



EXTENSIBLE PROVISIONING PROTOCOL MAPPING: <EMAIL FORWARDING>

Version 1.3

LEGAL DISCLAIMER

COPYRIGHT NOTIFICATION

Copyright © 2024, 2012 VeriSign, Inc. All rights reserved. VERISIGN; the Verisign logo; and other trademarks, service marks and Verisign designs are registered or unregistered trademarks of VeriSign Inc. and its subsidiaries in the United States and foreign countries. Copyright laws and international treaties protect this document, and any Verisign product to which it relates.

VERISIGN PROPRIETARY INFORMATION

This document is the property of VeriSign, Inc. and its subsidiaries (together "Verisign") It may be used by recipient only for the purpose for which it was transmitted and must be returned upon request or when no longer needed by recipient. It may not be copied or communicated without the prior written consent of Verisign.

DISCLAIMER AND LIMITATION OF LIABILITY

Verisign has made efforts to ensure the accuracy and completeness of the information in this document. However, Verisign makes no warranties of any kind (whether express, implied or statutory) with respect to the information contained herein. Verisign assumes no liability to any party for any loss or damage (whether direct or indirect) caused by any errors, omissions or statements of any kind contained in this document. Further, Verisign assumes no liability arising from the application or use of the product or service described herein and specifically disclaims any representation that the products or services described herein do not infringe upon any existing or future intellectual property rights. Nothing herein grants the reader any license to make, use or sell equipment or products constructed in accordance with this document. Finally, all rights and privileges related to any intellectual property right described herein are vested in the patent, trademark or service mark owner and no other person may exercise such rights without express permission, authority or license secured from the patent, trademark or service mark owner. Verisign reserves the right to make changes to any information herein without further notice.

Any statements contained within this document concerning Verisign's future prospects are "forward looking statements" under the Federal Securities laws. There can be no assurance that future results will be achieved and actual results could differ materially from forecasts, estimates, and summary information contained in the document. Important factors that could cause actual results to differ materially include but are not limited to factors discussed in Verisign's SEC filings.

NOTICE AND CAUTION

Concerning U.S. Patent or Trademark Rights

Verisign and other trademarks, service marks and logos are registered or unregistered trademarks of Verisign and its subsidiaries in the United States and in foreign countries. The inclusion in this document, the associated on-line file or the associated software of any information covered by any other patent, trademark or service mark rights does not constitute nor imply a grant of or authority to exercise, any right or privilege protected by such patent, trademark or service mark. All such rights and privileges are vested in the patent, trademark or service mark owner and no other person may exercise such rights without express permission, authority or license secured from the patent, trademark or service mark owner.

Table of Contents

1	INTRODUCTION	1
2	OBJECT ATTRIBUTES.....	2
2.1	Email Forwarding Name.....	2
2.2	Contact and Client Identifiers	2
2.3	Status Values.....	2
2.4	Dates and Times.....	4
2.5	Validity Periods	4
2.6	Authorization Information	4
3	EPP COMMAND MAPPING	5
3.1	EPP Query Commands	5
3.1.1	EPP <check> Command	5
3.1.2	EPP <info> Command.....	7
3.1.3	EPP <transfer> Command	11
3.2	EPP Transform Commands.....	13
3.2.1	EPP <create> Command.....	13
3.2.2	EPP <delete> Command.....	15
3.2.3	EPP <renew> Command.....	16
3.2.4	EPP <transfer> Command	18
3.2.5	EPP <update> Command.....	20
4	FORMAL SYNTAX.....	22
5	REFERENCES	29

1 Introduction

This document describes an internet Email Forwarding name mapping for version 1.0 of the Extensible Provisioning Protocol (EPP). This mapping is specified using the Extensible Markup Language (XML) 1.0 as described in [W3C.REC-xml20040204] and XML Schema notation as described in [W3C.REC-xmlschema-1-20041028] and [W3C.REC-xmlschema-2-20041028]. The basis for this mapping is the EPP domain name mapping [RFC3731].

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

XML is case sensitive. Unless stated otherwise, XML specifications and examples provided in this document MUST be interpreted in the character case presented to develop a conforming implementation.

