This document is being discussed on the "rrp" mailing list. To join the list, send a message to <majordomo@verisign-grs.com> with the words "subscribe rrp" in the body of the message. There is a web site for the list archives at <http://www.verisign-grs.com/maillist/rrp>.

2. Object Attributes

An EPP host object has attributes and associated values that may be viewed and modified by the sponsoring client or the server. This section describes each attribute type in detail.

2.1 Host Names

The syntax for host names described in this document MUST conform to [RFC952] as updated by [RFC1123]. These conformance requirements MAY change as a result of progressing work in developing standards for internationalized host names.

2.2 Client Identifiers

All EPP clients are identified by a server-unique identifier. Client identifiers use the "clIDType" syntax described in [EPP].

2.3 Dates and Times

Date and time attribute values MUST be represented in Universal Coordinated Time (UTC). Both extended and truncated date and time forms defined in [ISO8601] MAY be used, though a server SHOULD use one form or the other consistently.

2.4 Authorization Identifiers

Authorization identifiers are associated with host objects to facilitate transformation operations on host objects. Authorization identifiers use the transaction identifier syntax described in [EPP].

2.5 IP Addresses

The syntax for IPv4 addresses described in this document MUST conform to [RFC791]. The syntax for IPv6 addresses described in this document MUST conform to [RFC1883].

A server SHOULD reject assignments of IP addresses to hosts using IP addresses that have not been allocated for public use. A list of IPv4 address allocations is maintained in [IANAa]. A list of IPv6 address allocations is maintained in [IANAb]. Additional IPv4 address allocations that are not for public use are documented in [RFC1918].

3. EPP Command Mapping

A detailed description of the EPP syntax and semantics can be found in [EPP]. The command mappings described here are specifically for use in provisioning and managing internet host names via EPP.

3.1 EPP Query Commands

EPP provides two commands to retrieve host information: <check> to determine if a host object is known to the server, and <info> to retrieve detailed information associated with a host object.

3.1.1 EPP <check> Command

The EPP <check> command is used to determine if a host object is known to the server. In addition to the standard EPP command elements, the <check> command MUST contain a <host:check> element that identifies the host namespace and the location of the host schema. The <host:check> element SHALL contain the following child elements:

- One or more <host:name> elements that contain the fully qualified names of the host objects to be queried.

Example <check> command:

```
C:<?xml version="1.0" standalone="no"?>
C:<epp xmlns="urn:iana:xml:ns:epp"
C:     xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
C:     xsi:schemaLocation="urn:iana:xml:ns:epp epp.xsd">
C:  <command>
C:    <check>
C:      <host:check xmlns:host="urn:iana:xml:ns:host"
C:       xsi:schemaLocation="urn:iana:xml:ns:host host.xsd">
C:        <host:name>ns1.example.com</host:name>
C:        <host:name>ns2.example.com</host:name>
C:        <host:name>ns3.example.com</host:name>
C:      </host:check>
C:    </check>
C:    <unspec/>
C:    <clTRID>ABC-12346</clTRID>
C:  </command>
C:</epp>
```

When a <check> command has been processed successfully, the EPP <resData> element MUST contain a child <host:chkData> element that identifies the host namespace and the location of the host schema. The <host:chkData> element SHALL contain the following child elements:

- One or more <host:cd> elements that contain the fully qualified names for the queried host objects and an "x" attribute whose value identifies the object as either "+" for a known object or "-" for an unknown object.

Example <check> response:

```
S:<?xml version="1.0" standalone="no"?>
S:<epp xmlns="urn:iana:xml:ns:epp"
S:     xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
S:     xsi:schemaLocation="urn:iana:xml:ns:epp epp.xsd">
S:  <response>
S:    <result code="1000">
S:      <msg>Command completed successfully</msg>
S:    </result>
S:    <resData>
S:      <host:chkData xmlns:host="urn:iana:xml:ns:host"
S:       xsi:schemaLocation="urn:iana:xml:ns:host host.xsd">
S:        <host:cd x="+">ns1.example.com</host:cd>
S:        <host:cd x="-">ns2.example.com</host:cd>
S:        <host:cd x="+">ns3.example.com</host:cd>
S:      </host:chkData>
S:    </resData>
S:    <unspec/>
S:    <trID>
S:      <clTRID>ABC-12346</clTRID>
S:      <svTRID>54322-XYZ</svTRID>
S:    </trID>
S:  </response>
S:</epp>
```

An EPP error response MUST be returned if a <check> command can not be processed for any reason.

