Change Poll Extension for the Extensible Provisioning Protocol (EPP)  
draft-ietf-regext-change-poll-12

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 July 8, 2019.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.
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 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 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 [RFC0020], 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 that 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 that 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 had the "pendingDelete" status.
"custom": Custom operation that MUST set the "op" attribute with the custom operation name. The custom operations supported is 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 with 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 freeform value that is strictly meant for audit purposes and not meant 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 upper case "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 lower case "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 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 operation detail of EPP object mapping operations Section 2.1 to an EPP poll response, as described in [RFC5730]. The extension 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 [RFC0020], 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).

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:
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: <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:    <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:
<xml version="1.0" encoding="UTF-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1301">
      <msg>Command completed successfully; ack to dequeue</msg>
    </result>
    <msgQ id="201" count="1">
      <qDate>2013-10-22T14:25:57.0Z</qDate>
      <msg>Registry initiated Sync of Domain Expiration Date</msg>
    </msgQ>
    <resData>
      <domain:infData
        xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>domain.example</domain:name>
        <domain:roid>EXAMPLE1-REP</domain:roid>
        <domain:status s="ok"/>
        <domain:registrant>jd1234</domain:registrant>
        <domain:contact type="admin">sh8013</domain:contact>
        <domain:contact type="tech">sh8013</domain:contact>
        <domain:clID>ClientX</domain:clID>
        <domain:crID>ClientY</domain:crID>
        <domain:crDate>2012-04-03T22:00:00.0Z</domain:crDate>
        <domain:upID>ClientZ</domain:upID>
        <domain:upDate>2013-10-22T14:25:57.0Z</domain:upDate>
        <domain:exDate>2014-04-03T22:00:00.0Z</domain:exDate>
      </domain:infData>
    </resData>
    <extension>
      <changePoll:changeData
        xmlns:changePoll="urn:ietf:params:xml:ns:changePoll-1.0">
        <changePoll:operation op="sync">custom</changePoll:operation>
        <changePoll:date>2013-10-22T14:25:57.0Z</changePoll:date>
        <changePoll:svTRID>12345-XYZ</changePoll:svTRID>
        <changePoll:who>CSR</changePoll:who>
        <changePoll:reason lang="en">Customer sync request</changePoll:reason>
      </changePoll:changeData>
    </extension>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>54321-XYZ</svTRID>
    </trID>
  </response>
</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 "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">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 domain resulting in immediate purge.</msg>
S:    </msgQ>
S:    <resData>
S:      <domain:infData 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 xmlns:changePoll="urn:ietf:params:xml:ns:changePoll-1.0" state="before">
S:        <changePoll:operation op="purge">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 "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 purged domain with pendingDelete status.</msg>
S:     </msgQ>
S:     <resData>
S:       <domain:infData 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 xmlns:changePoll="urn:ietf:params:xml:ns:changePoll-1.0" state="before">
S:         <changePoll:operation>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>Past pendingDelete 5 day period</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:    </response>
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: </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
  </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>
<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>

<simpleType name="stateType">
  <restriction base="token">
    <enumeration value="before"/>
    <enumeration value="after"/>
  </restriction>
</simpleType>

<complexType name="operationType">
  <simpleContent>
    <extension base="changePoll:operationEnum">
      <attribute name="op" type="token"/>
    </extension>
  </simpleContent>
</complexType>

<complexType name="caseIdType">
  <simpleContent>
    <extension base="token">
      <attribute name="type" type="changePoll:caseTypeEnum" use="required"/>
      <attribute name="name" type="token"/>
    </extension>
  </simpleContent>
</complexType>
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.
-->
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 7942 [RFC7942] 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 7942 [RFC7942]. 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 7942 [RFC7942], "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


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

6.4. Neustar EPP SDK

Organisation: Neustar Inc.

Name: Neustar EPP SDK

Description: The Neustar EPP SDK includes a full client implementation of draft-ietf-regext-change-poll.

Level of maturity: Production

Coverage: All client side aspects of the protocol are implemented.

Licensing: GNU Lesser General Public License

Contact: quoc-anh.np@team.neustar

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 Scott Hollenbeck, Michael Holloway, and Patrick Mevzek.

9. References

9.1. Normative References


9.2. Informative References


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.

