

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: September 1, 2018

L. Zhou
N. Kong
G. Zhou
X. Lee
CNNIC
J. Gould
VeriSign, Inc.
February 28, 2018

Extensible Provisioning Protocol (EPP) Organization Mapping draft-ietf-regext-org-02

Abstract

This document describes an Extensible Provisioning Protocol (EPP) mapping for provisioning and management of organization objects stored in a shared central repository. Specified in Extensible Markup Language (XML), this extended mapping is applied to provide additional features required for the provisioning of organizations.

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 September 1, 2018.

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.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1.	Introduction	3
2.	Conventions Used in This Document	3
3.	Object Attributes	4
3.1.	Organization Identifier	4
3.2.	Organization Roles	4
3.2.1.	Role Type	4
3.2.2.	Role Identifier	4
3.2.3.	Example of Organization Roles	4
3.3.	Contact and Client Identifiers	5
3.4.	Organization Status Values	5
3.5.	Role Status Values	6
3.6.	Parent Identifier	6
3.7.	URL	7
3.8.		7
4.	EPP Command Mapping	7
4.1.	EPP Query Commands	7
4.1.1.	EPP <check> Command	7
4.1.2.	EPP <info> Command	9
4.1.3.	EPP <transfer> Command	15
4.2.	EPP Transform Commands	15
4.2.1.	EPP <create> Command	15
4.2.2.	EPP <delete> Command	19
4.2.3.	EPP <renew> Command	20
4.2.4.	EPP <transfer> Command	20
4.2.5.	EPP <update> Command	21
5.	Formal Syntax	24
6.	Internationalization Considerations	30
7.	IANA Considerations	30
7.1.	XML Namespace	30

7.2.	EPP Extension Registry	31
7.3.	Role Values Registry	31
8.	Implementation Status	32
8.1.	CNNIC Implementation	32
8.2.	Reseller Extension	33
9.	Security Considerations	33
10.	Acknowledgment	33
11.	Normative References	33
Appendix A.	Change Log	34
	Authors' Addresses	35

[1.](#) Introduction

There are many entities, such as registrars, resellers, DNS service operators, or privacy proxies involved in the domain registration business. These kind of entities have not been formally defined as an object in EPP which will be specified as "organization" in this document.

This document describes an organization object mapping for version 1.0 of the Extensible Provisioning Protocol (EPP) [[RFC5730](#)]. This mapping 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 [[RFC2119](#)].

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 presented to develop a conforming implementation.

"org-1.0" in is used as an abbreviation for "urn:ietf:params:xml:ns:org-1.0". The XML namespace prefix "org" 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.

3. Object Attributes

An EPP organization object has attributes and associated values that can be viewed and modified by the sponsoring client or the server. This section describes each attribute type in detail. The formal syntax for the attribute values described here can be found in the "Formal Syntax" section of this document and in the appropriate normative references.

3.1. Organization Identifier

All EPP organizations are identified by a server-unique identifier. Organization identifiers are character strings with a specific minimum length, a specified maximum length, and a specified format. Organization identifiers use the "clIDType" client identifier syntax described in [[RFC5730](#)]. Its corresponding element is <org:id>.

3.2. Organization Roles

The organization roles are used to represent the relationship an organization would have. Its corresponding element is <org:role>.

3.2.1. Role Type

An organization would support a list of roles. See [Section 7.3](#) for a list of values. Its corresponding element is <org:type> with a "roleStatus" attribute. The values of role statuses are defined in [Section 3.5](#).

3.2.2. Role Identifier

A role MAY have a third party assigned identifier such as the IANA ID for registrars. Its corresponding element is <org:roleid>.

3.2.3. Example of Organization Roles

Example of "Organization Roles"

```
S: <org:role>
S: <org:type roleStatus="ok">registrar</org:type>
S: <org:roleid>1362</org:roleid>
S: </org:role>
```


3.3. Contact and Client Identifiers

All EPP contacts are identified by a server-unique identifier. Contact identifiers are character strings with a specific minimum length, a specified maximum length, and a specified format. Contact identifiers use the "clIDType" client identifier syntax described in [\[RFC5730\]](#).

3.4. Organization Status Values

An organization object **MUST** always have at least one associated status value. The default value is "ok".

Status Value Descriptions:

- o ok: This is the normal status value for an object that has no pending operations or prohibitions. This value is set and removed by the server as other status values are added or removed.
- o hold: Organization transform commands and new links **MUST** be rejected.
- o terminated: The organization which has been terminated **MUST NOT** be linked. Organization transform commands and new links **MUST** be rejected.
- o linked: The organization object has at least one active association with another object. The "linked" status is not explicitly set by the client. Servers **SHOULD** provide services to determine existing object associations.
- o clientLinkProhibited, serverLinkProhibited: Requests to add new links to the organization **MUST** be rejected.
- o clientUpdateProhibited, serverUpdateProhibited: Requests to update the object (other than to remove this status) **MUST** be rejected.
- o clientDeleteProhibited, serverDeleteProhibited: Requests to delete the object **MUST** be rejected.
- o pendingCreate, pendingUpdate, pendingDelete: A transform command has been processed for the object, but the action has not been completed by the server. Server operators can delay action completion for a variety of reasons, such as to allow for human review or third-party action. A transform command that is processed, but whose requested action is pending, is noted with response code 1001.