3.1.2 EPP <info> Command

The EPP <info> command is used to retrieve information associated with a host object. In addition to the standard EPP command elements, the <info> command MUST contain a <host:info> element that identifies the host namespace and the location of the host schema. The <host:info> element SHALL contain the following child elements:

- A <host:name> element that contains the fully qualified name of the host object for which information is requested.

Example <info> command:

```
C:<?xml version="1.0" standalone="no"?>
C:<epp xmlns="urn:iana:xml:ns:epp"
C:     xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
C:     xsi:schemaLocation="urn:iana:xml:ns:epp epp.xsd">
C:  <command>
C:    <info>
C:      <host:info xmlns:host="urn:iana:xml:ns:host"
C:       xsi:schemaLocation="urn:iana:xml:ns:host host.xsd">
C:        <host:name>ns1.example.com</host:name>
C:      </host:info>
C:    </info>
C:    <unspec/>
C:    <clTRID>ABC-12346</clTRID>
C:  </command>
C:</epp>
```

When an <info> command has been processed successfully, the EPP <resData> element MUST contain a child <host:infData> element that identifies the host namespace and the location of the host schema. The <host:infData> element SHALL contain the following child elements:

- A <host:name> element that contains the fully qualified name of the host object to be queried.

- A <host:roid> element that contains the host object's repository object identifier.

- Zero or more <host:addr> elements that contain the IP addresses associated with the host object.

- A <host:clID> element that contains the identifier of the sponsoring client.

- A <host:crID> element that contains the identifier of the client that created the host object.

- A <host:crDate> element that contains the date and time of host object creation.

- A <host:upID> element that contains the identifier of the client that last updated the host object. This element MUST NOT be present if the host object has never been modified.

- A <host:upDate> element that contains the date and time of the most recent host object modification. This element MUST NOT be present if

the host object has never been modified.

- A <host:trDate> element that contains the date and time of the most recent successful host object transfer. This element MUST NOT be provided if the host object has never been transferred. Note that host objects MUST NOT be transferred directly; host objects MUST be transferred implicitly when the host object's hierarchical domain object is transferred. Host objects that are subject to transfer when transferring a domain object are listed in the response to an EPP <info> command performed on the domain object.

- A <host:auID> element derived from either the original host object creation transaction or the most recent successful hierarchical domain transfer transaction. This element MUST NOT be provided if the querying client is not the current sponsoring client.

Example <info> response:

```
S:<?xml version="1.0" standalone="no"?>
S:<epp xmlns="urn:iana:xml:ns:epp"
S:     xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
S:     xsi:schemaLocation="urn:iana:xml:ns:epp epp.xsd">
S:  <response>
S:    <result code="1000">
S:      <msg>Command completed successfully</msg>
S:    </result>
S:    <resData>
S:      <host:infData xmlns:host="urn:iana:xml:ns:host"
S:       xsi:schemaLocation="urn:iana:xml:ns:host host.xsd">
S:        <host:name>ns1.example.com</host:name>
S:        <host:roid>NS1EXAMPLE1-VRSN</host:roid>
S:        <host:addr ip="v4">192.1.2.3</host:addr>
S:        <host:addr ip="v6">1080:0:0:0:8:800:200C:417A</host:addr>
S:        <host:clID>ClientY</host:clID>
S:        <host:crID>ClientX</host:crID>
S:        <host:crDate>1999-04-03T22:00:00.0Z</host:crDate>
S:        <host:upID>ClientX</host:upID>
S:        <host:upDate>1999-12-03T09:00:00.0Z</host:upDate>
S:        <host:trDate>2000-04-08T09:00:00.0Z</host:trDate>
S:        <host:auID>
S:          <clTRID>ABC-12345</clTRID>
S:          <svTRID>54321-XYZ</svTRID>
S:        </host:auID>
S:      </host:infData>
S:    </resData>
S:    <unspec/>
S:    <trID>
S:      <clTRID>ABC-12346</clTRID>
S:      <svTRID>54322-XYZ</svTRID>
S:    </trID>
S:  </response>
S:</epp>
```

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

3.1.3 EPP <transfer> Command

Transfer semantics do not directly apply to host objects, so there is no mapping defined for the EPP <transfer> query command.

3.2 EPP Transform Commands

EPP provides three commands to transform host objects: <create> to create an instance of a host object, <delete> to delete an instance of a host object, and <update> to change information associated with a host object. This document does not define host object mappings for the EPP <renew> and <transfer> commands.

