EXTENSIBLE PROVISIONING PROTOCOL EXTENSION MAPPING: <RELATED DOMAIN>

Version 1.2

Table of Contents

1 Introduction

This document describes a Related Domain extension for version 1.0 of the Extensible Provisioning Protocol (EPP). This mapping, an extension of the domain name mapping described in [EPP-D], is specified using the Extensible Markup Language (XML) 1.0 as described in [XML] and XML Schema notation as described in [XMLS-1] and [XMLS-2], for managing client-side and server-side domain name relationships. A client-side domain name relationship can be managed by using the extension to the transform commands that enable transforming more than one domain name in a single transform command. A server-side domain name relationship (across top-level domains "tld" or variants within a top-level domain) is reflected in the extension to the info response, and can be managed by using the extension to the transform commands.

[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 is 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.

2 Object Attributes

This extension adds additional elements to the domain name mapping described in [EPP-D]. Only new element descriptions are described here.

3 EPP Command Mapping

A detailed description of the EPP syntax and semantics can be found in [EPP].

3.1 EPP Query Commands

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

3.1.1 EPP <check> Command

This extension does not add any elements to the EPP <check> command or <check> response described in [EPP-D].

3.1.2 EPP <info> Command

This extension defines additional elements for the EPP <info> command described in [EPP-D].

There are two forms of the extension to the EPP <info> command based on the "type" attribute: The Domain Info Form and the Related Info Form.

3.1.2.1 <relDom:infData> Element

The <relDom:infData> element is returned to a successfully processed <info> command for both the Domain Info Form, described in section 3.1.2.1, and the Related Info Form, described in section 3.1.2.3. The <relDom:infData> element contains the following child elements:

- <relDom:group> - One or more <relDom:group> elements describing the group of related domains currently associated with the object. The <relDomain:group> element MUST contain a "type" attribute that defines the type of the related domains with the possible values of "tld" and "variant". The "tld" type represents a set of related domains across Top Level Domains (TLDs) and the "variant" type represents a set of related variant domains within a TLD. The <relDom:group> element contains the following child elements:
    - <relDom:fields> - Element containing the set of fields that MUST be synchronized across the related domains. The <relDom:fields> element MUST contain an "inSync" boolean attribute that defines whether all of the fields are synchronized. The <relDom:fields> elements contains the following child elements:
        - <relDom:field> - One or more elements that MUST be the same across all of the related domains. The <relDom:field> element MUST contain a "name" attribute that defines the name of the field and an "inSync" boolean attribute that defines the field is synchronized across all of the related domains.
    - <relDom:registered> - An OPTIONAL element containing one or more <relDom:name> elements specifying the related domains that are registered.
    - <relDom:available> - An OPTIONAL element containing one or more <relDom:name> elements specifying the related domains that are available for registration.

3.1.2.2 Domain Info Form

The Domain Info Form, defined with the <relDom:info> "type" attribute value of "domain", is used to get the domain information of an existing domain along with the related domain information. This is the default form when the <relDom:info> "type" attribute is not explicitly defined. With the Domain Info Form, in addition to the EPP info command elements described in [EPP-D], the command MUST contain an <extension> element. The <extension> element MUST contain a child <relDom:info> element, with the "type" attribute value of "domain" explicitly or by default, to indicate to the server to include the related domain information in an extension to the EPP info response described in [EPP-D].

