

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: June 8, 2019

L. Zhou
CNNIC
N. Kong
Consultant
J. Wei
J. Yao
CNNIC
J. Gould
Verisign, Inc.
December 5, 2018

Organization Extension for the Extensible Provisioning Protocol (EPP)
draft-ietf-regext-org-ext-11

Abstract

This document describes an extension to Extensible Provisioning Protocol (EPP) object mappings, which is designed to support assigning an organization to any existing object (domain, host, contact) as well as any future objects.

Status of This Memo

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

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

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

This Internet-Draft will expire on June 8, 2019.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include 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	2
2. Conventions Used in This Document	3
3. Object Attributes	4
3.1. Organization Identifier	4
4. EPP Command Mapping	4
4.1. EPP Query Commands	4
4.1.1. EPP <check> Command	4
4.1.2. EPP <info> Command	4
4.1.3. EPP <transfer> Query Command	7
4.2. EPP Transform Commands	8
4.2.1. EPP <create> Command	8
4.2.2. EPP <delete> Command	10
4.2.3. EPP <renew> Command	10
4.2.4. EPP <transfer> Command	11
4.2.5. EPP <update> Command	11
5. Formal Syntax	15
6. Internationalization Considerations	18
7. IANA Considerations	18
7.1. XML Namespace	18
7.2. EPP Extension Registry	18
8. Implementation Status	19
8.1. Verisign EPP SDK	19
8.2. CNNIC Implementation	20
9. Security Considerations	20
10. Acknowledgment	20
11. References	20
11.1. Normative References	20
11.2. Informative References	22
Appendix A. Change Log	22
Authors' Addresses	25

1. Introduction

In the business model of domain registration, we usually have three roles of entities: a registrant, a registrar and a registry, as defined in section 9 of [ID.draft-ietf-dnsop-terminology-bis]. There may be other roles of entities involved in the domain registration process, such as resellers, DNS operators in section 9 of [ID.draft-ietf-dnsop-terminology-bis], privacy proxies, etc.

A domain reseller is an individual or a company that acts as an agent for accredited registrars. DNS operator is defined in section 9 of [ID.draft-ietf-dnsop-terminology-bis]. A privacy proxy is an entity used for domain registrations to protect the private information of the individuals and organizations. These kind of entities are defined as "organizations" with different role types in this document.

In order to facilitate provisioning and management of organization information in a shared central repository, this document proposes an organization extension mapping for any Extensible Provisioning Protocol (EPP) object like domain names in [RFC5731], hosts in [RFC5732] and contacts in [RFC5733]. The examples provided in this document are used for the domain object for illustration purpose. The host and contact object could be extended in the same way with the domain object.

Organization object identifiers defined in [ID.draft-ietf-regext-org] MUST be known to the server before the organization object can be associated with the EPP object.

This document is specified using the XML 1.0 as described in [W3C.REC-xml-20040204] and XML Schema notation as described in [W3C.REC-xmlschema-1-20041028] and [W3C.REC-xmlschema-2-20041028].

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

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

XML is case sensitive. Unless stated otherwise, XML specifications and examples provided in this document MUST be interpreted in the character case.

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

3. Object Attributes

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

3.1. Organization Identifier

Organization identifier provides the ID of an organization. Its corresponding element is `<orgext:id>` which refers to the `<org:id>` element defined in [ID.draft-ietf-regext-org]. All organization objects are identified by a server-unique identifier. A "role" attribute is used to represent the relationship that the organization has to the EPP object. Any given object MUST have at most one associated organization ID for any given role value.

4. EPP Command Mapping

A detailed description of the EPP syntax and semantics can be found in the EPP core protocol specification [RFC5730]. The command mappings described here are specifically for assigning organizations to EPP objects.

4.1. EPP Query Commands

EPP provides three commands to retrieve EPP object information: `<check>` to determine if an object can be provisioned within a repository, `<info>` to retrieve detailed information associated with an object, and `<transfer>` to retrieve object transfer status information.