3.2.1 EPP <create> Command

The EPP <create> command provides a transform operation that allows a client to create a host object. In addition to the standard EPP command elements, the <create> command MUST contain a <host:create> element that identifies the host namespace and the location of the host schema. The <host:create> element SHALL contain the following child elements:

- A <host:name> element that contains the fully qualified name of the host object to be created.

- Zero or more <host:addr> elements that contain the IP addresses to be associated with the host. Each element MAY contain an "ip" attribute to identify the IP address format. Attribute value "v4" is used to note IPv4 address format. Attribute value "v6" is used to note IPv6 address format. If the "ip" attribute is not specified, value "v4" is the default value.

If the host name exists in a TLD for which the server is not authoritative, then IP addresses MUST NOT be provided for the host. For example, if the server is authoritative for the ".com" TLD and the name of the server is "ns1.example.com.au", then IP addresses MUST NOT be provided. If the host name exists in a TLD for which the server is authoritative, then the hierarchical domain of the host MUST be known to the server before the host object can be created.

Example <create> command:

```
C:<?xml version="1.0" standalone="no"?>
C:<epp xmlns="urn:iana:xml:ns:epp"
C:     xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
C:     xsi:schemaLocation="urn:iana:xml:ns:epp epp.xsd">
C:  <command>
C:    <create>
C:      <host:create xmlns:host="urn:iana:xml:ns:host"
C:       xsi:schemaLocation="urn:iana:xml:ns:host host.xsd">
C:        <host:name>ns1.example.com</host:name>
C:        <host:addr ip="v4">192.1.2.3</host:addr>
C:        <host:addr ip="v4">198.1.2.3</host:addr>
C:        <host:addr ip="v6">1080:0:0:0:8:800:200C:417A</host:addr>
C:        <host:addr ip="v6">::FFFF:129.144.52.38</host:addr>
C:      </host:create>
C:    </create>
C:    <unspec/>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

When a <create> command has been processed successfully, the EPP <resData> element MUST contain a child <host:creData> element that identifies the host namespace and the location of the host schema. The <host:creData> element SHALL contain the following child elements:

- A <host:name> element that contains the fully qualified name of the host object.

- A <host:roid> element that contains the host object's repository object identifier.

Example <create> response:

```
S:<?xml version="1.0" standalone="no"?>
S:<epp xmlns="urn:iana:xml:ns:epp"
S:     xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
S:     xsi:schemaLocation="urn:iana:xml:ns:epp epp.xsd">
S:  <response>
S:    <result code="1000">
S:      <msg>Command completed successfully</msg>
S:    </result>
S:    <resData>
S:      <host:creData xmlns:host="urn:iana:xml:ns:host"
S:       xsi:schemaLocation="urn:iana:xml:ns:host host.xsd">
S:        <host:name>ns1.example.com</host:name>
S:        <host:roid>NS1EXAMPLE1-VRSN</host:roid>
S:      </host:creData>
S:    </resData>
S:    <unspec/>
S:    <trID>
S:      <clTRID>ABC-12346</clTRID>
S:      <svTRID>54322-XYZ</svTRID>
S:    </trID>
S:  </response>
S:</epp>
```

An EPP error response MUST be returned if a <create> command can not be processed for any reason.

3.2.2 EPP <delete> Command

The EPP <delete> command provides a transform operation that allows a client to delete a host object. In addition to the standard EPP command elements, the <delete> command MUST contain a <host:delete> element that identifies the host namespace and the location of the host schema. The <host:delete> element SHALL contain the following child elements:

- A <host:name> element that contains the fully qualified name of the host object to be deleted.

A host name object MUST NOT be deleted if the host object is associated with any other object. For example, if the host object is associated with a domain object, the host object MUST NOT be deleted until the existing association has been broken.

Example <delete> command:

```
C:<?xml version="1.0" standalone="no"?>
C:<epp xmlns="urn:iana:xml:ns:epp"
C:     xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
C:     xsi:schemaLocation="urn:iana:xml:ns:epp epp.xsd">
C:  <command>
C:    <delete>
C:      <host:delete xmlns:host="urn:iana:xml:ns:host"
C:       xsi:schemaLocation="urn:iana:xml:ns:host host.xsd">
C:        <host:name>ns1.example.com</host:name>
C:      </host:delete>
C:      <auID>
C:        <clTRID>ABC-12345</clTRID>
C:        <svTRID>54321-XYZ</svTRID>
C:      </auID>
C:    </delete>
C:    <unspec/>
C:    <clTRID>ABC-12346</clTRID>
C:  </command>
C:</epp>
```

When a <delete> command has been processed successfully, a server MUST respond with an EPP response with no <resData> element.

Example <delete> response:

```
S:<?xml version="1.0" standalone="no"?>
S:<epp xmlns="urn:iana:xml:ns:epp"
S:     xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
S:     xsi:schemaLocation="urn:iana:xml:ns:epp epp.xsd">
S:  <response>
S:    <result code="1000">
S:      <msg>Command completed successfully</msg>
S:    </result>
S:    <unspec/>
S:    <trID>
S:      <clTRID>ABC-12346</clTRID>
S:      <svTRID>54322-XYZ</svTRID>
S:    </trID>
S:  </response>
S:</epp>
```

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

3.2.3 EPP <renew> Command

Renewal semantics do not apply to host objects, so there is no mapping defined for the EPP <renew> command.

3.2.4 EPP <transfer> Command

Transfer semantics do not directly apply to host objects, so there is no mapping defined for the EPP <transfer> command. Host objects are subordinate to an existing hierarchical domain object, and as such they are subject to transfer when a domain object is transferred.

3.2.5 EPP <update> Command

The EPP <update> command provides a transform operation that allows a client to modify the attributes of a host object. In addition to the standard EPP command elements, the <update> command MUST contain a <host:update> element that identifies the host namespace and the location of the host schema. The <host:update> element SHALL contain the following child elements:

- A <host:name> element that contains the fully qualified name of the host object to be updated.

- An OPTIONAL <host:add> element that contains attribute values to be added to the host object.

- An OPTIONAL <host:rem> element that contains attribute values to be removed from the host object.

- An OPTIONAL <host:chg> element that contains object attribute values to be changed.

The <host:add> and <host:rem> elements SHALL contain the following child elements:

- One or more <host:addr> elements that contain IP addresses to be associated with or removed from the host object. IP address restrictions explained in the <create> command mapping apply here as well.

A <host:chg> element SHALL contain the following child elements:

- A <host:name> element that contains a new fully qualified host name by which the host object will be known.

Host name changes MAY require the addition or removal of IP addresses to be accepted by the server. If a new host name exists in a TLD for which the server is not authoritative, then the host object MUST NOT have any associated IP addresses. If a new host name exists in a TLD

for which the server is authoritative, then the host object MUST have associated IP addresses.

Host name changes MAY have an impact on associated objects that refer to the host object. A host name change SHOULD not require additional updates of associated objects to preserve existing associations.

Example <update> command:

```
C:<?xml version="1.0" standalone="no"?>
C:<epp xmlns="urn:iana:xml:ns:epp"
C:     xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
C:     xsi:schemaLocation="urn:iana:xml:ns:epp epp.xsd">
C:  <command>
C:    <update>
C:      <host:update xmlns:host="urn:iana:xml:ns:host"
C:       xsi:schemaLocation="urn:iana:xml:ns:host host.xsd">
C:        <host:name>ns1.example.com</host:name>
C:        <host:add>
C:        <host:addr ip="v4">192.1.2.3</host:addr>
C:        </host:add>
C:        <host:rem>
C:          <host:addr ip="v6">1080:0:0:0:8:800:200C:417A</host:addr>
C:        </host:rem>
C:        <host:chg>
C:          <host:name>ns2.example.com</host:name>
C:        </host:chg>
C:      </host:update>
C:      <auID>
C:        <clTRID>ABC-12345</clTRID>
C:        <svTRID>54321-XYZ</svTRID>
C:      </auID>
C:    </update>
C:    <unspec/>
C:    <clTRID>ABC-12346</clTRID>
C:  </command>
C:</epp>
```

When an <update> command has been processed successfully, a server MUST respond with an EPP response with no <resData> element.

Example <update> response:

```
S:<?xml version="1.0" standalone="no"?>
S:<epp xmlns="urn:iana:xml:ns:epp"
S:     xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
S:     xsi:schemaLocation="urn:iana:xml:ns:epp epp.xsd">
S:  <response>
S:    <result code="1000">
S:      <msg>Command completed successfully</msg>
S:    </result>
S:    <unspec/>
S:    <trID>
S:      <clTRID>ABC-12346</clTRID>
S:      <svTRID>54322-XYZ</svTRID>
S:    </trID>
S:  </response>
S:</epp>
```

An EPP error response MUST be returned if an <update> command could not be processed for any reason.

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.