Example <info> command for a domain with the <relDom:info> extension using the Domain Info Form:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0”>
  <command>
    <info>
      <domain:info
        xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>xn--test.tld1</domain:name>
      </domain:info>
    </info>
    <extension>
      <relDom:info
        xmlns:relDom=http://www.verisign.com/epp/relatedDomain-1.0
type=”domain”/>
    </extension>
    <clTRID>ABC-12345</clTRID>
  </command>
</epp>
```

When an <info> command has been processed successfully, the EPP <resData> element MUST contain child elements as described in [EPP-D]. In addition, the EPP <extension> element SHOULD contain a child <relDom:infData> element that identifies the extension namespace if the object has one or more related domains associated with it and based on server policy. The <relDom:infData> element contains the child elements defined in section 3.1.2.1.

Example <info> response for a domain with both “tld” and “variant” related domain information using the Domain Info Form:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0”>
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
  </result>
  <resData>
    <domain:infData xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
      <domain:name>xn--test.tld1</domain:name>
      <domain:roid>TEST1-REP</domain:roid>
      <domain:status s="ok"/>
      <domain:registrant>sh8013</domain:registrant>
      <domain:contact type="admin">sh8013</domain:contact>
      <domain:contact type="tech">sh8013</domain:contact>
      <domain:contact type="billing">sh8013</domain:contact>
      <domain:ns>
      <domain:hostObj>ns1.example.com</domain:hostObj>
      <domain:hostObj>ns2.example.com</domain:hostObj>
      </domain:ns>
      <domain:host>ns1.example.com</domain:host>
      <domain:host>ns2.example.com</domain:host>
      <domain:clID>ClientX</domain:clID>
      <domain:crID>ClientY</domain:crID>
      <domain:crDate>1999-04-03T22:00:00.0Z</domain:crDate>
      <domain:upID>ClientX</domain:upID>
      <domain:upDate>1999-12-03T09:00:00.0Z</domain:upDate>
      <domain:exDate>2005-04-03T22:00:00.0Z</domain:exDate>
      <domain:trDate>2000-04-08T09:00:00.0Z</domain:trDate>
      <domain:authInfo>
```

```
        <domain:pw>2fooBAR</domain:pw>
      </domain:authInfo>
    </domain:infData>
  </resData>
  <extension>
    <relDom:infData
      xmlns:relDom="http://www.verisign-grs.com/epp/relatedDomain-1.0">
      <relDom:group type="tld">
        <relDom:fields inSync="false">
          <relDom:field name="clID" inSync="false">
          <relDom:field name="registrant" inSync="true">
          <relDom:field name="ns" inSync="false">
        </relDom:fields>
        <relDom:registered>
          <relDom:name>xn--test.tld1</relDom:name>
          <relDom:name>xn--test.tld2</relDom:name>
        </relDom:registered>
        <relDom:available>
          <relDom:name>xn--test.tld3</relDom:name>
        </relDom:available>
      </relDom:group>
      <relDom:group type="variant">
        <relDom:fields inSync="true">
          <relDom:field name="clID" inSync="true">
          <relDom:field name="registrant" inSync="true">
          <relDom:field name="ns" inSync="true">
        </relDom:fields>
        <relDom:registered>
          <relDom:name>xn--test-variant1.tld1</relDom:name>
          <relDom:name>xn--test-variant2.tld1</relDom:name>
          <relDom:name>xn--test-variant3.tld1</relDom:name>
        </relDom:registered>
        <relDom:available>
          <relDom:name>xn--test-variant4.tld1</relDom:name>
          <relDom:name>xn--test-variant5.tld1</relDom:name>
          <relDom:name>xn--test-variant6.tld1</relDom:name>
        </relDom:available>
      </relDom:group>
    </relDom:infData>
    </extension>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>54322-XYZ</svTRID>
    </trID>
  </response>
</epp>
```

3.1.2.3 Related Info Form

The Related Info Form, defined with the <relDom:info> “type” attribute value of “related”, is a new command called the Related Domain Info Command. The command gets the related domain information of the <domain:name> info command element defined in [EPP-D], independent of the existence of the domain name. With the Related Info Form, in addition to the EPP info command elements defined in [EPP-D], the command MUST contain an <extension> element. The <extension> element MUST contain a child <relDom:info> element, with the “type” attribute value of “related”, to indicate to the server to include the related domain information in an extension to the EPP response described in [EPP].

Example <info> command for a domain with the <relDom:info> extension using the Related Info Form:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0”>
  <command>
    <info>
      <domain:info
        xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>xn--test.tld3</domain:name>
      </domain:info>
    </info>
    <extension>
      <relDom:info
        xmlns:relDom=http://www.verisign.com/epp/relatedDomain-1.0
type=”related”/>
    </extension>
    <clTRID>ABC-12345</clTRID>
  </command>
</epp>
```

When an <info> command has been processed successfully, the EPP <response> element MUST contain an <extension> <relDom:infData> element that identifies the related domain namespace. In addition, the EPP <extension> element SHOULD contain a child <relDom:infData> element that identifies the extension namespace if at least one related domain exists and based on server policy. The <relDom:infData> element contains the child elements defined in section 3.1.2.1.

Example <info> response for both “tld” and “variant” related domain information using the Related Info Form:

```
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0>
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <extension>
      <relDom:infData
        xmlns:relDom="http://www.verisign.com/epp/relatedDomain-1.0">
        <relDom:group type="tld">
          <relDom:fields inSync="false">
            <relDom:field name="clID" inSync="false"/>
            <relDom:field name="registrant" inSync="true"/>
            <relDom:field name="ns" inSync="false"/>
          </relDom:fields>
          <relDom:registered>
            <relDom:name>xn--test.tld1</relDom:name>
            <relDom:name>xn--test.tld2</relDom:name>
          </relDom:registered>
          <relDom:available>
            <relDom:name>xn--test.tld3</relDom:name>
          </relDom:available>
        </relDom:group>
        <relDom:group type="variant">
          <relDom:fields inSync="true">
```