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

2.1 Email Forwarding Name

The syntax for email forwarding names described in this document **MUST** conform to [RFC2822]. These conformance requirements **MAY** change as a result of progressing work in developing standards for internationalized email forwarding names. A server **MAY** restrict allowable email forwarding name to a particular top level domain, second level domain, or other domain for which the server is authoritative.

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 [RFC5730].

2.3 Status Values

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

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

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

Status Value Descriptions:

- clientDeleteProhibited, serverDeleteProhibited
 - Requests to delete the object **MUST** be rejected.
- clientHold, serverHold
 - Information **MUST NOT** be published for the object.
- 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 normal status value for an object that has no pending operations or prohibitions. This value is set and removed by the server as other status values are added or removed.
- pendingCreate, pendingDelete, pendingRenew, pendingTransfer, pendingUpdate
 - A transform command has been processed for the object, but the action has not been completed by the server. Server operators can delay action completion for a variety of reasons, such as to allow for human review or third-party action. A transform command that is processed, but whose requested action is pending, is noted with response code 1001.

With one exception, transform commands MUST be rejected when a pendingCreate, pendingDelete, pendingRenew, pendingTransfer, or pendingUpdate status is set. The only exception is that a <transfer> command to approve, reject, or cancel a transfer MAY be processed while an object is in "pendingTransfer" status.

When the requested action has been completed, the pendingCreate, pendingDelete, pendingRenew, pendingTransfer, or pendingUpdate status value MUST be removed. All clients involved in the transaction MUST be notified using a service message that the action has been completed and that the status of the object has changed.

"ok" status MUST NOT be combined with any other status.

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

"pendingRenew" status MUST NOT be combined with either "clientRenewProhibited" or "serverRenewProhibited" status.

"pendingTransfer" status MUST NOT be combined with either "clientTransferProhibited" or "serverTransferProhibited" status.

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

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

Other status combinations not expressly prohibited MAY be used.

2.4 Dates and Times

All date and time attribute values **MUST** be expressed in Universal Coordinated Time (UTC) using the Gregorian calendar. The date-time format defined in [RFC3339], with time-offset="Z" **MUST** be used.

2.5 Validity Periods

An email forwarding name object **MAY** have a specified validity period. If server policy supports email forwarding object validity periods, the validity period is defined when an email forwarding 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 an email forwarding 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 email forwarding objects to facilitate transfer operations. Authorization information is assigned when an email forwarding 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 [RFC5730]. The command mappings described here are specifically for use in provisioning and managing email forwarding via EPP.

3.1 EPP Query Commands

EPP provides three commands to retrieve email forwarding information: <check> to determine if an email forwarding object is available for provisioning, <info> to retrieve detailed information associated with an email forwarding object, and <transfer> to retrieve email forwarding 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 <create> 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 <emailFwd:check> element that identifies the email forwarding namespace. The <emailFwd:check> element contains the following child elements:

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

Example <check> command:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <check>
C:      <emailFwd:check
C:        xmlns:emailFwd="http://www.nic.name/epp/emailFwd-1.0">
C:          <emailFwd:name>john@doe.name</emailFwd:name>
C:          <emailFwd:name>johnny@doe.name</emailFwd:name>
C:          <emailFwd:name>jane@doe.name</emailFwd:name>
C:        </emailFwd: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 <emailFwd:chkData> element that identifies the email forwarding namespace. The <emailFwd:chkData> element contains one or more <emailFwd:cd> elements that contain the following child elements:

- A <emailFwd:name> element that contains the fully qualified name of the queried email forwarding 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 <emailFwd: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. This text MUST be represented in the response language previously negotiated with the client; an OPTIONAL "lang" attribute MAY be present to identify the language if the negotiated value is something other than a default value of "en" (English).