```
<?xml version="1.0"?>

<schema targetNamespace="urn:iana:xml:ns:host"
        xmlns:host="urn:iana:xml:ns:host"
        xmlns:epp="urn:iana:xml:ns:epp"
        xmlns:eppcom="urn:iana:xml:ns:eppcom"
        xmlns="http://www.w3.org/2000/10/XMLSchema"
        elementFormDefault="qualified">

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

  <annotation>
    <documentation>
      Extensible Provisioning Protocol v1.0
```

```
        host provisioning schema.
      </documentation>
    </annotation>

  <!--
  Types used within an EPP greeting.
  -->
    <element name="svc" type="eppcom:emptyType"/>

  <!--
  Child elements found in EPP commands.
  -->
    <element name="check" type="host:mNameType"/>
    <element name="create" type="host:createType"/>
    <element name="delete" type="host:sNameType"/>
    <element name="info" type="host:sNameType"/>
    <element name="update" type="host:updateType"/>

  <!--
  Child elements of the <create> command.
  -->
    <complexType name="createType">
      <sequence>
        <element name="name" type="eppcom:labelType"/>
        <element name="addr" type="host:addrType"
         minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
    </complexType>

    <complexType name="addrType">
      <simpleContent>
        <extension base="host:addrStringType">
          <attribute name="ip" type="host:ipType"
           use="default" value="v4"/>
        </extension>
      </simpleContent>
    </complexType>

    <simpleType name="addrStringType">
      <restriction base="string">
        <minLength value="3"/>
        <maxLength value="45"/>
      </restriction>
    </simpleType>

    <simpleType name="ipType">
      <restriction base="string">
        <enumeration value="v4"/>
```

```
        <enumeration value="v6"/>
      </restriction>
    </simpleType>

  <!--
  Child elements of the <delete> and <info> commands.
  -->
    <complexType name="sNameType">
      <sequence>
        <element name="name" type="eppcom:labelType"/>
      </sequence>
    </complexType>

  <!--
  Child element of commands that accept multiple names.
  -->
    <complexType name="mNameType">
      <sequence>
        <element name="name" type="eppcom:labelType"
         maxOccurs="unbounded"/>
      </sequence>
    </complexType>

  <!--
  Child elements of the <update> command.
  -->
    <complexType name="updateType">
      <sequence>
        <element name="name" type="eppcom:labelType"/>
        <element name="add" type="host:addRemType"
         minOccurs="0"/>
        <element name="rem" type="host:addRemType"
         minOccurs="0"/>
        <element name="chg" type="host:chgType"
         minOccurs="0"/>
      </sequence>
    </complexType>

    <complexType name="addRemType">
      <sequence>
        <element name="addr" type="host:addrType"
         maxOccurs="unbounded"/>
      </sequence>
    </complexType>

    <complexType name="chgType">
      <sequence>
        <element name="name" type="eppcom:labelType"/>
```

```
      </sequence>
    </complexType>

  <!--
  Child response elements.
  -->
    <element name="chkData" type="host:chkDataType"/>
    <element name="creData" type="host:creDataType"/>
    <element name="infData" type="host:infDataType"/>

  <!--
  <check> response elements.
  -->
    <complexType name="chkDataType">
      <sequence>
        <element name="cd" type="host:checkedType"
         maxOccurs="unbounded"/>
      </sequence>
    </complexType>

    <complexType name="checkedType">
      <simpleContent>
        <extension base = "eppcom:labelType">
          <attribute name="x" type="eppcom:checkStatusType"
           use="required"/>
        </extension>
      </simpleContent>
    </complexType>

  <!--
  <create> response elements.
  -->
    <complexType name="creDataType">
      <sequence>
        <element name="name" type="eppcom:labelType"/>
        <element name="roid" type="eppcom:roidType"/>
      </sequence>
    </complexType>

  <!--
  <info> response elements.
  -->
    <complexType name="infDataType">
      <sequence>
        <element name="name" type="eppcom:labelType"/>
        <element name="roid" type="eppcom:roidType"/>
        <element name="addr" type="host:addrType"
         maxOccurs="unbounded"/>
```