```
            <relDom:field name="clID" inSync="true"/>
            <relDom:field name="registrant" inSync="true"/>
            <relDom:field name="ns" inSync="true"/>
          </relDom:fields>
          <relDom:registered>
            <relDom:name>xn--test-variant1.tld1</relDom:name>
            <relDom:name>xn--test-variant1.tld2</relDom:name>
            <relDom:name>xn--test-variant1.tld3</relDom:name>
          </relDom:registered>
          <relDom:available>
            <relDom:name>xn--test-variant2.tld1</relDom:name>
            <relDom:name>xn--test-variant2.tld2</relDom:name>
            <relDom:name>xn--test-variant2.tld3</relDom:name>
          </relDom:available>
        </relDom:group>
      </relDom:infData>
    </extension>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>54322-XYZ</svTRID>
    </trID>
  </response>
</epp>
```

3.1.3 EPP <transfer> Query Command

This extension defines additional elements for the EPP <transfer> command described in [EPP-D]. The extension to the EPP <transfer> query command is defined in section 3.2.4 where the <transfer> command MUST contain an “op” attribute with value “query”. The extension to the EPP <transfer> query response is defined in section 3.2.4.

3.2 EPP Transform Commands

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

3.2.1 EPP <create> Command

This extension defines additional elements for the EPP <create> command described in [EPP-D].

In addition to the EPP command elements described in [EPP-D], the command MUST contain an <extension> element. The <extension> element MUST contain a child <relDom:create> element to create more than one related domain name in the <create> command. The <relDom:create> element contains the following child elements:

- <relDom:domain> - One or more <relDom:domain> elements to create along with the <domain:name> described in [EPP-D]. The <domain:ns>, <domain:registrant>, and <domain:contact> elements in [EPP-D] SHOULD be set on the <relDom:domain> objects by the server. The <relDom:domain> element contains the following child elements:
    - <relDom:name> - Element that contains the fully qualified name of the domain object to be created.
    - <relDom:authInfo> - Element that contains authorization information to be associated with the domain object as described in [EPP-D].
    - <relDom:period> - An OPTIONAL element containing the initial registration period of the domain object as described in [EPP-D]. A server MAY define a default initial registration period if not specified by the client.
    - <relDom:lang> - An OPTIONAL element containing language tag value, as defined in [EPP-ILANG], for an internationalized domain name (IDN).

Example <create> command for three related domain names (“example1.tld”, “example2.tld”, and “example3.tld”) with the <relDom:create> extension:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <create>
      <domain:create
            xmlns:domain="urn:ietf:params:xml:ns:domain-1.0”>
        <domain:name>example1.tld</domain:name>
        <domain:authInfo>
          <domain:pw>2fooBAR</domain:pw>
        </domain:authInfo>
      </domain:create>
    </create>
    <extension>
      <relDom:create
            xmlns:relDom="http://www.verisign.com/epp/relatedDomain-1.0">
        <relDom:domain>
          <relDom:name>example2.tld</relDom:name>
          <relDom:authInfo>
            <relDom:pw>relDom123!</relDom:pw>
          </relDom:authInfo>
          <relDom:period unit="y">5</relDom:period>
        </relDom:domain>
        <relDom:domain>
          <relDom:name>example3.tld</relDom:name>
          <relDom:authInfo>
            <relDom:pw>relDom456!</relDom:pw>
```

```
          </relDom:authInfo>
          <relDom:period unit="y">5</relDom:period>
        </relDom:domain>
      </relDom:create>
    </extension>
    <clTRID>ABC-12345</clTRID>
  </command>
</epp>
```

When an <create> command has been processed successfully, the EPP <resData> element MUST contain child elements as described in [EPP-D]. In addition, the EPP <extension> element MUST contain a child <relDom:creData> element. The <relDom:creData> element contains the following child elements:

- <relDom:domain> - One or more <relDom:domain> elements created along with the <domain:name> described in [EPP-D]. The <relDom:domain> element contains the following child elements:
  - <relDom:name> - Element that contains the fully qualified name of the domain object created.
  - <relDom:crDate> - Element that contains the date and time of domain object creation.
  - <relDom:exDate> - An OPTIONAL element that contains the date and time identifying the end of the domain object's registration period.

