Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

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.

Abstract

This document describes an Extensible Provisioning Protocol (EPP) mapping for the provisioning and management of Internet domain names stored in a shared central repository. Specified in XML, the mapping defines EPP command syntax and semantics as applied to domain names.

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 in examples is provided only to illustrate element relationships and is not a REQUIRED feature of this protocol.

XML protocol elements are case sensitive.
1. Introduction

This document describes an internet domain name 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 [XML] and XML Schema notation as described in [XML-SD] and [XML-SS].

The referenced XML Schema documents recently progressed from Working Draft status to Candidate Recommendation status. The references to these documents and the URIs used to refer to XML Schema namespaces MUST be changed once XML parsers that support the updated specifications are available.

[EPP] 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.

It is important to note that XML is case sensitive. XML specifications and examples provided in this document MUST be interpreted in the exact character case presented to develop a conforming implementation.

This document is being discussed on the "rrp" mailing list. To join the list, send a message to <majordomo@NSIRegistry.net> with the words "subscribe rrp" in the body of the message. There is a web site for the list archives at <http://www.NSIRegistry.net/maillist/rrp>. 
2. Object Attributes

An EPP domain object has attributes and associated values that may be viewed and modified by the sponsoring client or the server. This section describes each attribute type in detail.

2.1 Domain Name and Host Names

The syntax for domain and host names described in this document MUST conform to [RFC952] as updated by [RFC1123]. These conformance requirements MAY change as a result of progressing work in developing standards for internationalized domain names. A server MAY restrict allowable domain names to a particular top level domain, second level domain, or other domain for which the server is authoritative.

2.2 Client Identifiers

All EPP clients are identified by a server-unique identifier. Client identifiers use the contact identifier syntax described in [EPP-C].

2.3 Status Values

A domain name always has at least one associated status indicator. Status indicators MAY be set only by the client that sponsors a known domain object and by the server on which the object resides. A client MAY change the status of a domain object using the EPP <update> command.

Client-Managed Status Values:

CLIENT-HOLD

The domain MUST NOT be published in a zone for DNS resolution.

CLIENT-LOCK

The domain MUST NOT be modified through any action of the EPP <create>, <transfer>, or <update> commands, though the <update> command MAY be used to change the status value. The <renew> command MAY be applied to avoid exceeding the end of the domain validity period. Child objects MUST NOT be added or removed.

Server-Managed Status Values:

NEW

The domain has been delegated, can be modified, and has not been published in a zone.
ACTIVE
The domain has been delegated, can be modified, and appears in a zone.

INACTIVE
The domain has not been delegated, can be modified, and does not appear in a zone.

HOLD
The domain MUST NOT be published in a zone for DNS resolution.

LOCK
The domain MUST NOT be modified through any action of the EPP <create>, <transfer>, or <update> commands, though the <update> command MAY be used to change the status value. The <renew> command MAY be applied to avoid exceeding the end of the domain validity period. Child objects MUST NOT be added or removed.

PENDING-TRANSFER
A transfer request has been received for the domain, and processing of the request is pending.

PENDING-DELETE
A delete request has been received for this domain. The domain has been removed from the zone, but has not yet been purged from the server database.

2.4 Domain Contacts
The syntax for domain contact identifiers is described in [EPP-C]. A server MAY support a contact object service to allow contact objects to be associated with domain objects.

2.5 Dates and Times
Date and time attribute values MUST be represented in Universal Coordinated Time (UTC). Both extended and truncated date and time forms defined in [ISO8601] MAY be used.

Every domain name has an associated validity period. The validity period is determined when a domain object is created, and it MAY be extended by the action of the EPP <renew> and <transfer> commands. If the end of the validity period is reached without explicit client
action to extend the period, a server MAY extend the period automatically for one additional year or it MAY place the domain on HOLD status. A domain name MUST NOT be deleted automatically upon expiration of the validity period.

2.6 Authorization Identifiers

Authorization identifiers are associated with domain name objects to facilitate authorization of transfer requests. Authorization identifiers use the transaction identifier syntax described in [EPP].
3. EPP Command Mapping

A detailed description of the EPP syntax and semantics can be found in [EPP]. The command mappings described here are specifically for use in provisioning and managing internet domain names via EPP.

3.1 EPP Query Commands

EPP provides three commands to retrieve domain information: <info> to retrieve detailed information associated with a domain, <ping> to determine if a domain is known to the server, and <transfer> to retrieve domain transfer status information.

3.1.1 EPP <info> Command

The EPP <info> command is used to retrieve information associated with a domain. In addition to the standard EPP command elements, the <info> command MUST contain a <domain:info> element that identifies the domain namespace and the location of the domain schema. The <domain:info> element MUST contain the following child elements:

- A <domain:name> element that contains the fully qualified domain name for which information is requested.

Example <info> command:

```
C:<?xml version="1.0" standalone="no"?>
C:<epp xmlns="urn:iana:xmlns:epp"
     xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
     xsi:schemaLocation="urn:iana:xmlns:epp epp.xsd">
C:  <command>
C:    <info>
C:      <domain:info xmlns:domain="urn:iana:xmlns:domain"
       xsi:schemaLocation="urn:iana:xmlns:domain domain.xsd">
C:        <domain:name>example.com</domain:name>
C:      </domain:info>
C:    </info>
C:    <trans-id>
C:      <date>2000-06-08</date>
C:      <client-id>ClientX</client-id>
C:      <code>ABC-12345-XYZ</code>
C:    </trans-id>
C:  </command>
C:</epp>
```

If the domain is known to the server, the EPP <response-data> element MUST contain a child <domain:info-data> element that identifies the domain namespace and the location of the domain schema. The
<domain:info-data> element SHALL contain the following child elements:

- A <domain:name> element that contains the fully qualified name of the domain.

