EXTENSIBLE PROVISIONING PROTOCOL MAPPING:
<DEFENSIVE REGISTRATION>

Version 1.2

LEGAL DISCLAIMER

COPYRIGHT NOTIFICATION

VERISIGN PROPRIETARY INFORMATION

DISCLAIMER AND LIMITATION OF LIABILITY

NOTICE AND CAUTION

Concerning U.S. Patent or Trademark Rights

Table of Contents

1 Introduction

This document describes a Defensive Registration 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 [XMLS-1] and [XMLS-2].

[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.

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.

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.

2 Object Attributes

An EPP defensive registration object has following 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 Defensive Registration Name

Defensive Registration Name will block Personal Name registrations, which include Domain objects described in [RFC3731], and Defensive Registration objects described in [EPP-E]. Defensive Registration Name has Premium and Standard name levels. A Standard Defensive Registration Name is of the format "SAMPLE.MARK" and a Premium Defensive Registration Name is of the format of "TRADEMARK". Defensive Registration Name is a character string with a specified minimum length, a specified maximum length, and a specified format.

2.2 Contact and Client Identifiers

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

2.3 Status Values

A Defensive Registration object MUST always have at least one associated status value. Status values MAY be set only by the client that sponsors a Defensive Registration object and by the server on which the object resides. A client MAY change the status of a Defensive Registration object using the EPP <update> command. Each status value MAY be accompanied by a string of human-readable text that describes the rationale for the status applied to the object.

A client MUST NOT alter status values set by the server. A server MAY alter or override status values set by a client subject to local server policies.

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

  Status Value Descriptions:

  clientDeleteProhibited, serverDeleteProhibited

  Requests to delete the object MUST be rejected.

  clientRenewProhibited, serverRenewProhibited

  Requests to renew the object MUST be rejected.

  clientTransferProhibited, serverTransferProhibited

  Requests to transfer the object MUST be rejected.

  clientUpdateProhibited, serverUpdateProhibited

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

```
ok
```

This is the nominal status value for an object that has no pending operations or prohibitions.

```
pendingDelete
```

A delete request has been received for the object, but the object has not yet been purged from the server database.

```
pendingTransfer
```

A transfer request has been received for the object, and completion of the request is pending. Transform commands other than <transfer> MUST be rejected while an object is in this state.

"ok" status MUST NOT be combined with any other status.
"pendingDelete" status MUST NOT be combined with either
"clientDeleteProhibited" or "serverDeleteProhibited" status.
"pendingTransfer" status MUST NOT be combined with either
"clientTransferProhibited" or "serverTransferProhibited" status.
All other status value combinations are valid.

2.4 Dates and Times

Date and time attribute values MUST be represented in Universal Coordinated Time (UTC) using the Gregorian calendar. The extended date-time form defined in [ISO8601] MUST be used to represent date-time values as XML Schema does not support truncated date-time forms.

2.5 Validity Periods

A Defensive Registration object MAY have a specified validity period. If server policy supports Defensive Registration object validity periods, the validity period is defined when a Defensive Registration object is created, and it MAY be extended by the EPP <renew> or <transfer> commands. As a matter of server policy, this specification does not define actions to be taken upon expiration of a Defensive Registration object's validity period.

Validity periods are measured in years or months with the appropriate units specified using the "unit" attribute. Valid values for the "unit" attribute are "y" for years and "m" for months. The minimum allowable period value is one decimal (1). The maximum allowable value is ninety-nine decimal (99). A server MAY support a lower maximum value.

2.6 Authorization Information

Authorization information is associated with Defensive Registration objects to facilitate transfer operations. Authorization information is assigned when a Defensive Registration object is created, and it MAY be updated in the future. This specification describes password-based authorization information, though other mechanisms are possible.

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 Defensive Registration via EPP.

3.1 EPP Query Commands

EPP provides three commands to retrieve Defensive Registration information:
<check> to determine if a Defensive Registration object is available for
provisioning, <info> to retrieve detailed information associated with a Defensive
Registration object, and <transfer> to retrieve Defensive Registration object
transfer status information.

3.1.1 EPP <check> Command

The EPP <check> command is used to determine if an object may be provisioned
within a repository. It provides a hint that allows a client to anticipate the
success or failure of provisioning an object using the <check> command. Object
availability and provisioning conditions are a matter of server policy.

In addition to the standard EPP command elements, the <check> command MUST
contain a <defReg:check> element that identifies the Defensive Registration
namespace and the location of the Defensive Registration schema. The
<defReg:check> element contains the following child elements:

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

```
Example <check> command:
C:<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
C:    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
C:    xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
C:    <command>
C:        <check>
C:            <defReg:check
C:                  xmlns:defReg="http://www.nic.name/epp/defReg-1.0"
C:                  xsi:schemaLocation="http://www.nic.name/epp/defReg-1.0
C:                  defReg-1.0.xsd">
C:                  <defReg:name level="premium">doe</defReg:name>
C:                  <defReg:name level="standard">john.doe</defReg:name>
C:            </defReg:check>
C:        </check>
C:        <clTRID>ABC-12345</clTRID>
C:    </command>
C:</epp>
```

When a <check> command has been processed successfully, the EPP <resData> element
MUST contain a child <defReg:chkData> element that identifies the Defensive
Registration namespace and the location of the Defensive Registration schema.
The <defReg:chkData> element contains one or more <defReg:cd> elements that
contain the following child elements:

  - A <defReg:name> element that contains the name of the queried Defensive
Registration object. This element MUST contain an "avail" attribute whose value
indicates object availability at the moment the <check> command was completed. A

value of "1" or "true" means that the object is available. A value of "0" or
"false" means that the object is not available.

