

Extensible Provisioning Protocol Contact Mapping
<[draft-ietf-provreg-epp-contact-04.txt](#)>

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of RFC2026](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress".

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

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

Abstract

This document describes an Extensible Provisioning Protocol (EPP) mapping for the provisioning and management of individual or organizational social information identifiers (known as "contacts") stored in a shared central repository. Specified in XML, the mapping defines EPP command syntax and semantics as applied to contacts.

Conventions Used In This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [[RFC2119](#)].

In examples, "C:" represents lines sent by a protocol client and "S:" represents lines returned by a protocol server. Indentation and white space in examples is provided only to illustrate element relationships and is not a REQUIRED feature of this protocol.

Table of Contents

1. Introduction	3
2. Object Attributes	4
2.1 Contact and Client Identifiers	4
2.2 Status Values	4
2.3 Individual and Organizational Names	5
2.4 Address	5
2.4.1 Street, City, and State or Province	6
2.4.2 Postal Code	6
2.4.3 Country	6
2.5 Telephone Numbers	6
2.6 Email Addresses	6
2.7 Dates and Times	6
2.8 Authorization Information	6
3. EPP Command mapping	8
3.1 EPP Query Commands	8
3.1.1 EPP <check> Command	8
3.1.2 EPP <info> Command	10
3.1.3 EPP <transfer> Command	14
3.2 EPP Transform Commands	16
3.2.1 EPP <create> Command	16
3.2.2 EPP <delete> Command	19
3.2.3 EPP <renew> Command	21
3.2.4 EPP <transfer> Command	21
3.2.5 EPP <update> Command	22
4. Formal Syntax	27
5. Internationalization Considerations	34
6. IANA Considerations	34
7. Security Considerations	35
8. Acknowledgements	35
9. References	36
10. Author's Address	37
A. Revisions From Previous Version	38
B. Full Copyright Statement	39

1. Introduction

This document describes a personal and organizational identifier mapping for version 1.0 of the Extensible Provisioning Protocol (EPP). This mapping is specified using the Extensible Markup Language (XML) 1.0 as described in [[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.

This document is being discussed on the "ietf-provreg" mailing list. To join the list, send a message to <majordomo@cafax.se> with the words "subscribe ietf-provreg" in the body of the message. There is a web site for the list archives at <http://www.cafax.se/ietf-provreg>.

2. Object Attributes

An EPP contact object has attributes and associated values that can be viewed and modified by the sponsoring client or the server. This section describes each attribute type in detail. The formal syntax for the attribute values described here can be found in the "Formal Syntax" section of this document and in the appropriate normative references.

2.1 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 [\[EPP\]](#).

2.2 Status Values

A contact object **MUST** always have at least one associated status value. Status values can be set only by the client that sponsors a contact object and by the server on which the object resides. A client can change the status of a contact 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 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.

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.

linked

The contact object has at least one active association with another object, such as a domain object. Servers SHOULD provide services to determine existing object associations.

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

2.3 Individual and Organizational Names

Individual and organizational names associated with a contact are represented using character strings. These strings have a specified minimum length and a specified maximum length. Individual and organizational names MAY be provided in both UTF-8 [[RFC2279](#)] and a subset of UTF-8 that can be represented in 7-bit ASCII depending on local needs.

2.4 Address

Every contact has associated postal address information. A postal address contains OPTIONAL street information, city information, OPTIONAL state/province information, an OPTIONAL postal code, and a country identifier. Address information MAY be provided in both UTF-8

and a subset of UTF-8 that can be represented in 7-bit ASCII depending on local needs.

[2.4.1](#) Street, City, and State or Province

Contact street, city, and state or province information is represented using character strings. These strings have a specified minimum length and a specified maximum length.

[2.4.2](#) Postal Code

Contact postal codes are represented using character strings. These strings have a specified minimum length and a specified maximum length.

[2.4.3](#) Country

Contact country identifiers are represented using two-character identifiers specified in [[ISO3166](#)].

[2.5](#) Telephone Numbers

Contact telephone number structure is derived from structures defined in [[E164a](#)]. Telephone numbers described in this mapping are character strings that MUST begin with a plus sign ("+", ASCII value 0x002B), followed by a country code defined in [[E164b](#)], followed by a dot (".", ASCII value 0x002E), followed by a sequence of digits representing the telephone number. An optional "x" attribute is provided to note telephone extension information.