Example <check> response:

```
S:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S:  <response>
S:    <result code="1000">
S:      <msg>Command completed successfully</msg>
S:    </result>
S:    <resData>
S:      <emailFwd:chkData
S:        xmlns:emailFwd="http://www.nic.name/epp/emailFwd-1.0">
S:        <emailFwd:cd>
S:          <emailFwd:name avail="1">john@doe.name</emailFwd:name>
S:        </emailFwd:cd>
S:        <emailFwd:cd>
S:          <emailFwd:name avail="0">johnny@doe.name</emailFwd:name>
S:          <emailFwd:reason>In use</emailFwd:reason>
S:        </emailFwd:cd>
S:        <emailFwd:cd>
S:          <emailFwd:name avail="1">jane@doe.name</emailFwd:name>
S:        </emailFwd:cd>
S:      </emailFwd: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 can not be processed for any reason.

3.1.2 EPP <info> Command

The EPP <info> command is used to retrieve information associated with an email forwarding 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 <emailFwd:info> element that identifies the email forwarding namespace. The <emailFwd:info> element contains the following child elements:

- A <emailFwd:name> element that contains the fully qualified name of the email forwarding object to be queried.
- An OPTIONAL <emailFwd:authInfo> element that contains authorization information associated with the email forwarding object or authorization information associated with the email forwarding 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 email forwarding 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:  <command>
C:    <info>
C:      <emailFwd:info
C:        xmlns:emailFwd="http://www.nic.name/epp/emailFwd-1.0">
C:          <emailFwd:name>john@doe.name</emailFwd:name>
C:        </emailFwd: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 <emailFwd:infData> element that identifies the email forwarding namespace and the location of the email forwarding schema. The <emailFwd:infData> element contains the following child elements:

- A <emailFwd:name> element that contains the fully qualified name of the email forwarding object.
- A <emailFwd:roid> element that contains the Repository Object Identifier assigned to the email forwarding object when the object was created.
- Zero or more OPTIONAL <emailFwd:status> elements that contain the current status descriptors associated with the email forwarding.
- An OPTIONAL <emailFwd:registrant> element and one or more OPTIONAL <emailFwd:contact> elements that contain identifiers for the human or organizational social information objects associated with the email forwarding object.
- An OPTIONAL <emailFwd:fwdTo> element that contains the fully qualified forward to email address associated with the email forwarding object.
- A <emailFwd:clID> element that contains the identifier of the sponsoring client.
- An OPTIONAL <emailFwd:crID> element that contains the identifier of the client that created the email forwarding object.
- An OPTIONAL <emailFwd:crDate> element that contains the date and time of email forwarding object creation.
- An OPTIONAL <emailFwd:exDate> element that contains the date and time identifying the end of the email forwarding object's registration period.
- An OPTIONAL <emailFwd:upID> element that contains the identifier of the client that last updated the email forwarding object. This element MUST NOT be present if the email forwarding has never been modified.
- An OPTIONAL <emailFwd:upDate> element that contains the date and time of the most recent email forwarding object modification. This element MUST NOT be present if the email forwarding object has never been modified.
- An OPTIONAL <emailFwd:trDate> elements that contains the date and time of the most recent successful email forwarding object transfer. This element MUST NOT be provided if the email forwarding object has never been transferred.
- An OPTIONAL <emailFwd:authInfo> element that contains authorization information associated with the email forwarding object. This element MUST only be returned if the querying client is 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:  <response>
```

```

S:    <result code="1000">
S:      <msg>Command completed successfully</msg>
S:    </result>
S:    <resData>
S:      <emailFwd:infData
S:        xmlns:emailFwd="http://www.nic.name/epp/emailFwd-1.0">
S:          <emailFwd:name>john@doe.name</emailFwd:name>
S:          <emailFwd:roid>EXAMPLE1-VRSN</emailFwd:roid>
S:          <emailFwd:status s="ok"/>
S:          <emailFwd:registrant>jd1234</emailFwd:registrant>
S:          <emailFwd:contact type="admin">sh8013</emailFwd:contact>
S:          <emailFwd:contact type="tech">sh8013</emailFwd:contact>
S:          <emailFwd:fwdTo>jdoe@example.com</emailFwd:fwdTo>
S:          <emailFwd:clID>ClientX</emailFwd:clID>
S:          <emailFwd:crID>ClientY</emailFwd:crID>
S:          <emailFwd:crDate>1999-04-03T22:00:00.0Z</emailFwd:crDate>
S:          <emailFwd:upID>ClientX</emailFwd:upID>
S:          <emailFwd:upDate>1999-12-03T09:00:00.0Z</emailFwd:upDate>
S:          <emailFwd:exDate>2005-04-03T22:00:00.0Z</emailFwd:exDate>
S:          <emailFwd:trDate>2000-04-08T09:00:00.0Z</emailFwd:trDate>
S:          <emailFwd:authInfo>
S:            <emailFwd:pw>2fooBAR</emailFwd:pw>
S:          </emailFwd:authInfo>
S:        </emailFwd:infData>
S:      </resData>
S:    <trID>
S:      <clTRID>ABC-12345</clTRID>
S:      <svTRID>54322-XYZ</svTRID>
S:    </trID>
S:  </response>
S:</epp>

