Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 4, 2018

J. Gould
VeriSign, Inc.
K. Feher
Neustar
July 3, 2017

Change Poll Extension for the Extensible Provisioning Protocol (EPP) draft-ietf-regext-change-poll-03

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 post-operation object, and optionally a pre-operation object, with the operation meta-data of what, when, who, and why.

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 http://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 January 4, 2018.

Copyright Notice

Copyright (c) 2017 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 (http://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 Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction 3
1.1. Conventions Used in This Document 3
2. Object Attributes 3
2.1. Operation 4
2.2. Who 4
3. EPP Command Mapping 5
3.1. EPP Query Commands 5
3.1.1. EPP <check> Command 5
3.1.2. EPP <info> Command 5
3.1.3. EPP <transfer> Command 15
3.2. EPP Transform Commands 15
3.2.1. EPP <create> Command 15
3.2.2. EPP <delete> Command 15
3.2.3. EPP <renew> Command 15
3.2.4. EPP <transfer> Command 15
3.2.5. EPP <update> Command 15
4. Formal Syntax 15
4.1. Change Poll Extension Schema 16
5. IANA Considerations 18
5.1. XML Namespace 18
5.2. EPP Extension Registry 19
6. Implementation Status 19
6.1. Verisign EPP SDK 20
6.2. Verisign Consolidated Top Level Domain (CTLD) SRS 20
6.3. Verisign .COM / .NET SRS 20
7. Security Considerations 21
8. Acknowledgements 21
9. Normative References 21
Appendix A. Change History 22
A.1. Change from 00 to 01 22
A.2. Change from 01 to 02 22
A.3. Change from 02 to 03 22
A.4. Change from 03 to 04 22
A.5. Change from 04 to 05 22
A.6. Change from 05 to REGEXT 00 22
A.7. Change from REGEXT 00 to REGEXT 01 23

A.8. Change from REGEXT 01 to REGEXT 02 23
A.9. Change from REGEXT 02 to REGEXT 03 23
Authors' Addresses 23

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.

1.1. Conventions Used in This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

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.

"changePoll-1.0" is used as an abbreviation for "urn:ietf:params:xml:ns:changePoll-1.0". The XML namespace prefix "changePoll" is used, but implementations MUST NOT depend on it and instead 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 sponsers and SHOULD be notified of. The <changePoll:operation> element defines the operation. The OPTIONAL "op" attribute is used to define a sub-operation or the name of a "custom" operation. The enumerated list of <changePoll:operation> values include:

"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] with the OPTIONAL "op" attribute defining the transfer type with the possible values of "request", "approve", "cancel", and "reject".
"update" Update operation as defined in [RFC5730].
"restore" Restore operation as defined in [RFC3915] with the OPTIONAL "op" attribute defining the restore type with the possible values of "request" and "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 had the "pendingDelete" status.
"custom" Custom operation that uses the "op" attribute to define the custom operation name.

2.2. Who

Who defines who executed the operation for audit purposes, and is represented using the <changePoll:who> element. The scheme used for the possible set of Who values is up to server policy. The server MAY identify Who 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.

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 the [RFC5730].

3.1.2. EPP <info> Command

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

This extension adds transaction detail of the operations to the EPP <info> poll response, as described in [RFC5730], of an EPP Object Mapping like [RFC5731]. 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, using the OPTIONAL "state" attribute, with the possible values of "before" or "after", and with a default value of "after". The "state" attribute describes the state of the response data or <resData> block returned in the poll response. The server MAY support providing the "before" state and "after" state to the operation, by using one poll message for the "before" state and one poll message for the "after" state. When using the "before" state poll message, it MUST be inserted prior to the "after" state poll message. 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 2.2.

<changePoll:caseId> OPTIONAL case identifer associated with the operation. The required "type" attribute defines the type of case with an enumerated list of case types including:

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).

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>
```

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>
```

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>
```

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 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="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:          <changePoll:operation op="purge">
S:            delete</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>Court order</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>
```