- A <domain:client-id> element that contains the identifier of the sponsoring client.

- One or more <domain:status> elements that contain the current status descriptors associated with the domain. See the <update> command description for a list of valid status values.

- If supported by the server, one or more <domain:contact> elements that contain identifiers for the registrant, administrative, technical, and billing contacts.

- Zero or more <domain:server> elements that contain the fully qualified names of the name servers to which the domain is delegated.

- Zero or more <domain:child-server> elements that contain the fully qualified names of the child name servers known under this parent domain.

- A <domain:created-by> element that contains the identifier of the client that created the domain name.

- A <domain:created-date> element that contains the date and time of domain creation.

- A <domain:expiration-date> element that contains the date and time identifying the end of the domain’s registration period.

- A <domain:last-updated-by> element that contains the identifier of the client that last updated the domain name. This element MUST NOT be present if the domain has never been modified.

- A <domain:last-updated-date> element that contains the date and time of the most recent domain modification. This element MUST NOT be present if the domain has never been modified.

- A <domain:last-transfer-date> elements that contains the date and time of the most recent successful transfer. This element MUST NOT be provided if the domain has never been transferred.

- A <domain:auth-id> elements derived from either the original creation transaction or the most recent successful transfer transaction. This element MUST NOT be provided if the querying client is not the current sponsoring client.
Example <info> response:

```xml
S:<?xml version="1.0" standalone="no"?>
S:<epp xmlns="urn:iana:xmlns:epp"
    xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
    xsi:schemaLocation="urn:iana:xmlns:epp epp.xsd">
S:  <response>
S:    <result code="1000">
S:      <text>Command completed successfully</text>
S:    </result>
S:    <response-data>
S:      <domain:info-data xmlns:domain="urn:iana:xmlns:domain"
        xsi:schemaLocation="urn:iana:xmlns:domain domain.xsd">
S:        <domain:name>example.com</domain:name>
S:        <domain:client-id>ClientX</domain:client-id>
S:        <domain:status>ACTIVE</domain:status>
S:        <domain:contact type="registrant">SH0000</domain:contact>
S:        <domain:contact type="administrative">SH0000</domain:contact>
S:        <domain:contact type="technical">SH0000</domain:contact>
S:        <domain:contact type="billing">SH0000</domain:contact>
S:        <domain:server>ns1.example.com</domain:server>
S:        <domain:server>ns2.example.com</domain:server>
S:        <domain:child-server>ns1.example.com</domain:child-server>
S:        <domain:child-server>ns2.example.com</domain:child-server>
S:        <domain:created-by>ClientY</domain:created-by>
S:        <domain:created-date>1999-04-03T22:00:00.0Z</domain:created-date>
S:        <domain:expiration-date>2005-04-03T22:00:00.0Z</domain:expiration-date>
S:        <domain:last-updated-by>ClientX</domain:last-updated-by>
S:        <domain:last-updated-date>1999-12-03T09:00:00.0Z</domain:last-updated-date>
S:        <domain:last-transfer-date>2000-04-08T09:00:00.0Z</domain:last-transfer-date>
S:        <domain:auth-id>
S:          <domain:date>2000-04-08</domain:date>
S:          <domain:client-id>ClientX</domain:client-id>
S:          <domain:code>ABC-98765-XYZ</domain:code>
S:        </domain:auth-id>
S:      </domain:info-data>
S:    </response-data>
S:    <trans-id>
S:      <date>2000-06-08</date>
S:      <client-id>ClientX</client-id>
S:      <code>ABC-12345-XYZ</code>
S:    </trans-id>
S:  </response>
```
S:</epp>

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

3.1.2 EPP <ping> Command

The EPP <ping> command is used to determine if a domain name is known to the server. In addition to the standard EPP command elements, the <ping> command MUST contain a <domain:ping> element that identifies the domain namespace and the location of the domain schema. The <domain:ping> element MUST contain the following child elements:

- One or more (up to a maximum of sixteen) <domain:name> elements that contain the fully qualified name of the queried domains.

Example <ping> command:

C:<?xml version="1.0" standalone="no"?>
C:<epp xmlns="urn:iana:xmlns:epp"
C:     xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
C:     xsi:schemaLocation="urn:iana:xmlns:epp epp.xsd">
C:  <command>
C:    <ping>
C:      <domain:ping xmlns:domain="urn:iana:xmlns:domain"
C:       xsi:schemaLocation="urn:iana:xmlns:domain domain.xsd">
C:        <domain:name>example1.com</domain:name>
C:        <domain:name>example2.com</domain:name>
C:        <domain:name>example3.com</domain:name>
C:      </domain:ping>
C:    </ping>
C:    <trans-id>
C:      <date>2000-06-08</date>
C:      <client-id>ClientX</client-id>
C:      <code>ABC-12345-XYZ</code>
C:    </trans-id>
C:  </command>
C:</epp>

When a <ping> command has been processed successfully, the EPP <response-data> element MUST contain a child <domain:ping-data> element that identifies the domain namespace and the location of the domain schema. The <domain:ping-data> element SHALL contain the following child elements:

