

Network Working Group
Internet-Draft
Intended status: Informational
Expires: 16 September 2026

P. Garg
J. Gould
J. Colosi
VeriSign, Inc.
15 March 2026

Extensible Provisioning Protocol (EPP) Change Mapping
draft-garg-change-00

Abstract

This document describes an Extensible Provisioning Protocol (EPP) mapping for provisioning and management of change request objects in a shared central repository, where a change request is one unit of work that is processed by submitting to a workflow to execute the linked EPP transform commands in order. The change request is a container with meta-data to ensure that the contained commands are processed as a group.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 16 September 2026.

Copyright Notice

Copyright (c) 2026 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	2
1.1. Conventions Used in This Document	3
2. Object Attributes	3
2.1. Change Request Identifier	3
2.2. Change Status	4
2.3. Change Priority	4
2.4. Change Request Operations	4
2.5. Change Request Poll Message	5
3. EPP Command Mapping	6
3.1. EPP Query Commands	6
3.1.1. EPP <check> Command	7
3.1.2. EPP <info> Command	8
3.1.3. EPP <transfer> Query Command	12
3.2. EPP Transform Commands	12
3.2.1. EPP <create> Command	12
3.2.2. EPP <update> Command	13
3.2.3. EPP <delete> Command	18
3.2.4. EPP <renew> Command	19
3.2.5. EPP <transfer> Command	19
4. Formal Syntax	20
5. IANA Considerations	24
5.1. XML Namespace	24
5.2. EPP Extension Registry	25
6. Security Considerations	25
7. References	25
7.1. Normative References	25
7.2. Informative References	26
Authors' Addresses	26

1. Introduction

A change request object enables the creation, deletion, clearing, withdrawal, and submission of an ordered set of Extensible Provisioning Protocol (EPP) transform commands that are linked to a change request object with a change request identifier. The linked EPP transform commands will be executed only when the change request object is submitted.

This document describes a change object mapping for version 1.0 of the Extensible Provisioning Protocol (EPP) [6]. This mapping is specified using the XML 1.0 as described in [1] and XML Schema notation as described in [2] and [3].

1.1. Conventions Used in This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [4] when, and only when, they appear in all capitals, as shown here.

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 is provided only to illustrate element relationships and is not a REQUIRED feature of this specification.

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.

The XML namespace prefix "change" is used for the namespace "http://www.verisign-grs.com/epp/change-1.0", but implementations MUST NOT depend on it; instead, they should employ a proper namespace-aware XML parser and serializer to interpret and output the XML documents.

2. Object Attributes

An EPP change request 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. The formal syntax for the attribute values described here can be found in the "Formal Syntax" section of this document and in the appropriate normative references.

2.1. Change Request Identifier

All EPP Change Requests are identified by a server-unique identifier that is generated and provided by the client. Change Request Identifiers are character strings with a specific minimum length, a specified maximum length, and a specified format. Change Request Identifiers use the "clIDType" client identifier syntax described in [6]. Its corresponding element is <change:requestID>.

2.2. Change Status

The change request object contains a status value that can be made to support various state machine implementations. It is up to server policy to define a set of statuses. Each implementation will specify different behaviors for the submit operation of the <update> command as it is related directly to status modification.

The status values will be defined out-of-band of the protocol.

Some of the examples of status values are:

- * "initial": Change Request has been created
- * "submitted": Change Request has been submitted
- * "complete": Change Request has been completed

2.3. Change Priority

The change request object contains a priority value that can support order in which the change requests are implemented. It is up to server policy to define a set of priorities.

The priority values will be defined out-of-band of the protocol.

Some of the examples of priority values are:

- * "normal": Change Request need to be implemented within normal SLA
- * "urgent": Change Request need to be implemented within urgent SLA
- * "emergency": Change Request need to be implemented within emergency SLA

2.4. Change Request Operations

The change request object contains an operation value for the <update> command to define the operation being performed. Operation values are case sensitive.