Example <create> response for three related domain names ("example1.tld", "example2.tld", and "example3.tld") created with the <relDom:create> extension:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <resData>
      <domain:creData
            xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>example1.tld</domain:name>
        <domain:crDate>2013-07-10T00:00:00.0000Z</domain:crDate>
        <domain:exDate>2018-07-10T00:00:00.0000Z</domain:exDate>
      </domain:creData>
    </resData>
    <extension>
      <relDom:creData
            xmlns:relDom="http://www.verisign.com/epp/relatedDomain-1.0">
        <relDom:domain>
          <relDom:name>example2.tld</relDom:name>
          <relDom:crDate>2013-07-10T00:00:00.0000Z</relDom:crDate>
          <relDom:exDate>2018-07-10T00:00:00.0000Z</relDom:exDate>
        </relDom:domain>
        <relDom:domain>
          <relDom:name>example3.tld</relDom:name>
          <relDom:crDate>2013-07-10T00:00:00.0000Z</relDom:crDate>
          <relDom:exDate>2018-07-10T00:00:00.0000Z</relDom:exDate>
        </relDom:domain>
      </relDom:creData>
```

```
    </extension>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>54321-XYZ</svTRID>
    </trID>
  </response>
</epp>
```

3.2.2 EPP <delete> Command

This extension defines additional elements for the EPP <delete> command described in [EPP-D].

In addition to the EPP command elements described in [EPP-D], the command MUST contain an <extension> element. The <extension> element MUST contain a child <relDom:delete> element to delete more than one related domain name in the <delete> command. The <relDom:delete> element contains the following child elements:

- <relDom:name> - One or more <relDom:name> elements to delete along with the <domain:name> described in [EPP-D].

Example <delete> command for three related domain names ("example1.tld", "example2.tld", and "example3.tld") with the <relDom:delete> extension:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <delete>
      <domain:delete xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>example1.tld</domain:name>
      </domain:delete>
    </delete>
    <extension>
      <relDom:delete
            xmlns:relDom="http://www.verisign.com/epp/relatedDomain-1.0">
        <relDom:name>example2.tld</relDom:name>
        <relDom:name>example3.tld</relDom:name>
      </relDom:delete>
    </extension>
    <clTRID>ABC-12345</clTRID>
  </command>
</epp>
```

When an <delete> command has been processed successfully, the EPP <resData> element MUST contain child elements as described in [EPP-D]. In addition, the EPP <extension> element MUST contain a child <relDom:delData> element. The <relDom:delData> element contains the following child elements:

- <relDom:domain> - One or more <relDom:domain> elements containing the result of the delete command. The <relDom:domain> element contains the following child elements:
  - <relDom:name> - Element that contains the fully qualified name of the domain object.
  - <relDom:result> - Element that contains the result of the delete with the possible values of "deleted", to indicated that the domain object was immediately deleted, and "pendingDelete", to indicate that the domain object was updated with the "pendingDelete" status.

Example <delete> response for three related domain names ("example1.tld", "example2.tld", and "example3.tld") deleted with the <relDom:delete> extension:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
```

```
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1001">
      <msg>Command completed successfully; action pending</msg>
    </result>
    <extension>
      <relDom:delData
            xmlns:relDom="http://www.verisign.com/epp/relatedDomain-1.0">
        <relDom:domain>
          <relDom:name>domain1.com</relDom:name>
          <relDom:result>deleted</relDom:result>
        </relDom:domain>
        <relDom:domain>
          <relDom:name>domain2.com</relDom:name>
          <relDom:result>pendingDelete</relDom:result>
        </relDom:domain>
      </relDom:delData>
    </extension>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>54321-XYZ</svTRID>
    </trID>
  </response>
</epp>
```

3.2.3 EPP <renew> Command

This extension defines additional elements for the EPP <renew> command described in [EPP-D].

In addition to the EPP command elements described in [EPP-D], the command MUST contain an <extension> element. The <extension> element MUST contain a child <relDom:renew> element to renew more than one related domain name in the <renew> command. The <relDom:renew> element contains the following child elements:

- <relDom:domain> - One or more <relDom:domain> elements to renew along with the <domain:name> described in [EPP-D]. The <relDom:domain> element contains the following child elements:
  - <relDom:name> - Element that contains the fully qualified name of the domain object to be renewed.
  - <relDom:curExpDate> - Element that contains the date on which the current validity period ends as described in [EPP-D].
  - <relDom:period> - An OPTIONAL element that contains the number of units to be added to the registration period of the domain object as described in [EPP-D]. The number of units available MAY be subject to limits imposed by the server.

Example <renew> command for three related domain names (“example1.tld”, “example2.tld”, and “example3.tld”) with the <relDom:renew> extension:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <renew>
      <domain:renew
            xmlns:domain="urn:ietf:params:xml:ns:domain-1.0”>
        <domain:name>example1.tld</domain:name>
        <domain:curExpDate>2013-07-10</domain:curExpDate>
        <domain:period unit="y">5</domain:period>
      </domain:renew>
    </renew>
    <extension>
      <relDom:renew
            xmlns:relDom="http://www.verisign.com/epp/relatedDomain-1.0">
        <relDom:domain>
          <relDom:name>example2.tld</relDom:name>
          <relDom:curExpDate>2013-07-10</relDom:curExpDate>
          <relDom:period unit="y">5</relDom:period>
        </relDom:domain>
        <relDom:domain>
          <relDom:name>example3.tld</relDom:name>
          <relDom:curExpDate>2013-07-10</relDom:curExpDate>
          <relDom:period unit="y">5</relDom:period>
        </relDom:domain>
      </relDom:renew>
    </extension>
    <clTRID>ABC-12345</clTRID>
  </command>
</epp>
```