[2.6](#) Email Addresses

Email address syntax is defined in [[RFC2822](#)]. This mapping does not prescribe minimum or maximum lengths for character strings used to represent email addresses.

[2.7](#) 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 [[DATE-TIME](#)] MUST be used to represent date-time values as XML Schema does not support truncated date-time forms.

[2.8](#) Authorization Information

Authorization information is associated with contact objects to facilitate transfer operations. Authorization information is assigned when a contact object is created, and it might 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 contact objects via EPP.

3.1 EPP Query Commands

EPP provides three commands to retrieve contact information: <check> to determine if a contact object can be provisioned within a repository, <info> to retrieve detailed information associated with a contact object, and <transfer> to retrieve contact object transfer status information.

3.1.1 EPP <check> Command

The EPP <check> command is used to determine if an object can be provisioned within a repository. It provides a hint that allows a client to anticipate the success or failure of provisioning an object using the <create> command. 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 <contact:check> element that identifies the contact namespace and the location of the contact schema. The <contact:check> element contains the following child elements:

- One or more <contact:id> elements that contain the server-unique identifier of the contact 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
C:    epp-1.0.xsd">
C:  <command>
C:    <check>
C:      <contact:check
C:        xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
C:        xsi:schemaLocation="urn:ietf:params:xml:ns:contact-1.0
C:          contact-1.0.xsd">
C:          <contact:id>sh8013</contact:id>
C:          <contact:id>sah8013</contact:id>
C:          <contact:id>8013sah</contact:id>
C:        </contact: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 <contact:chkData> element that identifies the contact namespace and the location of the contact schema. The <contact:chkData> element contains one or more <contact:cd> elements that contain the following child elements:

- A <contact:id> element that identifies the queried 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 <contact: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 the 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:  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:      <contact:chkData
S:        xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
S:        xsi:schemaLocation="urn:ietf:params:xml:ns:contact-1.0
S:          contact-1.0.xsd">
S:        <contact:cd>
S:          <contact:id avail="1">sh8013</contact:id>
S:        </contact:cd>
S:        <contact:cd>
S:          <contact:id avail="0">sah8013</contact:id>
S:          <contact:reason>In use</contact:reason>
S:        </contact:cd>
S:        <contact:cd>
S:          <contact:id avail="1">8013sah</contact:id>
S:        </contact:cd>
S:      </contact: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 a contact object. In addition to the standard EPP command elements, the <info> command MUST contain a <contact:info> element that identifies the contact namespace and the location of the contact schema. The <contact:info> element contains the following child elements:

- A <contact:id> element that contains the server-unique identifier of the contact object to be queried.

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
C:    epp-1.0.xsd">
C:  <command>
C:    <info>
C:      <contact:info
C:        xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
C:        xsi:schemaLocation="urn:ietf:params:xml:ns:contact-1.0
C:          contact-1.0.xsd">
C:          <contact:id>sh8013</contact:id>
C:        </contact: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 <contact:infData> element that identifies the contact namespace and the location of the contact schema. The <contact:infData> element contains the following child elements:

- A <contact:id> element that contains the server-unique identifier of the contact object.
- A <contact:roid> element that contains the Repository Object Identifier assigned to the contact object when the object was created.
- One or more <contact:status> elements that describe the status of the contact object.
- One or two <contact:postalInfo> elements that contain postal address information. Two elements are provided so that address information can be provided in both internationalized and localized forms. If an internationalized form is provided, it MUST be listed first and element content MUST be represented in a subset of UTF-8 that can be represented in the 7-bit US-ASCII character set. If a localized form is provided, element content MAY be represented in unrestricted UTF-8. The <contact:postalInfo> element contains the following child elements:

- A <contact:name> element that contains the name of the individual or role represented by the contact.
- An OPTIONAL <contact:org> element that contains the name of the organization with which the contact is affiliated.
- A <contact:addr> element that contains address information associated with the contact. A <contact:addr> element contains the following child elements:
 - One, two, or three OPTIONAL <contact:street> elements that contain the contact's street address.
 - A <contact:city> element that contains the contact's city.
 - An OPTIONAL <contact:sp> element that contains the contact's state or province.
 - An OPTIONAL <contact:pc> element that contains the contact's postal code.
 - A <contact:cc> element that contains the contact's country code.
- An OPTIONAL <contact:voice> element that contains the contact's voice telephone number.
- An OPTIONAL <contact:fax> element that contains the contact's facsimile telephone number.
- A <contact:email> element that contains the contact's email address.
- A <contact:clID> element that contains the identifier of the sponsoring client.
- A <contact:crID> element that contains the identifier of the client that created the contact object.
- A <contact:crDate> element that contains the date and time of contact object creation.
- A <contact:upID> element that contains the identifier of the client that last updated the contact object. This element MUST NOT be present if the contact has never been modified.
- A <contact:upDate> element that contains the date and time of the most recent contact object modification. This element MUST NOT be present if the contact object has never been modified.

- A <contact:trDate> elements that contains the date and time of the most recent successful contact object transfer. This element MUST NOT be provided if the contact object has never been transferred.
- A <contact:authInfo> element that contains authorization information associated with the contact object. This element MUST NOT be provided if the querying client is not the current sponsoring client.

Example <info> 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:      <contact:infData
S:        xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
S:        xsi:schemaLocation="urn:ietf:params:xml:ns:contact-1.0
S:          contact-1.0.xsd">
S:        <contact:id>sh8013</contact:id>
S:        <contact:roid>SH8013-REP</contact:roid>
S:        <contact:status s="linked"/>
S:        <contact:status s="clientDeleteProhibited"/>
S:        <contact:postalInfo>
S:          <contact:name>John Doe</contact:name>
S:          <contact:org>Example Inc.</contact:org>
S:          <contact:addr>
S:            <contact:street>123 Example Dr.</contact:street>
S:            <contact:street>Suite 100</contact:street>
S:            <contact:city>Dulles</contact:city>
S:            <contact:sp>VA</contact:sp>
S:            <contact:pc>20166-6503</contact:pc>
S:            <contact:cc>US</contact:cc>
S:          </contact:addr>
S:        </contact:postalInfo>
S:        <contact:voice x="1234">+1.7035555555</contact:voice>
S:        <contact:fax>+1.7035555556</contact:fax>
S:        <contact:email>jdoe@example.tld</contact:email>
S:        <contact:clID>ClientY</contact:clID>
S:        <contact:crID>ClientX</contact:crID>
S:        <contact:crDate>1999-04-03T22:00:00.0Z</contact:crDate>
S:        <contact:upID>ClientX</contact:upID>
S:        <contact:upDate>1999-12-03T09:00:00.0Z</contact:upDate>
```



```
S:      <contact:trDate>2000-04-08T09:00:00.0Z</contact:trDate>
S:      <contact:authInfo type="pw">2fooBAR</contact:authInfo>
S:      </contact: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 <contact:transfer> element that identifies the contact namespace and the location of the contact schema. The <contact:transfer> element MUST contain the following child elements:

- A <contact:id> element that contains the server-unique identifier of the contact object to be queried.
- A <contact:authInfo> element that contains authorization information associated with the contact object.

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
C:    epp-1.0.xsd">
C:  <command>
C:    <transfer op="query">
C:      <contact:transfer
C:        xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
C:        xsi:schemaLocation="urn:ietf:params:xml:ns:contact-1.0
C:          contact-1.0.xsd">
C:        <contact:id>sh8013</contact:id>
C:        <contact:authInfo type="pw">2fooBAR</contact:authInfo>
C:      </contact: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 <contact:trnData> element that identifies the contact namespace and the location of the contact schema. The <contact:trnData> element contains the following child elements:

- A <contact:id> element that contains the server-unique identifier for the queried contact.
- A <contact:trStatus> element that contains the state of the most recent transfer request.
- A <contact:reID> element that contains the identifier of the client that requested the object transfer.
- A <contact:reDate> element that contains the date and time that the transfer was requested.
- A <contact:acID> element that contains the identifier of the client that SHOULD act upon the transfer request.
- A <contact: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 SHOULD be taken by the server. For all other status types, the value identifies the date and time when the

request was completed.