```

A server with a different information return policy MAY provide less information in a response.

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:  <response>
S:    <result code="1000">
S:      <msg>Command completed successfully</msg>
S:    </result>
S:    <resData>
S:      <emailFwd:infData
S:        xmlns:emailFwd="http://www.nic.name/epp/emailFwd-1.0">
S:          <emailFwd:name>john@doe.name</emailFwd:name>
S:          <emailFwd:roid>EXAMPLE1-VRSN</emailFwd:roid>
S:          <emailFwd:clID>ClientX</emailFwd:clID>
S:        </emailFwd:infData>
S:      </resData>

```

```
S:    <trID>
S:      <clTRID>ABC-12345</clTRID>
S:      <svTRID>54322-XYZ</svTRID>
S:    </trID>
S:  </response>
S:</epp>
```

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

3.1.3 EPP <transfer> Command

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

- A <emailFwd:name> element that contains the fully qualified name of the email forwarding object to be queried.
- An OPTIONAL <emailFwd:authInfo> element that contains authorization information associated with the email forwarding object or authorization information associated with the email forwarding 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 email forwarding 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:  <command>
C:    <transfer op="query">
C:      <emailFwd:transfer
C:        xmlns:emailFwd="http://www.nic.name/epp/emailFwd-1.0">
C:          <emailFwd:name>john@doe.name</emailFwd:name>
C:          <emailFwd:authInfo>
C:            <emailFwd:pw roid="JD1234-REP">2fooBAR</emailFwd:pw>
C:          </emailFwd:authInfo>
C:        </emailFwd: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 <emailFwd:trnData> element that identifies the email forwarding namespace and the location of the email forwarding schema. The <emailFwd:trnData> element contains the following child elements:

- A <emailFwd:name> element that contains the fully qualified name of the email forwarding object.
- A <emailFwd:trStatus> element that contains the state of the most recent transfer request.

- A <emailFwd:reID> element that contains the identifier of the client that requested the object transfer.
- A <emailFwd:reDate> element that contains the date and time that the transfer was requested.
- A <emailFwd:acID> element that contains the identifier of the client that SHOULD act upon the transfer request.
- A <emailFwd: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 will be taken by the server. For all other status types, the value identifies the date and time when the request was completed.
- An OPTIONAL <emailFwd:exDate> element that contains the end of the email forwarding 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:  <response>
S:    <result code="1000">
S:      <msg>Command completed successfully</msg>
S:    </result>
S:    <resData>
S:      <emailFwd:trnData
S:        xmlns:emailFwd="http://www.nic.name/epp/emailFwd-1.0">
S:          <emailFwd:name>john@doe.name</emailFwd:name>
S:          <emailFwd:trStatus>pending</emailFwd:trStatus>
S:          <emailFwd:reID>ClientX</emailFwd:reID>
S:          <emailFwd:reDate>2000-06-06T22:00:00.0Z</emailFwd:reDate>
S:          <emailFwd:acID>ClientY</emailFwd:acID>
S:          <emailFwd:acDate>2000-06-11T22:00:00.0Z</emailFwd:acDate>
S:          <emailFwd:exDate>2002-09-08T22:00:00.0Z</emailFwd:exDate>
S:        </emailFwd: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> query command cannot be processed for any reason.

3.2 EPP Transform Commands

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

3.2.1 EPP <create> Command

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

- A <emailFwd:name> element that contains the fully qualified name of the email forwarding object to be created.
- An OPTIONAL <emailFwd:period> element that contains the initial registration period of the email forwarding object. A server MAY define a default initial registration period if not specified by the client.
- A <emailFwd:fwdTo> element that contain the fully qualified email address associated with the email forwarding object.
- An OPTIONAL <emailFwd:registrant> element that contains the identifier for the human or organizational social information (contact) object to be associated with the email forwarding object as the object registrant. This object identifier MUST be known to the server before the contact object can be associated with the email forwarding object. The EPP mapping for contact objects is described in [RFC5733].
- Zero or more OPTIONAL <emailFwd:contact> elements that contain the identifiers for other contact objects to be associated with the email forwarding object. Contact object identifiers MUST be known to the server before the contact object can be associated with the email forwarding object.
- A <emailFwd:authInfo> element that contains authorization information to be associated with the email forwarding object.

Example <create> command:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <create>
C:      <emailFwd:create
C:        xmlns:emailFwd="http://www.nic.name/epp/emailFwd-1.0">
C:          <emailFwd:name>john@doe.name</emailFwd:name>
C:          <emailFwd:fwdTo>jdoe@example.com</emailFwd:fwdTo>
C:          <emailFwd:period unit="y">2</emailFwd:period>
```