Supported values are:

- * "upAttrs": Update the change request attributes
- * "clear": Remove all change action objects associated with this change request

- * "submit": Submit the change request to promote to next status
- * "withdraw": Withdraw the change request after submission

2.5. Change Request Poll Message

The EPP <poll> command and response is defined in section 2.9.2.3 of [6]. For servers that support a change request object, the Change Info Response, as defined in Section 3.1.2, is inserted into the poll queue.

Example <poll> command:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:<command xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <poll op="req"/>
C:  <clTRID>51364-CLI</clTRID>
C:</command>
C:</epp>
```

Figure 1

Example <poll> 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="1301">
S:      <msg>Command completed successfully; ack to dequeue</msg>
S:    </result>
S:    <msgQ count="1" id="12345">
S:      <qDate>2025-07-23T20:28:12.816Z</qDate>
S:      <msg>This Change Request has been completed</msg>
S:    </msgQ>
S:    <resData>
S:      <change:infData xmlns:change="http://www.verisign-grs.com/epp/change-1.0">
S:        <change:requestID>tk421</change:requestID>
S:        <change:priority>normal</change:priority>
S:        <change:category>EXAMPLE</change:category>
S:        <change:desc>A change request within .EXAMPLE</change:desc>
S:        <change:status>completed</change:status>
S:        <change:crDate>2025-07-11</change:crDate>
S:        <change:upDate>2025-07-23</change:upDate>
S:        <change:crID>userA</change:crID>
S:        <change:upID>userA</change:upID>
S:      </change:infData>
S:    </resData>
S:    <trID>
S:      <clTRID>51364-CLI</clTRID>
S:      <svTRID>SRV-43659</svTRID>
S:    </trID>
S:  </response>
S:</epp>
```

Figure 2

3. EPP Command Mapping

A detailed description of the EPP syntax and semantics can be found in the EPP core protocol specification [6]. The command mappings described here are specifically for use in provisioning and managing change information via EPP.

3.1. EPP Query Commands

This document provides two commands to retrieve change information:

<check> to determine if a change request object can be provisioned within a repository and <info> to retrieve detailed information associated with a change request object. This document does not define a mapping for the EPP <transfer> command to retrieve change-object transfer status information.

3.1.1.1. EPP <check> Command

The EPP <check> command is used to determine if an object can be provisioned within a repository. It provides a hint that allows a client to anticipate the success or failure of provisioning an object using the <create> command, as object-provisioning requirements are ultimately a matter of server policy.

In addition to the standard EPP command elements, the <check> command MUST contain an <change:check> element. This element or its ancestor element MUST identify the change namespace. The <change:check> element contains the following child elements:

- * One or more <change:requestID> elements that contain the server-unique identifier of the change request objects to be queried.

Example <check> 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:    <check>
C:      <change:check
C:        xmlns:change="http://www.verisign-grs.com/epp/change-1.0">
C:        <change:requestID>tk421</change:requestID>
C:        <change:requestID>thx1138</change:requestID>
C:      </change:check>
C:    </check>
C:    <clTRID>51364-CLI</clTRID>
C:  </command>
C:</epp>
```

Figure 3

When a <check> command has been processed successfully, the EPP <resData> element MUST contain a child <change:chkData> element that identifies the change namespace. The <change:chkData> element contains the following child elements:

one or more <change:cd> elements that contain the following child elements:

- * The element MUST contain an "exists" attribute whose value indicates the object existence at the moment the <check> command was completed. A value "1" or "true" means that the change request identifier exists and cannot be provisioned. A value of "0" or "false" means that the change request identifier does not exist and can be provisioned.

Example <check> 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"><msg>Command completed successfully</msg></result>
S:   <resData>
S:     <change:chkData
S:       xmlns:change="http://www.verisign-grs.com/epp/change-1.0">
S:       <change:cd exists="0">tk421</change:cd>
S:       <change:cd exists="1">thx1138</change:cd>
S:     </change:chkData>
S:   </resData>
S:   <trID>
S:     <clTRID>51364-CLI</clTRID>
S:     <svTRID>SRV-43659</svTRID>
S:   </trID>
S: </response>
S:</epp>
```

Figure 4

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

3.1.2. EPP <info> Command

The EPP <info> command is used to retrieve change request information based on the specified Change Request Identifier. In addition to the standard EPP command elements, the <info> command MUST contain a <change:info> element that identifies the change namespace. The <change:info> element contains the following child elements:

- * A <change:requestID> element that contain the server-unique identifier of the change request objects to be queried.