   - An OPTIONAL <defReg:reason> element that MAY be provided when an object is
not available for provisioning. If present, this element contains server-
specific text to help explain why the object is unavailable.

Example <check> response:

```
S:<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
S:    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
S:    xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
S:    <response>
S:         <result code="1000">
S:               <msg>Command completed successfully</msg>
S:         </result>
S:         <resData>
S:               <defReg:chkData
S:               xmlns:defReg="http://www.nic.name/epp/defReg-1.0"
S:               xsi:schemaLocation="http://www.nic.name/epp/defReg-
S:               1.0 defReg-1.0.xsd">
S:                     <defReg:cd>
S:                           <defReg:name level="premium"
S:                           avail="1">doe</defReg:name>
S:                     </defReg:cd>
S:                     <defReg:cd>
S:                           <defReg:name level="standard"
S:                           avail="0">john.doe</defReg:name>
S:                           <defReg:reason>Conflicting object
S:                           exists</defReg:reason>
S:                     </defReg:cd>
S:               </defReg:chkData>
S:         </resData>
S:         <trID>
S:               <clTRID>ABC-12345</clTRID>
S:               <svTRID>54322-XYZ</svTRID>
S:         </trID>
S:    </response>
S:</epp>
```

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

3.1.2   EPP <info> Command

The EPP <info> command is used to retrieve information associated with a
Defensive Registration object. The response to this command MAY vary depending
on the identity of the querying client, use of authorization information, and
server policy towards unauthorized clients. If the querying client is the
sponsoring client, all available information MUST be returned. If the querying
client is not the sponsoring client, but the client provides valid authorization
information, all available information MUST be returned. If the querying client
is not the sponsoring client, and the client does not provide valid authorization
information, server policy determines which OPTIONAL elements are returned.

In addition to the standard EPP command elements, the <info> command MUST contain
a <defReg:info> element that identifies the Defensive Registration namespace and
the location of the Defensive Registration schema.
The <defReg:info> element contains the following child element:

  - A <defReg:roid > element that contains the Repository Object Identifier
assigned to the Defensive Registration object when the object was created.

  - An OPTIONAL <defReg:authInfo> element that contains authorization information
associated with the Defensive Registration object or authorization information
associated with the Defensive Registration object's registrant or associated
contacts. An OPTIONAL "roid" attribute MUST be used to identify the registrant
or contact object if and only if the given authInfo is associated with a
registrant or contact object, and not the Defensive Registration object itself.
If this element is not provided or if the authorization information is invalid,
server policy determines if the command is rejected or if response information
will be returned to the client.

Example <info> command:
```
C:<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
C:    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
C:    xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
C:      <command>
C:          <info>
C:                <defReg:info xmlns:defReg="http://www.nic.name/epp/defReg-1.0"
C:                xsi:schemaLocation="http://www.nic.name/epp/defReg-1.0
C:                defReg-1.0.xsd">
C:                     <defReg:roid>EXAMPLE1-REP</defReg:roid>
C:                </defReg:info>
C:          </info>
C:          <clTRID>ABC-12345</clTRID>
C:      </command>
C:</epp>
```

  When an <info> command has been processed successfully, the EPP <resData>
element MUST contain a child <defReg:infData> element that identifies the
Defensive Registration namespace and the location of the Defensive
Registration schema.
  The <defReg:infData> element contains the following child elements:

  - A <defReg:roid> element that contains the Repository Object Identifier
assigned to the Defensive Registration object when the object was created.

  - A <defReg:name> element that contains the name of the Defensive Registration
object.

- An OPTIONAL <defReg:registrant> element that contains the identifier for the human or organizational social information (contact) object to be associated with the Defensive Registration object as the object registrant. This object identifier MUST be known to the server before the contact object can be associated with the Defensive Registration object.

   - An OPTIONAL <defReg:tm> element that contains the Trademark Identifier (ID) associated with the Defensive Registration Object.

   - An OPTIONAL <defReg:tmCountry> element that indicates the Country which issued the Trademark associated with the Defensive Registration Object.

   - An OPTIONAL <defReg:tmDate> element that indicates the date when the Trademark was issued.

   - An OPTIONAL <defReg:adminContact> element that contains identifier for the administrator associated with the Defensive Registration object.

   - Zero or more OPTIONAL <defReg:status> elements that contain the current status descriptors associated with the Defensive Registration.

   - A <defReg:clID> element that contains the identifier of the sponsoring client.

   - An OPTIONAL <defReg:crID> element that contains the identifier of the client that created the Defensive Registration object.

   - An OPTIONAL <defReg:crDate> element that contains the date and time of Defensive Registration object creation.

   - An OPTIONAL <defReg:upID> element that contains the identifier of the client that last updated the Defensive Registration object. This element MUST NOT be present if the Defensive Registration has never been modified.

   - An OPTIONAL <defReg:upDate> element that contains the date and time of the most recent Defensive Registration object modification. This element MUST NOT be present if the Defensive Registration object has never been modified.

   - An OPTIONAL <defReg:exDate> element that contains the date and time identifying the end of the Defensive Registration object's registration period.

   - An OPTIONAL <defReg:trDate> element that contains the date and time of the most recent successful Defensive Registration object transfer. This element MUST NOT be provided if the Defensive Registration object has never been transferred.

   - An OPTIONAL <defReg:authInfo> element that contains authorization information associated with the Defensive Registration object. This element MUST NOT be provided if the querying client is not the current sponsoring client, or if the client supplied valid authorization information with the command.