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
S:    epp-1.0.xsd">
S:  <response>
S:    <result code="1000">
S:      <msg>Command completed successfully</msg>
S:    </result>
S:    <resData>
S:      <contact:trnData
S:        xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
S:        xsi:schemaLocation="urn:ietf:params:xml:ns:contact-1.0
S:          contact-1.0.xsd">
S:          <contact:id>sh8013</contact:id>
S:          <contact:trStatus>pending</contact:trStatus>
S:          <contact:reID>ClientX</contact:reID>
S:          <contact:reDate>2000-06-06T22:00:00.0Z</contact:reDate>
S:          <contact:acID>ClientY</contact:acID>
S:          <contact:acDate>2000-06-11T22:00:00.0Z</contact:acDate>
S:        </contact: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 can not be processed for any reason.

3.2 EPP Transform Commands

EPP provides four commands to transform contact object information: <create> to create an instance of a contact object, <delete> to delete an instance of a contact object, <transfer> to manage contact object sponsorship changes, and <update> to change information associated with a contact object. This document does not define a mapping for the EPP <renew> command.

3.2.1 EPP <create> Command

The EPP <create> command provides a transform operation that allows a

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

- A <contact:id> element that contains the desired server-unique identifier for the contact to be created.
- One or two <contact:postalInfo> elements that contain postal address information. Two elements are provided so that address information can be provided in both internationalized and localized forms. If an internationalized form is provided, it MUST be listed first and element content MUST be represented in a subset of UTF-8 that can be represented in the 7-bit US-ASCII character set. If a localized form is provided, element content MAY be represented in unrestricted UTF-8. The <contact:postalInfo> element contains the following child elements:
 - A <contact:name> element that contains the name of the individual or role represented by the contact.
 - An OPTIONAL <contact:org> element that contains the name of the organization with which the contact is affiliated.
 - A <contact:addr> element that contains address information associated with the contact. A <contact:addr> element contains the following child elements:
 - One, two, or three OPTIONAL <contact:street> elements that contain the contact's street address.
 - A <contact:city> element that contains the contact's city.
 - An OPTIONAL <contact:sp> element that contains the contact's state or province.
 - An OPTIONAL <contact:pc> element that contains the contact's postal code.
 - A <contact:cc> element that contains the contact's country code.
 - An OPTIONAL <contact:voice> element that contains the contact's voice telephone number.
 - An OPTIONAL <contact:fax> element that contains the contact's facsimile telephone number.

- A <contact:email> element that contains the contact's email address.
- A <contact:authInfo> element that contains authorization information to be associated with the contact 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
C:    epp-1.0.xsd">
C:  <command>
C:    <create>
C:      <contact:create
C:        xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
C:        xsi:schemaLocation="urn:ietf:params:xml:ns:contact-1.0
C:          contact-1.0.xsd">
C:          <contact:id>sh8013</contact:id>
C:          <contact:postalInfo>
C:            <contact:name>John Doe</contact:name>
C:            <contact:org>Example Inc.</contact:org>
C:            <contact:addr>
C:              <contact:street>123 Example Dr.</contact:street>
C:              <contact:street>Suite 100</contact:street>
C:              <contact:city>Dulles</contact:city>
C:              <contact:sp>VA</contact:sp>
C:              <contact:pc>20166-6503</contact:pc>
C:              <contact:cc>US</contact:cc>
C:            </contact:addr>
C:          </contact:postalInfo>
C:          <contact:voice x="1234">+1.7035555555</contact:voice>
C:          <contact:fax>+1.7035555556</contact:fax>
C:          <contact:email>jdoe@example.tld</contact:email>
C:          <contact:authInfo type="pw">2fooBAR</contact:authInfo>
C:        </contact: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 <contact:creData> element that identifies the contact namespace and the location of the contact schema. The <contact:creData> element contains the following child elements:

- A <contact:id> element that contains the server-unique identifier

for the created contact.