VeriSign Inc. Proprietary Information


```

C:      <emailFwd:registrant>jd1234</emailFwd:registrant>
C:      <emailFwd:contact type="admin">sh8013</emailFwd:contact>
C:      <emailFwd:contact type="tech">sh8013</emailFwd:contact>
C:      <emailFwd:authInfo>
C:          <emailFwd:pw>2fooBAR</emailFwd:pw>
C:      </emailFwd:authInfo>
C:  </emailFwd:create>
C: </create>
C:  <clTRID>ABC-12345</clTRID>
C: </command>
C:</epp>

```

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

- A <emailFwd:name> element that contains the fully qualified name of the email forwarding object.
- A <emailFwd:crDate> element that contains the date and time of email forwarding object creation.
- An OPTIONAL <emailFwd:exDate> element that contains the date and time identifying the end of the email forwarding 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:  <response>
S:    <result code="1000">
S:      <msg>Command completed successfully</msg>
S:    </result>
S:    <resData>
S:      <emailFwd:creData
S:        xmlns:emailFwd="http://www.nic.name/epp/emailFwd-1.0">
S:          <emailFwd:name>john@doe.name</emailFwd:name>
S:          <emailFwd:crDate>1999-04-03T22:00:00.0Z</emailFwd:crDate>
S:          <emailFwd:exDate>2001-04-03T22:00:00.0Z</emailFwd:exDate>
S:        </emailFwd: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 can not be processed for any reason.

3.2.2 EPP <delete> Command

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

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

Example <delete> command:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <delete>
C:      <emailFwd:delete
C:        xmlns:emailFwd="http://www.nic.name/epp/emailFwd-1.0">
C:          <emailFwd:name>john@doe.name</emailFwd:name>
C:        </emailFwd:delete>
C:      </delete>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

When a <delete> email forwarding 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 an email forwarding object. In addition to the standard EPP command elements, the <renew> command MUST contain a <emailFwd:renew> element that identifies the email forwarding namespace. The <emailFwd:renew> element contains the following child elements:

- A <emailFwd:name> element that contains the fully qualified name of the email forwarding object whose validity period is to be extended.
- A <emailFwd: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 <emailFwd:period> element that contains the number of units to be added to the registration period of the email forwarding 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:  <command>
C:    <renew>
C:      <emailFwd:renew
C:        xmlns:emailFwd="http://www.nic.name/epp/emailFwd-1.0">
C:          <emailFwd:name>john@doe.name</emailFwd:name>
C:          <emailFwd:curExpDate>2000-04-03</emailFwd:curExpDate>
C:          <emailFwd:period unit="y">5</emailFwd:period>
C:        </emailFwd: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 <emailFwd:renData> element that identifies the email forwarding namespace. The <emailFwd:renData> element contains the following child elements:

- A <emailFwd:name> element that contains the fully qualified name of the email forwarding object.
- A <emailFwd:exDate> element that contains the date and time identifying the end of the email forwarding 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:  <response>
```

```
S:    <result code="1000">
S:      <msg>Command completed successfully</msg>
S:    </result>
S:    <resData>
S:      <emailFwd:renData
S:        xmlns:emailFwd="http://www.nic.name/epp/emailFwd-1.0">
S:          <emailFwd:name>john@doe.name</emailFwd:name>
S:          <emailFwd:exDate>2005-04-03T22:00:00.0Z</emailFwd:exDate>
S:        </emailFwd: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 an email forwarding object. In addition to the standard EPP command elements, the <transfer> command MUST contain a <emailFwd:transfer> element that identifies the email forwarding namespace. The <emailFwd:transfer> element contains the following child elements:

- A <emailFwd:name> element that contains the fully qualified name of the email forwarding object for which a transfer request is to be created, approved, rejected, or cancelled.
- An OPTIONAL <emailFwd:period> element that contains the number of units to be added to the registration period of the email forwarding 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. The number of units available MAY be subject to limits imposed by the server.
- A <emailFwd:authInfo> element that contains authorization information associated with the email forwarding object or authorization information associated with the email forwarding 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 email forwarding 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 [RFC5730].

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:  <command>
C:    <transfer op="request">
C:      <emailFwd:transfer
C:        xmlns:emailFwd="http://www.nic.name/epp/emailFwd-1.0">
C:          <emailFwd:name>john@doe.name</emailFwd:name>
C:          <emailFwd:period unit="y">1</emailFwd:period>
C:          <emailFwd:authInfo>
C:            <emailFwd:pw roid="JD1234-REP">2fooBAR</emailFwd:pw>
C:          </emailFwd:authInfo>
C:        </emailFwd: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 <emailFwd:trnData> element that identifies the email forwarding namespace. The <emailFwd: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:  <response>
S:    <result code="1000">
S:      <msg>Command completed successfully</msg>
S:    </result>
S:    <resData>
S:      <emailFwd:trnData
S:        xmlns:emailFwd="http://www.nic.name/epp/emailFwd-1.0">
S:        <emailFwd:name>john@doe.name</emailFwd:name>
S:        <emailFwd:trStatus>pending</emailFwd:trStatus>
S:        <emailFwd:reID>ClientX</emailFwd:reID>
S:        <emailFwd:reDate>2000-06-08T22:00:00.0Z</emailFwd:reDate>
S:        <emailFwd:acID>ClientY</emailFwd:acID>
S:        <emailFwd:acDate>2000-06-13T22:00:00.0Z</emailFwd:acDate>
S:        <emailFwd:exDate>2002-09-08T22:00:00.0Z</emailFwd:exDate>
S:      </emailFwd: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 can not be processed for any reason.

3.2.5 EPP <update> Command

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

- A <emailFwd:name> element that contains the fully qualified name of the email forwarding object to be updated.
- An OPTIONAL <emailFwd:add> element that contains attribute values to be added to the object.
- An OPTIONAL <emailFwd:rem> element that contains attribute values to be removed from the object.
- An OPTIONAL <emailFwd:chg> element that contains object attribute values to be changed.

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

- Zero or more <emailFwd:contact> elements that contain the identifiers for contact objects to be associated with or removed from the email forwarding object. Contact object identifiers MUST be known to the server before the contact object can be associated with the email forwarding object.
- Zero or more <emailFwd: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 <emailFwd:chg> element contains the following child elements:

- An OPTIONAL <emailFwd:registrant> element that contains the identifier for the human or organizational social information (contact) object to be associated with the email forwarding object as the object registrant. This object identifier MUST be known to the server before the contact object can be associated with the email forwarding object.
- An OPTIONAL <emailFwd:authInfo> element that contains authorization information associated with the email forwarding object. This mapping includes a password-based authentication mechanism, but the schema allows new mechanisms to be defined in new schemas. A <emailFwd:null> element can be used within the <emailFwd:authInfo> element to remove authorization information.
- An OPTIONAL <emailFwd:fwdTo> element that contains fully qualified forward to email address associated with the email forwarding object.