Example <info> 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:   <info>
C:     <change:info
C:       xmlns:change="http://www.verisign-grs.com/epp/change-1.0">
C:       <change:requestID>tk421</change:requestID>
C:     </change:info>
C:   </info>
C:   <clTRID>51364-CLI</clTRID>
C: </command>
C:</epp>
```

Figure 5

When an `<info>` command has been processed successfully, the EPP `<resData>` element MUST contain a child `<change:infData>` element that identifies the change namespace. The `<change:infData>` element contains the following child elements:

- * An `<change:requestID>` element that contains the change request identifier, defined in Section 2.1.
- * An `<change:priority>` element that contains the defined priority of the change request, described in Section 2.3.
- * One or more `<change:category>` elements that contain the Top Level Domain (TLD) for the change request. The TLD may contain a single dot character (0x2e) to represent a change to the root zone.
- * An `<change:desc>` element that contains a freeform description of the purpose or reason for the change request.
- * An `<change:status>` element that contains the server defined status of the change request, defined in Section 2.2.
- * An `<change:crDate>` element that contains the date of change request creation.
- * An `<change:upDate>` element that contains the date of the most recent change request modification.

The element MUST NOT be present if the change request has never been modified.

- * An `<change:crID>` element that contains the identifier of the client that created the change request object.

- * An <change:upID> element that contains the identifier of the client that last updated the change request object.

The element MUST NOT be present if the change request has never been modified.

- * Zero or more <change:action> elements with the following change action elements:
 - An OPTIONAL <change:requestID> element that contains the change request identifier, defined in Section 2.1.
 - An OPTIONAL <change:cltrid> element that contains client transaction identifier.
 - An <change:svtrid> elements that contains the server transaction identifier that is assigned by and unique to the server.
 - An OPTIONAL <change:crDate> element that contains the date of the change action.

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"><msg>Command completed successfully</msg></result>
S:   <resData>
S:     <change:infData
S:       xmlns:change="http://www.verisign-grs.com/epp/change-1.0">
S:       <change:requestID>tk421</change:requestID>
S:       <change:priority>normal</change:priority>
S:       <change:category>EXAMPLE</change:category>
S:       <change:category>.</change:category>
S:       <change:desc>A change request within .EXAMPLE</change:desc>
S:       <change:status>initial</change:status>
S:       <change:crDate>2026-09-21</change:crDate>
S:       <change:upDate>2026-10-30</change:upDate>
S:       <change:crID>userA</change:crID>
S:       <change:upID>userA</change:upID>
S:       <change:action>
S:         <change:requestID>tk421</change:requestID>
S:         <change:cltrid>51125-CLI</change:cltrid>
S:         <change:svtrid>SRV-10122</change:svtrid>
S:       </change:action>
S:       <change:action>
S:         <change:requestID>tk421</change:requestID>
S:         <change:svtrid>SRV-10321</change:svtrid>
S:       </change:action>
S:       <change:action>
S:         <change:requestID>tk421</change:requestID>
S:         <change:cltrid>51345-CLI</change:cltrid>
S:         <change:svtrid>SRV-10122</change:svtrid>
S:       </change:action>
S:     </change:infData>
S:   </resData>
S:   <trID>
S:     <clTRID>51364-CLI</clTRID>
S:     <svTRID>SRV-43659</svTRID>
S:   </trID>
S: </response>
S:</epp>

```

Figure 6

An EPP error response MUST be returned if a <info> command cannot be processed for any reason.

3.1.3. EPP <transfer> Query Command

The transfer semantics do not apply to change request objects. No EPP <transfer> query command is defined in this document.

3.2. EPP Transform Commands

This document provides three commands to transform change request object information:

<create> to create an instance of a change request object, <delete> to delete an instance of a change request object, and <update> to change information associated with a change request object. This document does not define a mapping for the EPP <transfer> and <renew> command.

3.2.1. EPP <create> Command

The EPP <create> command is used to construct a new change request object.