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

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
S:    epp-1.0.xsd">
S:  <response>
S:    <result code="1000">
S:      <msg>Command completed successfully</msg>
S:    </result>
S:    <resData>
S:      <contact:creData
S:        xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
S:        xsi:schemaLocation="urn:ietf:params:xml:ns:contact-1.0
S:          contact-1.0.xsd">
S:        <contact:id>sh8013</contact:id>
S:        <contact:crDate>1999-04-03T22:00:00.0Z</contact:crDate>
S:      </contact: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 a contact object. In addition to the standard EPP command elements, the <delete> command MUST contain a <contact:delete> element that identifies the contact namespace and the location of the contact schema. The <contact:delete> element MUST contain the following child elements:

- A <contact:id> element that contains the server-unique identifier of the contact object to be deleted.

A contact object SHOULD NOT be deleted if it is associated with other

known objects. An associated contact SHOULD NOT be deleted until associations with other known objects have been broken. A server SHOULD notify clients of object relationships when a <delete> command is attempted and fails due to existing object relationships.

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
C:    epp-1.0.xsd">
C:  <command>
C:    <delete>
C:      <contact:delete
C:        xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
C:        xsi:schemaLocation="urn:ietf:params:xml:ns:contact-1.0
C:          contact-1.0.xsd">
C:          <contact:id>sh8013</contact:id>
C:        </contact: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

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

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

- A <contact:id> element that contains the server-unique identifier of the contact object for which a transfer request is to be created, approved, rejected, or cancelled.
- A <contact:authInfo> element that contains authorization information associated with the contact object.

Every EPP <transfer> command MUST contain an "op" attribute that identifies the transfer operation to be performed as 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:      <contact:transfer
C:        xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
C:        xsi:schemaLocation="urn:ietf:params:xml:ns:contact-1.0
C:          contact-1.0.xsd">
C:        <contact:id>sh8013</contact:id>
C:        <contact:authInfo type="pw">2fooBAR</contact:authInfo>
C:      </contact: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 <contact:trnData> element that

identifies the contact namespace and the location of the contact schema. The <contact: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:      <contact:trnData
S:        xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
S:        xsi:schemaLocation="urn:ietf:params:xml:ns:contact-1.0
S:          contact-1.0.xsd">
S:        <contact:id>sh8013</contact:id>
S:        <contact:trStatus>pending</contact:trStatus>
S:        <contact:reID>ClientX</contact:reID>
S:        <contact:reDate>2000-06-08T22:00:00.0Z</contact:reDate>
S:        <contact:acID>ClientY</contact:acID>
S:        <contact:acDate>2000-06-13T22:00:00.0Z</contact:acDate>
S:      </contact: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 a contact object. In addition to the standard EPP command elements, the <update> command MUST contain a <contact:update> element that identifies the contact namespace and the location of the contact schema. The <contact:update> element contains the following child elements:

- A <contact:id> element that contains the server-unique identifier of

the contact object to be updated.

- An OPTIONAL <contact:add> element that contains attribute values to be added to the object.
- An OPTIONAL <contact:rem> element that contains attribute values to be removed from the object.
- An OPTIONAL <contact:chg> element that contains object attribute values to be changed.

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

- One or more <contact:status> elements that contain status values to be associated with 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 <contact:chg> element contains the following OPTIONAL child elements. At least one child element MUST be present:

- One or two <contact:postalInfo> elements that contain postal address information. Two elements are provided so that address information can be provided in both internationalized and localized forms. If an internationalized form is provided, it MUST be listed first and element content MUST be represented in a subset of UTF-8 that can be represented in the 7-bit US-ASCII character set. If a localized form is provided, element content MAY be represented in unrestricted UTF-8. The <contact:postalInfo> element contains the following child elements:

- A <contact:name> element that contains the name of the individual or role represented by the contact.
- A <contact:org> element that contains the name of the organization with which the contact is affiliated.
- A <contact:addr> element that contains address information associated with the contact. A <contact:addr> element contains the following child elements:
 - One, two, or three OPTIONAL <contact:street> elements that contain the contact's street address.
 - A <contact:city> element that contains the contact's city.

- An OPTIONAL <contact:sp> element that contains the contact's state or province.
- An OPTIONAL <contact:pc> element that contains the contact's postal code.
- A <contact:cc> element that contains the contact's country code.
- A <contact:voice> element that contains the contact's voice telephone number.
- A <contact:fax> element that contains the contact's facsimile telephone number.
- A <contact:email> element that contains the contact's email address.
- A <contact:authInfo> element that contains authorization information associated with the contact object.