When an <renew> command has been processed successfully, the EPP <resData> element MUST contain child elements as described in [EPP-D]. In addition, the EPP <extension> element MUST contain a child <relDom:renData> element. The <relDom:renData> element contains the following child elements:

- <relDom:domain> - One or more <relDom:domain> elements renewed along with the <domain:name> described in [EPP-D]. The <relDom:domain> element contains the following child elements:
  - <relDom:name> - Element that contains the fully qualified name of the domain object renewed.
  - <relDom:exDate> - An OPTIONAL element that contains the date and time identifying the end of the domain object's registration period.

Example <renew> response for three related domain names ("example1.tld", "example2.tld", and "example3.tld") renewed with the <relDom:renew> extension:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <resData>
      <domain:renData
            xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>example1.com</domain:name>
        <domain:exDate>2018-07-10T00:00:00.0000Z</domain:exDate>
      </domain:renData>
    </resData>
    <extension>
      <relDom:renData
            xmlns:relDom="http://www.verisign.com/epp/relatedDomain-1.0">
        <relDom:domain>
          <relDom:name>example2.com</relDom:name>
          <relDom:exDate>2018-07-10T00:00:00.0000Z</relDom:exDate>
        </relDom:domain>
        <relDom:domain>
          <relDom:name>example3.com</relDom:name>
          <relDom:exDate>2018-07-10T00:00:00.0000Z</relDom:exDate>
        </relDom:domain>
      </relDom:renData>
    </extension>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>54321-XYZ</svTRID>
    </trID>
  </response>
</epp>
```

3.2.4 EPP <transfer> Command

This extension defines additional elements for the EPP <transfer> command described in [EPP-D].

In addition to the EPP command elements described in [EPP-D], the command MUST contain an <extension> element. The <extension> element MUST contain a child <relDom:transfer> element to transfer more than one related domain name in the <transfer> command. The <relDom:transfer> element contains the following child elements:

- <relDom:domain> - One or more <relDom:domain> elements to apply the transfer operation to along with the <domain:name> described in [EPP-D]. The <relDom:domain> element contains the following child elements:
  - <relDom:name> - Element that contains the fully qualified name of the domain object.
  - <relDom:authInfo> - An OPTIONAL element that contains authorization information associated with the domain object as described in [EPP-D].
  - <relDom:period> - An OPTIONAL element that contains the number of units to be added to the registration period of the domain object as described in [EPP-D]. This element can only be used when a transfer is requested, and it MUST be ignored if used otherwise. The number of units available MAY be subject to limits imposed by the server.

Example <transfer> request command for three related domain names (“example1.tld”, “example2.tld”, and “example3.tld”) with the <relDom:transfer> extension:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <transfer op="request">
      <domain:transfer
            xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>example1.tld</domain:name>
        <domain:period unit="y">1</domain:period>
        <domain:authInfo>
          <domain:pw>2fooBAR</domain:pw>
        </domain:authInfo>
      </domain:transfer>
    </transfer>
    <extension>
      <relDom:transfer
            xmlns:relDom="http://www.verisign.com/epp/relatedDomain-1.0">
        <relDom:domain>
          <relDom:name>example2.tld</relDom:name>
          <relDom:authInfo>
            <relDom:pw>relDom123!</relDom:pw>
          </relDom:authInfo>
          <relDom:period unit="y">1</relDom:period>
        </relDom:domain>
        <relDom:domain>
          <relDom:name>example3.tld</relDom:name>
          <relDom:authInfo>
            <relDom:pw>relDom123!</relDom:pw>
          </relDom:authInfo>
```

```
        </relDom:domain>
      </relDom:transfer>
    </extension>
    <clTRID>ABC-12345</clTRID>
  </command>
</epp>
```

When a <transfer> command has been processed successfully, the EPP <resData> element MUST contain child elements as described in [EPP-D]. In addition, the EPP <extension> element MUST contain a child <relDom:trnData> element. The <relDom:trnData> element contains the following child elements:

- <relDom:domain> - One or more <relDom:domain> elements associated with the transfer along with the <domain:name> described in [EPP-D]. The <relDom:domain> element contains the following child elements:
    - <relDom:name> - Element that contains the fully qualified name of the domain object.
    - <relDom:trStatus> - Element that contains the state of the most recent transfer request.
    - <relDom:reID> - Element that contains the identifier of the client that requested the object transfer.
    - <relDom:reDate> - Element that contains the date and time that the transfer was requested.
    - <relDom:acID> - Element that contains the identifier of the client that SHOULD act upon a PENDING transfer request. For all other status types, the value identifies the client that took the indicated action.
    - <relDom:acDate> - Element that contains the date and time of a required or completed request as described in [EPP-D]. For a PENDING request, the value identifies the date and time by which a response is required before an automated response action will be taken by the server. For all other status types, the value identifies the date and time when the request was completed.
    - <relDom:exDate> - An OPTIONAL element that contains the date and time identifying the end of the domain object's registration period.