4.1.1. EPP `<check>` Command

This extension does not add any elements to the EPP `<check>` command or `<check>` response described in the EPP object mapping.

4.1.2. EPP `<info>` Command

This extension does not add any elements to the EPP `<info>` command described in the EPP object mapping. However, additional elements are defined for the `<info>` response in the EPP object mapping.

When an `<info>` command has been processed successfully, the EPP `<resData>` element MUST contain child elements as described in the EPP object extensions. In addition, the EPP `<extension>` element SHOULD contain a child `<orgext:infData>` element. This element or its ancestor element MUST identify the extension namespace "urn:ietf:params:xml:ns:epp:orgext-1.0" if the object has data

associated with this extension and based on server policy. The <orgext:infData> element contains the following child elements:

- o Zero or more <orgext:id> elements are allowed that contain the identifier of the organization, as defined in Section 3.1. The "role" attribute is used to represent the relationship that the organization has to the object. See Section 7.3 in [ID.draft-ietf-regext-org] for a list of values.

Example <info> response for an authorized client with multiple organizations:

```
S:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S:  <response>
S:    <result code="1000">
S:      <msg lang="en-US">Command completed successfully</msg>
S:    </result>
S:    <resData>
S:      <domain:infData
S:        xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
S:        <domain:name>example.com</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="billing">sh8013</domain:contact>
S:        <domain:contact type="tech">sh8013</domain:contact>
S:        <domain:ns>
S:          <domain:hostObj>ns1.example.com</domain:hostObj>
S:        </domain:ns>
S:        <domain:clID>ClientX</domain:clID>
S:        <domain:crID>ClientY</domain:crID>
S:        <domain:crDate>2015-02-06T04:01:21.0Z</domain:crDate>
S:        <domain:exDate>2018-02-06T04:01:21.0Z</domain:exDate>
S:        <domain:authInfo>
S:          <domain:pw>2fooBAR</domain:pw>
S:        </domain:authInfo>
S:      </domain:infData>
S:    </resData>
S:    <extension>
S:      <orgext:infData
S:        xmlns:orgext="urn:ietf:params:xml:ns:epp:orgext-1.0">
S:        <orgext:id role="reseller">reseller1523</orgext:id>
S:        <orgext:id role="privacyproxy">proxy2935</orgext:id>
S:      </orgext:infData>
S:    </extension>
S:    <trID>
S:      <clTRID>ngcl-IvJjzMZc</clTRID>
S:      <svTRID>test142AWQONJZ</svTRID>
S:    </trID>
S:  </response>
S:</epp>
```

Example <info> response for an authorized client with no organization:

```
S:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S:  <response>
S:    <result code="1000">
S:      <msg lang="en-US">Command completed successfully</msg>
S:    </result>
S:    <resData>
S:      <domain:infData
S:        xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
S:        <domain:name>example.com</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="billing">sh8013</domain:contact>
S:        <domain:contact type="tech">sh8013</domain:contact>
S:        <domain:ns>
S:          <domain:hostObj>ns1.example.com</domain:hostObj>
S:        </domain:ns>
S:        <domain:clID>ClientX</domain:clID>
S:        <domain:crID>ClientY</domain:crID>
S:        <domain:crDate>2015-02-06T04:01:21.0Z</domain:crDate>
S:        <domain:exDate>2018-02-06T04:01:21.0Z</domain:exDate>
S:        <domain:authInfo>
S:          <domain:pw>2fooBAR</domain:pw>
S:        </domain:authInfo>
S:      </domain:infData>
S:    </resData>
S:    <extension>
S:      <orgext:infData
S:        xmlns:orgext="urn:ietf:params:xml:ns:epp:orgext-1.0"/>
S:    </extension>
S:    <trID>
S:      <clTRID>ngcl-IvJjzMZc</clTRID>
S:      <svTRID>test142AWQONJZ</svTRID>
S:    </trID>
S:  </response>
S:</epp>
```

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

4.1.3. EPP <transfer> Query Command

This extension does not add any elements to the EPP <transfer> query command or <transfer> query response described in the EPP object mapping.