Example <info> response for an authorized client:

S:<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
S: xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
S: xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
S: <response>
S: <result code="1000">
S: <msg>Command completed successfully</msg>
S: </result>
S: <resData>

```
S:                    <defReg:infData
S:                    xmlns:defReg="http://www.nic.name/epp/defReg-1.0"
S:                    xsi:schemaLocation="http://www.nic.name/epp/defReg-1.0
S:                    defReg-1.0.xsd">
S:                        <defReg:roid>EXAMPLE1-REP</defReg:roid>
S:                        <defReg:name level="premium">doe</defReg:name>
S:                        <defReg:registrant>jd1234</defReg:registrant>
S:                        <defReg:tm>XYZ-123</defReg:tm>
S:                        <defReg:tmCountry>US</defReg:tmCountry>
S:                        <defReg:tmDate>1990-04-03</defReg:tmDate>
S:                        <defReg:adminContact>sh8013</defReg:adminContact>
S:                        <defReg:status s="ok" />
S:                        <defReg:clID>ClientX</defReg:clID>
S:                        <defReg:crID>ClientY</defReg:crID>
S:                        <defReg:crDate>1999-04-03T22:00:00.0Z</defReg:crDate>
S:                        <defReg:upID>ClientX</defReg:upID>
S:                        <defReg:upDate>1999-12-03T09:00:00.0Z</defReg:upDate>
S:                        <defReg:exDate>2000-04-03T22:00:00.0Z</defReg:exDate>
S:                        <defReg:trDate>2000-01-08T09:00:00.0Z</defReg:trDate>
S:                        <defReg:authInfo>
S:                            <defReg:pw>2fooBAR</defReg:pw>
S:                        </defReg:authInfo>
S:                    </defReg:infData>
S:            </resData>
S:            <trID>
S:                    <clTRID>ABC-12345</clTRID>
S:                    <svTRID>54322-XYZ</svTRID>
S:            </trID>
S:     </response>
S:</epp>


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

Example <info> response for an unauthorized client:

S:<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
S:     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
S:     xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
S:     <response>
S:            <result code="1000">
S:                    <msg>Command completed successfully</msg>
S:            </result>
S:            <resData>
S:                    <defReg:infData
S:                    xmlns:defReg="http://www.nic.name/epp/defReg-1.0"
S:                    xsi:schemaLocation="http://www.nic.name/epp/defReg-1.0
S:                    defReg-1.0.xsd">
S:                        <defReg:roid>EXAMPLE1-REP</defReg:roid>
S:                        <defReg:name level="premium">doe</defReg:name>
S:                        <defReg:clID>ClientX</defReg:clID>
S:                    </defReg:infData>
S:            </resData>
S:            <trID>
S:                    <clTRID>ABC-12345</clTRID>
S:                    <svTRID>54322-XYZ</svTRID>
S:            </trID>
S:     </response>
S:</epp>
```

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 <defReg:transfer> element that identifies the Defensive Registration namespace and the location of the Defensive Registration schema. The <defReg:transfer> element contains the following child elements:

 - A <defReg:roid> element that contains the Repository Object Identifier assigned to the Defensive Registration object when the object was created.

 - An OPTIONAL <defReg:authInfo> element that contains authorization information associated with the Defensive Registration object or authorization information associated with the Defensive Registration object's registrant or associated contacts. An OPTIONAL "roid" attribute MUST be used to identify the registrant or contact object if and only if the given authInfo is associated with a registrant or contact object, and not the Defensive Registration object itself. If this element is not provided or if the authorization information is invalid, server policy determines if the command is rejected or if response information will be returned to the client.

  Example <transfer> query command:
```
C:<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
C:    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
C:    xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
C:    <command>
C:        <transfer op="query">
C:            <defReg:transfer
C:            xmlns:defReg="http://www.nic.name/epp/defReg-1.0"
C:            xsi:schemaLocation="http://www.nic.name/epp/defReg-1.0
C:            defReg-1.0.xsd">
C:                <defReg:roid>EXAMPLE1-REP</defReg:roid>
C:                <defReg:authInfo>
C:                    <defReg:pw roid="JD1234-REP">2fooBAR</defReg:pw>
C:                </defReg:authInfo>
C:            </defReg:transfer>
C:        </transfer>
C:        <clTRID>ABC-12345</clTRID>
C:    </command>
C:</epp>
```

When a <transfer> query command has been processed successfully, the EPP <resData> element MUST contain a child <defReg:trnData> element that identifies the Defensive Registration namespace and the location of the Defensive Registration schema. The <defReg:trnData> element contains the following child elements:

 - A <defReg:roid> element that contains the Repository Object Identifier assigned to the Defensive Registration object when the object was created.

 - A <defReg:trStatus> element that contains the state of the most recent transfer request.

 - A <defReg:reID> element that contains the identifier of the client that requested the object transfer.

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

   - A <defReg:acID> element that contains the identifier of the client that
SHOULD act upon the transfer request.