Example <transfer> response for three related domain names ("example1.tld", "example2.tld", and "example3.tld") transferred with the <relDom:transfer> extension:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <resData>
      <domain:trnData
            xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>example1.tld</domain:name>
        <domain:trStatus>pending</domain:trStatus>
        <domain:reID>ClientX</domain:reID>
        <domain:reDate>2013-07-10T00:00:00.0000Z</domain:reDate>
        <domain:acID>ClientY</domain:acID>
        <domain:acDate>2013-07-10T00:00:00.0000Z</domain:acDate>
```

```
        <domain:exDate>2014-07-10T00:00:00.0000Z</domain:exDate>
      </domain:trnData>
    </resData>
    <extension>
      <relDom:trnData
            xmlns:relDom="http://www.verisign.com/epp/relatedDomain-1.0">
        <relDom:domain>
          <relDom:name>example2.tld</relDom:name>
          <relDom:trStatus>pending</relDom:trStatus>
          <relDom:reID>ClientX</relDom:reID>
          <relDom:reDate>2013-07-10T00:00:00.0000Z</relDom:reDate>
          <relDom:acID>ClientY</relDom:acID>
          <relDom:acDate>2013-07-10T00:00:00.0000Z</relDom:acDate>
          <relDom:exDate>2014-07-10T00:00:00.0000Z</relDom:exDate>
        </relDom:domain>
        <relDom:domain>
          <relDom:name>example3.tld</relDom:name>
          <relDom:trStatus>pending</relDom:trStatus>
          <relDom:reID>ClientX</relDom:reID>
          <relDom:reDate>2013-07-10T00:00:00.0000Z</relDom:reDate>
          <relDom:acID>ClientY</relDom:acID>
          <relDom:acDate>2013-07-10T00:00:00.0000Z</relDom:acDate>
          <relDom:exDate>2014-07-10T00:00:00.0000Z</relDom:exDate>
        </relDom:domain>
      </relDom:trnData>
    </extension>
    <trID>
      <clTRID>ABC-12345</clTRID>
      <svTRID>54321-XYZ</svTRID>
    </trID>
  </response>
</epp>
```

3.2.5 EPP <update> Command

This extension defines additional elements for the EPP <update> command described in [EPP-D].

In addition to the EPP command elements described in [EPP-D], the command MUST contain an <extension> element. The <extension> element MUST contain a child <relDom:update> element to update more than one related domain name in the <update> command. The updates specified in the EPP <update> command described in [EPP-D] MUST be applied to the domain objects specified in the <relDom:update>. The <relDom:update> element contains the following child elements:

- <relDom:name> - One or more <relDom:name> elements to update along with the <domain:name> described in [EPP-D].

Example <update> command for three related domain names (“example1.tld”, “example2.tld”, and “example3.tld”) with the <relDom:update> extension:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <update>
      <domain:update
            xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>example1.tld</domain:name>
        <domain:add>
          <domain:ns>
            <domain:hostObj>ns1.example.com</domain:hostObj>
          </domain:ns>
          <domain:status s="clientHold"/>
        </domain:add>
        <domain:rem>
          <domain:ns>
            <domain:hostObj>ns2.example.com</domain:hostObj>
          </domain:ns>
          <domain:status s="clientDeleteProhibited"/>
        </domain:rem>
        <domain:chg>
          <domain:registrant>jd1234</domain:registrant>
          <domain:authInfo>
            <domain:pw>2fooBAR</domain:pw>
          </domain:authInfo>
        </domain:chg>
      </domain:update>
    </update>
    <extension>
      <relDom:update
            xmlns:relDom="http://www.verisign.com/epp/relatedDomain-1.0">
        <relDom:name>example2.tld</relDom:name>
        <relDom:name>example3.tld</relDom:name>
      </relDom:update>
    </extension>
    <clTRID>ABC-12345-XYZ</clTRID>
  </command>
</epp>
```

This extension does not define any extension to the response of an <update> domain command. After processing the command, the server replies with a standard EPP response as defined in [EPP-D].

