Internet Engineering Task Force (IETF)
Request for Comments: 8590
Category: Standards Track
ISSN: 2070-1721

J. Gould
VeriSign, Inc.
K. Feher
Neustar

May 2019

Change Poll Extension for the Extensible Provisioning Protocol (EPP)

Abstract

This document describes an Extensible Provisioning Protocol (EPP) extension for notifying clients of operations on client-sponsored objects that were not initiated by the client through EPP. These operations may include contractual or policy requirements including, but not limited to, regular batch processes, customer support actions, Uniform Domain-Name Dispute-Resolution Policy (UDRP) or Uniform Rapid Suspension (URS) actions, court-directed actions, and bulk updates based on customer requests. Since the client is not directly involved or knowledgable of these operations, the extension is used along with an EPP object mapping to provide the resulting state of the postoperation object, and optionally a preoperation object, with the operation metadata of what, when, who, and why.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

This document describes an extension mapping for version 1.0 of the Extensible Provisioning Protocol (EPP) [RFC5730]. This mapping, an extension to EPP object mappings like the EPP domain name mapping [RFC5731], is used to notify clients of operations they are not directly involved in, on objects that the client sponsors. It is up to server policy to determine what transform operations and clients to notify. Using this extension, clients can more easily keep their systems in sync with the objects stored in the server. When a change occurs that a client needs to be notified of, a poll message can be inserted by the server for consumption by the client using the EPP <poll> command and response defined in [RFC5730]. The extension supports including a "before" operation poll message and an "after" operation poll message. The extension only extends the EPP <poll> response in [RFC5730] and does not extend the EPP <poll> command. Please refer to [RFC5730] for information and examples of the EPP <poll> command.

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 BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

XML is case sensitive. Unless stated otherwise, XML specifications and examples provided in this document MUST be interpreted in the character case presented in order to develop a conforming implementation.

In examples, "C:" represents lines sent by a protocol client, and "S:" represents lines returned by a protocol server. Indentation and white space in examples are provided only to illustrate element relationships and are not a REQUIRED feature of this protocol.

The XML namespace prefix "changePoll" is used for the namespace "urn:ietf:params:xml:ns:changePoll-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

This extension adds additional elements to EPP object mappings like the EPP domain name mapping [RFC5731]. Only those new elements are described here.

2.1. Operation

An operation consists of any transform operation that impacts objects that the client sponsors and should be notified of. The <changePoll:operation> element defines the operation. The OPTIONAL "op" attribute is an identifier, represented in the 7-bit US-ASCII character set defined in [RFC20], that is used to define a sub-operation or the name of a "custom" operation. The enumerated list of <changePoll:operation> values is:

"create": Create operation as defined in [RFC5730].

"delete": Delete operation as defined in [RFC5730]. If the delete operation results in an immediate purge of the object, then the "op" attribute MUST be set to "purge".

"renew": Renew operation as defined in [RFC5730].

"transfer": Transfer operation as defined in [RFC5730] that MUST set the "op" attribute with one of the possible transfer-type values; these include "request", "approve", "cancel", or "reject".

"update": Update operation as defined in [RFC5730].

"restore": Restore operation as defined in [RFC3915] that MUST set the "op" attribute with one of the possible restore-type values; these include "request" or "report".

"autoRenew": Auto renew operation executed by the server.

"autoDelete": Auto delete operation executed by the server. If the "autoDelete" operation results in an immediate purge of the object, then the "op" attribute MUST be set to "purge".

"autoPurge": Auto purge operation executed by the server when removing the object after it has had the status "pendingDelete".

"custom": Custom operation that MUST set the "op" attribute with the custom operation name. The custom operations supported are up to server policy.

2.2. State

The state attribute reflects the state of the object "before" or "after" the operation. The state is defined using the OPTIONAL "state" attribute of the <changePoll:changeData> element, with the possible values "before" or "after" and a default value of "after". The server MAY support both the "before" state and the "after" state of the operation by using one poll message for the "before" state and one poll message for the "after" state. The "before" state poll message MUST be inserted into the message queue prior to the "after" state poll message.

For operations in Section 2.1 that don’t have an "after" state, the server MUST use the "before" state poll message. For example, for the "delete" operation with the "op" attribute set to "purge", or the "autoPurge" operation, the server includes the state of the object prior to being purged in the "before" state poll message.