"pendingCreate", "ok", "hold", and "terminated" are mutually exclusive statuses. Organization MUST have only one of these statuses set.

"ok" status MAY only be combined with "linked" status.

"pendingDelete" status MUST NOT be combined with either "clientDeleteProhibited" or "serverDeleteProhibited" status.

The pendingCreate, pendingDelete, and pendingUpdate status values MUST NOT be combined with each other.

3.5. Role Status Values

A role SHOULD have at least one associated status value. Valid values include "ok", "linked", "clientLinkProhibited", and "serverLinkProhibited". The default value is "ok".

Status Value Descriptions:

- o ok: This is the normal status value for an role that has no pending operations or prohibitions. This value is set and removed by the server as other status values are added or removed.
- o linked: The role of an organization object has at least one active association with another object. The "linked" status is not explicitly set by the client. Servers SHOULD provide services to determine existing object associations.
- o clientLinkProhibited, serverLinkProhibited: Requests to add new links to the role MUST be rejected.

3.6. Parent Identifier

There can be more than one layer of organizations, such as a reseller. The parent identifier, as defined with the <org:parentId> element, represents the parent organization identifier in a child organization.

Take a reseller organization for example, the parent identifier is not defined for the top level reseller, namely the registrar of the registry. An N-tier reseller has a parent reseller and at least one child reseller. A reseller customer has a parent reseller and no child resellers.

Loops SHOULD be prohibited. If organization A has B as parent identifier, organization B must not have organization A as parent identifier.

3.7. URL

The URL represents the organization web home page, as defined with the `<org:url>` element.

3.8.

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.

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 use in provisioning and managing organization information via EPP.

4.1. EPP Query Commands

EPP provides two commands to retrieve domain information: `<check>` to determine if an organization object can be provisioned within a repository, and `<info>` to retrieve detailed information associated with an organization object. This document does not define a mapping for the EPP `<transfer>` command to retrieve domain-object transfer status information..

4.1.1. EPP `<check>` Command

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

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

- o One or more `<org:id>` elements that contain the server-unique identifier of the organization objects to be queried.

Example `<check>` command:


```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <check>
C:      <org:check
C:        xmlns:org="urn:ietf:params:xml:ns:org-1.0">
C:          <org:id>res1523</org:id>
C:          <org:id>re1523</org:id>
C:          <org:id>1523res</org:id>
C:        </org:check>
C:      </check>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

When a `<check>` command has been processed successfully, the EPP `<resData>` element MUST contain a child `<org:chkData>` element that identifies the organization namespace. The `<org:chkData>` element contains one or more `<org:cd>` elements that contain the following child elements:

- o A `<org:id>` element that identifies the queried object. This element MUST contain an "avail" attribute whose value indicates object availability (can it be provisioned or not) at the moment the `<check>` command was completed. A value of "1" or "true" means that the object can be provisioned. A value of "0" or "false" means that the object cannot be provisioned.
- o An OPTIONAL `<org:reason>` element that MAY be provided when an object cannot be provisioned. If present, this element contains server-specific text to help explain why the object cannot be provisioned. 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 `<check>` response:


```
S:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S:  <response>
S:    <result code="1000">
S:      <msg>Command completed successfully</msg>
S:    </result>
S:    <resData>
S:      <org:chkData
S:        xmlns:org="urn:ietf:params:xml:ns:org-1.0">
S:        <org:cd>
S:          <org:id avail="1">res1523</org:id>
S:        </org:cd>
S:        <org:cd>
S:          <org:id avail="0">re1523</org:id>
S:          <org:reason>In use</org:reason>
S:        </org:cd>
S:        <org:cd>
S:          <org:id avail="1">1523res</org:id>
S:        </org:cd>
S:      </org:chkData>
S:    </resData>
S:    <trID>
S:      <clTRID>ABC-12345</clTRID>
S:      <svTRID>54322-XYZ</svTRID>
S:    </trID>
S:  </response>
S:</epp>
```

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

4.1.1.2. EPP <info> Command

The EPP <info> command is used to retrieve information associated with an organization object. In addition to the standard EPP command elements, the <info> command MUST contain a <org:info> element that identifies the organization namespace. The <org:info> element contains the following child elements:

- o A <org:id> element that contains the server-unique identifier of the organization object to be queried.

Example <info> command:


```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <info>
C:      <org:info
C:        xmlns:org="urn:ietf:params:xml:ns:org-1.0">
C:          <org:id>res1523</org:id>
C:        </org:info>
C:      </info>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