- One or more (up to a maximum of sixteen) <domain:name> elements that contain the fully qualified name of the queried domains and a "result" attribute whose value identifies the object as either "known" or
Example <ping> response:

S: <?xml version="1.0" standalone="no"?>
S: <epp xmlns="urn:iana:xmlns:epp"
S:     xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
S:     xsi:schemaLocation="urn:iana:xmlns:epp epp.xsd">
S:  <response>
S:    <result code="1000">
S:      <text>Command completed successfully</text>
S:    </result>
S:    <response-data>
S:      <domain:ping-data xmlns:domain="urn:iana:xmlns:domain"
S:       xsi:schemaLocation="urn:iana:xmlns:domain domain.xsd">
S:        <domain:name result="known">example1.com</domain:name>
S:        <domain:name result="unknown">example2.com</domain:name>
S:        <domain:name result="known">example3.com</domain:name>
S:      </domain:ping-data>
S:    </response-data>
S:    <trans-id>
S:      <date>2000-06-08</date>
S:      <client-id>ClientX</client-id>
S:      <code>ABC-12345-XYZ</code>
S:    </trans-id>
S:  </response>
S:</epp>

An EPP error response MUST be returned if a <ping> command could not be processed for any reason.

3.1.3 EPP <transfer> Command

The EPP <transfer> command provides a query operation that allows a client to determine real-time status of pending and completed transfer requests. In addition to the standard EPP command elements, the <transfer> command MUST contain an "op" attribute with value "query", and a <domain:transfer-query> element that identifies the domain namespace and the location of the domain schema. The <domain:transfer-query> element MUST contain the following child elements:

- A <domain:name> element that contains the fully qualified name of the domain.
Example <transfer> query command:

C:<?xml version="1.0" standalone="no"?>
C:<epp xmlns="urn:iana:xmlns:epp"
C:     xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
C:     xsi:schemaLocation="urn:iana:xmlns:epp epp.xsd">
C:  <command>
C:    <transfer op="query">
C:      <domain:transfer-query xmlns:domain="urn:iana:xmlns:domain"
C:       xsi:schemaLocation="urn:iana:xmlns:domain domain.xsd">
C:        <domain:name>example.com</domain:name>
C:      </domain:transfer-query>
C:      <auth-id>
C:        <date>1999-06-08</date>
C:        <client-id>ClientX</client-id>
C:        <code>ABC-98765-XYZ</code>
C:      </auth-id>
C:    </transfer>
C:    <trans-id>
C:      <date>2000-06-08</date>
C:      <client-id>ClientX</client-id>
C:      <code>ABC-12345-XYZ</code>
C:    </trans-id>
C:  </command>
C:</epp>

When a <transfer> command has been processed successfully, a server
MUST respond with an EPP <response-data> element that MUST contain a
child <domain:transfer-data> element that identifies the domain
namespace and the location of the domain schema. The
<domain:transfer-data> element SHALL contain the following child
elements:

- A <domain:name> element that contains the fully qualified domain
  name used in the query.

- A <domain:request-client> element that contains the identifier of
  the client that initiated the transfer request.

- A <domain:action-client> element that contains the identifier of the
  client that SHOULD respond to the transfer request.

- A <domain:transfer-status> element that contains the state of the
  most recent transfer request. Valid values are "PENDING", "APPROVED",
  "REJECTED", "AUTO-APPROVED", "AUTO-REJECTED", and "CANCELLED".

- A <domain:request-date> element that contains the date and time that
the transfer was requested.

- A `<domain:action-date>` element that contains the date and time of a required or completed response. For a PENDING request, the value identifies the date and time by which a response is required before an automated response action MUST be taken by the server. For all other status types, the value identifies the date and time when the request was completed.

- An OPTIONAL `<domain:expiration-date>` element that contains the end of the domain’s validity period if the `<transfer>` command caused or causes a change in the validity period.

Example `<transfer>` query response:

S: <?xml version="1.0" standalone="no"?><br>
S: <epp xmlns="urn:iana:xmlns:epp"
S:     xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
S:     xsi:schemaLocation="urn:iana:xmlns:epp epp.xsd">
S:  <response>
S:    <result code="1000">
S:      <text>Command completed successfully</text>
S:    </result>
S:    <response-data>
S:      <domain:transfer-data xmlns:domain="urn:iana:xmlns:domain"
S:       xsi:schemaLocation="urn:iana:xmlns:domain domain.xsd">
S:        <domain:name>example.com</domain:name>
S:        <domain:request-client>ClientX</domain:request-client>
S:        <domain:action-client>ClientY</domain:action-client>
S:        <domain:transfer-status>PENDING</domain:transfer-status>
S:        <domain:request-date>
S:          2000-06-06T22:00:00.0Z
S:        </domain:request-date>
S:        <domain:action-date>
S:          2000-06-11T22:00:00.0Z
S:        </domain:action-date>
S:        <domain:expiration-date>
S:          2002-09-08T22:00:00.0Z
S:        </domain:expiration-date>
S:      </domain:transfer-data>
S:    </response-data>
S:    <trans-id>
S:      <date>2000-06-08</date>
S:      <client-id>ClientX</client-id>
S:      <code>ABC-12345-XYZ</code>
S:    </trans-id>
S:  </response>
S:</epp>
An EPP error response MUST be returned if a <transfer> query command could not be processed for any reason.

