

Internet Engineering Task Force	Y. Chen
Internet Draft	J. Xie
Intended status: Experimental	Z. Li
Expires: February 24, 2021	Z. Fan
China Academy of Information and Communications Technology	
August 17, 2020	

**Extensible Provisioning Protocol (EPP) Industrial Internet
Identifier Mapping
draft-chen-epp-identifier-mapping-03**

Abstract

This document describes an Extensible Provisioning Protocol (EPP) mapping for the provisioning and management of Industrial Internet Identifiers. Specified in XML, the mapping defines EPP command syntax and semantics as applied to identifiers.

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), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/1id-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>

This Internet-Draft will expire on February 24, 2021.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of

publication of this document. Please review these documents
Chen, et al. Expires February 17,2021 [Page 1]

Internet-Draft EPP Identifier Mapping August 17, 2020

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.

1.	Introduction	4
1.1.	Conventions Used in This Document	4
1.2.	Scope of Experimentation	4
2.	Object Attributes	4
2.1.	Industrial Internet Identifier Object	5
2.2.	Client Identifiers	5
2.3.	Status Values	5
2.4.	Dates and Times.....	7
2.5.	IP Addresses	7
3.	EPP Command Mapping	7
3.1.	EPP Query Commands	7
3.1.1.	EPP <check> Command	8
3.1.2.	EPP <info> Command	10
3.1.3.	EPP <transfer> Query Command	14
3.2.	EPP Transform Commands	14
3.2.1.	EPP <create> Command	15
3.2.2.	EPP <delete> Command	19
3.2.3.	EPP <renew> Command	20
3.2.4.	EPP <transfer> Command	20
3.2.5.	EPP <update> Command	20
4.	Formal Syntax	25
5.	Internationalization Considerations	33
6.	Security Considerations	34
7.	IANA Considerations	34
8.	Acknowledgments	35
9.	References	35
9.1.	Normative References	35
9.2.	Informative References	36

1. Introduction

Industrial Internet Identifiers are character strings with a specified format that may consist of digits, letters or notations being structured in a way that is interpretable by one or more computational facilities.

This document describes an Industrial Internet Identifier mapping for version 1.0 of the Extensible Provisioning Protocol (EPP). This mapping is specified using the Extensible Markup Language (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](#)].

[[RFC5730](#)] provides a complete description of EPP command and response structures. A thorough understanding of the base protocol specification is necessary to understand the mapping described in this document.

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.

1.1. Conventions Used in This Document

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.

1.2. Scope of Experimentation

This document describes an experimental extension to EPP protocol. This section specifies scope of this experiment and how it can yield useful information.

This EPP extension is designed to manage the registration, modification and resolution of digital objects, handles, OID, for example throughout the Industrial Internet. According to the definition of EPP, this extension is an XML text protocol that permits multiple service providers to perform object-provisioning operations using a shared central object repository. It is designed for use in a layered protocol environment. Bindings to specific transport and security protocols are outside the scope of this specification.

Given the above points, the experiment can be run on the open Internet between consenting client and server implementations.

2. Object Attributes

An EPP identifier 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.

2.1. Industrial Internet Identifier Object

Industrial Internet Identifiers are character strings with a specified format that may consist of digits, letters or notations being structured in a way that is interpretable by one or more computational facilities.

It is an unique persistent set of bits used to identify and obtain state information about physical resource such as machines, products, or digital resources such as algorithms, manufacturing process, etc.

This document provides an overview of the EPP mapping of Industrial Internet Identification. Handle mapping is specified as an example, while description in this document applies to other identification techniques as well.

The syntax for handle namespace described in this document MUST conform to [[RFC3650](#)], [[RFC3651](#)], [[RFC3652](#)]. Handle identifiers are character strings with a specified length and a specified format.

All handle identifiers are of the form prefix/suffix where, by default, the prefix may first be resolved to locate the specific identifier service and the suffix may be any bit sequence. Epp mapping on the prefix examples are provided in this document while it MAY also apply to handle identifiers with suffix.

These conformance requirements might change in the future as a result of progressing work in developing standards for internationalized digital object identification.

2.2. Client Identifiers

All EPP clients are identified by a server-unique identifier. Client identifiers conform to the "clIDType" syntax described in [[RFC5730](#)].

2.3. Status Values

An EPP identifier object MUST always have at least one associated

status value. Status values MAY be set only by the client that
sponsors an identifier object and by the server on which the object
Chen, et al. Expires February 17,2021 [Page 5]

resides. A client can change the status of object using the EPP <update> command. Each status value MAY be accompanied by a string of human-readable text that describes the rationale for the status applied to the object.

A client MUST NOT alter status values set by the server. A server MAY alter or override status values set by a client, subject to local server policies. The status of an object MAY change as a result of either a client-initiated transform command or an action performed by a server operator.

Status values that can be added or removed by a client are prefixed with "client". Corresponding status values that can be added or removed by a server are prefixed with "server". Status values that do not begin with either "client" or "server" are server-managed.

Status Value Descriptions:

- clientDeleteProhibited, serverDeleteProhibited

Requests to delete the object MUST be rejected.

- clientUpdateProhibited, serverUpdateProhibited

Requests to update the object (other than to remove this status) MUST be rejected.

- linked

The identifier object has at least one active association with another object. Servers SHOULD provide services to determine existing object associations.

- 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.

- pendingCreate, pendingDelete, pendingTransfer, pendingUpdate

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.

When the requested action has been completed, the pendingCreate,

pendingDelete, pendingTransfer, or pendingUpdate status value MUST
be removed. All clients involved in the transaction MUST be
Chen, et al. Expires February 17,2021 [Page 6]

Internet-Draft EPP Identifier Mapping August 17, 2020
notified using a service message that the action has been completed
and that the status of the object has changed.

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

"linked" status MAY be combined with any status.

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

"pendingUpdate" status MUST NOT be combined with either
"clientUpdateProhibited" or "serverUpdateProhibited" status.

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

Other status combinations not expressly prohibited MAY be used.

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.