4.2. EPP Transform Commands

EPP provides five commands to transform EPP 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 the object sponsorship changes, and <update> to change information associated with an object.

4.2.1. EPP <create> Command

This extension defines additional elements for the EPP <create> command described in the EPP object extensions. No additional elements are defined for the EPP <create> response.

The EPP <create> command provides a transform operation that allows a client to create an object. In addition to the EPP command elements described in the EPP object extensions, the command MUST contain an <extension> element, and the <extension> element MUST contain a child <orgext:create> element. This element or its ancestor element MUST identify the extension namespace "urn:ietf:params:xml:ns:epp:orgext-1.0" if the client wants to associate data defined in this extension to the object. The <orgext:create> element contains the following child elements:

- o One or more <orgext:id> elements that contain the identifier of the organization, as defined in Section 3.1. The "role" attribute is used to represent the relationship that the organization has to the object. See Section 7.3 in [ID.draft-ietf-regext-org] for a list of values.

Example <create> Command with only one organization:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <create>
C:      <domain:create
C:        xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
C:          <domain:name>example.com</domain:name>
C:          <domain:period unit="y">3</domain:period>
C:          <domain:ns>
C:            <domain:hostObj>ns1.example.com</domain:hostObj>
C:          </domain:ns>
C:          <domain:registrant>jd1234</domain:registrant>
C:          <domain:contact type="tech">sh8013</domain:contact>
C:          <domain:contact type="billing">sh8013</domain:contact>
C:          <domain:contact type="admin">sh8013</domain:contact>
C:          <domain:authInfo>
C:            <domain:pw>fooBAR</domain:pw>
C:          </domain:authInfo>
C:        </domain:create>
C:      </create>
C:    <extension>
C:      <orgext:create
C:        xmlns:orgext="urn:ietf:params:xml:ns:epp:orgext-1.0">
C:          <orgext:id role="reseller">reseller1523</orgext:id>
C:        </orgext:create>
C:      </extension>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

Example <create> Command with multiple organizations:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <create>
C:      <domain:create>
C:        xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
C:          <domain:name>example.com</domain:name>
C:          <domain:period unit="y">3</domain:period>
C:          <domain:ns>
C:            <domain:hostObj>ns1.example.com</domain:hostObj>
C:          </domain:ns>
C:          <domain:registrant>jd1234</domain:registrant>
C:          <domain:contact type="tech">sh8013</domain:contact>
C:          <domain:contact type="billing">sh8013</domain:contact>
C:          <domain:contact type="admin">sh8013</domain:contact>
C:          <domain:authInfo>
C:            <domain:pw>fooBAR</domain:pw>
C:          </domain:authInfo>
C:        </domain:create>
C:      </create>
C:    <extension>
C:      <orgext:create>
C:        xmlns:orgext="urn:ietf:params:xml:ns:epp:orgext-1.0">
C:          <orgext:id role="reseller">reseller1523</orgext:id>
C:          <orgext:id role="privacyproxy">proxy2935</orgext:id>
C:        </orgext:create>
C:      </extension>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

When a <create> command has been processed successfully, the EPP response is as described in the EPP object extension.

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

4.2.2. EPP <delete> Command

This extension does not add any elements to the EPP <delete> command or <delete> response described in the EPP object mapping.

4.2.3. EPP <renew> Command

This extension does not add any elements to the EPP <renew> command or <renew> response described in the EPP object mapping.

4.2.4. EPP <transfer> Command

This extension does not add any elements to the EPP <transfer> command or <transfer> response described in the EPP object mapping, but after a successful transfer of an object with an assigned organization, the handling of the assigned organization is dependent on the organization roles and server policy.

4.2.5. EPP <update> Command

This extension defines additional elements for the EPP <update> command described in the EPP domain mapping [RFC5731], host mapping [RFC5732] and contact mapping [RFC5733]. No additional elements are defined for the EPP <update> response.