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

- o A `<org:id>` element that contains the server-unique identifier of the organization object, as defined in [Section 3.1](#).
- o A `<org:roid>` element that contains the Repository Object Identifier assigned to the organization object when the object was created.
- o One or more `<org:role>` elements that contains the role type and optional role id of the organization.
 - * A `<org:type>` element that contains the type of the organization, as defined in [Section 3.2](#). A "roleStatus" attribute is used to describe the status value of a role type.
 - * An OPTIONAL `<org:roleid>` element that contains a third party assigned identifier, such as IANA ID for registrars, as defined in [Section 3.2.2](#).
- o One or more `<org:status>` elements that contains the operational status of the organization, as defined in [Section 3.4](#).
- o An OPTIONAL `<org:parentId>` element that contains the identifier of the parent object, as defined in [Section 3.6](#).
- o One or two `<org:postalInfo>` elements that contain postal-address information. Two elements are provided so that address information can be provided in both internationalized and localized forms; a "type" attribute is used to identify the two forms. If an internationalized form (type="int") is provided, element content MUST be represented in a subset of UTF-8 that can

be represented in the 7-bit US-ASCII character set. If a localized form (type="loc") is provided, element content MAY be represented in unrestricted UTF-8. The <org:postalInfo> element contains the following child elements:

- * A <org:name> element that contains the name of the organization.
- * A <org:addr> element that contains address information associated with the organization. A <org:addr> element contains the following child elements:
 - + One, two, or three OPTIONAL <org:street> elements that contain the organization's street address.
 - + A <org:city> element that contains the organization's city.
 - + An OPTIONAL <org:sp> element that contains the organization's state or province.
 - + An OPTIONAL <org:pc> element that contains the organization's postal code.
 - + A <org:cc> element that contains the organization's country code.
- o An OPTIONAL <org:voice> element that contains the organization's voice telephone number.
- o An OPTIONAL <org:fax> element that contains the organization's facsimile telephone number.
- o An OPTIONAL <org:email> element that contains the organization's email address.
- o An OPTIONAL <org:url> element that contains the URL to the website of the organization.
- o Zero or more OPTIONAL <org:contact> elements that contain identifiers for the contact objects to be associated with the organization object. Contact object identifiers MUST be known to the server before the contact object can be associated with the organization object. The required "type" is used to represent contact types. The type values include "admin", "tech", "billing", "abuse", and "custom". The OPTIONAL "typeName" attribute is used to define the name of a "custom" type.

- o An OPTIONAL <org:clID> element that contains the organization identifier of the sponsoring client. There is no <org:clID> element if the organization is managed by the registry.
- o A <org:crID> element that contains the identifier of the client that created the organization object.
- o A <org:crDate> element that contains the date and time of organization object creation.
- o A <org:upID> element that contains the identifier of the client that last updated the organization object. This element MUST NOT be present if the organization has never been modified.
- o A <org:upDate> element that contains the date and time of the most recent organization object modification. This element MUST NOT be present if the organization object has never been modified.

Example <info> response for "Example Registrar Inc." organization object with registrar identifier "1362":


```
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>Command completed successfully</msg>
S:    </result>
S:    <resData>
S:      <org:infData
S:        xmlns:org="urn:ietf:params:xml:ns:org-1.0">
S:        <org:id>registrar1362</org:id>
S:        <org:roid>registrar1362-REP</org:roid>
S:        <org:role>
S:          <org:type roleStatus="ok">registrar</org:type>
S:          <org:roleid>1362</org:roleid>
S:        </org:role>
S:        <org:status>ok</org:status>
S:        <org:postalInfo type="int">
S:          <org:name>Example Registrar Inc.</org:name>
S:          <org:addr>
S:            <org:street>123 Example Dr.</org:street>
S:            <org:street>Suite 100</org:street>
S:            <org:city>Dulles</org:city>
S:            <org:sp>VA</org:sp>
S:            <org:pc>20166-6503</org:pc>
S:            <org:cc>US</org:cc>
S:          </org:addr>
S:        </org:postalInfo>
S:        <org:voice x="1234">+1.7035555555</org:voice>
S:        <org:fax>+1.7035555556</org:fax>
S:        <org:email>contact@organization.example</org:email>
S:        <org:url>http://organization.example</org:url>
S:        <org:contact type="admin">sh8013</org:contact>
S:        <org:contact type="billing">sh8013</org:contact>
S:        <org:contact type="custom"
S:          typeName="legal">sh8013</org:contact>
S:        <org:crID>ClientX</org:crID>
S:        <org:crDate>1999-04-03T22:00:00.0Z</org:crDate>
S:        <org:upID>ClientX</org:upID>
S:        <org:upDate>1999-12-03T09:00:00.0Z</org:upDate>
S:      </org:infData>
S:    </resData>
S:    <trID>
S:      <clTRID>ABC-12345</clTRID>
S:      <svTRID>54322-XYZ</svTRID>
S:    </trID>
S:  </response>
S:</epp>
```