3.2 EPP Transform Commands

EPP provides five commands to transform domain object information: <create> to create an instance of a domain object, <delete> to delete an instance of a domain object, <renew> to extend the validity period of a domain object, <transfer> to manage domain object sponsorship changes, and <update> to change information associated with a domain object.

3.2.1 EPP <create> Command

The EPP <create> command provides a transform operation that allows a client to create a domain object. In addition to the standard EPP command elements, the <create> command MUST contain a <domain:create> element that identifies the domain namespace and the location of the domain schema. The <domain:create> element MUST contain the following child elements:

- A <domain:name> element that contains the fully qualified domain name of the object to be created.

- An OPTIONAL <domain:period> element that contains the initial registration period of the domain, measured in years. If not specified, the initial registration period MUST be one year. The minimum allowable value is one (1) year. The maximum allowable value is ninety-nine (99) years. A server MAY support a lower maximum value.

- Zero or more (up to a maximum of thirteen) <domain:server> elements that contain the fully qualified host name of a known host object to provide resolution services for the domain. A host object MUST be known to the server before a domain can be delegated to the host. A server MUST provide host object services to provide domain name services. The EPP mapping for host objects is described in [EPP-H].

- Zero or more (up to a maximum of four) <domain:contact> elements that contain the registrant, administrative, technical, and billing contact identifiers to be associated with the domain. A contact identifier MUST be known to the server before the contact can be associated with the domain. Only one contact identifier of each type MAY be specified. A server MAY provide contact object services when providing domain name object services. The EPP mapping for contact objects is described in [EPP-C].

It is important to note that the transaction identifier associated
with successful creation of a domain object becomes the authorization identifier required to transfer sponsorship of the domain object. A client MUST retain all transaction identifiers associated with domain object creation and protect them from disclosure. A client MUST also provide a copy of the transaction identifier information to the domain registrant, who will need this information to request a domain transfer through a different client.

Example <create> command:

```xml
<?xml version="1.0" standalone="no"?>
<epp xmlns="urn:iana:xmlns:epp"
     xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
     xsi:schemaLocation="urn:iana:xmlns:epp epp.xsd">
  <command>
    <create>
      <domain:create xmlns:domain="urn:iana:xmlns:domain"
                      xsi:schemaLocation="urn:iana:xmlns:domain domain.xsd">
        <domain:name>example.com</domain:name>
        <domain:period>2</domain:period>
        <domain:server>ns1.example.com</domain:server>
        <domain:server>ns2.example.com</domain:server>
        <domain:contact type="registrant">SH0000</domain:contact>
        <domain:contact type="administrative">SH0000</domain:contact>
        <domain:contact type="technical">SH0000</domain:contact>
        <domain:contact type="billing">SH0000</domain:contact>
      </domain:create>
    </create>
    <trans-id>
      <date>2000-06-08</date>
      <client-id>ClientX</client-id>
      <code>ABC-12345-XYZ</code>
    </trans-id>
  </command>
</epp>
```

When a <create> command has been processed successfully, a server MUST respond with an EPP <response-data> element that MUST contain a child <domain:expire-data> element that identifies the domain namespace and the location of the domain schema. The <domain:expire-data> element SHALL contain the following child elements:

- A <domain:name> element that contains the fully qualified domain name that has been created.
- A <domain:expiration-date> element that contains the end of the domain’s validity period.
Example <create> response:

S:<?xml version="1.0" standalone="no"?>
S:<epp xmlns="urn:iana:xmlns:epp"
S:     xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
S:     xsi:schemaLocation="urn:iana:xmlns:epp epp.xsd">
S:  <response>
S:    <result code="1000">
S:      <text>Command completed successfully</text>
S:    </result>
S:    <response-data>
S:      <domain:expire-data xmlns:domain="urn:iana:xmlns:domain"
S:       xsi:schemaLocation="urn:iana:xmlns:domain domain.xsd">
S:        <domain:name>example.com</domain:name>
S:        <domain:expiration-date>
S:          2002-06-08T22:00:00.0Z
S:        </domain:expiration-date>
S:      </domain:expire-data>
S:    </response-data>
S:    <trans-id>
S:      <date>2000-06-08</date>
S:      <client-id>ClientX</client-id>
S:      <code>ABC-12345-XYZ</code>
S:    </trans-id>
S:  </response>
S:</epp>

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

3.2.2 EPP <delete> Command

The EPP <delete> command provides a transform operation that allows a
client to delete a domain object. In addition to the standard EPP
command elements, the <delete> command MUST contain a <domain:delete>
element that identifies the domain namespace and the location of the
domain schema. The <domain:delete> element SHALL contain the
following child elements:

- A <domain:name> element that contains the fully qualified domain
  name of the object to be deleted.