For operations in Section 2.1 that don’t have a "before" state, the server MUST use the "after" state poll message. For example, for the "create" operation, the server includes the state of the object after creation in the "after" state poll message.

2.3. Who

The <changePoll:who> element defines who executed the operation, for audit purposes. It is a free-form value that is meant strictly for audit purposes and not to drive client-side logic. The scheme used for the possible set of <changePoll:who> element values is up to server policy. The server MAY identify the <changePoll:who> element value based on:

"Identifier": Unique user identifier of the user that executed the operation. An example is "ClientX".

"Name": Name of the user that executed the operation. An example is "John Doe".

"Role": Role of the user that executed operation. An example is "CSR" for a Customer Support Representative or "Batch" for a server batch.

2.4. Dates and Times

Date and time attribute values MUST be represented in Universal Coordinated Time (UTC) using the Gregorian calendar. The extended date-time form using uppercase "T" and "Z" characters defined in

[W3C.REC-xmlschema-2-20041028] MUST be used to represent date-time values, as XML Schema does not support truncated date-time forms or lowercase "T" and "Z" characters.

3. EPP Command Mapping

A detailed description of the EPP syntax and semantics can be found in the EPP core protocol specification [RFC5730].

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 [RFC5730].

3.1.2. EPP <info> Command

This extension does not add any elements to the EPP <info> command described in [RFC5730].

This extension adds operational detail of EPP object-mapping operations (Section 2.1) to an EPP poll response, as described in [RFC5730]. It is an extension of the EPP object-mapping info response. Any transform operation to an object defined in an EPP object mapping by a client other than the sponsoring client MAY result in extending the <info> response of the object for inserting an EPP poll message with the operation detail. The sponsoring client will then receive the state of the object with operation detail like what, who, when, and why the object was changed. The <changePoll:changeData> element contains the operation detail along with an indication of whether the object reflects the state before or after the operation as defined in Section 2.2. The <changePoll:changeData> element includes the operation detail with the following child elements:

<changePoll:operation>: Transform operation executed on the object, as defined in Section 2.1.

<changePoll:date>: Date and time when the operation was executed.

<changePoll:svTRID>: Server transaction identifier of the operation.

<changePoll:who>: Who executed the operation, as defined in Section 2.3.

<changePoll:caseId>: OPTIONAL case identifier associated with the operation. The required "type" attribute defines the type of case. The OPTIONAL "name" attribute is an identifier, represented in the 7-bit US-ASCII character set defined in [RFC20], that is used to define the name of the "custom" case type. The enumerated list of case types is:

udrp: A Uniform Domain-Name Dispute-Resolution Policy (UDRP) case.

urs: A Uniform Rapid Suspension (URS) case.

custom: A custom case that is defined using the "name" attribute.

<changePoll:reason>: OPTIONAL reason for executing the operation. If present, this element contains the server-specific text to help explain the reason the operation was executed. This text MUST be represented in the response language previously negotiated with the client; an OPTIONAL "lang" attribute MAY be present to identify the language if the negotiated value is something other than the default value of "en" (English).

The following is an example poll <info> response with the <changePoll:changeData> extension for a URS lock transaction on the domain.example domain name, with the "before" state. The "before" state is reflected in the <resData> block:

```
S:<?xml version="1.0" encoding="UTF-8"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S:   <response>
S:      <result code="1301">
S:         <msg lang="en-US">
S:       Command completed successfully; ack to dequeue</msg>
S:      </result>
S:      <msgQ id="201" count="1">
S:         <qDate>2013-10-22T14:25:57.0Z</qDate>
S:         <msg>Registry initiated update of domain.</msg>
S:      </msgQ>
S:    <resData>
S:      <domain:infData
S:       xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
S:        <domain:name>domain.example</domain:name>
S:        <domain:roid>EXAMPLE1-REP</domain:roid>
S:        <domain:status s="ok"/>
```

```
   S:        <domain:registrant>jd1234</domain:registrant>
   S:        <domain:contact type="admin">sh8013</domain:contact>
   S:        <domain:contact type="tech">sh8013</domain:contact>
   S:        <domain:clID>ClientX</domain:clID>
   S:        <domain:crID>ClientY</domain:crID>
   S:        <domain:crDate>2012-04-03T22:00:00.0Z</domain:crDate>
   S:        <domain:exDate>2014-04-03T22:00:00.0Z</domain:exDate>
   S:      </domain:infData>
   S:    </resData>
   S:    <extension>
   S:      <changePoll:changeData
   S:        xmlns:changePoll="urn:ietf:params:xml:ns:changePoll-1.0"
   S:        state="before">
   S:        <changePoll:operation>update</changePoll:operation>
   S:        <changePoll:date>2013-10-22T14:25:57.0Z</changePoll:date>
   S:        <changePoll:svTRID>12345-XYZ</changePoll:svTRID>
   S:        <changePoll:who>URS Admin</changePoll:who>
   S:        <changePoll:caseId type="urs">urs123</changePoll:caseId>
   S:        <changePoll:reason>URS Lock</changePoll:reason>
   S:      </changePoll:changeData>
   S:    </extension>
   S:    <trID>
   S:      <clTRID>ABC-12345</clTRID>
   S:      <svTRID>54321-XYZ</svTRID>
   S:    </trID>
   S:   </response>
   S:</epp>
```