2.5. IP Addresses

The syntax for IPv4 addresses described in this document MUST
conform To[RFC5730]. The syntax for IPv6 addresses described in
this document MUST conform to [[RFC4291](#)]. Practical considerations
for publishing IPv6 address information in zone files are documented
in [[RFC2874](#)] and [[RFC3596](#)]. A server MAY reject IP addresses that
have not been allocated for public use by IANA.

3. EPP Command Mapping

A detailed description of the EPP syntax and semantics is specified
in [[RFC5730](#)]. The command mappings described here are specifically
for use in provisioning and managing Industrial Internet identifiers
via EPP.

3.1. EPP Query Commands

EPP provides two commands to retrieve object information: <check> to
determine if an EPP object can be provisioned within a repository,
Chen, et al. Expires February 17,2021 [Page 7]

and <info> to retrieve detailed information associated with an EPP object.

3.1.1. 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 an <identifier: check> element that recognizes the identifier namespace. The <identifier: check> element contains the following child elements:

- o One or more <identifier:name> elements that contain the fully qualified names of the identifier objects to be queried.

example <check> command:

```
C:<?xml version="1.0" encoding="utf-8"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0" xmlns:xsi=
C:"http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation=
C:"urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
C:  <command>
C:    <check>
C:      <identifier:check xmlns:identifier="urn:ietf:params:xml:ns:
C:        identifier-1.0" xsi:schemaLocation="urn:ietf:params:xml:ns:
C:        identifier-1.0 identifier-1.0.xsd">
C:        <identifier:name>88.1000.1</identifier:name>
C:        <identifier:name>88.1000.2</identifier:name>
C:      </identifier:check>
C:    </check>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

When a <check> command has been processed successfully, a server MUST respond with an EPP <resData> element that MUST contain a child element that identifies the identifier object namespace. The child elements of the <resData> element are identifier-specific, though the EPP <resData> element MUST contain a child <identifier:chkData> element that contains one or more <identifier:cd> (check data) elements. Each <identifier:cd> element contains the following child elements:

- o An identifier-specific element that identifies the queried identifier.

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 identifier can be provisioned. A value of "0" or "false" means that the identifier cannot be provisioned.

- o An <identifier:reason> element that is provided when an identifier cannot be provisioned. This element contains server-specific text to help explain why the identifier 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"?>
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:      <identifier:chkData xmlns:identifier="urn:ietf:params:
S:        xml:ns:identifier-1.0">
S:        <identifier:cd>
S:          <identifier:name avail="false">88.1000.1
S:          </identifier:name>
S:          <identifier:reason>The identifier already exists
S:          </identifier:reason>
S:        </identifier:cd>
S:        <identifier:cd>
S:          <identifier:name avail="true">88.1000.1
S:          </identifier:name>
S:        </identifier:cd>
S:      </identifier:chkData>
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 <check> command cannot be processed for any reason.

3.1.2. EPP <info> Command

The EPP <info> command is used to retrieve information associated with an Industrial Internet Identifier object. In addition to the standard EPP command elements, the <info> command MUST contain an <identifier:info> element that identifies the identifier namespace. The <identifier:info> element contains one child element:

An <identifier:name> element that contains the fully qualified name of the identifier object for which information is requested.

Example <info> command:

```
C:<?xml version="1.0" encoding="utf-8"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0" xmlns:xsi=
C:"http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation
C:"urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
C:  <command>
C:    <info>
C:      <identifier:info xmlns:identifier="urn:ietf:params:xml:
C:      ns:identifier-1.0" xsi:schemaLocation="urn:ietf:params:xml
C:      :ns:identifier-1.0 identifier-1.0.xsd">
C:        <identifier:name>88.1000.1</identifier:name>
C:      </identifier: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 <identifier:infData> element that identifies the identifier namespace. The <identifier:infData> element contains the following child elements:

- o An <identifier:name> element that contains the fully qualified name of the identifier object to be created. The identifier name with a minimum length of 1 byte and a maximum length of 255 bytes SHOULD be unique and SHOULD NOT be reused.

- o An <identifier:type> element that specifies type of identification technique of the identifier object. Handle is taken as an example in this document.
- o Zero or more OPTIONAL <identifier:contact> elements that contain contact information of the enterprise that applies for the identifier to be queried.
- o Zero or more OPTIONAL <identifier:URL> elements that contain the URL associated with the identifier object to be queried.
- o An <identifier:administratorList> element that contains one or more <identifier:administrator> elements that specify administrator information of the identifier object. Identifier administrators are entitled to create identifier or sub-naming authorities under the handle prefix according to the permission defined by its <identifier:permissionList> sub-element.

Each <identifier:administrator> element includes the following child elements:

An <identifier:adminIndex> element that provides the reference to the authentication key that can be used to authenticate the administrator.

An <identifier:pubkey> element that contains the authentication key of the administrator and information of the type of the technique used to authenticate administrator. The public key is processed with base64 encoding schemes.

Three types of algorithms are recommended to authenticate the identifier administrator: Digital Signature Algorithm (DSA) public-key cryptography, Rivest-Shamir-Adleman(RSA) public-key cryptography, or the password-based authentication mechanism.

The Digital Signature Algorithm (DSA) is a typical kind of cryptographic algorithm to generate pairs of keys used in public-key system: public keys which may be stored in the server, and private keys which are known only to the client.

The RSA is one of the first public-key cryptosystems and is another kind of cryptographic algorithm used for secure data transmission.

The password is a word or string of characters used for user authentication to prove identity of the administrator.

An <identifier:permissionList> element MAY contain zero or more <identifier:permission> elements that specify information about the administration authority of the administrator. A set of administration functions that include adding, deleting, and modifying identifier or identifier values are supported by the

identifier service. Before fulfilling any administration request,

Chen, et al.

Expires February 17, 2021

[Page 11]

the server must authenticate the client as the identifier administrator that is authorized for the administrative operation.

List of all the permissions see the "Formal Syntax" section of this document.