A domain name MUST NOT be deleted if the domain is associated with
child name servers. For example, if domain "example.com" is known,
and name server "ns1.example.com" is also known, then domain
"example.com" MUST NOT be deleted until server "ns1.example.com" has
been either deleted or renamed to exist in a different parent domain.
Example <delete> command:

C:<?xml version="1.0" standalone="no"?><
C:    <epp xmlns="urn:iana:xmlns:epp"
C:        xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
C:        xsi:schemaLocation="urn:iana:xmlns:epp epp.xsd">
C:    <command>
C:      <delete>
C:        <domain:delete xmlns:domain="urn:iana:xmlns:domain"
C:            xsi:schemaLocation="urn:iana:xmlns:domain domain.xsd">
C:          <domain:name>example.com</domain:name>
C:        </domain:delete>
C:      </delete>
C:      <trans-id>
C:        <date>2000-06-08</date>
C:        <client-id>ClientX</client-id>
C:        <code>ABC-12345-XYZ</code>
C:      </trans-id>
C:    </command>
C:</epp>

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

Example <delete> response:

S:<?xml version="1.0" standalone="no"?><
S:    <epp xmlns="urn:iana:xmlns:epp"
S:        xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
S:        xsi:schemaLocation="urn:iana:xmlns:epp epp.xsd">
S:    <response>
S:      <result code="1000">
S:        <text>Command completed successfully</text>
S:      </result>
S:      <trans-id>
S:        <date>2000-06-08</date>
S:        <client-id>ClientX</client-id>
S:        <code>ABC-12345-XYZ</code>
S:      </trans-id>
S:    </response>
S:</epp>

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

3.2.3 EPP <renew> Command
The EPP <renew> command provides a transform operation that allows a client to extend the validity period of a domain object. In addition to the standard EPP command elements, the <renew> command MUST contain a <domain:renew> element that identifies the domain namespace and the location of the domain schema. The <domain:renew> element SHALL contain the following child elements:

- A <domain:name> element that contains the fully qualified domain name of the object whose validity period is to be extended.

- A <domain:current-expiration-year> element that contains the year in which the current validity period ends. This value ensures that repeated <renew> commands do not result in multiple unanticipated successful renewals.

- A <domain:period> element that contains the number of years to be added to the validity period of the domain. The minimum allowable value is one (1) year. The maximum allowable value is ninety-nine (99) years. A server MAY support a lower maximum value.

Example <renew> command:

```
C:<?xml version="1.0" standalone="no"?>
C:<epp xmlns="urn:iana:xmlns:epp"
C:     xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
C:     xsi:schemaLocation="urn:iana:xmlns:epp epp.xsd">
C:  <command>
C:    <renew>
C:      <domain:renew xmlns:domain="urn:iana:xmlns:domain"
C:       xsi:schemaLocation="urn:iana:xmlns:domain domain.xsd">
C:        <domain:name>example.com</domain:name>
C:        <domain:current-expiration-year>2000</domain:current-expiration-year>
C:        <domain:period>5</domain:period>
C:      </domain:renew>
C:    </renew>
C:    <trans-id>
C:      <date>2000-06-08</date>
C:      <client-id>ClientX</client-id>
C:      <code>ABC-12345-XYZ</code>
C:    </trans-id>
C:  </command>
C:</epp>
```

When a <renew> command has been processed successfully, a server MUST respond with an EPP <response-data> element that MUST contain a child <domain:expire-data> element that identifies the domain namespace and the location of the domain schema. The <domain:expire-data> element
SHALL contain the following child elements:

- A <domain:name> element that contains the fully qualified domain name whose validity period has been extended.

- A <domain:expiration-date> element that contains the end of the domain’s validity period.

Example <renew> response:

S:<?xml version="1.0" standalone="no"?>
S:<epp xmlns="urn:iana:xmlns:epp"
S:     xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
S:     xsi:schemaLocation="urn:iana:xmlns:epp epp.xsd">
S:  <response>
S:    <result code="1000">
S:      <text>Command completed successfully</text>
S:    </result>
S:    <response-data>
S:      <domain:expire-data xmlns:domain="urn:iana:xmlns:domain"
S:       xsi:schemaLocation="urn:iana:xmlns:domain domain.xsd">
S:        <domain:name>example.com</domain:name>
S:        <domain:expiration-date>
S:          2005-04-03T22:00:00.0Z
S:        </domain:expiration-date>
S:      </domain:expire-data>
S:    </response-data>
S:    <trans-id>
S:      <date>2000-06-08</date>
S:      <client-id>ClientX</client-id>
S:      <code>ABC-12345-XYZ</code>
S:    </trans-id>
S:  </response>
S:</epp>

An EPP error response MUST be returned if a <renew> command could not be processed for any reason.

3.2.4 EPP <transfer> Command

The EPP <transfer> command provides a transform operation that allows a client to manage requests to transfer the sponsorship of a domain object. In addition to the standard EPP command elements, the <transfer> command MUST contain a <domain:transfer> element that identifies the domain namespace and the location of the domain schema. The <domain:transfer> element SHALL contain the following child elements:
- A `<domain:name>` element that contains the fully qualified domain name of the object for which a transfer request is to be created, approved, rejected, or cancelled.

- An OPTIONAL `<domain:period>` element that contains a number of years to be added to the registration period of the domain if the transfer is successful. If not specified, the registration period MUST NOT be changed as a result of the transfer. The minimum allowable value is one (1) year. The maximum allowable value is ninety-nine (99) years. A server MAY support a lower maximum value. This element MUST be consistently present or absent for all associated request, approval, rejection, or cancellation operations.

Every EPP `<transfer>` command MUST contain an "op" attribute that identifies the transfer operation to be performed. Valid values, definitions, and authorizations for all attribute values are defined in [EPP].