The following is an example poll <info> response with the <changePoll:changeData> extension for a URS lock transaction on the domain.example domain name, with the "after" state. The "after" state is reflected in the <resData> block:

```
   S:<?xml version="1.0" encoding="UTF-8"?>
   S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
   S:   <response>
   S:      <result code="1301">
   S:         <msg lang="en-US">
   S:       Command completed successfully; ack to dequeue</msg>
   S:      </result>
   S:      <msgQ id="202" count="1">
   S:         <qDate>2013-10-22T14:25:57.0Z</qDate>
   S:         <msg>Registry initiated update of domain.</msg>
   S:      </msgQ>
   S:    <resData>
   S:      <domain:infData
   S:       xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
   S:        <domain:name>domain.example</domain:name>
```

```
S:        <domain:roid>EXAMPLE1-REP</domain:roid>
S:        <domain:status s="serverUpdateProhibited"/>
S:        <domain:status s="serverDeleteProhibited"/>
S:        <domain:status s="serverTransferProhibited"/>
S:        <domain:registrant>jd1234</domain:registrant>
S:        <domain:contact type="admin">sh8013</domain:contact>
S:        <domain:contact type="tech">sh8013</domain:contact>
S:        <domain:clID>ClientX</domain:clID>
S:        <domain:crID>ClientY</domain:crID>
S:        <domain:crDate>2012-04-03T22:00:00.0Z</domain:crDate>
S:        <domain:upID>ClientZ</domain:upID>
S:        <domain:upDate>2013-10-22T14:25:57.0Z</domain:upDate>
S:        <domain:exDate>2014-04-03T22:00:00.0Z</domain:exDate>
S:      </domain:infData>
S:    </resData>
S:    <extension>
S:      <changePoll:changeData
S:        xmlns:changePoll="urn:ietf:params:xml:ns:changePoll-1.0"
S:        state="after">
S:        <changePoll:operation>update</changePoll:operation>
S:        <changePoll:date>2013-10-22T14:25:57.0Z</changePoll:date>
S:        <changePoll:svTRID>12345-XYZ</changePoll:svTRID>
S:        <changePoll:who>URS Admin</changePoll:who>
S:        <changePoll:caseId type="urs">urs123</changePoll:caseId>
S:        <changePoll:reason>URS Lock</changePoll:reason>
S:      </changePoll:changeData>
S:    </extension>
S:    <trID>
S:      <clTRID>ABC-12345</clTRID>
S:      <svTRID>54321-XYZ</svTRID>
S:    </trID>
S:   </response>
S:</epp>
```