   - A <defReg:acDate> 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 <defReg:exDate> element that contains the end of the Defensive
Registration object'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" encoding="UTF-8" standalone="no" ?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
S:    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
S:    xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
S:    <response>
S:         <result code="1000">
S:              <msg>Command completed successfully</msg>
S:         </result>
S:         <resData>
S:              <defReg:trnData
S:              xmlns:defReg="http://www.nic.name/epp/defReg-1.0"
S:              xsi:schemaLocation="http://www.nic.name/epp/defReg-1.0
S:              defReg-1.0.xsd">
S:                   <defReg:roid>EXAMPLE1-REP</defReg:roid>
S:                   <defReg:trStatus>pending</defReg:trStatus>
S:                   <defReg:reID>ClientX</defReg:reID>
S:                   <defReg:reDate>2000-06-06T22:00:00.0Z</defReg:reDate>
S:                   <defReg:acID>ClientY</defReg:acID>
S:                   <defReg:acDate>2000-06-11T22:00:00.0Z</defReg:acDate>
S:                   <defReg:exDate>2002-09-08T22:00:00.0Z</defReg:exDate>
S:              </defReg:trnData>
S:         </resData>
S:         <trID>
S:              <clTRID>ABC-12345</clTRID>
S:              <svTRID>54322-XYZ</svTRID>
S:    </response>
S:</epp>
```

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

3.2 EPP Transform Commands

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

3.2.1 EPP <create> Command

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

   - A <defReg:name> element that contains the name of the Defensive Registration object to be created.

   - A <defReg:registrant> element that contains the identifier for the human or organizational social information (contact) object to be associated with the Defensive Registration object as the object registrant. This object identifier MUST be known to the server before the contact object can be associated with the Defensive Registration object.

   - An OPTIONAL <defReg:tm> element contains the Trademark Identifier (ID) associated with the Defensive Registration Object..

   - An OPTIONAL <defReg:tmCountry> element that indicates the Country which issued the Trademark associated with the Defensive Registration Object.

   - An OPTIONAL <defReg:tmDate> element that indicates the date when the Trademark was issued.

   - An OPTIONAL <defReg:period> element that contains the initial registration period of the Defensive Registration object. A server MAY define a default initial registration period if not specified by the client.

   - A <defReg:adminContact> element that contains identifier for the administrator associated with the Defensive Registration object.

   - A <defReg:authInfo> element that contains authorization information to be associated with the Defensive Registration object.

   Example <create> command:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
C:    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
C:    xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
C:    <command>
C:         <create>
C:              <defReg:create
C:              xmlns:defReg="http://www.nic.name/epp/defReg-1.0"
C:              xsi:schemaLocation="http://www.nic.name/epp/defReg-1.0
```

```
C:                    defReg-1.0.xsd">
C:                        <defReg:name level="premium">doe</defReg:name>
C:                        <defReg:registrant>jd1234</defReg:registrant>
C:                        <defReg:tm>XYZ-123</defReg:tm>
C:                        <defReg:tmCountry>US</defReg:tmCountry>
C:                        <defReg:tmDate>1990-04-03</defReg:tmDate>
C:                        <defReg:adminContact>sh8013</defReg:adminContact>
C:                        <defReg:authInfo>
C:                            <defReg:pw>2fooBAR</defReg:pw>
C:                        </defReg:authInfo>
C:                </defReg:create>
C:            </create>
C:            <clTRID>ABC-12345</clTRID>
C:        </command>
C:</epp>
```

If a Defensive Registration object needs to be created for IDN, the EPP command MUST contain an <extension> element in addition to above EPP Command elements and the <extension> element MUST contain a child <idnLang:tag> element as shown below.

Example <create> command for **IDN:**

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
C:    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
C:    xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
C:    <command>
C:        <create>
C:            <defReg:create
C:            xmlns:defReg="http://www.nic.name/epp/defReg-1.0"
C:            xsi:schemaLocation="http://www.nic.name/epp/defReg-1.0
C:            defReg-1.0.xsd">
C:                        <defReg:name level="premium">xn--gya</defReg:name>
C:                        <defReg:registrant>jd1234</defReg:registrant>
C:                        <defReg:tm>XYZ-123</defReg:tm>
C:                        <defReg:tmCountry>US</defReg:tmCountry>
C:                        <defReg:tmDate>1990-04-03</defReg:tmDate>
C:                        <defReg:adminContact>sh8013</defReg:adminContact>
C:                        <defReg:authInfo>
C:                            <defReg:pw>2fooBAR</defReg:pw>
C:                        </defReg:authInfo>
C:                </defReg:create>
C:        </create>
C:        <extension>
C:            <idnLang:tag
C:             xmlns:idnLang="http://www.verisign.com/epp/idnLang-1.0"
C:    xsi:schemaLocation="http://www.verisign.com/epp/idnLang-1.0 idnLang-
C:    1.0.xsd">GRE</idnLang:tag>
C:        </extension>
C:        <clTRID>ABC-12345</clTRID>
C:    </command>
C:</epp>
```

When a <create> command has been processed successfully, the EPP <resData> element MUST contain a child <defReg:creData> element that identifies the Defensive Registration namespace and the location of the Defensive Registration schema.
   The <defReg:creData> element contains the following child elements:

- A <defReg:roid> element that contains the Repository Object Identifier
assigned to the Defensive Registration object when the object was created.

  - A <defReg:name> element that contains the string of the Defensive
Registration object.

  - A <defReg:crDate> element that contains the date and time of Defensive
Registration object creation.

  - A <defReg:exDate> element that contains the date and time identifying the
end of the Defensive Registration object's registration period.

Example <create> response:

```
S:<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
S:    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
S:    xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
S:    <response>
S:         <result code="1000">
S:              <msg>Command completed successfully</msg>
S:         </result>
S:         <resData>
S:              <defReg:creData
S:              xmlns:defReg="http://www.nic.name/epp/defReg-1.0"
S:              xsi:schemaLocation="http://www.nic.name/epp/defReg-1.0
S:              defReg-1.0.xsd">
S:                    <defReg:roid>EXAMPLE1-REP</defReg:roid>
S:                    <defReg:name level="premium">doe</defReg:name>
S:                    <defReg:crDate>1999-04-03T22:00:00.0Z</defReg:crDate>
S:                    <defReg:exDate>2000-04-03T22:00:00.0Z</defReg:exDate>
S:              </defReg:creData>
S:         </resData>
S:         <trID>
S:              <clTRID>ABC-12345</clTRID>
S:              <svTRID>54321-XYZ</svTRID>
S:         </trID>
S:    </response>
S:</epp>
```

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

3.2.2 EPP <delete> Command

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

  - A <defReg:roid> element that contains the Repository Object Identifier
assigned to the Defensive Registration object when the object was created.
  Example <delete> command:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
C:    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
C:    xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
C:    <command>
C:        <delete>
C:            <defReg:delete
C:            xmlns:defReg="http://www.nic.name/epp/defReg-1.0"
C:            xsi:schemaLocation="http://www.nic.name/epp/defReg-1.0
C:            defReg-1.0.xsd">
C:                <defReg:roid>EXAMPLE1-REP</defReg:roid>
C:            </defReg:delete>
C:        </delete>
C:        <clTRID>ABC-12345</clTRID>
C:    </command>
C:</epp>
```

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

```
  Example <delete> response:
S:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
S:    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
S:    xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0
S:    epp-1.0.xsd">
S:  <response>
S:    <result code="1000">
S:      <msg>Command completed successfully</msg>
S:    </result>
S:    <trID>
S:      <clTRID>ABC-12345</clTRID>
S:      <svTRID>54321-XYZ</svTRID>
S:    </trID>
S:  </response>
S:</epp>
```

An EPP error response MUST be returned if a <delete> command can 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 Defensive Registration object. In addition to the standard EPP command elements, the <renew> command MUST contain a <defReg:renew> element that identifies the Defensive Registration namespace and the location of the Defensive Registration schema. The <defReg:renew> element contains the following child elements:

  - A <defReg:roid> element that contains the Repository Object Identifier assigned to the Defensive Registration object when the object was created.

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

  - An OPTIONAL <defReg:period> element that contains the number of units to be added to the registration period of the Defensive Registration object.

  Example <renew> command:
```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
C:     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
C:     xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0
C:     epp-1.0.xsd">
C:  <command>
C:    <renew>
C:      <defReg:renew
C:       xmlns:defReg="http://www.nic.name/epp/defReg-1.0"
C:       xsi:schemaLocation="http://www.nic.name/epp/defReg-1.0
C:          defReg-1.0.xsd">
C:       <defReg:roid>EXAMPLE1-REP</defReg:roid>
C:       <defReg:curExpDate>2000-04-03</defReg:curExpDate>
C:       <defReg:period unit="y">1</defReg:period>
C:      </defReg:renew>
C:    </renew>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

When a <renew> command has been processed successfully, the EPP <resData> element MUST contain a child <defReg:renData> element that identifies the Defensive Registration namespace and the location of the Defensive Registration schema. The <defReg:renData> element contains the following child elements:

  - A <defReg:roid> element that contains the Repository Object Identifier assigned to the Defensive Registration object when the object was created.

  - A <defReg:exDate> element that contains the date and time identifying the end of the Defensive Registration object's registration period.

  Example <renew> response:
```
S:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
S:     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
S:     xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0
S:     epp-1.0.xsd">
```

```
S:    <response>
S:      <result code="1000">
S:         <msg>Command completed successfully</msg>
S:      </result>
S:      <resData>
S:        <defReg:renData
S:         xmlns:defReg="http://www.nic.name/epp/defReg-1.0"
S:         xsi:schemaLocation="http://www.nic.name/epp/defReg-1.0
S:           defReg-1.0.xsd">
S:          <defReg:roid>EXAMPLE1-REP</defReg:roid>
S:          <defReg:exDate>2001-04-03T22:00:00.0Z</defReg:exDate>
S:        </defReg:renData>
S:      </resData>
S:      <trID>
S:        <clTRID>ABC-12345</clTRID>
S:        <svTRID>54322-XYZ</svTRID>
S:      </trID>
S:    </response>
S:</epp>
```

An EPP error response MUST be returned if a <renew> command can 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 Defensive Registration object. In addition to the standard EPP command elements, the <transfer> command MUST contain a <defReg:transfer> element that identifies the Defensive Registration namespace and the location of the Defensive Registration schema.
  The <defReg:transfer> element contains the following child elements:

  - A <defReg:roid> element that contains the Repository Object Identifier assigned to the Defensive Registration object when the object was created.

  - An OPTIONAL <defReg:period> element that contains the number of units to be added to the registration period of the Defensive Registration object at completion of the transfer process. This element MAY only be used when a transfer is requested, and it MUST be ignored if used otherwise.

  - A <defReg:authInfo> element that contains authorization information associated with the Defensive Registration object or authorization information associated with the Defensive Registration object's registrant or associated contacts. An OPTIONAL "roid" attribute MUST be used to identify the registrant or contact object if and only if the given authInfo is associated with a registrant or contact object, and not the Defensive Registration object itself.

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].

  Example <transfer> request command:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
C:     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
C:     xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0
C:     epp-1.0.xsd">
C:  <command>
C:    <transfer op="request">
C:      <defReg:transfer
C:       xmlns:defReg="http://www.nic.name/epp/defReg-1.0"
C:       xsi:schemaLocation="http://www.nic.name/epp/defReg-1.0
C:       defReg-1.0.xsd">
C:       <defReg:roid>EXAMPLE1-REP</defReg:roid>
C:       <defReg:period unit="y">1</defReg:period>
C:       <defReg:authInfo>
C:           <defReg:pw>2fooBAR</defReg:pw>
C:       </defReg:authInfo>
C:      </defReg:transfer>
C:    </transfer>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

When a <transfer> command has been processed successfully, the EPP <resData> element MUST contain a child <defReg:trnData> element that identifies the Defensive Registration namespace and the location of the Defensive Registration

schema. The <defReg:trnData> element contains the same child elements defined for a transfer query response.

Example <transfer> response:

```
S:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
S:     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
S:     xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0
S:     epp-1.0.xsd">
S:  <response>
S:    <result code="1000">
S:      <msg>Command completed successfully</msg>
S:    </result>
S:    <resData>
S:      <defReg:trnData
S:       xmlns:defReg="http://www.nic.name/epp/defReg-1.0"
S:       xsi:schemaLocation="http://www.nic.name/epp/defReg-1.0
S:       defReg-1.0.xsd">
S:        <defReg:roid>EXAMPLE1-REP</defReg:roid>
S:        <defReg:trStatus>pending</defReg:trStatus>
S:        <defReg:reID>ClientX</defReg:reID>
S:        <defReg:reDate>2000-06-08T22:00:00.0Z</defReg:reDate>
S:        <defReg:acID>ClientY</defReg:acID>
S:        <defReg:acDate>2000-06-13T22:00:00.0Z</defReg:acDate>
S:        <defReg:exDate>2002-09-08T22:00:00.0Z</defReg:exDate>
S:      </defReg:trnData>
S:    </resData>
S:    <trID>
S:      <clTRID>ABC-12345</clTRID>
S:      <svTRID>54322-XYZ</svTRID>
S:    </trID>
S:  </response>
S:</epp>
```

An EPP error response MUST be returned if a <transfer> command cannot 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 Defensive Registration object. In addition to the standard EPP command elements, the <update> command MUST contain a <defReg:update> element that identifies the Defensive Registration namespace and the location of the Defensive Registration schema. The <defReg:update> element contains the following child elements:

  - A <defReg:roid> element that contains the Repository Object Identifier assigned to the Defensive Registration object when the object was created.

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

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

  - An OPTIONAL <defReg:chg> element that contains object attribute values to be changed.

At least one <defReg:add>, <defReg:rem>, or <defReg:chg> element MUST be provided. The <defReg:add> and <defReg:rem> elements contain the following child elements:

  - Zero or more <defReg:status> elements that contain status values to be applied to or removed from the object. When specifying a value to be removed, only the attribute value is significant; element text is not required to match a value for removal.

A <defReg:chg> element contains the following child elements:

  - A <defReg:registrant> element that contains the identifier for the human or organizational social information (contact) object to be associated with the Defensive Registration object as the object registrant. This object identifier MUST be known to the server before the contact object can be associated with the Defensive Registration object.

  - A <defReg:tm> element contains the Trademark Identifier (ID) associated with the Defensive Registration Object..

  - A <defReg:tmCountry> element that indicates the Country which issued the Trademark associated with the Defensive Registration Object.

  - A <defReg:tmDate> element that indicates the date when the Trademark was issued.

  - A <defReg:adminContact> element that contains identifier for the administrator associated with the Defensive Registration object.

  - A <defReg:authInfo> element that contains authorization information associated with the Defensive Registration object. This mapping includes a password-based authentication mechanism, but the schema allows new mechanisms to be defined in new schemas. A <defReg:null> element can be used within the <defReg:authInfo> element to remove authorization information.

Example <update> command:

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

```
C: <epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
C:    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
C:    xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0
C:    epp-1.0.xsd">
C:  <command>
C:    <update>
C:      <defReg:update
C:       xmlns:defReg="http://www.nic.name/epp/defReg-1.0"
C:       xsi:schemaLocation="http://www.nic.name/epp/defReg-1.0
C:       defReg-1.0.xsd">
C:        <defReg:roid>EXAMPLE1-REP</defReg:roid>
C:        <defReg:add>
C:          <defReg:status s="clientDeleteProhibited" lang="en">
C:                Deletions not desired.
C:          </defReg:status>
C:        </defReg:add>
C:        <defReg:rem>
C:          <defReg:status s="clientUpdateProhibited"/>
C:        </defReg:rem>
C:        <defReg:chg>
C:          <defReg:registrant>sh8013</defReg:registrant>
C:          <defReg:tm>XYZ-123</defReg:tm>
C:          <defReg:tmCountry>US</defReg:tmCountry>
C:          <defReg:tmDate>1990-04-03</defReg:tmDate>
C:          <defReg:adminContact>sh8013</defReg:adminContact>
C:          <defReg:authInfo>
C:              <defReg:pw>2BARfoo</defReg:pw>
C:          </defReg:authInfo>
C:        </defReg:chg>
C:      </defReg:update>
C:    </update>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

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

Example <update> rsponse:

```
S:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
S:    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
S:    xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0
S:    epp-1.0.xsd">
S:  <response>
S:    <result code="1000">
S:      <msg>Command completed successfully</msg>
S:    </result>
S:    <trID>
S:      <clTRID>ABC-12345</clTRID>
S:      <svTRID>54321-XYZ</svTRID>
S:    </trID>
S:  </response>
S:</epp>
```