Every EPP `<transfer>` command MUST also contain an authorization identifier as described in [EPP]. It is important to note that the transaction identifier associated with successful transfer of a domain object becomes the authorization identifier required to authorize subsequent transfers of sponsorship of the domain object. A client MUST retain all transaction identifiers associated with successful domain object transfers and protect them from disclosure. A client MUST provide a copy of the transaction identifier information to the domain registrant, who will need this information to request a domain transfer through a different client.
Example <transfer> request command:

```xml
C: <?xml version="1.0" standalone="no"?>
C: <epp xmlns="urn:iana:xmlns:epp"
C:     xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
C:     xsi:schemaLocation="urn:iana:xmlns:epp epp.xsd">
C:   <command>
C:     <transfer op="request">
C:       <domain:transfer xmlns:domain="urn:iana:xmlns:domain"
C:         xsi:schemaLocation="urn:iana:xmlns:domain domain.xsd">
C:         <domain:name>example.com</domain:name>
C:         <domain:period>1</domain:period>
C:       </domain:transfer>
C:       <auth-id>
C:         <date>1999-06-08</date>
C:         <client-id>ClientX</client-id>
C:         <code>ABC-98765-XYZ</code>
C:       </auth-id>
C:     </transfer>
C:     <trans-id>
C:       <date>2000-06-08</date>
C:       <client-id>ClientY</client-id>
C:       <code>ABC-12345-XYZ</code>
C:     </trans-id>
C:   </command>
C: </epp>
```

When a <transfer> command has been processed successfully, a server MUST respond with an EPP <response-data> element that MUST contain a child <domain:transfer-data> element that identifies the domain namespace and the location of the domain schema. The <domain:transfer-data> element SHALL contain the same child elements defined for a transfer query response.
Example <transfer> response:

S: <?xml version="1.0" standalone="no"?>
S: <epp xmlns="urn:iana:xmlns:epp"
S:     xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
S:     xsi:schemaLocation="urn:iana:xmlns:epp epp.xsd">
S:  <response>
S:    <result code="1000">
S:      <text>Command completed successfully</text>
S:    </result>
S:    <response-data>
S:      <domain:transfer-data xmlns:domain="urn:iana:xmlns:domain"
S:       xsi:schemaLocation="urn:iana:xmlns:domain domain.xsd">
S:        <domain:name>example.com</domain:name>
S:        <domain:request-client>ClientX</domain:request-client>
S:        <domain:action-client>ClientY</domain:action-client>
S:        <domain:transfer-status>PENDING</domain:transfer-status>
S:        <domain:request-date>2000-06-08T22:00:00.0Z</domain:request-date>
S:        <domain:action-date>2000-06-13T22:00:00.0Z</domain:action-date>
S:        <domain:expiration-date>2002-09-08T22:00:00.0Z</domain:expiration-date>
S:      </domain:transfer-data>
S:    </response-data>
S:    <trans-id>
S:      <date>2000-06-08</date>
S:      <client-id>ClientX</client-id>
S:      <code>ABC-12345-XYZ</code>
S:    </trans-id>
S:  </response>
S:</epp>

An EPP error response MUST be returned if a <transfer> command could
not be processed for any reason.

3.2.5 EPP <update> Command

The EPP <update> command provides a transform operation that allows a
client to modify the attributes of a domain object. In addition to
the standard EPP command elements, the <update> command MUST contain a
<domain:update> element that identifies the domain namespace and the
location of the domain schema. The <domain:update> element SHALL
contain the following child elements:
- A <domain:name> element that contains the fully qualified domain name of the object to be updated.

- An OPTIONAL <domain:add> element that contains attribute values to be added to the domain object.

- An OPTIONAL <domain:remove> element that contains attribute values to be removed from the domain object.

Either a <domain:add> element or a <domain:remove> element MUST be provided. The <domain:add> and <domain:remove> elements SHALL contain the following child elements:

- Zero or more <domain:server> elements that contain the fully qualified host name of a known host object. A host object MUST be known to the server before a server attribute can be added or removed from a domain object. The EPP mapping for host objects is described in [EPP-H].

- Zero or more (up to a maximum of four) <domain:contact> elements that contain the registrant, administrative, technical, and billing contact identifiers to be associated with the domain. A contact identifier MUST be known to the server before the contact can be associated with the domain. Only one contact identifier of each type MAY be specified. A server MAY provide contact object services when providing domain name object services. The EPP mapping for contact objects is described in [EPP-C].

- One or two <domain:status> elements that contain status values to be applied to or removed from the domain object.

It is important to note that the maximum number of domain attribute elements is subject to the number of values currently associated with the domain object. For example, if a domain object currently has "n" server attribute elements, the maximum number of server attribute elements that can be added is 13 - "n". Likewise, the maximum number of contact elements that can be added is 4 - "n", with the added restriction that only one contact of each type (registrant, administrative, technical, or billing) MAY be associated with a domain object.
Example <update> command:

```xml
<epp xmlns="urn:iana:xmlns:epp"
     xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
     xsi:schemaLocation="urn:iana:xmlns:epp epp.xsd">
  <command>
    <update>
      <domain:update xmlns:domain="urn:iana:xmlns:domain"
                      xsi:schemaLocation="urn:iana:xmlns:domain domain.xsd">
        <domain:name>example.com</domain:name>
        <domain:add>
          <domain:server>ns1.example.com</domain:server>
          <domain:contact type="registrant">SH0000</domain:contact>
          <domain:status>CLIENT-HOLD</domain:status>
        </domain:add>
        <domain:remove>
          <domain:server>ns2.example.com</domain:server>
          <domain:contact type="billing">SH0000</domain:contact>
          <domain:status>CLIENT-LOCK</domain:status>
        </domain:remove>
      </domain:update>
    </update>
    <trans-id>
      <date>2000-06-08</date>
      <client-id>ClientX</client-id>
      <code>ABC-12345-XYZ</code>
    </trans-id>
  </command>
</epp>
```

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

S:<xml version="1.0" standalone="no"/>
S:<epp xmlns="urn:iana:xmlns:epp"
S:     xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
S:     xsi:schemaLocation="urn:iana:xmlns:epp epp.xsd">
S:  <response>
S:    <result code="1000">
S:      <text>Command completed successfully</text>
S:    </result>
S:    <trans-id>
S:      <date>2000-06-08</date>
S:      <client-id>ClientX</client-id>
S:      <code>ABC-12345-XYZ</code>
S:    </trans-id>
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.