The EPP <update> command provides a transform operation that allows a client to modify the attributes of an object. In addition to the EPP <update> command elements, the command MUST contain an <extension> element, and the <extension> element MUST contain a child <orgext:update> element. This element or its ancestor element MUST identify the extension namespace "urn:ietf:params:xml:ns:epp:orgext-1.0" if the client wants to update the object with data defined in this extension. The <orgext:update> element contains the following child elements:

- o An OPTIONAL <orgext:add> element that contains one or more <orgext:id> elements, as defined in Section 3.1, that add non-existent organization roles to the object. The <orgext:id> element MUST have a non-empty organization identifier value. The server SHOULD validate that the <orgext:id> element role does not exist.
- o An OPTIONAL <orgext:rem> element that contains one or more <orgext:id> elements, as defined in Section 3.1, that remove organization roles from the object. The <orgext:id> element MAY have an empty organization identifier value. The server SHOULD validate the existence of the <orgext:id> element role and the organization identifier if provided.
- o An OPTIONAL <orgext:chg> element that contains one or more <orgext:id> elements, as defined in Section 3.1, that change organization role identifiers for the object. The existing organization identifier value will be replaced for the defined role. The server SHOULD validate the existence of the <orgext:id> element role.

At least one <orgext:add>, <orgext:rem> or <orgext:chg> element MUST be provided.

Example <update> command, adding a reseller:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <update>
C:      <domain:update>
C:        xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
C:        <domain:name>example.com</domain:name>
C:      </domain:update>
C:    </update>
C:  <extension>
C:    <orgext:update>
C:      xmlns:orgext="urn:ietf:params:xml:ns:epp:orgext-1.0">
C:      <orgext:add>
C:        <orgext:id role="reseller">reseller1523</orgext:id>
C:      </orgext:add>
C:    </orgext:update>
C:  </extension>
C:  <clTRID>ABC-12345</clTRID>
C: </command>
C:</epp>
```

Example <update> command, adding multiple organizations:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <update>
C:      <domain:update>
C:        xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
C:        <domain:name>example.com</domain:name>
C:      </domain:update>
C:    </update>
C:  <extension>
C:    <orgext:update>
C:      xmlns:orgext="urn:ietf:params:xml:ns:epp:orgext-1.0">
C:      <orgext:add>
C:        <orgext:id role="reseller">reseller1523</orgext:id>
C:        <orgext:id role="privacyproxy">proxy2935</orgext:id>
C:      </orgext:add>
C:    </orgext:update>
C:  </extension>
C:  <clTRID>ABC-12345</clTRID>
C: </command>
C:</epp>
```

Example <update> command, removing a reseller:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <update>
C:      <domain:update>
C:        xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
C:          <domain:name>example.com</domain:name>
C:        </domain:update>
C:      </update>
C:    <extension>
C:      <orgext:update>
C:        xmlns:orgext="urn:ietf:params:xml:ns:epp:orgext-1.0">
C:          <orgext:rem>
C:            <orgext:id role="reseller"/>
C:          </orgext:rem>
C:        </orgext:update>
C:      </extension>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

Example <update> command, removing multiple organizations:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <update>
C:      <domain:update>
C:        xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
C:          <domain:name>example.com</domain:name>
C:        </domain:update>
C:      </update>
C:    <extension>
C:      <orgext:update>
C:        xmlns:orgext="urn:ietf:params:xml:ns:epp:orgext-1.0">
C:          <orgext:rem>
C:            <orgext:id role="reseller"/>
C:            <orgext:id role="privacyproxy"/>
C:          </orgext:rem>
C:        </orgext:update>
C:      </extension>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