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:      <contact:update
C:        xmlns:contact="urn:ietf:params:xml:ns:contact-1.0"
C:        xsi:schemaLocation="urn:ietf:params:xml:ns:contact-1.0
C:          contact-1.0.xsd">
C:        <contact:id>sh8013</contact:id>
C:        <contact:add>
C:          <contact:status s="clientDeleteProhibited"/>
C:        </contact:add>
C:        <contact:chg>
C:          <contact:postalInfo>
C:            <contact:org/>
C:            <contact:addr>
C:              <contact:street>124 Example Dr.</contact:street>
C:              <contact:street>Suite 200</contact:street>
C:              <contact:city>Dulles</contact:city>
C:              <contact:sp>VA</contact:sp>
C:              <contact:pc>20166-6503</contact:pc>
C:              <contact:cc>US</contact:cc>
C:            </contact:addr>
C:          </contact:postalInfo>
C:          <contact:voice>+1.7034444444</contact:voice>
C:          <contact:fax/>
C:          <contact:authInfo type="pw">2BARfoo</contact:authInfo>
C:        </contact:chg>
C:      </contact: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:  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 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="urn:ietf:params:xml:ns:contact-1.0"
  xmlns:contact="urn:ietf:params:xml:ns:contact-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
      contact provisioning schema.
    </documentation>
  </annotation>

  <!--
  Child elements found in EPP commands.
  -->
  <element name="check" type="contact:mIDType"/>
  <element name="create" type="contact:createType"/>
  <element name="delete" type="contact:sIDType"/>
  <element name="info" type="contact:sIDType"/>
  <element name="transfer" type="contact:transferType"/>
  <element name="update" type="contact:updateType"/>

  <!--
  Utility types.
  -->
  <simpleType name="ccType">
```



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

<complexType name="e164Type">
  <simpleContent>
    <extension base="contact:e164StringType">
      <attribute name="x" type="token"/>
    </extension>
  </simpleContent>
</complexType>

<simpleType name="e164StringType">
  <restriction base="token">
    <pattern value="(\+[0-9]{1,3}\.[0-9]{1,14})?"/>
    <maxLength value="17"/>
  </restriction>
</simpleType>

<simpleType name="pcType">
  <restriction base="token">
    <maxLength value="10"/>
  </restriction>
</simpleType>

<simpleType name="postalLineType">
  <restriction base="normalizedString">
    <minLength value="1"/>
    <maxLength value="255"/>
  </restriction>
</simpleType>

<simpleType name="optPostalLineType">
  <restriction base="normalizedString">
    <maxLength value="255"/>
  </restriction>
</simpleType>

<!--
Child elements of the <create> and <info> commands.
-->
<complexType name="createType">
  <sequence>
    <element name="id" type="eppcom:clIDType"/>
    <element name="postalInfo" type="contact:postalInfoType"
      maxOccurs="2"/>
    <element name="voice" type="contact:e164Type"
```



```
        minOccurs="0"/>
        <element name="fax" type="contact:e164Type"
        minOccurs="0"/>
        <element name="email" type="eppcom:minTokenType"/>
        <element name="authInfo" type="eppcom:authInfoType"/>
    </sequence>
</complexType>

<complexType name="postalInfoType">
    <sequence>
        <element name="name" type="contact:postalLineType"/>
        <element name="org" type="contact:optPostalLineType"
        minOccurs="0"/>
        <element name="addr" type="contact:addrType"/>
    </sequence>
</complexType>

<complexType name="addrType">
    <sequence>
        <element name="street" type="contact:optPostalLineType"
        minOccurs="0" maxOccurs="3"/>
        <element name="city" type="contact:postalLineType"/>
        <element name="sp" type="contact:optPostalLineType"
        minOccurs="0"/>
        <element name="pc" type="contact:pcType"
        minOccurs="0"/>
        <element name="cc" type="contact:ccType"/>
    </sequence>
</complexType>

<!--
Child element of commands that require only an identifier.
-->
<complexType name="sIDType">
    <sequence>
        <element name="id" type="eppcom:clIDType"/>
    </sequence>
</complexType>

<!--
Child element of commands that accept multiple identifiers.
-->
<complexType name="mIDType">
    <sequence>
        <element name="id" type="eppcom:clIDType"
        maxOccurs="unbounded"/>
    </sequence>
</complexType>
```



```
<!--
Child elements of the <transfer> command.
-->
<complexType name="transferType">
  <sequence>
    <element name="id" type="eppcom:clIDType"/>
    <element name="authInfo" type="eppcom:authInfoType"/>
  </sequence>
</complexType>

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

<!--
Data elements that can be added or removed.
-->
<complexType name="addRemType">
  <sequence>
    <element name="status" type="contact:statusType"
      maxOccurs="9"/>
  </sequence>
</complexType>

<!--
Data elements that can be changed.
-->
<complexType name="chgType">
  <sequence>
    <element name="postalInfo" type="contact:chgPostalInfoType"
      minOccurs="0" maxOccurs="2"/>
    <element name="voice" type="contact:e164Type"
      minOccurs="0"/>
    <element name="fax" type="contact:e164Type"
      minOccurs="0"/>
    <element name="email" type="token"
      minOccurs="0"/>
```