Example <update> command:

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

VeriSign Inc. Proprietary Information

```

C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <update>
C:      <emailFwd:update
C:        xmlns:emailFwd="http://www.nic.name/epp/emailFwd-1.0">
C:          <emailFwd:name>john@doe.name</emailFwd:name>
C:          <emailFwd:add>
C:            <emailFwd:contact type="tech">mak21</emailFwd:contact>
C:            <emailFwd:status s="clientHold" lang="en">
C:              Payment overdue.
C:            </emailFwd:status>
C:          </emailFwd:add>
C:          <emailFwd:rem>
C:            <emailFwd:contact type="tech">sh8013</emailFwd:contact>
C:            <emailFwd:status s="clientUpdateProhibited"/>
C:          </emailFwd:rem>
C:          <emailFwd:chg>
C:            <emailFwd:fwdTo > johnny@example.com</emailFwd:fwdTo>
C:            <emailFwd:registrant>sh8013</emailFwd:registrant>
C:            <emailFwd:authInfo>
C:              <emailFwd:pw>2BARfoo</emailFwd:pw>
C:            </emailFwd:authInfo>
C:          </emailFwd:chg>
C:        </emailFwd:update>
C:      </update>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>

```

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

Example `<update>` response:

```

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

```

An EPP error response **MUST** be returned if an `<update>` command 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 BEGIN and END tags are not part of the schema; they are used to note the beginning and ending of the schema for URI registration purposes.

BEGIN

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

<schema targetNamespace="http://www.nic.name/epp/emailFwd-1.0"
  xmlns:emailFwd="http://www.nic.name/epp/emailFwd-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"/>
  <import namespace="urn:ietf:params:xml:ns:epp-1.0"/>

  <annotation>
    <documentation>
      Extensible Provisioning Protocol v1.0
      email forwarding provisioning schema.
    </documentation>
  </annotation>

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

  <!--
  Child elements of the <create> command.
  -->
  <complexType name="createType">
    <sequence>
      <element name="name" type="emailFwd:emailAddrType"/>
      <element name="fwdTo" type="emailFwd:emailAddrType"/>
      <element name="period" type="emailFwd:periodType"
        minOccurs="0"/>
      <element name="registrant" type="eppcom:clIDType"
        minOccurs="0"/>
      <element name="contact" type="emailFwd:contactType"
        minOccurs="0" maxOccurs="unbounded"/>
      <element name="authInfo" type="emailFwd:authInfoType"/>
    </sequence>
  </complexType>
```

```

<!--
This isn't a complete pattern; additional out-of-parser
processing is required to validate address format.
-->
<simpleType name="emailAddrType">
  <restriction base="token">
    <pattern value=".+@.+"/>
  </restriction>
</simpleType>

<complexType name="periodType">
  <simpleContent>
    <extension base="emailFwd:pLimitType">
      <attribute name="unit" type="emailFwd: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>

<complexType name="contactType">
  <simpleContent>
    <extension base="eppcom:clIDType">
      <attribute name="type" type="emailFwd:contactAttrType"/>
    </extension>
  </simpleContent>
</complexType>

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

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

```

<!--

Child element of commands that require a single name.

VeriSign Inc. Proprietary Information

```

-->
<complexType name="sNameType">
  <sequence>
    <element name="name" type="emailFwd:emailAddrType"/>
  </sequence>
</complexType>

<!--
Child element of commands that accept multiple names.
-->
<complexType name="mNameType">
  <sequence>
    <element name="name" type="emailFwd:emailAddrType"
      maxOccurs="unbounded"/>
  </sequence>
</complexType>

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

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

<!--
Child elements of the <transfer> command.
-->
<complexType name="transferType">
  <sequence>
    <element name="name" type="emailFwd:emailAddrType"/>
    <element name="period" type="emailFwd:periodType"
      minOccurs="0"/>
    <element name="authInfo" type="emailFwd:authInfoType"
      minOccurs="0"/>
  </sequence>
</complexType>