In addition to the standard EPP command elements, the <create> command MUST contain a <change:create> element that identifies the change namespace. The <change:create> element contains the following child elements:

- * An <change:requestID> element that contains the change request identifier, described in Section 2.1.
- * An OPTIONAL <change:priority> element that contains the defined priority of the change request, described in Section 2.3.
- * One or more <change:category> elements that contain the Top Level Domain (TLD) for the change request. The TLD may contain a single dot character (0x2e) to represent a change to the root zone.
- * An <change:desc> element that contains a freeform description of the purpose or reason for the change request.

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:     <change:create
C:       xmlns:change="http://www.verisign-grs.com/epp/change-1.0">
C:       <change:requestID>tk421</change:requestID>
C:       <change:priority>emergency</change:priority>
C:       <change:category>EXAMPLE</change:category>
C:       <change:desc>A new request within .EXAMPLE</change:desc>
C:     </change:create>
C:   </create>
C:   <clTRID>51364-CLI</clTRID>
C: </command>
C:</epp>
```

Figure 7

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

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"><msg>Command completed successfully</msg></result>
S:   <trID>
S:     <clTRID>51364-CLI</clTRID>
S:     <svTRID>SRV-43659</svTRID>
S:   </trID>
S: </response>
S:</epp>
```

Figure 8

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

3.2.2. EPP <update> Command

The EPP <update> command is used to modify an existing change request object.

In addition to the standard EPP command elements, the <update> command MUST contain a <change:update> element that identifies the change namespace. The <change:update> element contains the following child elements:

- * An <change:requestID> element that contains the change request identifier, described in Section 2.1.
- * A choice of one of the following change request operation elements, described in Section 2.4:
 - An <change:upAttrs> element to update the change request attributes that contains the following child elements, where at least one child element MUST be set:
 - o An OPTIONAL <change:priority> element that contains the defined priority of the change request, described in Section 2.3.
 - o Zero or more <change:category> elements that contain the Top Level Domain (TLD) for the change request. The TLD may contain a single dot character (0x2e) to represent a change to the root zone.
 - o An OPTIONAL <change:desc> that contains a freeform description of the purpose or reason for the change request.
 - An <change:clear> element to clear the change request actions. After execution of the clear operation, the change request will no longer be associated with any change request actions.
 - An <change:submit> element to submit the change request. Once the change request has been submitted, it can only be withdrawn.
 - An <change:withdraw> element to withdraw the change request. Once the submitted change request has been withdrawn, it can be deleted.

If the <change:upAttrs> element is present, then the change request record will be modified as specified by the <change:priority>, <change:category>, and <change:desc> elements.

NOTE: The <change:upAttrs> element MUST not be empty. The schema will not allow a <change:upAttrs> element without at least one of the <change:priority>, <change:category> or <change:desc> elements, described in Section 2.3.

If instead the <change:clear> element is present, then all change action objects associated with this change request will be removed. After execution the specified change request record will no longer be associated with any change action records.

If instead the `<change:submit>` element is present, then the specified change request record is promoted to a new status. Statuses supported will be implementation specific.

Example `<update>` command with `<upAttrs>`:

```
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:      <change:update
C:        xmlns:change="http://www.verisign-grs.com/epp/change-1.0">
C:        <change:requestID>tk421</change:requestID>
C:        <change:upAttrs>
C:          <change:category>EXAMPLE</change:category>
C:          <change:category>.</change:category>
C:          <change:desc>A change request within .EXAMPLE</change:desc>
C:        </change:upAttrs>
C:      </change:update>
C:    </update>
C:    <clTRID>51364-CLI</clTRID>
C:  </command>
C:</epp>
```

Figure 9

Example `<update>` command with `<clear>`:

```
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:      <change:update
C:        xmlns:change="http://www.verisign-grs.com/epp/change-1.0">
C:        <change:requestID>tk421</change:requestID>
C:        <change:clear/>
C:      </change:update>
C:    </update>
C:    <clTRID>51364-CLI</clTRID>
C:  </command>
C:</epp>
```