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

<complexType name="chgPostalInfoType">
  <sequence>
    <element name="name" type="contact:postalLineType"
      minOccurs="0"/>
    <element name="org" type="contact:optPostalLineType"
      minOccurs="0"/>
    <element name="addr" type="contact:addrType"
      minOccurs="0"/>
  </sequence>
</complexType>

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

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

  <complexType name="checkType">
    <sequence>
      <element name="id" type="contact:checkIDType"/>
      <element name="reason" type="eppcom:reasonType"
        minOccurs="0"/>
    </sequence>
  </complexType>

  <complexType name="checkIDType">
    <simpleContent>
      <extension base="eppcom:clIDType">
        <attribute name="avail" type="boolean"
          use="required"/>
      </extension>
    </simpleContent>
  </complexType>
```



```
        </simpleContent>
    </complexType>

<!--
<create> response elements.
-->
    <complexType name="creDataType">
        <sequence>
            <element name="id" type="eppcom:clIDType"/>
            <element name="crDate" type="dateTime"/>
        </sequence>
    </complexType>

<!--
<info> response elements.
-->
    <complexType name="infDataType">
        <sequence>
            <element name="id" type="eppcom:clIDType"/>
            <element name="roid" type="eppcom:roidType"/>
            <element name="status" type="contact:statusType"
                maxOccurs="8"/>
            <element name="postalInfo" type="contact:postalInfoType"
                maxOccurs="2"/>
            <element name="voice" type="contact:e164Type"
                minOccurs="0"/>
            <element name="fax" type="contact:e164Type"
                minOccurs="0"/>
            <element name="email" type="token"/>
            <element name="clID" type="eppcom:clIDType"/>
            <element name="crID" type="eppcom:clIDType"/>
            <element name="crDate" type="dateTime"/>
            <element name="upID" type="eppcom:clIDType"
                minOccurs="0"/>
            <element name="upDate" type="dateTime"
                minOccurs="0"/>
            <element name="trDate" type="dateTime"
                minOccurs="0"/>
            <element name="authInfo" type="eppcom: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.
-->
    <complexType name="statusType">
```



```
<simpleContent>
  <extension base="normalizedString">
    <attribute name="s" type="contact:statusValueType"
      use="required"/>
    <attribute name="lang" type="language"
      default="en"/>
  </extension>
</simpleContent>
</complexType>

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

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

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


5. Internationalization Considerations

EPP is represented in XML, which provides native support for encoding information using the Unicode character set and its more compact representations including UTF-8. Compliant XML processors are REQUIRED to understand both UTF-8 and UTF-16. Though XML includes provisions to identify other character set encodings through use of an "encoding" attribute in an `<?xml?>` declaration, EPP use with character sets other than UTF-8 is NOT RECOMMENDED.