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.

```
BEGIN
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns:relDom="http://www.verisign.com/epp/relatedDomain-1.0"
   xmlns="http://www.w3.org/2001/XMLSchema"
   xmlns:eppcom="urn:ietf:params:xml:ns:eppcom-1.0"
   targetNamespace="http://www.verisign.com/epp/relatedDomain-1.0"
   elementFormDefault="qualified">
   <annotation>
      <documentation>
         Extensible Provisioning Protocol v1.0
         Related Domain extension
       </documentation>
   </annotation>
   <!--
   Import common element types.
   -->
   <import namespace="urn:ietf:params:xml:ns:eppcom-1.0"
      schemaLocation="eppcom-1.0.xsd"/>
   <!--
   Related Domain info command extension root element
   -->
   <element name="info" type="relDom:infoType"/>

   <!--
   Info type attribute values
   -->
   <simpleType name="infoTypeType">
      <restriction base="string">
         <enumeration value="domain"/>
         <enumeration value="related"/>
      </restriction>
   </simpleType>

   <!--
   Related Domain Info type
   -->
   <complexType name="infoType">
      <attribute name="type" type="relDom:infoTypeType"
         default="domain"/>
   </complexType>

   <!--
   Related Domain info response extension root element
   -->
   <element name="infData" type="relDom:infDataType"/>
   <complexType name="infDataType">
      <sequence>
         <element name="group" type="relDom:relatedDomainGroupType"
            maxOccurs="unbounded"/>
      </sequence>
```

```
</complexType>
<simpleType name="fieldNameType">
   <restriction base="normalizedString">
      <minLength value="1"/>
      <maxLength value="64"/>
   </restriction>
</simpleType>
<!--
Field that needs to be synchronized.
-->
<complexType name="fieldType">
   <attribute name="name" type="relDom:fieldNameType"
      use="required"/>
   <attribute name="inSync" type="boolean"
      use="required"/>
</complexType>
<!--
Related Domain group types
-->
<simpleType name="groupType">
   <restriction base="string">
      <enumeration value="tld"/>
      <enumeration value="variant"/>
   </restriction>
</simpleType>
<!--
Fields that need to be synchronized
-->
<complexType name="fieldsType">
   <sequence>
      <element name="field" type="relDom:fieldType"
         maxOccurs="unbounded"/>
   </sequence>
   <attribute name="inSync" use="required"/>
</complexType>
<!--
Domain names that are registered or available.
-->
<complexType name="domainListType">
   <sequence>
      <element name="name" type="eppcom:labelType"
         maxOccurs="unbounded"/>
   </sequence>
</complexType>
<!--
Related Domain Group
-->
<complexType name="relatedDomainGroupType">
   <sequence>
      <element name="fields" type="relDom:fieldsType"/>
      <element name="registered" type="relDom:domainListType"
         minOccurs="0"/>
      <element name="available" type="relDom:domainListType"
         minOccurs="0"/>
   </sequence>
   <attribute name="type" type="relDom:groupType"
      use="required"/>
</complexType>
<!--
```

```
Related Domain Auth Info Type
-->
<complexType name="authInfoType">
   <choice>
      <element name="pw" type="eppcom:pwAuthInfoType"/>
      <element name="ext" type="eppcom:extAuthInfoType"/>
   </choice>
</complexType>
<!--
Related Domain Period Type
-->
<complexType name="periodType">
   <simpleContent>
      <extension base="relDom:pLimitType">
         <attribute name="unit" type="relDom:pUnitType"
            use="required"/>
      </extension>
   </simpleContent>
</complexType>
<!--
Related Domain Period Limit Type
-->
<simpleType name="pLimitType">
   <restriction base="unsignedShort">
      <minInclusive value="1"/>
      <maxInclusive value="99"/>
   </restriction>
</simpleType>
<!--
Related Domain Period Unit Type
-->
<simpleType name="pUnitType">
   <restriction base="token">
      <enumeration value="y"/>
      <enumeration value="m"/>
   </restriction>
</simpleType>
<!--
Related Domain Create Request Type
-->
<complexType name="createRequestType">
   <sequence>
      <element name="name" type="eppcom:labelType"/>
      <element name="authInfo" type="relDom:authInfoType"/>
      <element name="period" type="relDom:periodType"
         minOccurs="0"/>
      <element name="lang" type="language"
         minOccurs="0"/>
   </sequence>
</complexType>
<!--
Related Domain Create Request element
-->
<element name="create">
   <complexType>
      <sequence>
         <element name="domain" type="relDom:createRequestType"
            maxOccurs="unbounded"/>
      </sequence>
```

```
   </complexType>
</element>
<!--
Related Domain Create Response type
-->
<complexType name="creDataType">
   <sequence>
      <element name="name" type="eppcom:labelType"/>
      <element name="crDate" type="dateTime"/>
      <element name="exDate" type="dateTime"
         minOccurs="0"/>
   </sequence>
</complexType>
<!--
Related Domain Create Request element
-->
<element name="creData">
   <complexType>
      <sequence>
         <element name="domain" type="relDom:creDataType"
            maxOccurs="unbounded"/>
      </sequence>
   </complexType>
</element>
<!--
Related Domain Delete Request element
-->
<element name="delete" type="relDom:domainListType"/>
<simpleType name="deleteResultType">
   <restriction base="string">
      <enumeration value="deleted"/>
      <enumeration value="pendingDelete"/>
   </restriction>
</simpleType>
<!--
Related Domain Delete Response type
-->
<complexType name="delDataType">
   <sequence>
      <element name="name" type="eppcom:labelType"/>
      <element name="result" type="relDom:deleteResultType"/>
   </sequence>
</complexType>
<!--
Related Domain Delete Response element
-->
<element name="delData">
   <complexType>
      <sequence>
         <element name="domain" type="relDom:delDataType"
            maxOccurs="unbounded"/>
      </sequence>
   </complexType>
</element>
<!--
Related Domain Update Request element
-->
<element name="update" type="relDom:domainListType"/>
<!--
```

```
Related Domain Renew type
-->
<complexType name="renewType">
   <sequence>
      <element name="name" type="eppcom:labelType"/>
      <element name="curExpDate" type="date"/>
      <element name="period" type="relDom:periodType"
         minOccurs="0"/>
   </sequence>
</complexType>
<!--
Related Domain Renew element
-->
<element name="renew">
   <complexType>
      <sequence>
         <element name="domain" type="relDom:renewType"
            maxOccurs="unbounded"/>
      </sequence>
   </complexType>
</element>
<!--
Related Domain Renew Data type
-->
<complexType name="renDataType">
   <sequence>
      <element name="name" type="eppcom:labelType"/>
      <element name="exDate" type="dateTime"/>
   </sequence>
</complexType>
<!--
Related Domain Renew Data element
-->
<element name="renData">
   <complexType>
      <sequence>
         <element name="domain" type="relDom:renDataType"
            maxOccurs="unbounded"/>
      </sequence>
   </complexType>
</element>
<!--
Related Domain Transfer type
-->
<complexType name="transferType">
   <sequence>
      <element name="name" type="eppcom:labelType"/>
      <element name="authInfo" type="relDom:authInfoType"
         minOccurs="0"/>
      <element name="period" type="relDom:periodType"
         minOccurs="0"/>
   </sequence>
</complexType>
<!--
Related Domain Transfer element
-->
<element name="transfer">
   <complexType>
      <sequence>
```

```
            <element name="domain" type="relDom:transferType"
               maxOccurs="unbounded"/>
         </sequence>
      </complexType>
   </element>
   <!--
   Related Domain Transfer Data Type
   -->
   <complexType name="trnDataType">
      <sequence>
         <element name="name" type="eppcom:labelType"/>
         <element name="trStatus" type="eppcom:trStatusType"/>
         <element name="reID" type="eppcom:clIDType"/>
         <element name="reDate" type="dateTime"/>
         <element name="acID" type="eppcom:clIDType"/>
         <element name="acDate" type="dateTime"/>
         <element name="exDate" type="dateTime" minOccurs="0"/>
      </sequence>
   </complexType>
   <!--
   Related Domain Transfer Data element
   -->
   <element name="trnData">
      <complexType>
         <sequence>
            <element name="domain" type="relDom:trnDataType"
               maxOccurs="unbounded"/>
         </sequence>
      </complexType>
   </element>
</schema>
END
```