A.10. Change from REGEXT 03 to REGEXT 04

1. Added Neustar implementation to the Implementation Status section.

A.11. Change from REGEXT 04 to REGEXT 05

1. Updates based on feedback from Patrick Mevzek, that include:
   1. Added a missing comma to "Using this extension, clients" in the Introduction section.
   2. Modified the description of the "transfer", "restore", and "custom" operations to include "MUST set the "op" attribute" language.
   3. Rephrased the first sentence of the Who section.
   4. Added references to the <changePoll:who> element in the Who section.
   5. Revise the sentence that describes how the extension extends the info response in the EPP <info> Command section.
   6. Refer to EPP Object Mapping as EPP object mapping throughout the document.
   7. Add a Dates and Times section to the Object Attributes section.

A.12. Change from REGEXT 05 to REGEXT 06

1. Added the "State" sub-section to the "Object Attributes" section to describe the expected behavior for the "before" and "after" states, based on feedback from Patrick Mevzek.
2. Added a colon suffix to each hangText entry to provide better separation.

A.13. Change from REGEXT 06 to REGEXT 07

1. Updates based on feedback from Scott Hollenbeck, that include:
   1. Changed MAY to may in the Abstract.
   2. Revised the "IANA Considerations" section to include the registration of the XML schema.
3. Revised the description of the `<changePoll:caseId>` "name" attribute and the "<changePoll:operation>" op" attribute as containing 7-bit US-ASCII identifiers for the case type or the operation type, respectively.

A.14. Change from REGEXT 07 to REGEXT 08

1. Updated obsoleted RFC 6982 to RFC 7942.
2. Moved RFC 7451 to an informational reference based on a check done by the Idnits Tool.
3. Changed Kal Feher’s contact e-mail address.
4. Changed Neustar’s Implementation Status contact e-mail address.

A.15. Change from REGEXT 08 to REGEXT 09

1. Fixed Section 1.1 (Conventions) to contain the updated language (e.g. "NOT RECOMMENDED", RFC 8174, BCP 14), based on feedback from the Document Shepherd.

A.16. Change from REGEXT 09 to REGEXT 10

1. Updates based on the AD review by Adam Roach, that include:
   1. Fix the "purge" and "autoPurge" examples to use the normative "before" state instead of the default "after" state.
   2. Added the sentences "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." in the "Introduction" to clarify what is extended and reference [RFC5730] for the EPP <poll> command.
   3. Added missing hyphens to "client-sponsored" and "court-directed".
   4. Removed "changePoll-1.0" is used as an abbreviation for "urn:ietf:params:xml:ns:changePoll-1.0" and replaced the paragraph based on what was done in draft-ietf-regext-allocation-token.
   5. Changed normative "SHOULD" to non-normative "should" in "An operation consists of any transform operation that impacts objects that the client sponsers and should be notified of."
   6. Added normative reference to [RFC0020] to define "7-bit US-ASCII".
   7. Added the sentence "The custom operations supported is up to server policy." to the description of the "custom" operation.
8. Broke up the "This extension adds operation detail..." sentence into two separate sentences to address the "does" and the "is" separately.
9. Removed the commas from "Any transform operation to an object..." sentence.
10. Changed to use an IPv6 address from the documentation-only prefix "2001:DB8::/32" in RFC 3849. The IPv6 address 2001:db8:0:0:1:0:0:1 was used.

A.17. Change from REGEXT 10 to REGEXT 11

1. Updates based on the review by Benjamin Kaduk, that include:
   1. Change references of "The enumerated list ... include:" to "The enumerated list ... is:"
   2. In section 2.2, explicitly state what the message is inserted into, with the change of "... MUST be inserted prior to ..." to "... MUST be inserted into the message queue prior to ...

A.18. Change from REGEXT 11 to REGEXT 12

1. Added clarification for the <changePoll:who> element based on the feedback from Benjamin Kaduk.

Authors’ Addresses

James Gould
VeriSign, Inc.
12061 Bluemont Way
Reston, VA 20190
US
Email: jgould@verisign.com
URI: http://www.verisign.com

Kal Feher
Neustar
1vl 8/10 Queens Road
Melbourne, VIC 3004
AU
Email: ietf@feherfamily.org
URI: http://www.neustar.biz