Example <update> command, updating reseller identifier:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <update>
C:      <domain:update>
C:        xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
C:        <domain:name>example.com</domain:name>
C:      </domain:update>
C:    </update>
C:  <extension>
C:    <orgext:update>
C:      xmlns:orgext="urn:ietf:params:xml:ns:epp:orgext-1.0">
C:      <orgext:chg>
C:        <orgext:id role="reseller">reseller1523</orgext:id>
C:      </orgext:chg>
C:    </orgext:update>
C:  </extension>
C:  <clTRID>ABC-12345</clTRID>
C: </command>
C:</epp>
```

Example <update> command, updating multiple organization identifiers:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <update>
C:      <domain:update>
C:        xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
C:        <domain:name>example.com</domain:name>
C:      </domain:update>
C:    </update>
C:  <extension>
C:    <orgext:update>
C:      xmlns:orgext="urn:ietf:params:xml:ns:epp:orgext-1.0">
C:      <orgext:chg>
C:        <orgext:id role="reseller">reseller1523</orgext:id>
C:        <orgext:id role="privacyproxy">proxy2935</orgext:id>
C:      </orgext:chg>
C:    </orgext:update>
C:  </extension>
C:  <clTRID>ABC-12345</clTRID>
C: </command>
C:</epp>
```

When an extended <update> command has been processed successfully, the EPP response is as described in the EPP object extension.

An EPP error response MUST be returned if an <update> command cannot be processed for any reason. An attempt to add one organization ID or multiple organization IDs with a particular role value when at least one of them already exists does not change the object at all. A server SHOULD notify clients that object relationships exist by sending a 2305 error response code. An attempt to remove an organization ID or multiple organization IDs with a particular role value when at least one of them does not exist does not change the object at all. A server SHOULD notify clients that object relationships does not exist by sending a 2305 error response code. An attempt to change an organization ID or multiple organization IDs with a particular role value when at least one of them does not exist does not change the object at all. A server SHOULD notify clients that object relationships does not exist by sending a 2305 error response code. Response format with error value elements is defined in Section 2.6 of [RFC5730].