- o An <identifier:siteList> element that contains one or more <identifier:siteInfo> elements that provide information to locate the site to implement provisions and resolution of the identifier. In this section, the element defines a handle service site by identifying the server computers that comprise the site along with their service configurations (e.g., port numbers).

Each <identifier:siteInfo> contains the following child elements:
An <identifier:siteIndex> element that indicates the specific index of a site.

An <identifier:protocolVersion> element that indicates handle protocol version used to create the handle identifier.

One or more <identifier:serviceInfo> elements that contain the following elements:

An <identifier:serverID> element defines the number of servers in the service site.

One or more <identifier:addr> elements that describe IP address of the identifier service. Each <identifier:addr> element MAY contain an "ip" attribute to identify the IP address format. Attribute value "v4" is used to note IPv4 address format. Attribute value "v6" is used to note IPv6 address format. If the "ip" attribute is not specified, "v4" is the default attribute value.

An <identifier:pubkey> element that contains the server's public key with a "type" attribute that specifies algorithms used to generate the public key. Public key in the <identifier:serviceInfo> can be used to authenticate any service response from the handle server.

One or more <identifier:serviceInterfaces> elements that have three child elements: an <identifier:serviceType> element that indicates whether the service is for query or for administration, an <identifier:protocol> element that specifies transmission protocol, where UDP and HTTP could be considered as alternative protocols, and the <identifier:port> element that represents service port of specific the service component.

Example <info> response:


```
S:<?xml version="1.0" encoding="utf-8"?>
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:      <identifier:infData
S:        xmlns:identifier="urn:ietf:params:xml:ns:identifier-1.0">
S:        <identifier:name>88.1000.1</identifier:name>
S:        <identifier:type>handle</identifier:type>
S:        <identifier:status s="clientUpdateProhibited"/>
S:        <identifier:contact>jd1234</identifier:contact>
S:        <identifier:url>www.caict.ac.cn</identifier:url>
S:        <identifier:administratorList>
S:          <identifier:administrator>
S:            <identifier:adminIndex>100</identifier:adminIndex>
S:            <identifier:pubkey type="dsa_pub_key">
S:              AAAAB3NzaC1yc2EAAAADAQABAAQACprNl4N4e175lVnv03Qf
S:              wYFTfB05hhLDC1...</identifier:pubkey>
S:            <identifier:permissionList>
S:              <identifier:permission>add_handle
S:              </identifier:permission>
S:              <identifier:permission>delete_handle
S:              </identifier:permission>
S:              <identifier:permission>add_value
S:              </identifier:permission>
S:              <identifier:permission>modify_admin
S:              </identifier:permission>
S:              <identifier:permission>remove_admin
S:              </identifier:permission>
S:            </identifier:permissionList>
S:          </identifier:administrator>
S:        </identifier:administratorList>
S:        <identifier:siteList>
S:          <identifier:siteInfo>
S:            <identifier:siteIndex>500</identifier:siteIndex>
S:            <identifier:protocolVersion>2.10
S:            </identifier:protocolVersion>
S:          <identifier:serviceInfo>
S:            <identifier:serverID>1</identifier:serverID>
S:            <identifier:addr ip="v4">192.0.2.2</identifier:addr>
S:            <identifier:addr ip="v6">
S:              1080:0:0:0:8:800:200C:417A
S:            </identifier:addr>
```