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 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="200" count="1">
S:         <qDate>2013-10-22T14:25:57.0Z</qDate>
S:    <msg>Registry purged domain with pendingDelete status.</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:          <changePoll:operation>
S:            autoPurge</changePoll:operation>
S:          <changePoll:date>2013-10-22T14:25:57.0Z</changePoll:date>
S:          <changePoll:svTRID>12345-XYZ</changePoll:svTRID>
S:          <changePoll:who>Batch</changePoll:who>
S:          <changePoll:reason>
S:            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>
```

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">1080:0:0:0:8:800:200C:417A</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 the [RFC5730].

3.2.2. EPP <delete> Command

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

3.2.3. EPP <renew> Command

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

3.2.4. EPP <transfer> Command

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

3.2.5. EPP <update> Command

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

4. Formal Syntax

One schema is presented here that is 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 of 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 is requested of IANA:

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

Registrant Contact: See the "Author's Address" section of this document.

XML: See the "Formal Syntax" section of this document.

5.2. EPP Extension Registry

The EPP extension described in this document should be registered by the IANA in the EPP Extension 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: (insert reference to RFC version of this document)

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

TLDs: Any

IPR Disclosure: None

Status: Active

Notes: None

6. Implementation Status

Note to RFC Editor: Please remove this section and the reference to RFC 6982 [RFC6982] before publication.

This section records the status of known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft, and is based on a proposal described in RFC 6982 [RFC6982]. The description of implementations in this section is intended to assist the IETF in its decision processes in progressing drafts to RFCs. Please note that the listing of any individual implementation here does not imply endorsement by the IETF. Furthermore, no effort has been spent to verify the information presented here that was supplied by IETF contributors. This is not intended as, and must not be construed to be, a catalog of available implementations or their features. Readers are advised to note that other implementations may exist.

According to RFC 6982 [RFC6982], "this will allow reviewers and working groups to assign due consideration to documents that have the benefit of running code, which may serve as evidence of valuable experimentation and feedback that have made the implemented protocols more mature. It is up to the individual working groups to use this information as they see fit".

6.1. Verisign EPP SDK

Organization: Verisign Inc.

Name: Verisign EPP SDK

Description: The Verisign EPP SDK includes both a full client implementation and a full server stub implementation of draft-ietf-regext-change-poll.

Level of maturity: Production

Coverage: All aspects of the protocol are implemented.

Licensing: GNU Lesser General Public License

Contact: jgould@verisign.com

URL: https://www.verisign.com/en_US/channel-resources/domain-registry-products/epp-sdks

6.2. Verisign Consolidated Top Level Domain (CTLD) SRS

Organization: Verisign Inc.

Name: Verisign Consolidated Top Level Domain (CTLD) Shared Registry System (SRS)

Description: The Verisign Consolidated Top Level Domain (CTLD) Shared Registry System (SRS) implements the server-side of draft-ietf-regext-change-poll for a variety of Top Level Domains (TLD's).

Level of maturity: Production

Coverage: The "after" state poll message for an "update" transform operation of a domain name due to server policy.

Licensing: Proprietary

Contact: jgould@verisign.com

6.3. Verisign .COM / .NET SRS

Organization: Verisign Inc.

Name: Verisign .COM / .NET Shared Registry System (SRS)

Description: The Verisign Shared Registry System (SRS) for .COM and .NET implements the server-side of draft-ietf-regext-change-poll.

Level of maturity: Production

Coverage: The "after" state poll message for an "update" transform operation of a domain name due to server policy.

Licensing: Proprietary

Contact: jgould@verisign.com

7. 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.

8. Acknowledgements

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

Special suggestions that have been incorporated into this document were provided by Michael Holloway.

9. Normative References

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

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <http://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, <http://www.rfc-editor.org/info/rfc3915>.

[RFC5730] Hollenbeck, S., "Extensible Provisioning Protocol (EPP)", STD 69, RFC 5730, DOI 10.17487/RFC5730, August 2009, <http://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, <http://www.rfc-editor.org/info/rfc5731>.

[RFC6982] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", RFC 6982, DOI 10.17487/RFC6982, July 2013, <http://www.rfc-editor.org/info/rfc6982>.

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

Appendix A. Change History

A.1. Change from 00 to 01

1. Added an optional caseId element that defines the case identifier from UDRP, URS, or custom case, based on feedback from Michael Holloway.

A.2. Change from 01 to 02

1. Amended XML Namespace section of IANA Considerations, added EPP Extension Registry section.
2. Moved Change History to the back section as an Appendix.

A.3. Change from 02 to 03

1. Fixed "before" state example to use the "before" state value based on feedback from Patrick Mevzek.

A.4. Change from 03 to 04

1. Updated the authors for the draft.

A.5. Change from 04 to 05

1. Ping update.

A.6. Change from 05 to REGEXT 00

1. Changed to regext working group draft by changing draft-gould-change-poll to draft-ietf-regext-change-poll.

A.7. Change from REGEXT 00 to REGEXT 01

1. Ping update.

A.8. Change from REGEXT 01 to REGEXT 02

1. Added the Implementation Status section.

A.9. Change from REGEXT 02 to REGEXT 03

1. Changed Neustar author to Kal Feher.

Authors' Addresses

James Gould
VeriSign, Inc.
12061 Bluemont Way
Reston, VA 20190
US

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

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

Email: kal.feher@team.neustar
URI: http://www.neustar.biz