<?xml version="1.0"?>
<schema xmlns="http://www.w3.org/1999/XMLSchema"
xmlns:domain="urn:iana:xmlns:domain"
xmlns:epp="urn:iana:xmlns:epp"
targetNamespace="urn:iana:xmlns:domain"
elementFormDefault="qualified">
<annotation>
<documentation>Extensible Provisioning Protocol v1.0 domain provisioning schema.</documentation>
</annotation>

<!-- Types used within an EPP greeting. -->
<complexType name="emptyType" content="empty"/>
<element name="service" type="domain:emptyType"/>
<!--
Child elements found in EPP commands.
-->  
<element name="create" type="domain:createType"/>
<element name="delete" type="domain:nameOnlyType"/>
<element name="info" type="domain:nameOnlyType"/>
<element name="ping" type="domain:multiNameType"/>
<element name="renew" type="domain:renewType"/>
<element name="transfer" type="domain:transferType"/>
<element name="transfer-query" type="domain:nameOnlyType"/>
<element name="update" type="domain:updateType"/>

<!--
Various simple type definitions.
-->  
<simpleType name="clientIDType" base="string">
  <minLength value="3"/>
  <maxLength value="16"/>
  <pattern value="[A-Za-z0-9]{3,16}"/>
</simpleType>

<simpleType name="nameType" base="string">
  <minLength value="1"/>
  <maxLength value="255"/>
</simpleType>

<simpleType name="codeType" base="string">
  <minLength value="8"/>
  <maxLength value="16"/>
  <pattern value="[A-Za-z0-9][A-Za-z0-9-]{6,14}[A-Za-z0-9]"/>
</simpleType>

<complexType name="xidType" content="elementOnly">
  <element name="date" type="date"/>
  <element name="client-id" type="domain:clientIDType"/>
  <element name="code" type="domain:codeType"/>
</complexType>

<!--
Child elements of the <create> and <info> commands.
-->  
<simpleType name="periodType" base="unsignedShort">
  <minInclusive value="1"/>
  <maxInclusive value="99"/>
</simpleType>

<simpleType name="contactType" base="string">
  <enumeration value="administrative"/>
  <enumeration value="billing"/>
<enumeration value="registrant"/>
<enumeration value="technical"/>
</simpleType>
<complexType name="contactBaseType" content="textOnly">
 <attribute name="type" type="domain:contactType"/>
</complexType>
<complexType name="createType" content="elementOnly">
 <element name="name" type="domain:nameType"/>
 <element name="period" type="domain:periodType" minOccurs="0" maxOccurs="1"/>
 <element name="server" type="domain:nameType" minOccurs="0" maxOccurs="13"/>
 <element name="contact" type="domain:contactBaseType" minOccurs="0" maxOccurs="4"/>
</complexType>

<!--
Child elements of the <transfer> transform command.
-->
<complexType name="transferType" content="elementOnly">
 <element name="name" type="domain:nameType"/>
 <element name="period" type="domain:periodType" minOccurs="0" maxOccurs="1"/>
</complexType>

<!--
Child element of commands that require only a name.
-->
<complexType name="nameOnlyType" content="elementOnly">
 <element name="name" type="domain:nameType"/>
</complexType>

<!--
Child element of commands that accept multiple names.
-->
<complexType name="multiNameType" content="elementOnly">
 <element name="name" type="domain:nameType" minOccurs="1" maxOccurs="16"/>
</complexType>

<!--
Child elements of the <renew> command.
-->
<complexType name="renewType" content="elementOnly">
 <element name="name" type="domain:nameType"/>
 <element name="current-expiration-year" type="year"/>
 <element name="period" type="domain:periodType"/>
</complexType>
<complexType name="updateType" content="elementOnly">
  <element name="name" type="domain:nameType"/>
  <element name="add" type="domain:addRemoveType"
    minOccurs="0" maxOccurs="1"/>
  <element name="remove" type="domain:addRemoveType"
    minOccurs="0" maxOccurs="1"/>
</complexType>

<complexType name="addRemoveType" content="elementOnly">
  <element name="server" type="string"
    minOccurs="0" maxOccurs="13"/>
  <element name="contact" type="domain:contactBaseType"
    minOccurs="0" maxOccurs="4"/>
  <element name="status" type="domain:statusType"
    minOccurs="0" maxOccurs="2"/>
</complexType>

<complexType name="expireDataType" content="elementOnly">
  <element name="name" type="domain:nameType"/>
  <element name="expiration-date" type="timeInstant"/>
</complexType>

<complexType name="pingNameType" content="textOnly">
  <minLength value="1"/>
  <maxLength value="255"/>
  <attribute name="result" use="required" type="domain:pingResultType"/>
</complexType>