Internet-Draft EPP Identifier Mapping August 17, 2020

```
S:      <identifier:pubkey type="dsa_pub_key">
S:      AAAAB3NzaC1yc2EAAAADAQABAAQACprNl4N4e175lVnv03QfwY
S:      FTfB05hhLDC1...</identifier:pubkey>
S:      <identifier:serviceInterfaces>
S:      <identifier:serviceType>query
S:      </identifier:serviceType>
S:      <identifier:protocol>tcp</identifier:protocol>
S:      <identifier:port>2641</identifier:port>
S:      </identifier:serviceInterfaces>
S:      </identifier:serviceInfo>
S:      </identifier:siteInfo>
S:      </identifier:siteList>
S:      </identifier: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.

3.1.3. EPP <transfer> Query Command

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

3.2. EPP Transform Commands

EPP provides three commands to transform identifier objects: <create> to create an instance of an identifier object, <delete> to delete an instance of an identifier object, and <update> to change information associated with an identifier object. This document does not define identifier-object mappings for the EPP <renew> and <transfer> commands.

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.

3.2.1. EPP <create> Command

The EPP <create> command provides an operation that allows a client to create an identifier object. In addition to the standard EPP command elements, the <create> command MUST contain an <identifier:create> element that identifies the identifier to be created. The <identifier:create> element contains the following child elements:

- o An <identifier:name> element that contains the fully qualified name of the identifier object to be created. The identifier name with a minimum length of 1 byte and a maximum length of 255 bytes SHOULD be unique and SHOULD NOT be reused.
- o An <identifier:type> element that specifies type of identification technique of the identifier object. Handle is taken as an example in this document.
- o Zero or more OPTIONAL <identifier:contact> elements that contain contact information of the enterprise that applies for the identifier to be created.
- o Zero or more OPTIONAL <identifier:URL> elements that contain the URL associated with the identifier object to be created.
- o An <identifier:administratorList> element that contains one or more <identifier:administrator> elements that specify administrator information of the identifier object. Identifier administrators are entitled to administrate or resolve identifier or identifier values according to the permission defined by its <identifier:permissionList> sub-element.

Each <identifier:administrator> element includes the following child elements:

An <identifier:adminIndex> element that provides the reference to the authentication key that can be used to authenticate the administrator.

An <identifier:pubkey> element that contains the authentication key of the administrator and information of the type of the technique used to authenticate administrator. The public key is processed with base64 encoding schemes.

Three types of algorithms are recommended to authenticate the identifier administrator: Digital Signature Algorithm (DSA) public-key cryptography, Rivest-Shamir-Adleman(RSA) public-key cryptography, or the password-based authentication mechanism.

The Digital Signature Algorithm (DSA) is a typical kind of cryptographic algorithm to generate pairs of keys used in public-key system: public keys which may be stored in the server, and private keys which are known only to the client.

The RSA is one of the first public-key cryptosystems and is another kind of cryptographic algorithm used for secure data transmission.

The password is a word or string of characters used for user authentication to prove identity of the administrator.

An <identifier:permissionList> element MAY contain zero or more <identifier:permission> elements that specify information about the administration authority of the administrator. A set of administration functions that include adding, deleting, and modifying identifier or identifier values are supported by the identifier service. Before fulfilling any administration request, the server must authenticate the client as the identifier administrator that is authorized for the administrative operation.

List of all the permissions see the "Formal Syntax" section of this document.

- o An <identifier:siteList> element that contains one or more <identifier:siteInfo> elements that provide information to locate

the site to implement provisions and resolution of the identifier. In this section, the element defines a handle service site by identifying the server computers that comprise the site along with their service configurations (e.g., port numbers).

Each <identifier:siteInfo> contains the following child elements:
An <identifier:siteIndex> element that indicates the specific index of a site.

An <identifier:protocolVersion> element that indicates handle protocol version used to create the handle identifier.

One or more <identifier:serviceInfo> elements that contain the following elements:

An <identifier:serverID> element defines the number of servers in the service site.

One or more <identifier:addr> elements that describe IP address of

the identifier service. Each <identifier:addr> element MAY contain an "ip" attribute to identify the IP address format.

Chen, et al.

Expires February 17,2021

[Page 16]

Attribute value "v4" is used to note IPv4 address format.

Attribute value "v6" is used to note IPv6 address format. If the "ip" attribute is not specified, "v4" is the default attribute value.

An <identifier:pubkey> element that contains the server's public key with a "type" attribute that specifies algorithms used to generate the public key. Public key in the <identifier:serviceInfo> can be used to authenticate any service response from the handle server.

One or more <identifier:serviceInterfaces> elements that have three child elements: an <identifier:serviceType> element that indicates whether the service is for query or for administration, an <identifier:protocol> element that specifies transmission protocol, where UDP and HTTP could be considered as alternative protocols, and the <identifier:port> element that represents service port of specific the service component.

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:xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
C:xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
C:  <command>
C:    <create>
C:      <identifier:create
C:        xmlns:identifier="urn:ietf:params:xml:ns:identifier-1.0"
C:        xsi:schemaLocation="urn:ietf:params:xml:ns:identifier-1.0
C:        identifier-1.0.xsd">
C:        <identifier:name>88.1000.1</identifier:name>
C:        <identifier:type>handle</identifier:type>
C:        <identifier:contact>jd1234</identifier:contact>
C:        <identifier:url>www.caict.ac.cn</identifier:url>
C:        <identifier:administratorList>
C:          <identifier:administrator>
C:            <identifier:adminIndex>100</identifier:adminIndex>
C:            <identifier:pubkey type="dsa_pub_key">
C:              AAAAB3NzaC1yc2EAAAADAQABAAQACprNl4N4
C:              e175lVnv03QfwYFTfB05hhLDC1...</identifier:pubkey>
C:            <identifier:permissionList>
C:              <identifier:permission>add_handle
C:              </identifier:permission>
C:              <identifier:permission>delete_handle
C:              </identifier:permission>
C:              <identifier:permission>add_value
```

```

C:      </identifier:permission>
C:      <identifier:permission>modify_admin
C:      </identifier:permission>
C:      <identifier:permission>remove_admin
C:      </identifier:permission>
C:      </identifier:permissionList>
C:      </identifier:administrator>
C:      </identifier:administratorList>
C:      <identifier:siteList>
C:      <identifier:siteInfo>
C:      <identifier:siteIndex>500</identifier:siteIndex>
C:      <identifier:protocolVersion>2.10
C:      </identifier:protocolVersion>
C:      <identifier:serviceInfo>
C:      <identifier:serverID>1</identifier:serverID>
C:      <identifier:addr ip="v4">192.0.2.2</identifier:addr>
C:      <identifier:addr ip="v6">1080:0:0:0:8:800:200C:417A
C:      </identifier:addr>
C:      <identifier:pubkey type="dsa_pub_key">
C:      AAAAB3NzaC1yc2EAAAADAQABAAQACprNl4N4e
C:      175lVnv03QfwYFTfB05hhLDC1...</identifier:pubkey>
C:      <identifier:serviceInterfaces>
C:      <identifier:serviceType>query
C:      </identifier:serviceType>
C:      <identifier:protocol>tcp</identifier:protocol>
C:      <identifier:port>2641</identifier:port>
C:      </identifier:serviceInterfaces>
C:      </identifier:serviceInfo>
C:      </identifier:siteInfo>
C:      </identifier:siteList>
C:      </identifier:create>
C:      </create>
C:      <clTRID>ABC-12345</clTRID>
C: </command>
C:</epp>

```

When a <create> command has been processed successfully, the EPP <response> element MUST contain a child <result code> element that identifies the result of processing.

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: <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.

3.2.2. EPP <delete> Command

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

- o An <identifier:name> element that contains the fully qualified name of the identifier object to be deleted.