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

4 Formal Syntax

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

```
BEGIN

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

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

<!--
Import common element types.
-->
  <import namespace="urn:ietf:params:xml:ns:eppcom-1.0"
          schemaLocation="eppcom-1.0.xsd"/>
  <import namespace="urn:ietf:params:xml:ns:epp-1.0"
          schemaLocation="epp-1.0.xsd"/>

  <annotation>
    <documentation>
      Extensible Provisioning Protocol v1.0
      .name defensive registration provisioning schema.
    </documentation>
  </annotation>

<!--
Child elements found in EPP commands.
-->
  <element name="check" type="defReg:checkType"/>
  <element name="create" type="defReg:createType"/>
  <element name="delete" type="defReg:roidOnlyType"/>
  <element name="info" type="defReg:infoType"/>
  <element name="renew" type="defReg:renewType"/>
  <element name="transfer" type="defReg:transferType"/>
  <element name="update" type="defReg:updateType"/>

<!--
Child elements of the <check> command.
-->
  <complexType name="checkType">
    <sequence>
      <element name="name" type="defReg:nameType"
       maxOccurs="unbounded"/>
    </sequence>
  </complexType>

  <complexType name="nameType">
    <simpleContent>
      <extension base="eppcom:labelType">
        <attribute name="level" type="defReg:levelType"
```

VeriSign Inc. Proprietary Information

```
          use="required"/>
        </extension>
      </simpleContent>
    </complexType>

    <simpleType name="levelType">
      <restriction base="token">
        <enumeration value="premium"/>
        <enumeration value="standard"/>
      </restriction>
    </simpleType>

<!--
Child elements of the <create> command.
-->
    <complexType name="createType">
      <sequence>
        <element name="name" type="defReg:nameType"/>
        <element name="registrant" type="eppcom:clIDType"/>
        <element name="tm" type="defReg:tmType"
         minOccurs="0"/>
        <element name="tmCountry" type="defReg:ccType"
         minOccurs="0"/>
        <element name="tmDate" type="date"
         minOccurs="0"/>
        <element name="adminContact" type="eppcom:clIDType"/>
        <element name="period" type="defReg:periodType"
         minOccurs="0"/>
        <element name="authInfo" type="defReg:authInfoType"/>
      </sequence>
    </complexType>

    <complexType name="authInfoType">
      <choice>
            <element name="pw" type="eppcom:pwAuthInfoType"/>
            <element name="ext" type="eppcom:extAuthInfoType"/>
      </choice>
    </complexType>

    <simpleType name="ownerType">
      <restriction base="token">
        <minLength value="1"/>
        <maxLength value="64"/>
      </restriction>
    </simpleType>

    <simpleType name="tmType">
      <restriction base="token">
        <minLength value="1"/>
        <maxLength value="64"/>
      </restriction>
    </simpleType>

    <simpleType name="ccType">
      <restriction base="token">
        <length value="2"/>
      </restriction>
    </simpleType>

    <complexType name="periodType">
      <simpleContent>
```

```
        <extension base="defReg:pLimitType">
          <attribute name="unit" type="defReg:pUnitType"
           use="required"/>
        </extension>
      </simpleContent>
    </complexType>

    <simpleType name="pLimitType">
      <restriction base="unsignedShort">
        <minInclusive value="1"/>
        <maxInclusive value="99"/>
      </restriction>
    </simpleType>

    <simpleType name="pUnitType">
      <restriction base="token">
        <enumeration value="y"/>
        <enumeration value="m"/>
      </restriction>
    </simpleType>

<!--
Child element of commands that require a ROID.
-->
    <complexType name="roidOnlyType">
      <sequence>
        <element name="roid" type="eppcom:roidType"/>
      </sequence>
    </complexType>


<!--
Child elements of the <info> command.
-->
<complexType name="infoType">
      <sequence>
              <element name="roid" type="eppcom:roidType"/>
              <element name="authInfo" type="defReg:authInfoType"
               minOccurs="0"/>
      </sequence>
</complexType>


<!--
Child elements of the <renew> command.
-->
    <complexType name="renewType">
      <sequence>
        <element name="roid" type="eppcom:roidType"/>
        <element name="curExpDate" type="date"/>
        <element name="period" type="defReg:periodType"
         minOccurs="0"/>
      </sequence>
    </complexType>

<!--
Child elements of the <transfer> command.
-->
    <complexType name="transferType">
      <sequence>
        <element name="roid" type="eppcom:roidType"/>
```

```
        <element name="period" type="defReg:periodType"
         minOccurs="0"/>
        <element name="authInfo" type="defReg:authInfoType" minOccurs="0"/>
      </sequence>
    </complexType>

<!--
Child elements of the <update> command.
-->
    <complexType name="updateType">
      <sequence>
        <element name="roid" type="eppcom:roidType"/>
        <element name="add" type="defReg:addRemType"
         minOccurs="0"/>
        <element name="rem" type="defReg:addRemType"
         minOccurs="0"/>
        <element name="chg" type="defReg:chgType"
         minOccurs="0"/>
      </sequence>
    </complexType>

<!--
Data elements that can be added or removed.
-->
    <complexType name="addRemType">
      <sequence>
        <element name="status" type="defReg:statusType"
         minOccurs="0" maxOccurs="12"/>
      </sequence>
    </complexType>


<!--
Data elements that can be changed.
-->
    <complexType name="chgType">
      <sequence>
        <element name="registrant" type="eppcom:clIDType"
         minOccurs="0"/>
        <element name="tm" type="defReg:tmType"
         minOccurs="0"/>
        <element name="tmCountry" type="defReg:ccType"
         minOccurs="0"/>
        <element name="tmDate" type="date"