5 References

[EPP] Hollenbeck, S., "Extensible Provisioning Protocol (EPP)", STD 69, RFC 5730, August 2009.

[EPP-D] Hollenbeck, S., "Extensible Provisioning Protocol (EPP) Domain Name Mapping", STD 69, RFC 5731, August 2009.

[EPP-ILANG] "Extensible Provisioning Protocol Extension Mapping <IDN Language Tag>", Version 1.1

[XML] Maler, E., Sperberg-McQueen, C., Bray, T., and J. Paoli, "Extensible Markup Language (XML) 1.0 (Second Edition)", World Wide Web Consortium FirstEdition REC-xml-20001006, October 2000, <http://www.w3.org/TR/2000/REC-xml-20001006>.

[XMLS-1] Beech, D., Thompson, H., Mendelsohn, N., and M. Maloney, "XML Schema Part 1: Structures", World Wide Web Consortium FirstEdition REC-xmlschema-1-20010502, May 2001, <http://www.w3.org/TR/2001/REC-xmlschema-1-20010502>.

[XMLS-2] Malhotra, A. and P. Biron, "XML Schema Part 2: Datatypes", World Wide Web Consortium FirstEdition REC-xmlschema-2-20010502, May 2001, <http://www.w3.org/TR/2001/REC-xmlschema-2-20010502>.