Figure 10

Example `<update>` command with `<submit>`:

```
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:     <change:update
C:       xmlns:change="http://www.verisign-grs.com/epp/change-1.0">
C:       <change:requestID>tk421</change:requestID>
C:       <change:submit/>
C:     </change:update>
C:   </update>
C:   <clTRID>51364-CLI</clTRID>
C: </command>
C:</epp>
```

Figure 11

When an <update> command has been processed successfully, the EPP <resData> element MUST contain a child <change:updData> element that identifies the change namespace. The <change:updData> element contains the following child elements:

- * An OPTIONAL <change:receipt> element that contains a submission receipt from change request submit operation. No <change:receipt> element will be returned for non-submit operations.

NOTE: The <change:receipt> element will appear only in a response to a <update> command with the <change:submit> element. No receipt is required when updating attributes or clearing a Change Request.

Example <update> response from an <update> command with a <change:submit> element:


```
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"><msg>Command completed successfully</msg></result>
S:   <resData>
S:     <change:updData
S:       xmlns:change="http://www.verisign-grs.com/epp/change-1.0">
S:       <change:receipt>
S:         [+] Begin Change Request Summary: DO NOT EDIT BELOW
S:
S:         Change Request Id:           tk421
S:         Request Priority:             NORMAL
S:         Top-level domain:            EXAMPLE
S:         Purpose/Description:         A change request within .EXAMPLE
S:
S:         Operations:
S:
S:         Domain Create
S:           Name:                      EXAMPLE
S:
S:         [-] End Change Request Summary: DO NOT EDIT ABOVE
S:       </change:receipt>
S:     </change:updData>
S:   </resData>
S:   <trID>
S:     <clTRID>51364-CLI</clTRID>
S:     <svTRID>SRV-43659</svTRID>
S:   </trID>
S: </response>
S:</epp>
```

Figure 12

Example <update> response from an <update> command with a <change:clear> or <change:upAttrs> element:

```
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"><msg>Command completed successfully</msg></result>
S:   <resData>
S:     <change:updData
S:       xmlns:change="http://www.verisign-grs.com/epp/change-1.0">
S:     </change:updData>
S:   </resData>
S:   <trID>
S:     <clTRID>51364-CLI</clTRID>
S:     <svTRID>SRV-43659</svTRID>
S:   </trID>
S: </response>
S:</epp>
```

Figure 13

An EPP error response MUST be returned if a <update> command cannot be processed for any reason.

3.2.3. EPP <delete> Command

The EPP <delete> command is used to delete a change request object before it is submitted.

In addition to the standard EPP command elements, the <delete> command MUST contain a <change:delete> element that identifies the change namespace. The <change:delete> element contains the following child elements:

- * An <change:requestID> element that contains the change request identifier, described in Section 2.1, for the change request object to be deleted.

Example <delete> 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:    <delete>
C:      <change:delete
C:        xmlns:change="http://www.verisign-grs.com/epp/change-1.0">
C:          <change:requestID>tk421</change:requestID>
C:        </change:delete>
C:      </delete>
C:    <clTRID>51364-CLI</clTRID>
C:  </command>
C:</epp>
```

Figure 14

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

Example <delete> 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"><msg>Command completed successfully</msg></result>
S:    <trID>
S:      <clTRID>51364-CLI</clTRID>
S:      <svTRID>SRV-43659</svTRID>
S:    </trID>
S:  </response>
S:</epp>
```

Figure 15

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

3.2.4. EPP <renew> Command

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

3.2.5. EPP <transfer> Command

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

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.

BEGIN

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.verisign-grs.com/epp/change-1.0"
  xmlns:change="http://www.verisign-grs.com/epp/change-1.0"
  xmlns:epp="urn:ietf:params:xml:ns:epp-1.0"
  xmlns:eppcom="urn:ietf:params:xml:ns:eppcom-1.0"
  xmlns="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">

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

  <annotation>
    <documentation>
      Extensible Provisioning Protocol v1.0
      Change provisioning schema
    </documentation>
  </annotation>

  <!-- Check command -->
  <element name="check" type="change:checkType"/>
  <complexType name="checkType">
    <sequence>
      <element name="requestID" type="epp:trIDStringType"
        maxOccurs="unbounded"/>
    </sequence>
  </complexType>