5. Formal Syntax

An EPP object mapping is specified in XML Schema notation. The formal syntax presented here is a complete schema representation of the object mapping suitable for automated validation of EPP XML instances. The BEGIN and END tags are not part of the schema; they are used to note the beginning and ending of the schema for URI registration purposes.

```
BEGIN
<?xml version="1.0" encoding="UTF-8"?>

<schema
  targetNamespace="urn:ietf:params:xml:ns:epp:orgext-1.0"
  xmlns:orgext="urn:ietf:params:xml:ns:epp:orgext-1.0"
  xmlns="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified"
>

  <annotation>
    <documentation>
      Extensible Provisioning Protocol v1.0
      Organization Extension Schema v1.0
    </documentation>
  </annotation>

  <!-- Child elements found in EPP commands. -->
```

```
<element
  name="create"
  type="orgext:createType"/>
<element
  name="update"
  type="orgext:updateType"/>

<!--
  Organization identifier with required role
-->
<complexType name="orgIdType">
  <simpleContent>
    <extension base="token">
      <attribute
        name="role"
        type="token"
        use="required"/>
    </extension>
  </simpleContent>
</complexType>

<!--
  Child elements of the <orgext:create> command
  All elements must be present at time of creation
-->
<complexType name="createType">
  <sequence>
    <!-- agent identifier or the organization,
         e.g. registrar, reseller, privacy proxy, etc. -->
    <element
      name="id"
      type="orgext:orgIdType"
      maxOccurs="unbounded"/>
  </sequence>
</complexType>

<!--
  Child elements of <orgext:update> command
-->
<complexType name="updateType">
  <sequence>
    <element
      name="add"
      type="orgext:addRemChgType"
      minOccurs="0"
    />
    <element
      name="rem"
```

```
        type="orgext:addRemChgType"
        minOccurs="0"
    />
    <element
        name="chg"
        type="orgext:addRemChgType"
        minOccurs="0"
    />
</sequence>
</complexType>

<complexType name="addRemChgType">
    <sequence>
        <!-- agent identifier of the organization,
             e.g. registrar, reseller, privacy proxy, etc. -->
        <element
            name="id"
            type="orgext:orgIdType"
            maxOccurs="unbounded"/>
    </sequence>
</complexType>

<!-- Child response element -->
<element
    name="infData"
    type="orgext:infDataType"/>

<!-- <orgext:infData> response elements -->
<complexType name="infDataType">
    <sequence>
        <!-- agent identifier the organization,
             e.g. registrar, reseller, privacy proxy, etc. -->
        <element
            name="id"
            type="orgext:orgIdType"
            minOccurs="0"
            maxOccurs="unbounded"/>
    </sequence>
</complexType>

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

6. Internationalization Considerations

EPP is represented in XML, which provides native support for encoding information using the Unicode character set and its more compact representations including UTF-8. Conformant XML processors recognize both UTF-8 and UTF-16. Though XML includes provisions to identify and use other character encodings through use of an "encoding" attribute in an `<?xml?>` declaration, use of UTF-8 is RECOMMENDED.

As an extension of the EPP object mapping, the elements, element content described in this document MUST inherit the internationalization conventions used to represent higher-layer domain and core protocol structures present in an XML instance that includes this extension.

7. IANA Considerations

7.1. XML Namespace

This document uses URNs to describe XML namespaces and XML schemas conforming to a registry mechanism described in [RFC3688]. IANA is requested to assign the following URI.

Registration request for the organization extension namespace:

URI: urn:ietf:params:xml:ns:epp:orgext-1.0

Registrant Contact: IESG

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

Registration request for the organization XML schema:

URI: urn:ietf:params:xml:schema:epp:orgext-1.0

Registrant Contact: IESG

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

7.2. 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: Organization Extension for the Extensible Provisioning Protocol (EPP)

Document status: Standards Track

Reference: RFCXXXX (please replace "XXXX" with the RFC number for this document after a number is assigned by the RFC Editor)

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

TLDs: Any

IPR Disclosure: None

Status: Active

Notes: None

8. Implementation Status

Note to RFC Editor: Please remove this section and the reference to [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 [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 [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".

8.1. 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-org-ext.

Level of maturity: Development

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

8.2. CNNIC Implementation

Organization: CNNIC

Name: Organization Extension for EPP

Description: CNNIC is trying to update organization extension from previous reseller extension according to this document.

Level of maturity: Development

Coverage: Organization extension for EPP

Contact: zhouguiqing@cnnic.cn

9. Security Considerations

The object mapping extension described in this document does not provide any other security services or introduce any additional considerations beyond those described by [RFC5730], [RFC5731], [RFC5732] and [RFC5733] or those caused by the protocol layers used by EPP.