Example <info> response for "Example Reseller Inc." organization object of reseller type managed by registrar identifier "1362":

```
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>Command completed successfully</msg>
S:    </result>
S:    <resData>
S:      <org:infData
S:        xmlns:org="urn:ietf:params:xml:ns:org-1.0">
S:        <org:id>reseller1523</org:id>
S:        <org:roid>reseller1523-REP</org:roid>
S:        <org:role>
S:          <org:type roleStatus="linked">reseller</org:type>
S:        </org:role>
S:        <org:status>ok</org:status>
S:        <org:parentId>registrar1362</org:parentId>
S:        <org:postalInfo type="int">
S:          <org:name>Example Reseller Inc.</org:name>
S:          <org:addr>
S:            <org:street>123 Example Dr.</org:street>
S:            <org:street>Suite 100</org:street>
S:            <org:city>Dulles</org:city>
S:            <org:sp>VA</org:sp>
S:            <org:pc>20166-6503</org:pc>
S:            <org:cc>US</org:cc>
S:          </org:addr>
S:        </org:postalInfo>
S:        <org:fax>+1.7035555556</org:fax>
S:        <org:url>http://organization.example</org:url>
S:        <org:contact type="admin">sh8013</org:contact>
S:        <org:clID>1362</org:clID>
S:        <org:crID>ClientX</org:crID>
S:        <org:crDate>1999-04-03T22:00:00.0Z</org:crDate>
S:        <org:upID>ClientX</org:upID>
S:        <org:upDate>1999-12-03T09:00:00.0Z</org:upDate>
S:      </org:infData>
S:    </resData>
S:    <trID>
S:      <clTRID>ABC-12345</clTRID>
S:      <svTRID>54322-XYZ</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>` Command

The transfer semantics does not apply to organization object. No EPP `<transfer>` command is defined in this document.

4.2. EPP Transform Commands

EPP provides four commands to transform organization object information: `<create>` to create an instance of an organization object, `<delete>` to delete an instance of an organization object, `<transfer>` to manage organization-object sponsorship changes, and `<update>` to change information associated with an organization object. This document does not define a mapping for the EPP `<transfer>` and `<renew>` command.

Transform commands are typically processed and completed in real time. Server operators **MAY** receive and process transform commands but defer completing the requested action if human or third-party review is required before the requested action can be completed. In such situations, the server **MUST** return a 1001 response code to the client to note that the command has been received and processed but that the requested action is pending. The server **MUST** also manage the status of the object that is the subject of the command to reflect the initiation and completion of the requested action. Once the action has been completed, all clients involved in the transaction **MUST** be notified using a service message that the action has been completed and that the status of the object has changed. Other notification methods **MAY** be used in addition to the required service message.

Server operators **SHOULD** confirm that a client is authorized to perform a transform command on a given object. Any attempt to transform an object by an unauthorized client **MUST** be rejected, and the server **MUST** return a 2201 response code to the client to note that the client lacks privileges to execute the requested command.

4.2.1. EPP `<create>` Command

The EPP `<create>` command provides a transform operation that allows a client to create an organization object. In addition to the standard EPP command elements, the `<create>` command **MUST** contain a `<org:create>` element that identifies the organization namespace. The `<org:create>` element contains the following child elements:

- o A <org:id> element that contains the desired server-unique identifier for the organization to be created, as defined in [Section 3.1](#).
- o One or more <org:role> elements that contains the role type and optional role id of the organization.
 - * A <org:type> element that contains the type of the organization, as defined in [Section 3.2](#). A "roleStatus" attribute is used to describe the status value of a role type.
 - * An OPTIONAL <org:roleid> element that contains a third party assigned identifier, such as IANA ID for registrars, as defined in [Section 3.2.2](#).
- o A <org:status> element that contains the operational status of the organization, as defined in [Section 3.4](#).
- o An OPTIONAL <org:parentId> element that contains the identifier of the parent object, as defined in [Section 3.6](#).
- o One or two <org:postalInfo> elements that contain postal-address information. Two elements are provided so that address information can be provided in both internationalized and localized forms; a "type" attribute is used to identify the two forms. If an internationalized form (type="int") is provided, element content MUST be represented in a subset of UTF-8 that can be represented in the 7-bit US-ASCII character set. If a localized form (type="loc") is provided, element content MAY be represented in unrestricted UTF-8. The <org:postalInfo> element contains the following child elements:
 - * A <org:name> element that contains the name of the organization.
 - * A <org:addr> element that contains address information associated with the organization. A <org:addr> element contains the following child elements:
 - + One, two, or three OPTIONAL <org:street> elements that contain the organization's street address.
 - + A <org:city> element that contains the organization's city.
 - + An OPTIONAL <org:sp> element that contains the organization's state or province.

- + An OPTIONAL <org:pc> element that contains the organization's postal code.
- + A <org:cc> element that contains the organization's country code.
- o An OPTIONAL <org:voice> element that contains the organization's voice telephone number.
- o An OPTIONAL <org:fax> element that contains the organization's facsimile telephone number.
- o An OPTIONAL <org:email> element that contains the organization's email address.
- o An OPTIONAL <org:url> element that contains the URL to the website of the organization.
- o Zero or more OPTIONAL <org:contact> elements that contain identifiers for the contact objects associated with the organization object.

Example <create> command:


```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <create>
C:      <org:create
C:        xmlns:org="urn:ietf:params:xml:ns:org-1.0">
C:        <org:id>res1523</org:id>
C:        <org:role>
C:          <org:type roleStatus="ok">reseller</org:type>
C:          <org:type roleStatus="ok">privacyproxy</org:type>
C:        </org:role>
C:        <org:status>ok</org:status>
C:        <org:parentId>1523res</org:parentId>
C:        <org:postalInfo type="int">
C:          <org:name>Example Organization Inc.</org:name>
C:          <org:addr>
C:            <org:street>123 Example Dr.</org:street>
C:            <org:street>Suite 100</org:street>
C:            <org:city>Dulles</org:city>
C:            <org:sp>VA</org:sp>
C:            <org:pc>20166-6503</org:pc>
C:            <org:cc>US</org:cc>
C:          </org:addr>
C:        </org:postalInfo>
C:        <org:voice x="1234">+1.7035555555</org:voice>
C:        <org:fax>+1.7035555556</org:fax>
C:        <org:email>contact@organization.example</org:email>
C:        <org:url>http://organization.example</org:url>
C:        <org:contact type="admin">sh8013</org:contact>
C:        <org:contact type="billing">sh8013</org:contact>
C:      </org:create>
C:    </create>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

When a <create> command has been processed successfully, the EPP <resData> element MUST contain a child <org:creData> element that identifies the organization namespace. The <org:creData> element contains the following child elements:

- o A <org:id> element that contains the server-unique identifier for the created organization, as defined in [Section 3.1](#).
- o A <org:crDate> element that contains the date and time of organization-object creation.

Example <create> response:


```
S:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S:  <response>
S:    <result code="1000">
S:      <msg>Command completed successfully</msg>
S:    </result>
S:    <resData>
S:      <org:creData
S:        xmlns:org="urn:ietf:params:xml:ns:org-1.0">
S:        <org:id>res1523</org:id>
S:        <org:crDate>1999-04-03T22:00:00.0Z</org:crDate>
S:      </org:creData>
S:    </resData>
S:    <trID>
S:      <clTRID>ABC-12345</clTRID>
S:      <svTRID>54321-XYZ</svTRID>
S:    </trID>
S:  </response>
S:</epp>
```

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

4.2.2. EPP <delete> Command

The EPP <delete> command provides a transform operation that allows a client to delete an organization object. In addition to the standard EPP command elements, the <delete> command MUST contain a <org:delete> element that identifies the organization namespace. The <org:delete> element MUST contain the following child element:

- o A <org:id> element that contains the server-unique identifier of the organization object to be deleted, as defined in [Section 3.1](#).

An organization object MUST NOT be deleted if it is associated with other known objects. An associated organization MUST NOT be deleted until associations with other known objects have been broken. A server MUST notify clients that object relationships exist by sending a 2305 error response code when a <delete> command is attempted and fails due to existing object relationships.

Example <delete> command:


```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <delete>
C:      <org:delete
C:        xmlns:org="urn:ietf:params:xml:ns:org-1.0">
C:          <org:id>res1523</org:id>
C:        </org:delete>
C:      </delete>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

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

Example <delete> response:

```
S:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S:  <response>
S:    <result code="1000">
S:      <msg>Command completed successfully</msg>
S:    </result>
S:    <trID>
S:      <clTRID>ABC-12345</clTRID>
S:      <svTRID>54321-XYZ</svTRID>
S:    </trID>
S:  </response>
S:</epp>
```

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

4.2.3. EPP <renew> Command

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

4.2.4. EPP <transfer> Command

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

4.2.5. EPP <update> Command

The EPP <update> command provides a transform operation that allows a client to modify the attributes of an organization object. In addition to the standard EPP command elements, the <update> command MUST contain a <org:update> element that identifies the organization namespace. The <org:update> element contains the following child elements:

- o A <org:id> element that contains the server-unique identifier of the organization object to be updated, as defined in [Section 3.1](#).
- o An OPTIONAL <org:add> element that contains attribute values to be added to the object.
- o An OPTIONAL <org:rem> element that contains attribute values to be removed from the object.
- o An OPTIONAL <org:chg> element that contains attribute values to be changed.

At least one <org:add>, <org:rem> or <org:chg> element MUST be provided if the command is not being extended. All of these elements MAY be omitted if an <update> extension is present. The <org:add> and <org:rem> elements contain the following child element:

- o Zero or more <org:contact> elements that contain the identifiers for contact objects to be associated with or removed from the organization object. Contact object identifiers MUST be known to the server before the contact object can be associated with the organization object.

A <org:chg> element contains the following OPTIONAL child elements. At least one child element MUST be present:

- o One or more <org:role> elements that contains the role type and optional role id of the organization.
 - * A <org:type> element that contains the type of the organization, as defined in [Section 3.2](#). A "roleStatus" attribute is used to describe the status value of a role type.
 - * An OPTIONAL <org:roleid> element that contains a third party assigned identifier, such as IANA ID for registrars, as defined in [Section 3.2.2](#).
- o A <org:status> element that contains the operational status of the organization.

- o A <org:parentId> element that contains the identifier of the parent object.
- o One or two <org:postalInfo> elements that contain postal-address information. Two elements are provided so that address information can be provided in both internationalized and localized forms; a "type" attribute is used to identify the two forms. If an internationalized form (type="int") is provided, element content MUST be represented in a subset of UTF-8 that can be represented in the 7-bit US-ASCII character set. If a localized form (type="loc") is provided, element content MAY be represented in unrestricted UTF-8. The change of the postal info is defined as a replacement of that postal info element with the contents of the sub-elements included in the update command. An empty <org:postalInfo> element is supported to allow a type of postal info to be removed. The <org:postalInfo> element contains the following child elements:
 - * A <org:name> element that contains the name of the organization.
 - * A <org:addr> element that contains address information associated with the organization. A <org:addr> element contains the following child elements:
 - + One, two, or three OPTIONAL <org:street> elements that contain the organization's street address.
 - + A <org:city> element that contains the organization's city.
 - + An OPTIONAL <org:sp> element that contains the organization's state or province.
 - + An OPTIONAL <org:pc> element that contains the organization's postal code.
 - + A <org:cc> element that contains the organization's country code.
- o A <org:voice> element that contains the organization's voice telephone number.
- o A <org:fax> element that contains the organization's facsimile telephone number.
- o A <org:email> element that contains the organization's email address.

- o A <org:url> element that contains the URL to the website of the organization.

Example <update> command:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <update>
C:      <org:update
C:        xmlns:org="urn:ietf:params:xml:ns:org-1.0">
C:        <org:id>res1523</org:id>
C:        <org:add>
C:          <org:contact type="tech">sh8013</org:contact>
C:        </org:add>
C:        <org:chg>
C:          <org:role>
C:            <org:type roleStatus="linked">reseller</org:type>
C:            <org:type roleStatus="clientLinkProhibited">privacyproxy</
org:type>
C:          </org:role>
C:          <org:status>ok</org:status>
C:          <org:postalInfo type="int">
C:            <org:addr>
C:              <org:street>124 Example Dr.</org:street>
C:              <org:street>Suite 200</org:street>
C:              <org:city>Dulles</org:city>
C:              <org:sp>VA</org:sp>
C:              <org:pc>20166-6503</org:pc>
C:              <org:cc>US</org:cc>
C:            </org:addr>
C:          </org:postalInfo>
C:          <org:voice>+1.7034444444</org:voice>
C:          <org:fax/>
C:        </org:chg>
C:      </org:update>
C:    </update>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

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

Example <update> response:


```
S:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S:  <response>
S:    <result code="1000">
S:      <msg>Command completed successfully</msg>
S:    </result>
S:    <trID>
S:      <clTRID>ABC-12345</clTRID>
S:      <svTRID>54321-XYZ</svTRID>
S:    </trID>
S:  </response>
S:</epp>
```

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

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:org-1.0"
  xmlns:org="urn:ietf:params:xml:ns:org-1.0"
  xmlns:epp="urn:ietf:params:xml:ns:epp-1.0"
  xmlns:eppcom="urn:ietf:params:xml:ns:eppcom-1.0"
  xmlns:contact="urn:ietf:params:xml:ns:contact-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"/>
  <import namespace="urn:ietf:params:xml:ns:contact-1.0"/>
  <import namespace="urn:ietf:params:xml:ns:domain-1.0"/>

  <annotation>
    <documentation>
      Extensible Provisioning Protocol v1.0
```



```
        organization provisioning schema.
    </documentation>
</annotation>

<!--
Child elements found in EPP commands.
-->
<element name="create" type="org:createType"/>
<element name="delete" type="org:sIDType"/>
<element name="update" type="org:updateType"/>
<element name="check" type="org:mIDType"/>
<element name="info" type="org:infoType"/>

<!--
Utility types.
-->
<simpleType name="statusType">
  <restriction base="token">
    <enumeration value="ok"/>
    <enumeration value="hold"/>
    <enumeration value="terminated"/>
    <enumeration value="clientDeleteProhibited"/>
    <enumeration value="clientUpdateProhibited"/>
    <enumeration value="clientLinkProhibited"/>
    <enumeration value="linked"/>
    <enumeration value="pendingCreate"/>
    <enumeration value="pendingUpdate"/>
    <enumeration value="pendingDelete"/>
    <enumeration value="serverDeleteProhibited"/>
    <enumeration value="serverUpdateProhibited"/>
    <enumeration value="serverLinkProhibited"/>
  </restriction>
</simpleType>

<simpleType name="roleStatusType">
  <restriction base="token">
    <enumeration value="ok"/>
    <enumeration value="clientLinkProhibited"/>
    <enumeration value="linked"/>
    <enumeration value="serverLinkProhibited"/>
  </restriction>
</simpleType>

<complexType name="roleType">
  <sequence>
    <element name="type" type="token"/>
    <element name="roleid" type="positiveInteger" minOccurs="0"/>
  </sequence>
```



```
    <attribute nmae="roleStatus" type="roleStatusType" default="ok"/>
  </complexType>

  <complexType name="postalInfoType">
    <sequence>
      <element name="name"
        type="contact:postalLineType"/>
      <element name="addr"
        type="contact:addrType"/>
    </sequence>
    <attribute name="type"
      type="contact:postalInfoEnumType"
      use="required"/>
  </complexType>

  <complexType name="contactType">
    <simpleContent>
      <extension base="eppcom:clIDType">
        <attribute name="type" type="org:contactAttrType" use="required"/>
        <attribute name="typeName"
          type="token"/>
      </extension>
    </simpleContent>
  </complexType>

  <simpleType name="contactAttrType">
    <restriction base="token">
      <enumeration value="admin"/>
      <enumeration value="billing"/>
      <enumeration value="tech"/>
      <enumeration value="abuse"/>
      <enumeration value="custom"/>
    </restriction>
  </simpleType>

  <!--
  Child element of commands that require only an identifier.
  -->
  <complexType name="sIDType">
    <sequence>
      <element name="id"
        type="eppcom:clIDType"/>
    </sequence>
  </complexType>

  <!--
  Child element of commands that accept multiple identifiers.
  -->
```



```
<complexType name="mIDType">
  <sequence>
    <element name="id"
      type="eppcom:clIDType" maxOccurs="unbounded"/>
  </sequence>
</complexType>

<!--
Child elements of the <info> commands.
-->
<complexType name="infoType">
  <sequence>
    <element name="id"
      type="eppcom:clIDType"/>
  </sequence>
</complexType>

<!--
Child elements of the <create> command.
-->
<complexType name="createType">
  <sequence>
    <element name="id"
      type="eppcom:clIDType"/>
    <element name="role"
      type="org:roleType"/>
    <element name="status"
      type="org:statusType" default="ok" maxOccurs="9"/>
    <element name="parentId"
      type="eppcom:clIDType" minOccurs="0"/>
    <element name="postalInfo"
      type="org:postalInfoType" maxOccurs="2"/>
    <element name="voice"
      type="contact:e164Type" minOccurs="0"/>
    <element name="fax"
      type="contact:e164Type" minOccurs="0"/>
    <element name="email"
      type="eppcom:minTokenType"/>
    <element name="url"
      type="anyURI" minOccurs="0"/>
    <element name="contact"
      type="org:contactType"
      minOccurs="0" maxOccurs="5"/>
  </sequence>
</complexType>

<!--
Child elements of the <update> command.
```



```
-->
<complexType name="updateType">
  <sequence>
    <element name="id"
      type="eppcom:clIDType"/>
    <element name="add"
      type="org:addRemType" minOccurs="0"/>
    <element name="rem"
      type="org:addRemType" minOccurs="0"/>
    <element name="chg"
      type="org:chgType" minOccurs="0"/>
  </sequence>
</complexType>

<!--
Data elements that can be added or removed.
-->
<complexType name="addRemType">
  <sequence>
    <element name="contact"
      type="domain:contactType" minOccurs="0"/>
  </sequence>
</complexType>

<!--
Data elements that can be changed.
-->
<complexType name="chgType">
  <sequence>
    <element name="status"
      type="org:statusType"/>
    <element name="parentId"
      type="eppcom:clIDType" minOccurs="0"/>
    <element name="postalInfo"
      type="org:chgPostalInfoType"
      minOccurs="0" maxOccurs="2"/>
    <element name="voice"
      type="contact:e164Type" minOccurs="0"/>
    <element name="fax"
      type="contact:e164Type" minOccurs="0"/>
    <element name="email"
      type="eppcom:minTokenType" minOccurs="0"/>
    <element name="url"
      type="anyURI" minOccurs="0"/>
  </sequence>
</complexType>

<complexType name="chgPostalInfoType">
```



```
<sequence>
  <element name="name"
    type="contact:postalLineType" minOccurs="0"/>
  <element name="addr"