The following is an example poll <info> response with the <changePoll:changeData> extension for a custom "sync" operation on the domain.example domain name, with the default "after" state. The "after" state is reflected in the <resData> block:

```
S:<?xml version="1.0" encoding="UTF-8"?>
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 id="201" count="1">
S:         <qDate>2013-10-22T14:25:57.0Z</qDate>
S:    <msg>Registry initiated Sync of Domain Expiration Date</msg>
```

```
   S:      </msgQ>
   S:    <resData>
   S:      <domain:infData
   S:       xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
   S:        <domain:name>domain.example</domain:name>
   S:        <domain:roid>EXAMPLE1-REP</domain:roid>
   S:        <domain:status s="ok"/>
   S:        <domain:registrant>jd1234</domain:registrant>
   S:        <domain:contact type="admin">sh8013</domain:contact>
   S:        <domain:contact type="tech">sh8013</domain:contact>
   S:        <domain:clID>ClientX</domain:clID>
   S:        <domain:crID>ClientY</domain:crID>
   S:        <domain:crDate>2012-04-03T22:00:00.0Z</domain:crDate>
   S:        <domain:upID>ClientZ</domain:upID>
   S:        <domain:upDate>2013-10-22T14:25:57.0Z</domain:upDate>
   S:        <domain:exDate>2014-04-03T22:00:00.0Z</domain:exDate>
   S:      </domain:infData>
   S:    </resData>
   S:    <extension>
   S:      <changePoll:changeData
   S:        xmlns:changePoll="urn:ietf:params:xml:ns:changePoll-1.0">
   S:        <changePoll:operation op="sync">custom
   S:        </changePoll:operation>
   S:        <changePoll:date>2013-10-22T14:25:57.0Z</changePoll:date>
   S:        <changePoll:svTRID>12345-XYZ</changePoll:svTRID>
   S:        <changePoll:who>CSR</changePoll:who>
   S:        <changePoll:reason lang="en">Customer sync request
   S:        </changePoll:reason>
   S:      </changePoll:changeData>
   S:    </extension>
   S:    <trID>
   S:      <clTRID>ABC-12345</clTRID>
   S:      <svTRID>54321-XYZ</svTRID>
   S:    </trID>
   S:   </response>
   S:</epp>
```

The following is an example poll <info> response with the <changePoll:changeData> extension for a "delete" operation on the domain.example domain name that is immediately purged, with the "before" state. The "before" state is reflected in the <resData> block:

```
S:<?xml version="1.0" encoding="UTF-8"?>
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 id="200" count="1">
S:      <qDate>2013-10-22T14:25:57.0Z</qDate>
S:      <msg>Registry initiated delete of
S:         domain resulting in immediate purge.</msg>
S:    </msgQ>
S:    <resData>
S:      <domain:infData
S:        xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
S:        <domain:name>domain.example</domain:name>
S:        <domain:roid>EXAMPLE1-REP</domain:roid>
S:        <domain:clID>ClientX</domain:clID>
S:      </domain:infData>
S:    </resData>
S:    <extension>
S:      <changePoll:changeData
S:        xmlns:changePoll="urn:ietf:params:xml:ns:changePoll-1.0"
S:        state="before">
S:        <changePoll:operation op="purge">delete
S:        </changePoll:operation>
S:        <changePoll:date>2013-10-22T14:25:57.0Z
S:        </changePoll:date>
S:        <changePoll:svTRID>12345-XYZ
S:        </changePoll:svTRID>
S:        <changePoll:who>ClientZ
S:        </changePoll:who>
S:        <changePoll:reason>Court order
S:        </changePoll:reason>
S:      </changePoll:changeData>
S:    </extension>
S:    <trID>
S:      <clTRID>ABC-12345</clTRID>
S:      <svTRID>54321-XYZ</svTRID>
S:    </trID>
S:  </response>
S:</epp>
```

The following is an example poll <info> response with the <changePoll:changeData> extension for an "autoPurge" operation on the domain.example domain name that previously had the "pendingDelete" status, with the "before" state. The "before" state is reflected in the <resData> block:

```
S:<?xml version="1.0" encoding="UTF-8"?>
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
S:      </msg>
S:    </result>
S:    <msgQ id="200" count="1">
S:      <qDate>2013-10-22T14:25:57.0Z</qDate>
S:      <msg>Registry purged domain with pendingDelete status.
S:      </msg>
S:    </msgQ>
S:    <resData>
S:      <domain:infData
S:        xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
S:        <domain:name>domain.example</domain:name>
S:        <domain:roid>EXAMPLE1-REP</domain:roid>
S:        <domain:status s="pendingDelete"/>
S:        <domain:clID>ClientX</domain:clID>
S:      </domain:infData>
S:    </resData>
S:    <extension>
S:      <changePoll:changeData
S:        xmlns:changePoll="urn:ietf:params:xml:ns:changePoll-1.0"
S:        state="before">
S:        <changePoll:operation>autoPurge
S:        </changePoll:operation>
S:        <changePoll:date>2013-10-22T14:25:57.0Z
S:        </changePoll:date>
S:        <changePoll:svTRID>12345-XYZ
S:        </changePoll:svTRID>
S:        <changePoll:who>Batch
S:        </changePoll:who>
S:        <changePoll:reason>Past pendingDelete 5 day period
S:        </changePoll:reason>
S:      </changePoll:changeData>
S:    </extension>
S:    <trID>
S:      <clTRID>ABC-12345</clTRID>
S:      <svTRID>54321-XYZ</svTRID>
S:    </trID>
```

```
S:  </response>
S:</epp>
```