All date-time values presented via EPP MUST be expressed in Universal Coordinated Time using the Gregorian calendar. XML Schema allows use of time zone identifiers to indicate offsets from the zero meridian, but this option MUST NOT be used with EPP. The extended date-time form defined in [\[DATE-TIME\]](#) MUST be used to represent date-time values as XML Schema does not support truncated date-time forms.

Humans, organizations, and other entities often need to represent social information in both a commonly understood character set and a locally optimized character set. This specification provides features allowing representation of social information in both a subset of UTF-8 for broad readability and unrestricted UTF-8 for local optimization.

6. IANA Considerations

This document uses URNs to describe XML namespaces and XML schemas conforming to a registry mechanism described in [\[IETF-XML\]](#). Two URI assignments are requested.

Registration request for the contact namespace:

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

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

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

Registration request for the contact XML schema:

URI: urn:ietf:params:xml:schema:contact-1.0

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

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

7. Security Considerations

An authorization token is REQUIRED to create a contact object. This token is used in transfer operations as an additional means of determining client authorization to perform the command. Failure to protect authorization tokens can result in unauthorized transfer operations. Both client and server MUST ensure that tokens are stored and exchanged with high-grade encryption mechanisms to provide privacy services.

The object mapping described in this document does not provide any other security services or introduce any additional considerations beyond those described by [[EPP](#)] and protocol layers used by EPP.

8. Acknowledgements

This document was originally written as an individual submission Internet-Draft. The provreg working group later adopted it as a working group document and provided many invaluable comments and suggested improvements. The author wishes to acknowledge the efforts of WG chairs Edward Lewis and Jaap Akkerhuis for their process and editorial contributions.

Specific suggestions that have been incorporated into this document were provided by Chris Bason, Eric Brunner-Williams, Jordyn Buchanan, Dave Crocker, Ayesha Damaraju, Anthony Eden, Sheer El-Showk, Dipankar Ghosh, Klaus Malorny, Dan Manley, Michael Mealling, Patrick Mevzek, Asbjorn Steira, and Rick Wesson.

9. References

Normative References:

[DATE-TIME] G. Klyne, C. Newman: "Date and Time on the Internet: Timestamps", work in progress.

[EPP] S. Hollenbeck: "Extensible Provisioning Protocol", work in progress.

[IETF-XML] M. Mealling: "The IETF XML Registry", work in progress.

[ISO3166] ISO 3166-1: "Codes for the representation of names of countries and their subdivisions - Part 1: Country codes", October 1997.

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

[RFC2279] F. Yergeau: "UTF-8, a transformation format of ISO 10646", [RFC 2279](#), January 1998.

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

Informative References:

[E164a] ITU-T Recommendation E.164: "The International Public Telecommunication Numbering Plan", May 1997.

[E164b] Complement To ITU-T Recommendation E.164 (05/1997): "List of ITU-T Recommendation E.164 assigned country codes", June 2000.

[XML] Editor 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.

10. Author's Address

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

A. Revisions From Previous Version

(Note to RFC editor: please remove this section completely before publication as an RFC.)

-03 to -04 (WG last call updates):

Fixed typos.

Fixed the e164StringType to allow full-length numbers with country codes of less than three characters.

Increased the maximum postalLineType length from 64 to 255.

Changed type of email and telephone number types to "token" from "normalizedString" since these values should never contain certain acceptable normalizedString characters.

Fixed schema to prevent multiple choices in updates, replacing <choice> with <sequence>.

Made <authInfo> mandatory for transfer queries.

Added a sentence to [section 3.2.2](#) to encourage server implementers to notify clients when a delete command fails due to existing object relationships.

Modified the schema to collapse the <ascii> and <i15d> elements into a single <postalInfo> element that may occur once or twice depending on context.

Modified the schema to ensure that required string fields can't be specified as empty strings.

Changed some lower-case "must"s, "may"s, etc. to avoid confusion with [RFC 2119](#) directives.

Updated the security considerations section to note the need to protect authorization tokens.

Separated references into normative and informative subsections.

Removed unneeded references to ISO 11180 and US-ASCII.

Replaced reference to ISO 8601 with an IMPP I-D that will hopefully be an RFC soon.

B. Full Copyright Statement

Copyright (C) The Internet Society 2002. All Rights Reserved.

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

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

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

Acknowledgement

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