  <!-- Check response -->
  <element name="chkData" type="change:chkDataType"/>
  <complexType name="chkDataType">
    <sequence>
      <element name="cd" type="change:cdType"
        maxOccurs="unbounded"/>
    </sequence>
  </complexType>
```

```
<complexType name="cdType">
  <simpleContent>
    <extension base="epp:trIDStringType">
      <attribute name="exists" type="boolean" use="required"/>
    </extension>
  </simpleContent>
</complexType>

<!-- Info command -->
<element name="info" type="change:infoType"/>
<complexType name="infoType">
  <sequence>
    <element name="requestID" type="epp:trIDStringType"/>
  </sequence>
</complexType>

<!-- Info response -->
<element name="infData" type="change:infDataType"/>
<complexType name="infDataType">
  <sequence>
    <element name="requestID" type="epp:trIDStringType"/>
    <element name="priority" type="change:priorityType"/>
    <element name="category" type="change:categoryType"
      maxOccurs="unbounded"/>
    <element name="desc" type="change:descType"/>
    <element name="status" type="change:statusType"/>
    <element name="crDate" type="date"/>
    <element name="upDate" type="date"/>
    <element name="crID" type="epp:trIDStringType"/>
    <element name="upID" type="epp:trIDStringType"/>
    <element name="action" type="change:actionType"
      minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
</complexType>

<simpleType name="statusType">
  <restriction base="token">
    <minLength value="4"/>
    <maxLength value="32"/>
  </restriction>
</simpleType>

<complexType name="actionType">
  <sequence>
    <element name="requestID" type="epp:trIDStringType"
      minOccurs="0"/>
    <element name="cltrid" type="epp:trIDStringType"
      minOccurs="0"/>
  </sequence>
</complexType>
```

```
<element name="svtrid" type="epp:trIDStringType"/>
<element name="def" type="string"
minOccurs="0"/>
<element name="crDate" type="date"
minOccurs="0"/>
<element name="upDate" type="date"
minOccurs="0"/>
</sequence>
</complexType>

<!-- Create command -->
<element name="create" type="change:createType"/>
<complexType name="createType">
  <sequence>
    <element name="requestID" type="epp:trIDStringType"/>
    <element name="priority" type="change:priorityType"
minOccurs="0"/>
    <element name="category" type="change:categoryType"
maxOccurs="unbounded"/>
    <element name="desc" type="change:descType"/>
  </sequence>
</complexType>

<!-- Update command -->
<element name="update" type="change:updateType"/>
<complexType name="updateType">
  <sequence>
    <element name="requestID" type="epp:trIDStringType"/>
    <choice>
      <element name="upAttrs" type="change:upAttrsType"/>
      <element name="clear" type="change:emptyType"/>
      <element name="submit" type="change:emptyType"/>
      <element name="withdraw" type="change:emptyType"/>
    </choice>
  </sequence>
</complexType>

<complexType name="emptyType"/>

<complexType name="upAttrsType">
  <choice>
    <sequence>
      <element name="priority" type="change:priorityType"/>
      <element name="category" type="change:categoryType"
minOccurs="0" maxOccurs="unbounded"/>
      <element name="desc" type="change:descType"
minOccurs="0"/>
    </sequence>
  </choice>
</complexType>
```

```
<sequence>
  <element name="category" type="change:categoryType"
    maxOccurs="unbounded"/>
  <element name="desc" type="change:descType"
    minOccurs="0"/>
</sequence>
<sequence>
  <element name="desc" type="change:descType"/>
</sequence>
</choice>
</complexType>

<!-- Update response -->
<element name="updData" type="change:updDataType"/>
<complexType name="updDataType">
  <sequence>
    <element name="receipt" type="change:receiptType"
      minOccurs="0"/>
  </sequence>
</complexType>

<!-- No schema level length restrictions on the receipt -->
<simpleType name="receiptType">
  <restriction base="string">
    <whiteSpace value="preserve"/>
  </restriction>
</simpleType>