The following is an example poll <info> response with the <changePoll:changeData> extension for an "update" operation on the ns1.domain.example host, with the default "after" state. The "after" state is reflected in the <resData> block:

```
S:<?xml version="1.0" encoding="UTF-8"?>
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 id="201" count="1">
S:         <qDate>2013-10-22T14:25:57.0Z</qDate>
S:         <msg>Registry initiated update of host.</msg>
S:      </msgQ>
S:    <resData>
S:      <host:infData
S:       xmlns:host="urn:ietf:params:xml:ns:host-1.0">
S:        <host:name>ns1.domain.example</host:name>
S:        <host:roid>NS1_EXAMPLE1-REP</host:roid>
S:        <host:status s="linked"/>
S:        <host:status s="serverUpdateProhibited"/>
S:        <host:status s="serverDeleteProhibited"/>
S:        <host:addr ip="v4">192.0.2.2</host:addr>
S:        <host:addr ip="v6">2001:db8:0:0:1:0:0:1</host:addr>
S:        <host:clID>ClientX</host:clID>
S:        <host:crID>ClientY</host:crID>
S:        <host:crDate>2012-04-03T22:00:00.0Z</host:crDate>
S:        <host:upID>ClientY</host:upID>
S:        <host:upDate>2013-10-22T14:25:57.0Z</host:upDate>
S:      </host:infData>
S:    </resData>
S:    <extension>
S:      <changePoll:changeData
S:        xmlns:changePoll="urn:ietf:params:xml:ns:changePoll-1.0">
S:        <changePoll:operation>update</changePoll:operation>
S:        <changePoll:date>2013-10-22T14:25:57.0Z</changePoll:date>
S:        <changePoll:svTRID>12345-XYZ</changePoll:svTRID>
S:        <changePoll:who>ClientZ</changePoll:who>
S:        <changePoll:reason>Host Lock</changePoll:reason>
S:      </changePoll:changeData>
S:    </extension>
S:    <trID>
S:      <clTRID>ABC-12345</clTRID>
S:      <svTRID>54321-XYZ</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> response described in the [RFC5730].

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 does not add any elements to the EPP <create> command or <create> response described in [RFC5730].

3.2.2. EPP <delete> Command

This extension does not add any elements to the EPP <delete> command or <delete> response described in [RFC5730].

3.2.3. EPP <renew> Command

This extension does not add any elements to the EPP <renew> command or <renew> response described in [RFC5730].

3.2.4. EPP <transfer> Command

This extension does not add any elements to the EPP <transfer> command or <transfer> response described in [RFC5730].

3.2.5. EPP <update> Command

This extension does not add any elements to the EPP <update> command or <update> response described in [RFC5730].

4. Formal Syntax

One schema is presented here: the EPP Change Poll Extension schema.

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.

4.1. Change Poll Extension Schema

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

  <!--
  Import common element types.
  -->
  <import namespace="urn:ietf:params:xml:ns:eppcom-1.0"/>
  <import namespace="urn:ietf:params:xml:ns:epp-1.0"/>


   <annotation>
     <documentation>
       Extensible Provisioning Protocol v1.0
       Change Poll Mapping Schema.
     </documentation>
   </annotation>

   <!--
    Change element.
   -->
   <element name="changeData" type="changePoll:changeDataType"/>

   <!--
    Attributes associated with the change.
   -->
   <complexType name="changeDataType">
     <sequence>
       <element name="operation" type="changePoll:operationType"/>
       <element name="date" type="dateTime"/>
       <element name="svTRID" type="epp:trIDStringType"/>
```

```
          <element name="who" type="changePoll:whoType"/>
          <element name="caseId" type="changePoll:caseIdType"
            minOccurs="0"/>
          <element name="reason" type="eppcom:reasonType"
            minOccurs="0"/>
        </sequence>
       <attribute name="state" type="changePoll:stateType"
         default="after"/>
      </complexType>

      <!--
       Enumerated list of operations, with extensibility via "custom".
      -->
      <simpleType name="operationEnum">
        <restriction base="token">
          <enumeration value="create"/>
          <enumeration value="delete"/>
          <enumeration value="renew"/>
          <enumeration value="transfer"/>
          <enumeration value="update"/>