Example <delete> command:

```
C: <?xml version="1.0" encoding="utf-8"?>
C: <epp xmlns="urn:ietf:params:xml:ns:epp-1.0" xmlns:xsi=
C: "http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation=
C: "urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
C:   <command>
C:     <delete>
C:       <identifier:delete xmlns:identifier="urn:ietf:params:
C:       xml:ns:identifier-1.0" xsi:schemaLocation="urn:ietf:
C:       params:xml:ns:identifier-1.0 identifier-1.0.xsd">
C:         <identifier:name>88.1000.1</identifier:name>
C:       </identifier: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

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

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.

3.2.3. EPP <renew> Command

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

3.2.4. EPP <transfer> Command

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

3.2.5. EPP <update> Command

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

- o An <identifier:name> element that contains the fully qualified name of the identifier object to be updated.
- o An OPTIONAL <identifier:add> element that contains attribute values to be added to the identifier object.
- o An OPTIONAL <identifier:rem> element that contains attribute values to be removed from the object. It has the following child elements: An OPTIONAL <identifier:contact> element that contains contact information that is to be removed from the identifier.

An optional `<identifier:url>` element that contains the URL to be removed. An OPTIONAL `<identifier:adminIndex>` element that specifies the index of the identifier administrator to be deleted. An OPTIONAL `<identifier:siteIndex>` element that contains information about index of the site to be removed from the identifier object. At least one child element of MUST be provided if the `<identifier:rem>` element is present.

- o An OPTIONAL `<identifier:chg>` element that contains object attribute values to be changed. The name of an identifier MUST NOT be changed, due to impacts on associated identifier objects.

At least one `<identifier:add>`, `<identifier:rem>`, or `<identifier: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 `<identifier:add>` and `<identifier:chg>` elements share two common child elements: `<identifier:administrator>` and the `<identifier:siteInfo>` element.

The `<identifier:add>` element has two additional child elements: `<identifier:contact>` and `<identifier:url>` other than the common element.

Whereas the `<identifier:chg>` has an additional `<identifier:status>` element that specifies status of the identifier object. Description of the common child elements of `<identifier:add>` and `<identifier:chg>` goes as follows:

- An `<identifier:administrator>` element that specifies administrator information of the identifier object. Identifier administrators are entitled to administrate or resolve identifier or identifier values according to the permission defined by its `<identifier:permissionList>` sub-element. An `<identifier:administrator>` element includes the following

child elements:

An `<identifier:adminIndex>` element that provides the reference to the authentication key that can be used to authenticate the administrator.

An `<identifier:pubkey>` element that contains the authentication key of the administrator and information of the type of the technique used to authenticate administrator. The public key is processed with base64 encoding schemes.

Three types of algorithms are recommended to authenticate the identifier administrator: Digital Signature Algorithm (DSA) public-key cryptography, Rivest-Shamir-Adleman(RSA) public-key cryptography, or the password-based authentication mechanism.

An <identifier:permissionList> element MAY contain zero or more
<identifier:permission> elements that specify information about
Chen, et al. Expires February 17,2021 [Page 21]

the administration authority of the administrator. A set of administration functions that include adding, deleting, and modifying identifier or identifier values are supported by the identifier service. Before fulfilling any administration request, the server must authenticate the client as the identifier administrator that is authorized for the administrative operation.

Lists of all the permissions see the "Formal Syntax" section of this document.

- An <identifier:siteInfo> element that provides information to locate the site to implement provisions and resolution of the identifier. The <identifier:siteInfo> element defines a handle service site by identifying the server computers that comprise the site along with their service configurations (e.g., port numbers). It contains the following child elements:

An <identifier:siteIndex> element that indicates the specific index of a site that is added or modified.

An <identifier:protocolVersion> element that indicates handle protocol version used to create the handle identifier.

One or more <identifier:serviceInfo> elements that contain the following elements: An <identifier:serverID> element defines the number of servers in the service site. One or more <identifier:addr> elements that describe IP address of the identifier service. Each <identifier:addr> element MAY contain an "ip" attribute to identify the IP address format. Attribute value "v4" is used to note IPv4 address format. Attribute value "v6" is used to note IPv6 address format. If the "ip" attribute is not specified, "v4" is the default attribute value. An <identifier:pubkey> element that contains the server's public key with a "type" attribute that specifies algorithms used to generate the public key. Public key in the <identifier:serviceInfo> can be used to authenticate any service response from the handle server. One or more <identifier:serviceInterfaces> elements that have three child elements: an <identifier:serviceType> element that indicates whether the service is for query or for administration, an <identifier:protocol> element that specifies transmission protocol, where UDP and HTTP could be considered as alternative protocols, and the <identifier:port> element that represents service port of specific the service component.

Example <update> command:

```
C:<?xml version="1.0" encoding="utf-8"?>
```

C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"

C:xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

Chen, et al.

Expires February 17,2021

[Page 22]

```
C: xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
C:   <command>
C:     <update>
C:       <identifier:update
C:         xmlns:identifier="urn:ietf:params:xml:ns:identifier-1.0"
C:         xsi:schemaLocation="urn:ietf:params:xml:ns:identifier-1.0
C:           identifier-1.0.xsd">
C:           <identifier:name>88.1000.1</identifier:name>
C:           <identifier:add>
C:             <identifier:contact>jd12345</identifier:contact>
C:             <identifier:url>www.abc.com</identifier:url>
C:             <identifier:administrator>
C:               <identifier:adminIndex>101</identifier:adminIndex>
C:               <identifier:pubkey type="dsa_pub_key">
C:                 AAAAB3NzaC1yc2EAAAADAQABAAQACprNl4N4e175lVnv03Qf
C:                 wYFTfB05hhLDC1...</identifier:pubkey>
C:               <identifier:permissionList>
C:                 <identifier:permission>add_handle
C:                 </identifier:permission>
C:                 <identifier:permission>delete_handle
C:                 </identifier:permission>
C:                 <identifier:permission>add_value
C:                 </identifier:permission>
C:                 <identifier:permission>modify_admin
C:                 </identifier:permission>
C:                 <identifier:permission>remove_admin
C:                 </identifier:permission>
C:               </identifier:permissionList>
C:             </identifier:administrator>
C:           <identifier:siteInfo>
C:             <identifier:siteIndex>501</identifier:siteIndex>
C:             <identifier:protocolVersion>2.10
C:             </identifier:protocolVersion>
C:           <identifier:serviceInfo>
C:             <identifier:serverID>1</identifier:serverID>
C:             <identifier:addr ip="v4">192.0.2.2</identifier:addr>
C:             <identifier:addr ip="v6">1080:0:0:0:8:800:200C:417A
C:             </identifier:addr>
C:             <identifier:pubkey type="dsa_pub_key">
C:               AAAAB3NzaC1yc2EAAAADAQABAAQACprNl4N4e
C:               175lVnv03QfwYFTfB05hhLDC1...</identifier:pubkey>
C:             <identifier:serviceInterfaces>
C:               <identifier:serviceType>admin
C:               </identifier:serviceType>
C:             <identifier:protocol>tcp</identifier:protocol>
```