<!-- Delete command -->
<element name="delete" type="change:deleteType"/>
<complexType name="deleteType">
  <sequence>
    <element name="requestID" type="epp:trIDStringType"/>
  </sequence>
</complexType>

<!-- Common elements -->
<simpleType name="categoryType">
  <union memberTypes="change:labelType change:rootType"/>
</simpleType>

<simpleType name="rootType">
  <restriction base="token">
    <enumeration value="."/>
  </restriction>
</simpleType>

<simpleType name="labelType">
```

```
<restriction base="token">
  <!-- [2-63] characters -->
  <pattern value="[a-zA-Z0-9][a-zA-Z0-9\-\_]{0,61}[a-zA-Z0-9]"/>
</restriction>
</simpleType>

<simpleType name="descType">
  <restriction base="token">
    <minLength value="1"/>
    <maxLength value="256"/>
  </restriction>
</simpleType>

<simpleType name="priorityType">
  <restriction base="token">
    <minLength value="4"/>
    <maxLength value="20"/>
  </restriction>
</simpleType>

</schema>
END
```

5. IANA Considerations

5.1. XML Namespace

This document uses URNs to describe XML namespaces and XML schemas conforming to a registry mechanism described in [5]. The following URI assignment has been made by IANA:

Registration request for the Change Mapping namespace:

URI: <http://www.verisign-grs.com/epp/change-1.0>
Registrant Contact: VeriSign Inc., <epp-registry@verisign.com>
XML: None. Namespace URIs do not represent an XML specification.

Registration request for the Change Mapping XML Schema:

URI: <http://www.verisign-grs.com/epp/change-1.0>
Registrant Contact: VeriSign Inc., <epp-registry@verisign.com>
XML: See the "Formal Syntax" section of this document.

5.2. EPP Extension Registry

The EPP extension described in this document has been registered by IANA in the "Extensions for the Extensible Provisioning Protocol (EPP)" registry described in [7]. The details of the registration are as follows:

Name of Extension: "Extensible Provisioning Protocol (EPP) Change Mapping"

Document Status: Informational

Reference: (insert reference to RFC version of this document)

Registrant Name and Email Address: VeriSign Inc., <epp-registry@verisign.com>

TLDs: Any

IPR Disclosure: None

Status: Active

Notes: None

6. Security Considerations

The mapping extensions described in this document do not provide any security services beyond those described by EPP [6] and protocol layers used by EPP. The security considerations described in these other specifications apply to this specification as well.

7. References

7.1. Normative References

- [1] Bray, T., Paoli, J., Sperberg-McQueen, C., Maler, E., and F. Yergeau, "Extensible Markup Language (XML) 1.0 (Third Edition)", World Wide Web Consortium First Edition REC-xml-20040204", February 2004, <<http://www.w3.org/TR/2004/REC-xml-20040204>>.
- [2] Thompson, H., Beech, D., Maloney, M., and N. Mendelsohn, "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>>.
- [3] 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>>.
- [4] Best Current Practice 14, <<https://www.rfc-editor.org/info/bcp14>>.

At the time of writing, this BCP comprises the following:

Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

[5] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.

[6] Hollenbeck, S., "Extensible Provisioning Protocol (EPP)", STD 69, RFC 5730, DOI 10.17487/RFC5730, August 2009, <<https://www.rfc-editor.org/info/rfc5730>>.

7.2. Informative References

[7] Hollenbeck, S., "Extension Registry for the Extensible Provisioning Protocol", RFC 7451, DOI 10.17487/RFC7451, February 2015, <<https://www.rfc-editor.org/info/rfc7451>>.

Authors' Addresses

Poonam Garg
VeriSign, Inc.
12061 Bluemont Way
Reston, VA 20190
United States of America
Email: pogarg@verisign.com
URI: <http://www.verisign.com>

James Gould
VeriSign, Inc.
12061 Bluemont Way
Reston, VA 20190
United States of America
Email: jgould@verisign.com
URI: <http://www.verisign.com>

John Colosi
VeriSign, Inc.
12061 Bluemont Way
Reston, VA 20190
United States of America
Email: jcolosi@verisign.com
URI: <http://www.verisign.com>