          <enumeration value="restore"/>
          <enumeration value="autoRenew"/>
          <enumeration value="autoDelete"/>
          <enumeration value="autoPurge"/>
          <enumeration value="custom"/>
        </restriction>
      </simpleType>

      <!--
       Enumerated state of the object in the poll message.
      -->
      <simpleType name="stateType">
        <restriction base="token">
          <enumeration value="before"/>
          <enumeration value="after"/>
        </restriction>
      </simpleType>

      <!--
       Transform operation type
      -->
     <complexType name="operationType">
       <simpleContent>
         <extension base="changePoll:operationEnum">
           <attribute name="op" type="token"/>
         </extension>
       </simpleContent>
     </complexType>
```

```
      <!--
       Case identifier type
      -->
     <complexType name="caseIdType">
       <simpleContent>
         <extension base="token">
           <attribute name="type" type="changePoll:caseTypeEnum"
             use="required"/>
           <attribute name="name" type="token"
             use="optional"/>
         </extension>
       </simpleContent>
     </complexType>

      <!--
       Enumerated list of case identifier types
      -->
      <simpleType name="caseTypeEnum">
        <restriction base="token">
          <enumeration value="udrp"/>
          <enumeration value="urs"/>
          <enumeration value="custom"/>
        </restriction>
      </simpleType>

      <!--
       Who type
      -->
     <simpleType name="whoType">
       <restriction base="normalizedString">
         <minLength value="1"/>
         <maxLength value="255"/>
       </restriction>
     </simpleType>

    <!--
    End of schema.
    -->
    </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 [RFC3688]. The following URI assignment has been made by IANA:

Registration for the changePoll namespace:

URI: urn:ietf:params:xml:ns:changePoll-1.0

Registrant Contact: IESG

XML: None. Namespace URIs do not represent an XML specification.

Registration for the changePoll XML schema:

URI: urn:ietf:params:xml:ns:changePoll-1.0

Registrant Contact: IESG

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 the IANA in the Extensions for the "Extensible Provisioning Protocol (EPP)" registry described in [RFC7451]. The details of the registration are as follows:

Name of Extension: Change Poll Extension for the Extensible Provisioning Protocol (EPP)

Document Status: Standards Track

Reference: RFC 8590

Registrant Name and Email Address: IESG <iesg@ietf.org>

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 [RFC5730] 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

[RFC20] Cerf, V., "ASCII format for network interchange", STD 80, RFC 20, DOI 10.17487/RFC0020, October 1969, <https://www.rfc-editor.org/info/rfc20>.

[RFC2119] 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>.

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

[RFC3915] Hollenbeck, S., "Domain Registry Grace Period Mapping for the Extensible Provisioning Protocol (EPP)", RFC 3915, DOI 10.17487/RFC3915, September 2004, <https://www.rfc-editor.org/info/rfc3915>.

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

[RFC5731] Hollenbeck, S., "Extensible Provisioning Protocol (EPP) Domain Name Mapping", STD 69, RFC 5731, DOI 10.17487/RFC5731, August 2009, <https://www.rfc-editor.org/info/rfc5731>.

[RFC8174] 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>.

[W3C.REC-xmlschema-2-20041028]
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>.

7.2. Informative References

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

Acknowledgments

The authors wish to acknowledge the original concept for this document and the efforts in the initial versions of this document by Trung Tran and Sharon Wodjenski.

Special suggestions that have been incorporated into this document were provided by Scott Hollenbeck, Michael Holloway, and Patrick Mevzek.

Authors' Addresses

James Gould
VeriSign, Inc.
12061 Bluemont Way
Reston, VA 20190
United States of America

Email: jgould@verisign.com
URI: http://www.verisign.com

Kal Feher
Neustar
lvl 8/10 Queens Road
Melbourne, VIC 3004
Australia

Email: ietf@feherfamily.org
URI: http://www.neustar.biz