```
C:      <identifier:port>2641</identifier:port>
C:      </identifier:serviceInterfaces>
C:      </identifier:serviceInfo>
C:      </identifier:siteInfo>
C:    </identifier:add>
C:    <identifier:rem>
C:      <identifier:contact>jd12345</identifier:contact>
C:      <identifier:url>www.abc.com</identifier:url>
C:      <identifier:adminIndex>101</identifier:adminIndex>
C:      <identifier:siteIndex>500</identifier:siteIndex>
C:    </identifier:rem>
C:    <identifier:chg>
C:      <identifier:status s="clientUpdateProhibited"/>
C:      <identifier:administrator>
C:        <identifier:adminIndex>102</identifier:adminIndex>
C:        <identifier:pubkey type="dsa_pub_key">
C:          AAAAB3NzaC1yc2EAAAADAQABAAQACprNl4N4e175lVnv03Qf
C:          wYFTfB05hhLDC1...</identifier:pubkey>
C:        <identifier:permissionList>
C:          <identifier:permission>add_handle
C:          </identifier:permission>
C:          <identifier:permission>delete_handle
C:          </identifier:permission>
C:          <identifier:permission>add_value
C:          </identifier:permission>
C:        </identifier:permissionList>
C:      </identifier:administrator>
C:    <identifier:siteInfo>
C:      <identifier:siteIndex>500</identifier:siteIndex>
C:      <identifier:protocolVersion>2.10
C:      </identifier:protocolVersion>
C:    <identifier:serviceInfo>
C:      <identifier:serverID>2</identifier:serverID>
C:      <identifier:addr ip="v4">192.0.2.2</identifier:addr>
C:      <identifier:addr ip="v6">1080:0:0:0:8:800:200C:417A
C:      </identifier:addr>
C:    <identifier:serviceInterfaces>
C:      <identifier:serviceType>query
C:      </identifier:serviceType>
C:      <identifier:protocol>tcp</identifier:protocol>
C:      <identifier:port>2641</identifier:port>
C:    </identifier:serviceInterfaces>
C:  </identifier:serviceInfo>
C: </identifier:siteInfo>
C: </identifier:chg>
```

```
C:      </identifier: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"?>
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 could not be processed for any reason.

4. 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.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- o Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- o Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- o Neither the name of Internet Society, IETF or IETF Trust, nor the names of specific contributors, may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

BEGIN

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

<schema xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:identifier="urn:ietf:params:xml:ns:identifier-1.0"
  xmlns:epp="urn:ietf:params:xml:ns:epp-1.0"
  xmlns:eppcom="urn:ietf:params:xml:ns:eppcom-1.0"
  targetNamespace="urn:ietf:params:xml:ns:identifier-1.0"
  elementFormDefault="qualified">
  <!--
    Import common element types.
  -->
  <import namespace="urn:ietf:params:xml:ns:eppcom-1.0"
    schemaLocation="eppcom-1.0.xsd"/>
  <import namespace="urn:ietf:params:xml:ns:epp-1.0"
    schemaLocation="epp-1.0.xsd"/>
  <annotation>
    <documentation>Extensible Provisioning Protocol v1.0
      identifier provisioning schema.</documentation>
  </annotation>
  <!--
    Child elements found in EPP commands.
  -->
  <element name="check" type="identifier:mNameType"/>
  <element name="create" type="identifier:createType"/>
  <element name="update" type="identifier:updateType"/>
  <element name="info" type="identifier:sNameType"/>
  <element name="delete" type="identifier:sNameType"/>
  <!--
```

Child elements of the <create> command.

-->

```
<complexType name="createType">
  <sequence>
    <element name="name" type="eppcom:labelType"/>
    <element name="type" type="identifier:typeEnumType"/>
    <element name="contact" type="identifier:contactType"
      maxOccurs="unbounded"/>
    <element name="url" type="anyURI" maxOccurs="unbounded"/>
    <element name="administratorList"
      type="identifier:administratorListType" minOccurs="0"/>
    <element name="siteList"
      type="identifier:siteListType" minOccurs="0"/>
  </sequence>
</complexType>
```

<!--

Child elements of the <delete> and <info> commands.

-->

```
<complexType name="sNameType">
  <sequence>
    <element name="name" type="eppcom:labelType"/>
  </sequence>
</complexType>
```

<!--

Child element of commands that accept multiple names.

-->

```
<complexType name="mNameType">
  <sequence>
    <element name="name" type="eppcom:labelType"
      maxOccurs="unbounded"/>
  </sequence>
</complexType>
<simpleType name="typeEnumType">
  <restriction base="token">
    <enumeration value="handle"/>
    <enumeration value="oid"/>
    <enumeration value="ecode"/>
    <enumeration value="epc"/>
    <enumeration value="other"/>
  </restriction>
</simpleType>
<complexType name="contactType">
  <simpleContent>
    <extension base="eppcom:clIDType">
      <attribute name="type" type="identifier:contactAttrType"/>
    </extension>
  </simpleContent>
</complexType>
```

```
</extension>
</simpleContent>
</complexType>
<simpleType name="contactAttrType">
  <restriction base="token">
    <enumeration value="admin"/>
    <enumeration value="billing"/>
    <enumeration value="tech"/>
  </restriction>
</simpleType>
<complexType name="administratorListType">
  <sequence>
    <element name="administrator"
      type="identifier:administratorType" minOccurs="0"
      maxOccurs="unbounded"/>
  </sequence>
</complexType>
<complexType name="administratorType">
  <sequence>
    <element name="adminIndex" type="unsignedInt"/>
    <element name="pubkey" type="identifier:pubkeyType"/>
    <element name="permissionList" type="identifier:
      permissionListType"/>
  </sequence>
</complexType>
<complexType name="pubkeyType">
  <simpleContent>
    <extension base="base64Binary">
      <attribute name="type" type="identifier:
        pubkeyTypeType"/>
    </extension>
  </simpleContent>
</complexType>
<simpleType name="pubkeyTypeType">
  <restriction base="token">
    <enumeration value="dsa_pub_key"/>
    <enumeration value="rsa_pub_key"/>
    <enumeration value="secret_key"/>
  </restriction>
</simpleType>
<complexType name="permissionListType">
  <sequence>
    <element name="permission" type="identifier:permissionType"
      minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
```