         minOccurs="0"/>
        <element name="adminContact" type="eppcom:clIDType"
         minOccurs="0"/>
        <element name="authInfo" type="defReg:authInfoChgType"
         minOccurs="0"/>
      </sequence>
    </complexType>



<!--
Allow the authInfo value to be nullified by including an
empty element within the choice.
-->
    <complexType name="authInfoChgType">
        <choice>
              <element name="pw" type="eppcom:pwAuthInfoType"/>
```

```
                <element name="ext" type="eppcom:extAuthInfoType"/>
                <element name="null"/>
            </choice>
        </complexType>

<!--
Child response elements.
-->
    <element name="chkData" type="defReg:chkDataType"/>
    <element name="creData" type="defReg:creDataType"/>
    <element name="infData" type="defReg:infDataType"/>
    <element name="renData" type="defReg:renDataType"/>
    <element name="trnData" type="defReg:trnDataType"/>

<!--
<check> response elements.
-->
    <complexType name="chkDataType">
        <sequence>
            <element name="cd" type="defReg:checkResType"
             maxOccurs="unbounded"/>
        </sequence>
    </complexType>

    <complexType name="checkResType">
        <sequence>
            <element name="name" type="defReg:checkNameType"/>
            <element name="reason" type="eppcom:reasonType"
             minOccurs="0"/>
        </sequence>
    </complexType>

    <complexType name="checkNameType">
        <simpleContent>
            <extension base="defReg:nameType">
                <attribute name="avail" type="boolean"
                 use="required"/>
            </extension>
        </simpleContent>
    </complexType>

<!--
<create> response elements.
-->
    <complexType name="creDataType">
        <sequence>
            <element name="roid" type="eppcom:roidType"/>
            <element name="name" type="defReg:nameType"/>
            <element name="crDate" type="dateTime"/>
            <element name="exDate" type="dateTime"
             minOccurs="0"/>
        </sequence>
    </complexType>

<!--
<info> response elements.
-->
    <complexType name="infDataType">
        <sequence>
            <element name="roid" type="eppcom:roidType"/>
            <element name="name" type="defReg:nameType"/>
```

```xml
    <element name="registrant" type="eppcom:clIDType"
     minOccurs="0"/>
    <element name="tm" type="defReg:tmType"
     minOccurs="0"/>
    <element name="tmCountry" type="defReg:ccType"
     minOccurs="0"/>
    <element name="tmDate" type="date"
     minOccurs="0"/>
    <element name="adminContact" type="eppcom:clIDType"
     minOccurs="0"/>
    <element name="status" type="defReg:statusType"
        minOccurs="0" maxOccurs="12"/>
    <element name="clID" type="eppcom:clIDType"/>
    <element name="crID" type="eppcom:clIDType"
     minOccurs="0"/>
    <element name="crDate" type="dateTime"
     minOccurs="0"/>
    <element name="upID" type="eppcom:clIDType"
     minOccurs="0"/>
    <element name="upDate" type="dateTime"
     minOccurs="0"/>
    <element name="exDate" type="dateTime"
     minOccurs="0"/>
    <element name="trDate" type="dateTime"
     minOccurs="0"/>
    <element name="authInfo" type="defReg:authInfoType"
     minOccurs="0"/>
   </sequence>
  </complexType>

<!--
Status is a combination of attributes and an optional human-readable
message that may be expressed in languages other than English.
A list of messages should be defined in the Programmer's Guide for
this mapping.
-->
  <complexType name="statusType">
    <simpleContent>
      <extension base="normalizedString">
        <attribute name="s" type="defReg:statusValueType"
         use="required"/>
        <attribute name="lang" type="language"
         default="en"/>
      </extension>
    </simpleContent>
  </complexType>

  <simpleType name="statusValueType">
    <restriction base="token">
      <enumeration value="clientDeleteProhibited"/>
      <enumeration value="clientRenewProhibited"/>
      <enumeration value="clientTransferProhibited"/>
      <enumeration value="clientUpdateProhibited"/>
       <enumeration value="ok"/>
      <enumeration value="pendingDelete"/>
      <enumeration value="pendingTransfer"/>
      <enumeration value="serverDeleteProhibited"/>
      <enumeration value="serverRenewProhibited"/>
      <enumeration value="serverTransferProhibited"/>
      <enumeration value="serverUpdateProhibited"/>
    </restriction>
```

```
    </simpleType>

<!--
<renew> response elements.
-->
  <complexType name="renDataType">
    <sequence>
      <element name="roid" type="eppcom:roidType"/>
      <element name="exDate" type="dateTime"
       minOccurs="0"/>
    </sequence>
  </complexType>

<!--
<transfer> response elements.
-->
  <complexType name="trnDataType">
    <sequence>
      <element name="roid" type="eppcom:roidType"/>
      <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>

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

END
```

5 References

[RFC3730] Hollenbeck, S., "Extensible Provisioning Protocol (EPP)", RFC 3730, March 2004.

[RFC3731] Hollenbeck, S., "Extensible Provisioning Protocol Domain Name Mapping", RFC 3731, March 2004.

[EPP-E] "Extensible Provisioning Protocol Defensive Registration Mapping".

[IETF-XML] M. Mealling: "The IETF XML Registry".

[ISO8601] ISO 8601:1988 (E): "Data elements and interchange formats -Information interchange - Representation of dates and times - The International Organization for Standardization".

[RFC2119] S. Bradner: "Key Words for Use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[XML] Editors T. Bray et al.: "Extensible Markup Language (XML) 1.0 (Second Edition)", W3C Recommendation 6 October 2000.

[XMLS-1] Editors H. Thompson et al.: "XML Schema Part 1: Structures", W3C Recommendation 2 May 2001.

[XMLS-2] Editors P. Biron, A. Malhotra: "XML Schema Part 2: Datatypes", W3C Recommendation 2 May 2001.