<!--
Child elements of the <update> command.
-->
<complexType name="updateType">
  <sequence>
    <element name="name" type="emailFwd:emailAddrType"/>

```

```

        <element name="add" type="emailFwd:addRemType"
          minOccurs="0"/>
        <element name="rem" type="emailFwd:addRemType"
          minOccurs="0"/>
        <element name="chg" type="emailFwd:chgType"
          minOccurs="0"/>
      </sequence>
    </complexType>

<!--
Data elements that can be added or removed.
-->
<complexType name="addRemType">
  <sequence>
    <element name="contact" type="emailFwd:contactType"
      minOccurs="0" maxOccurs="unbounded"/>
    <element name="status" type="emailFwd:statusType"
      minOccurs="0" maxOccurs="11"/>
  </sequence>
</complexType>

<!--
Data elements that can be changed.
-->
<complexType name="chgType">
  <sequence>
    <element name="fwdTo" type="emailFwd:emailAddrType"
      minOccurs="0"/>
    <element name="registrant" type="emailFwd:clIDChgType"
      minOccurs="0"/>
    <element name="authInfo" type="emailFwd:authInfoChgType"
      minOccurs="0"/>
  </sequence>
</complexType>

<!--
Allow the registrant value to be nullified by changing the
minLength restriction to "0".
-->
<simpleType name="clIDChgType">
  <restriction base="token">
    <minLength value="0"/>
    <maxLength value="16"/>
  </restriction>
</simpleType>

<!--
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="emailFwd:chkDataType"/>
<element name="creData" type="emailFwd:creDataType"/>
<element name="infData" type="emailFwd:infDataType"/>
<element name="panData" type="emailFwd:panDataType"/>
<element name="renData" type="emailFwd:renDataType"/>
<element name="trnData" type="emailFwd:trnDataType"/>

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

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

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

<!--
<create> response elements.
-->
<complexType name="creDataType">
  <sequence>
    <element name="name" type="emailFwd:emailAddrType"/>
    <element name="crDate" type="dateTime"/>
    <element name="exDate" type="dateTime"
      minOccurs="0"/>
  </sequence>
</complexType>

<!--
<info> response elements.
-->
<complexType name="infDataType">
  <sequence>
    <element name="name" type="emailFwd:emailAddrType"/>
    <element name="roid" type="eppcom:roidType"/>
    <element name="status" type="emailFwd:statusType"
      minOccurs="0" maxOccurs="11"/>
    <element name="registrant" type="emailFwd:contactType"
      minOccurs="0"/>
    <element name="contact" type="emailFwd:contactType"

```

```

    minOccurs="0" maxOccurs="unbounded"/>
<element name="fwdTo" type="emailFwd:emailAddrType"
  minOccurs="0"/>
<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="emailFwd: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="emailFwd:statusValueType"
        use="required"/>
      <attribute name="lang" type="language"
        default="en"/>
    </extension>
  </simpleContent>
</complexType>

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

<!--
Pending action notification response elements.

```

```

-->
<complexType name="panDataType">
  <sequence>
    <element name="name" type="emailFwd:paNameType"/>
    <element name="paTRID" type="epp:trIDType"/>
    <element name="paDate" type="dateTime"/>
  </sequence>
</complexType>

<complexType name="paNameType">
  <simpleContent>
    <extension base="eppcom:labelType">
      <attribute name="paResult" type="boolean"
        use="required"/>
    </extension>
  </simpleContent>
</complexType>

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

<!--
<transfer> response elements.
-->
<complexType name="trnDataType">
  <sequence>
    <element name="name" type="emailFwd:emailAddrType"/>
    <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

Document all 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.

[RFC3733] Hollenbeck, S., " Extensible Provisioning Protocol Contact Mapping", RFC 3733, March 2004.

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

[RFC952] K. Harrenstien et al.: "DOD Internet Host Table Specification", RFC 952, October 1985.

[RFC1123] R. Braden: "Requirements for Internet Hosts -- Application and Support", RFC 1123, October 1989.

[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.

[RFC2822] P. Resnick: "Internet Message Format", RFC 2822, April 2001.