```
</complexType>
<simpleType name="permissionType">
  <restriction base="token">
    <enumeration value="add_handle"/>
    <enumeration value="delete_handle"/>
    <enumeration value="add_na"/>
    <enumeration value="delete_na"/>
    <enumeration value="modify_value"/>
    <enumeration value="delete_value"/>
    <enumeration value="add_value"/>
    <enumeration value="modify_admin"/>
    <enumeration value="remove_admin"/>
    <enumeration value="add_admin"/>
    <enumeration value="authorized_read"/>
    <enumeration value="list_handle"/>
  </restriction>
</simpleType>
<complexType name="siteListType">
  <sequence>
    <element name="siteInfo" type="identifier:siteInfoType"
      minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
</complexType>
<complexType name="siteInfoType">
  <sequence>
    <element name="siteIndex" type="unsignedInt"/>
    <element name="protocolVersion" type="token"/>
    <element name="serviceInfo" type="identifier:serviceInfoType"
      minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
</complexType>
<complexType name="serviceInfoType">
  <sequence>
    <element name="serverID" type="unsignedInt"/>
    <element name="addr" type="identifier:addrType" minOccurs="0"
      maxOccurs="unbounded"/>
    <element name="pubkey" type="identifier:pubkeyType"
      minOccurs="0" maxOccurs="1"/>
    <element name="serviceInterfaces"
      type="identifier:serviceInterfacesType"
      minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
</complexType>
<complexType name="addrType">
  <simpleContent>
```

```
<extension base="identifier:addrStringType">
  <attribute name="ip" type="identifier:ipType" default="v4"/>
</extension>
</simpleContent>
</complexType>
<simpleType name="addrStringType">
  <restriction base="token">
    <minLength value="3"/>
    <maxLength value="45"/>
  </restriction>
</simpleType>
<simpleType name="ipType">
  <restriction base="token">
    <enumeration value="v4"/>
    <enumeration value="v6"/>
  </restriction>
</simpleType>
<complexType name="serviceInterfacesType">
  <sequence>
    <element name="serviceType"
type="identifier:serviceTypeType"/>
    <element name="protocol" type="identifier:protocolType"/>
    <element name="port" type="unsignedShort"/>
  </sequence>
</complexType>
<simpleType name="serviceTypeType">
  <restriction base="token">
    <enumeration value="query"/>
    <enumeration value="admin"/>
  </restriction>
</simpleType>
<simpleType name="protocolType">
  <restriction base="token">
    <enumeration value="tcp"/>
    <enumeration value="udp"/>
    <enumeration value="http"/>
  </restriction>
</simpleType>
<!--
  Child elements of the <update> command.
-->
<complexType name="updateType">
  <sequence>
    <element name="name" type="eppcom:labelType"/>
    <element name="add" type="identifier:addType" minOccurs="0"/>
```

```
<element name="rem" type="identifier:remType" minOccurs="0"/>
<element name="chg" type="identifier:chgType" minOccurs="0"/>
</sequence>
</complexType>
<complexType name="addType">
  <sequence>
    <element name="contact" type="identifier:contactType"
      minOccurs="0" maxOccurs="unbounded"/>
    <element name="url" type="eppcom:labelType" minOccurs="0"
      maxOccurs="unbounded"/>
    <element name="administrator"
      type="identifier:administratorType"
      minOccurs="0" maxOccurs="unbounded"/>
    <element name="siteInfo" type="identifier:siteInfoType"
      minOccurs="0" maxOccurs="unbounded"/>
    <element name="cert" type="token" minOccurs="0"
      maxOccurs="1"/>
    <element name="signature" type="token" minOccurs="0"
      maxOccurs="1"/>
  </sequence>
</complexType>
<complexType name="remType">
  <sequence>
    <element name="contact" type="identifier:contactType"
      minOccurs="0" maxOccurs="unbounded"/>
    <element name="url" type="eppcom:labelType" minOccurs="0"
      maxOccurs="unbounded"/>
    <element name="adminIndex" type="unsignedInt" minOccurs="0"
      maxOccurs="unbounded"/>
    <element name="siteIndex" type="unsignedInt" minOccurs="0"
      maxOccurs="unbounded"/>
  </sequence>
</complexType>
<complexType name="chgType">
  <sequence>
    <element name="status" type="identifier:statusType"
      minOccurs="0"/>
    <element name="administrator"
      type="identifier:administratorType" minOccurs="0"
      maxOccurs="unbounded"/>
    <element name="siteInfo" type="identifier:siteInfoType"
      minOccurs="0" maxOccurs="unbounded"/>
    <element name="cert" type="token" minOccurs="0"
      maxOccurs="1"/>
    <element name="signature" type="token" minOccurs="0"/>
```

```
      maxOccurs="1"/>
    </sequence>
  </complexType>
<!--
  Status is a combination of attributes and an optional
  human-readable message that may be expressed in languages other
  than English.
-->
<complexType name="statusType">
  <simpleContent>
    <extension base="normalizedString">
      <attribute name="s" type="identifier:statusValueType"
        use="required"/>
      <attribute name="lang" type="language" default="en"/>
    </extension>
  </simpleContent>
</complexType>
<simpleType name="statusValueType">
  <restriction base="token">
    <enumeration value="clientDeleteProhibited"/>
    <enumeration value="clientHold"/>
    <enumeration value="clientRenewProhibited"/>
    <enumeration value="clientTransferProhibited"/>
    <enumeration value="clientUpdateProhibited"/>
    <enumeration value="inactive"/>
    <enumeration value="ok"/>
    <enumeration value="pendingCreate"/>
    <enumeration value="pendingDelete"/>
    <enumeration value="pendingRenew"/>
    <enumeration value="pendingTransfer"/>
    <enumeration value="pendingUpdate"/>
    <enumeration value="serverDeleteProhibited"/>
    <enumeration value="serverHold"/>
    <enumeration value="serverRenewProhibited"/>
    <enumeration value="serverTransferProhibited"/>
    <enumeration value="serverUpdateProhibited"/>
  </restriction>
</simpleType>
<!--
  Child response elements.
-->
<element name="chkData" type="identifier:chkDataType"/>
<element name="infData" type="identifier:infDataType"/>
<!--
  <check> response elements.
```