    type="contact:addrType" minOccurs="0"/>
</sequence>
<attribute name="type"
  type="contact:postalInfoEnumType" use="required"/>
</complexType>

<!--
Child response elements.
-->
<element name="chkData" type="contact:chkDataType"/>
<element name="creData" type="contact:creDataType"/>
<element name="infData" type="org:infDataType"/>

<!--
<info> response elements.
-->
<complexType name="infDataType">
  <sequence>
    <element name="id"
      type="eppcom:clIDType"/>
    <element name="roid"
      type="eppcom:roidType"/>
    <element name="role"
      type="org:roleType"/>
    <element name="status"
      type="org:statusType"/>
    <element name="parentId"
      type="eppcom:clIDType" minOccurs="0"/>
    <element name="postalInfo"
      type="org:postalInfoType" maxOccurs="2"/>
    <element name="voice"
      type="contact:e164Type" minOccurs="0"/>
    <element name="fax"
      type="contact:e164Type" minOccurs="0"/>
    <element name="email"
      type="eppcom:minTokenType"/>
    <element name="url"
      type="anyURI" minOccurs="0"/>
    <element name="contact"
      type="domain:contactType" minOccurs="0" maxOccurs="3"/>
    <element name="clID"
      type="eppcom:clIDType" minOccurs="0"/>
    <element name="crID"
      type="eppcom:clIDType"/>
```



```
<element name="crDate"
  type="dateTime"/>
<element name="upID"
  type="eppcom:clIDType" minOccurs="0"/>
<element name="upDate"
  type="dateTime" minOccurs="0"/>
</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 organization object mapping, the elements and 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 namespace:

- o URI: urn:ietf:params:xml:ns:org-1.0
- o Registrant Contact: See the "Author's Address" section of this document.
- o 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 Object Extension

Document status: Standards Track

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

Registrant Name and Email Address: See the "Author's Address" section of this document.

TLDs: any

IPR Disclosure: none

Status: active

Notes: none

7.3. Role Values Registry

The following values should be registered by the IANA in the "EPP Organization Role Values" registry:

Value: registrar

Type: role

Description: The entity object instance represents the authority responsible for the registration in the registry.

Registrant Name: IESG

Registrant Contact Information: iesg@ietf.org

Value: reseller

Type: role

Description: The entity object instance represents a third party through which the registration was conducted (i.e., not the registry or registrar).

Registrant Name: IESG

Registrant Contact Information: `iesg@ietf.org`

Value: `privacyproxy`

Type: `role`

Description: The entity object instance represents a third-party who could help to register a domain without exposing the registrants' private information..

Registrant Name: `IESG`

Registrant Contact Information: `iesg@ietf.org`

8. Implementation Status

Note to RFC Editor: Please remove this section and the reference to [\[RFC6982\]](#) 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 [\[RFC6982\]](#). 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 [\[RFC6982\]](#), "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".

CNNIC is in the process of development research to update EPP organization mapping from reseller mapping. Verisign is also planning to implement this document.

8.1. CNNIC Implementation

Organization: `CNNIC`

Name: `EPP Organization Mapping`

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

Level of maturity: Research.

Coverage: EPP organization mapping.

Contact: zhouguiqing@cnnic.cn

8.2. Reseller Extension

This document was updated from [draft-ietf-regext-reseller](#). CNNIC, Verisign and Net::DRI (Patrick Mevzek) have already implemented the reseller mapping.

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 [\[RFC5730\]](#) or those caused by the protocol layers used by EPP. The security considerations described in these other specifications apply to this specification as well.

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. 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>>.
- [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>>.

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

[Appendix A](#). Change Log

Initial -00: Individual document submitted.

-01:

- * Updated abstract text.
- * Added sentences to avoid loop of parent identifiers in [section 3.4](#).
- * Revised typos in [section 3.6](#).
- * Added explanation of contact type attribute in [section 4.1.2](#).
- * Updated <info> responses.
- * Deleted description of <transfer> command in [section 4.1](#) and 4.2.

- * Deleted whoisInfo disclose type in XML schema.
- * Deleted maxOccurs of addRemType.
- * Deleted extra "OPTIONAL" in [section 4.2.5](#).
- * Updated typos in <update> response.

-02:

- * Changed author information.
- * Updated url definition.
- * Updated XML schema.

-03:

- * Changed author information.
- * Updated [section 3.1](#).
- * Refactoried the XSD file. Added <chgPostalInfoType> element.
- * Added acknowledgment.

WG document-00: WG document submitted

WG document-01: Keep document alive for further discussion.
Reseller object or entity object with multiple roles?

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

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

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

Authors' Addresses

Linlin Zhou

CNNIC

4 South 4th Street, Zhongguancun, Haidian District
Beijing, Beijing 100190
China

Phone: +86 10 5881 2677

Email: zhoulinlin@cnnic.cn

Ning Kong

CNNIC

4 South 4th Street, Zhongguancun, Haidian District
Beijing, Beijing 100190
China

Phone: +86 10 5881 3147

Email: nkong@cnnic.cn

Guiqing Zhou

CNNIC

4 South 4th Street, Zhongguancun, Haidian District
Beijing, Beijing 100190
China

Phone: +86 10 5881 2692

Email: zhouguiqing@cnnic.cn

Xiaodong Lee

CNNIC

4 South 4th Street, Zhongguancun, Haidian District
Beijing, Beijing 100190
China

Phone: +86 10 5881 3020

Email: xl@cnnic.cn

James Gould

VeriSign, Inc.

12061 Bluemont Way

Reston, VA 20190

US

Email: jgould@verisign.com