```
        <element name="clID" type="eppcom:clIDType"/>
        <element name="crID" type="eppcom:clIDType"/>
        <element name="crDate" type="timeInstant"/>
        <element name="upID" type="eppcom:clIDType"
         minOccurs="0"/>
        <element name="upDate" type="timeInstant"
         minOccurs="0"/>
        <element name="trDate" type="timeInstant"
         minOccurs="0"/>
        <element name="auID" type="epp:trIDType"
         minOccurs="0"/>
      </sequence>
    </complexType>

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

5. Internationalization Considerations

EPP is represented in XML, which provides native support for encoding information using the double-byte Unicode character set and its more compact representations including UTF-8. Compliant XML processors are required to understand both UTF-8 and raw Unicode character sets; XML also includes a provision for identifying other character sets through use of an "encoding" attribute in an <?xml?> processing instruction. The complete list of character set encoding identifiers is maintained by IANA and is described in [CHARSET] and [RFC1700].

All date-time values presented via EPP MUST be expressed in Universal Coordinated Time. The XML Schema "date" format allows use of time zone identifiers to indicate offsets from the zero meridian, but this option MUST NOT be used within EPP. Both extended and truncated date and time forms defined in [ISO8601] MAY be used.

This document requires host name syntax as specified in [RFC952] as updated by [RFC1123]. These conformance requirements MAY change as a result of progressing work in developing standards for internationalized host names.

6. IANA Considerations

XML schemas require a URI for unique identification. Schemas MUST be registered to ensure URI uniqueness, but the IETF does not currently have a recommended repository for the registration of XML schemas. This document uses URNs to describe XML namespaces and XML schemas conforming to a registry mechanism described in [IANA-XML].

IANA SHOULD maintain a registry of XML namespace and schema URI assignments. Per policies described in [IANA], URI assignment requests SHOULD be reviewed by a designated expert, and values SHOULD be assigned only as a result of standards action taken by the IESG.

This document requests assignment of the following URIs:

urn:iana:xml:ns:host: The XML namespace URI for this EPP host mapping.

urn:iana:xml:xmlschema:host: The XML Schema URI for this EPP host mapping.

7. Security Considerations

The object mapping described in this document does not provide any security services beyond those specified by [EPP] and protocol layers used by EPP.

8. References

[CHARSET] ftp://ftp.isi.edu/in-notes/iana/assignments/character-sets

[EPP] S. Hollenbeck: "Extensible Provisioning Protocol", draft-ietf-provreg-epp-01.txt, work in progress.

[IANA] T. Narten, H. Alvestrand: "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.

[IANA-XML] M. Mealling: "The IANA XML Registry", draft-mealling-iana-xmlns-registry-00.txt, work in progress.

[IANAa] http://www.isi.edu/in-notes/iana/assignments/ipv4-address-space

[IANAb] http://www.isi.edu/in-notes/iana/assignments/ipv6-address-space.txt

[ISO8601] ISO 8601:1988 (E): "Data elements and interchange formats - Information interchange - Representation of dates and times - The International Organization for Standardization".

[RFC791] J. Postel: "Internet Protocol", RFC 791, September 1981.

[RFC952] K. Harrenstien et al.: "DOD Internet Host Table Specification", RFC 952, October 1985.

[RFC1123] R. Braden: "Requirements for Internet Hosts -- Application and Support", RFC 1123, October 1989.

[RFC1700] J. Reynolds, J. Postel: "Assigned Numbers", STD 2, RFC 1700, October 1994.

[RFC1883] S. Deering and R. Hinden: "Internet Protocol, Version 6 (IPv6) Specification", RFC 1883, December 1995.

[RFC1918] Y. Rekhter et al.: "Address Allocation for Private Internets", BCP 5, RFC 1918, February 1996.

[RFC2119] S. Bradner: "Key Words for Use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[XML] Editor T. Bray et al.: "Extensible Markup Language (XML) 1.0 (Second Edition)", http://www.w3.org/TR/REC-xml, W3C Recommendation 6 October 2000.

[XMLS-1] Editor H. Thompson et al.: "XML Schema Part 1: Structures",

http://www.w3.org/TR/xmlschema-1, W3C Candidate Recommendation 24 October 2000.

[XMLS-2] Editors P. Biron and A. Malhotra: "XML Schema Part 2: Datatypes", http://www.w3.org/TR/xmlschema-2, W3C Candidate Recommendation 24 October 2000.

9. Author's Address

Scott Hollenbeck
VeriSign Global Registry Services
21345 Ridgetop Circle
Dulles, VA 20166-6503
USA
shollenbeck@verisign.com

A. Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.