```

-->
<complexType name="chkDataType">
  <sequence>
    <element name="cd" type="identifier:checkType"
      maxOccurs="unbounded"/>
  </sequence>
</complexType>
<complexType name="checkType">
  <sequence>
    <element name="name" type="identifier:checkNameType"/>
    <element name="reason" type="eppcom:reasonType"
minOccurs="0"/>
  </sequence>
</complexType>
<complexType name="checkNameType">
  <simpleContent>
    <extension base="eppcom:labelType">
      <attribute name="avail" type="boolean" use="required"/>
    </extension>
  </simpleContent>
</complexType>
<complexType name="infDataType">
  <sequence>
    <element name="name" type="eppcom:labelType"/>
    <element name="type" type="identifier:typeEnumType"/>
    <element name="status" type="identifier:statusType"/>
    <element name="contact" type="identifier:contactType"
      maxOccurs="unbounded"/>
    <element name="url" type="anyURI" maxOccurs="unbounded"/>
    <element name="administratorList"
      type="identifier:administratorListType" minOccurs="0"/>
    <element name="siteList" type="identifier:siteListType"
      minOccurs="0"/>
  </sequence>
</complexType>
<!--
  End of schema.
-->
</schema>
END

```

5. 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 [[RFC2781](#)]. 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 in environments where parser encoding support incompatibility exists.

All date-time values presented via EPP MUST be expressed in Universal Coordinated Time using the Gregorian calendar. XML Schema allows use of time zone identifiers to indicate offsets from the zero meridian, but this option MUST NOT be used with EPP. 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.

The syntax for handle identifiers described in this document MUST conform to [[RFC3650](#)], [[RFC3651](#)], [[RFC3652](#)]. The conformance requirements might change as a result of progressing work in developing standards for internationalized identifier techniques.

6. Security Considerations

Authorization information as described in [Section 3.2](#) is REQUIRED to create an identifier object. This information is used in some query and transfer operations as an additional means of determining client authorization to perform the command. Failure to protect authorization information from inadvertent disclosure can result in unauthorized transfer operations and unauthorized information release. Both client and server MUST ensure that authorization information is stored and exchanged with high-grade encryption mechanisms to provide privacy services.

The object mapping 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.

7. IANA Considerations

This document uses URNs to describe XML namespaces and XML schemas conforming to a registry mechanism described in [[RFC3688](#)]. Two URI assignments have been registered by the IANA.

Registration request for the identifier namespace:

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

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

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

Registration request for the identifier XML schema:

Chen, et al.

Expires February 17, 2021

[Page 34]

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

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

8. Acknowledgments

This document is based on an identifier application of EPP. Thus, the authors would like to thank J. Xie who suggested improvements and provided many invaluable comments. This document are individual submissions, based on the work done in [RFC 5730](#).

This document was prepared using 2-Word-v2.0.template.dot.

9. References

9.1. Normative References

- [W3C.REC-xml-20040204] Sperberg-McQueen, C., Maler, E., Yergeau, F., Paoli, J., and T. Bray, "Extensible Markup Language (XML) 1.0 (Third Edition)", World Wide Web Consortium First Edition REC-xml-20040204, February 2004, <<http://www.w3.org/TR/2004/REC-xml-20040204>>.
- [W3C.REC-xmlschema-1-20041028] Maloney, M., Thompson, H., Mendelsohn, N., and D. Beech, "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] Malhotra, A. and P. Biron, "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>>.
- [[RFC0791](#)] Postel, J., "Internet Protocol", STD 5, [RFC 791](#), September 1981.
- [RFC5730] Hollenbeck, S., "Extensible Provisioning Protocol (EPP)", STD 69, [RFC 5730](#), August 2009.
- [RFC3650] Sun, S. and L. Lannom, "Handle System Overview", November 2003.
- [RFC3651] Sun, S., Reilly, S. and L. Lannom, "Handle System Namespace and Service Definition", November 2003.
- [RFC3652] Sun, S., Reilly, S. and L. Lannom, "Handle System Protocol (ver 2.1) Specification", November 2003.

9.2. Informative References

- [RFC1558] T. Howes, "A String Representation of LDAP Search Filters", [RFC 1558](#), December 1993.
- [RFC2781] Hoffman, P. and F. Yergeau, "UTF-16, an encoding of ISO 10646", [RFC 2781](#), February 2000.
- [RFC2874] Crawford, M. and C. Huitema, "DNS Extensions to Support IPv6 Address Aggregation and Renumbering", [RFC 2874](#), July 2000.
- [RFC3596] Thomson, S., Huitema, C., Ksinant, V., and M. Souissi, "DNS Extensions to Support IP Version 6", [RFC 3596](#), October 2003.
- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", [RFC 4291](#), February 2006.

Author's Address

Yuying Chen
CAICT
No.52 Huayuan North Road, Haidian District
Beijing, Beijing, 100191
China

Phone: +86 188 1008 2358
Email: chenyyuying@caict.ac.cn

Jiagui Xie
CAICT
No.52 Huayuan North Road, Haidian District
Beijing, Beijing, 100191
China

Phone: +86 150 0138 5070
Email: xiejiagui@caict.ac.cn

Zhiping Li
CAICT
No.52 Huayuan North Road, Haidian District
Beijing, Beijing, 100191
China

Phone: +86 185 1107 1386
Email: lizhiping@caict.ac.cn

Zhipeng Fan
CAICT
No.52 Huayuan North Road, Haidian District
Beijing, Beijing, 100191
China

Phone: +86 159 1112 3285
Email: fanzhipeng@caict.ac.cn