<complexType name="pingDataType" content="elementOnly">
  <element name="name" type="domain:pingNameType"
    minOccurs="1" maxOccurs="16"/>
</complexType>
Child elements of the <info> and <transfer> query commands.

```xml
<simpleType name="statusType" base="string">
  <enumeration value="NEW"/>
  <enumeration value="ACTIVE"/>
  <enumeration value="INACTIVE"/>
  <enumeration value="HOLD"/>
  <enumeration value="LOCK"/>
  <enumeration value="CLIENT-HOLD"/>
  <enumeration value="CLIENT-LOCK"/>
  <enumeration value="PENDING-TRANSFER"/>
  <enumeration value="PENDING-DELETE"/>
</simpleType>

<simpleType name="transferStatusType" base="string">
  <enumeration value="PENDING"/>
  <enumeration value="APPROVED"/>
  <enumeration value="CANCELLED"/>
  <enumeration value="REJECTED"/>
  <enumeration value="AUTO-APPROVED"/>
  <enumeration value="AUTO-CANCELLED"/>
</simpleType>

<complexType name="infoDataType" content="elementOnly">
  <element name="name" type="domain:nameType"/>
  <element name="client-id" type="domain:clientIDType" minOccurs="1" maxOccurs="4"/>
  <element name="status" type="domain:statusType" minOccurs="1" maxOccurs="4"/>
  <element name="contact" type="domain:contactBaseType" minOccurs="0" maxOccurs="4"/>
  <element name="server" type="string" minOccurs="0" maxOccurs="13"/>
  <element name="child-server" type="string" minOccurs="0" maxOccurs="unbounded"/>
  <element name="created-by" type="domain:clientIDType"/>
  <element name="created-date" type="timeInstant"/>
  <element name="expiration-date" type="timeInstant"/>
  <element name="last-updated-by" type="domain:clientIDType" minOccurs="0" maxOccurs="1"/>
  <element name="last-updated-date" type="timeInstant" minOccurs="0" maxOccurs="1"/>
  <element name="last-transfer-date" type="timeInstant" minOccurs="0" maxOccurs="1"/>
  <element name="auth-id" type="domain:xidType" minOccurs="0" maxOccurs="1"/>
</complexType>
```
<complexType name="transferDataType" content="elementOnly">
  <element name="name" type="domain:nameType"/>
  <element name="request-client" type="domain:clientIDType"/>
  <element name="action-client" type="domain:clientIDType"/>
  <element name="transfer-status" type="domain:transferStatusType"/>
  <element name="request-date" type="timeInstant"/>
  <element name="action-date" type="timeInstant"/>
  <element name="expiration-date" type="timeInstant" minOccurs="0" maxOccurs="1"/>
</complexType>

<!--
End of schema.
-->
</schema>
5. Internationalization Considerations

EPP is represented in XML, which provides native support for encoding information using the double-byte Unicode character set and its more compact representations including UTF-8. Compliant XML processors are required to understand both UTF-8 and raw Unicode character sets; XML also includes a provision for identifying other character sets through use of an "encoding" attribute in an <?xml?> processing instruction. The complete list of character set encoding identifiers is maintained by IANA and is described in [CHARSET] and [RFC1700].

All date-time values presented via EPP MUST be expressed in Universal Coordinated Time. The XML Schema "date" format allows use of time zone identifiers to indicate offsets from the zero meridian, but this option MUST NOT be used within EPP. Both extended and truncated date and time forms defined in [ISO8601] MAY be used.

This document requires domain and host name syntax as specified in [RFC952] as updated by [RFC1123]. These conformance requirements MAY change as a result of progressing work in developing standards for internationalized domain names.

6. IANA Considerations

XML schemas require a URI for unique identification. Schemas MUST be registered to ensure URI uniqueness, but the IETF does not currently have a recommended repository for the registration of XML schemas. This document uses URNs to describe XML namespaces and XML schemas. IANA SHOULD maintain a registry of XML namespace and schema URI assignments. Per policies described in [IANA], URI assignment requests SHOULD be reviewed by a designated expert, and values SHOULD be assigned only as a result of standards action taken by the IESG.

7. Security Considerations

The object mapping described in this document does not provide any security services beyond those specified by [EPP].
8. References


[EPP] S. Hollenbeck: "Extensible Provisioning Protocol", draft-
hollenbeck-epp-00.txt, work in progress.

Mapping", draft-hollenbeck-epp-contact-00.txt, work in progress.

Mapping", draft-hollenbeck-epp-host-00.txt, work in progress.

Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.

Information interchange – Representation of dates and times – The
International Organization for Standardization".

[RFC952] K. Harrenstien et al.: "DOD Internet Host Table

[RFC1123] R. Braden: "Requirements for Internet Hosts -- Application

October 1994.

[RFC2119] S. Bradner: "Key Words for Use in RFCs to Indicate

/XML] Editor T. Bray et al.: "Extensible Markup Language (XML) 1.0",

/XML-SD] Editors P. Biron and A. Malhotra: "XML Schema Part 2:
Datatypes", http://www.w3.org/TR/xmlschema-2/, W3C Working Draft April
2000

http://www.w3.org/TR/xmlschema-1/, W3C Working Draft April 2000
9. Author’s Address

Scott Hollenbeck
VeriSign Global Registry Services
21345 Ridgetop Circle
Dulles, VA 20166-6503
USA
shollenbeck@verisign.com
10. Full Copyright Statement

Copyright (C) The Internet Society 2000. All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.