10. Acknowledgment

The authors would like to thank Rik Ribbers, Marc Groeneweg, Patrick Mevzek, Antoin Verschuren and Scott Hollenbeck for their careful review and valuable comments.

11. References

11.1. 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, <<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>>.
- [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>>.
- [RFC5732] Hollenbeck, S., "Extensible Provisioning Protocol (EPP) Host Mapping", STD 69, RFC 5732, DOI 10.17487/RFC5732, August 2009, <<https://www.rfc-editor.org/info/rfc5732>>.
- [RFC5733] Hollenbeck, S., "Extensible Provisioning Protocol (EPP) Contact Mapping", STD 69, RFC 5733, DOI 10.17487/RFC5733, August 2009, <<https://www.rfc-editor.org/info/rfc5733>>.
- [RFC7942] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", BCP 205, RFC 7942, DOI 10.17487/RFC7942, July 2016, <<https://www.rfc-editor.org/info/rfc7942>>.
- [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-xml-20040204]
Bray, T., Paoli, J., Sperberg-McQueen, C., Maler, E., and F. Yergeau, "Extensible Markup Language (XML) 1.0 (Third Edition)", World Wide Web Consortium FirstEdition REC-xml-20040204, February 2004, <<http://www.w3.org/TR/2004/REC-xml-20040204>>.
- [W3C.REC-xmlschema-1-20041028]
Thompson, H., Beech, D., Maloney, M., and N. Mendelsohn, "XML Schema Part 1: Structures Second Edition", World Wide Web Consortium Recommendation REC-xmlschema-1-20041028, October 2004, <<http://www.w3.org/TR/2004/REC-xmlschema-1-20041028>>.

[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>>.

11.2. Informative References

[ID.draft-ietf-dnsop-terminology-bis]

Hoffman, P., Sullivan, A., and K. Fujiwara, "DNS Terminology", September 2018, <<http://tools.ietf.org/html/draft-ietf-dnsop-terminology-bis>>.

[ID.draft-ietf-regex-org]

Zhou, L., Kong, N., Zhou, G., Yao, J., and J. Gould, "Extensible Provisioning Protocol (EPP) Reseller Mapping", November 2018, <<http://tools.ietf.org/html/draft-ietf-regex-org>>.

[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>>.

Appendix A. Change Log

Initial -00: Individual document submitted.

-01:

- * Updated abstract and introduction.
- * Revised typos in info response.
- * Added explanations on how to process reseller extension after successful transfer operation.
- * Modified <update> explanation.
- * Deleted reseller name element in <create> and <update> commands.
- * Removed some inaccurate comments from xml schema.
- * Modified the element name of reseller id and reseller name.

-02:

- * Changed author information.

- * Updated xml typos <reseller:infData> to <resellerext:infData> in <info> response.

-03:

- * Changed author information.
- * Updated section 3.1.
- * Removed reseller name element in <info> response.
- * Added acknowledgment.
- * Revised the typo "resellerr" to "resellerext".

WG document-00: WG document submitted

WG document-01: Keep document alive for further discussion. The requirement of reseller information is clear for both registrar and registry. What we should reach a consensus is whether the extension should support only a name or ID and name.

Organization WG document-00: Change to a generic organization object extension.

Organization WG document-01: Added "Implementation Status" section.

Organization WG document-02: Accepted some of the feedbacks on the mailing list. Modified the examples in the document.

Organization WG document-03:

- * Updated typos.
- * Changed some descriptions about <orgext:id> and role attribute.
- * Modified the example of "domain with no organization".
- * Updated section 8, adding implementation status of Verisign.

Organization WG document-04:

- * Updated typos.
- * Removed the example of <update> command, domain with no organization.
- * Updated references.

- * Updated section 8 of implementation status.

Organization WG document-05:

- * Removed the minOccurs="0" from the addRemChgType type of the XML schema
- * Removed the third paragraph of "Implementation Status".
- * Remove the Informative Reference to draft-ietf-regext-reseller-ext from the draft.

Organization WG document-06:

- * Updated "Abstraction".
- * Added "Query" for "<Transfer> Query Command".
- * Change "Registrant Contact" to IESG in section 7.1.
- * Modified section 7.2.

Organization WG document-07:

- * Updated "Abstraction".

Organization WG document-08:

- * Updated error codes of <update> response.
- * Modified XML namespace and schema.

Organization WG document-09:

- * Modified XML namespace and schema.
- * Changed "Exactly one" to "At least one" in section 4.2.5.

Organization WG document-10:

- * Updated the reseller id and dns proxy id in the document.
- * Updated the full xml namespace.
- * Updated the text of EPP <orgext:add>, <orgext:rem> and <orgext:chg>.

- * Added "Document satus" and "Reference" in section "EPP Extension Registry".

Organization WG document-11:

- * Added the reference of draft-ietf-dnsop-terminology-bis.

Authors' Addresses

Linlin Zhou
CNNIC
4 South 4th Street, Zhongguancun, Haidian District
Beijing, Beijing 100190
China

Email: zhoulinlin@cnnic.cn

Ning Kong
Consultant

Email: ietfing@gmail.com

Junkai Wei
CNNIC
4 South 4th Street, Zhongguancun, Haidian District
Beijing, Beijing 100190
China

Email: weijunkai@cnnic.cn

Jiankang Yao
CNNIC
4 South 4th Street, Zhongguancun, Haidian District
Beijing, Beijing 100190
China

Email: yaojk@cnnic.cn

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

Email: jgould@verisign.com