

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: December 14, 2013

J. XIE
CONAC
June 12, 2013

Extensible Provisioning Protocol (EPP) Trademark Mapping
draft-xie-epp-trademark-mapping-00

Abstract

This document describes an Extensible Provisioning Protocol (EPP) mapping for the provisioning and management of trademark stored in a trademark clearinghouse. Specified in XML, the mapping defines EPP command syntax and semantics as applied to trademark.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on December 14, 2013.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1.	Introduction	3
2.	Terminology	3
3.	Object Attributes	3
3.1.	Trademark	3
3.2.	Domain Label	3
3.3.	Contact and Client Identifiers	4
3.4.	Status Values	4
3.5.	Dates and Times	5
3.6.	Validity Periods	5
3.7.	Authorization Information	6
4.	EPP Command Mapping	6
4.1.	EPP Query Commands	6
4.1.1.	EPP <check> Command	6
4.1.2.	EPP <info> Command	8
4.1.3.	EPP <transfer> Query Command	12
4.2.	EPP Transform Commands	14
4.2.1.	EPP <create> Command	14
4.2.2.	EPP <delete> Command	16
4.2.3.	EPP <renew> Command	17
4.2.4.	EPP <transfer> Command	19
4.2.5.	EPP <update> Command	21
5.	Formal Syntax	21
6.	Internationalization Considerations	31
7.	IANA Considerations	31
8.	Security considerations	31
9.	Acknowledgements	31
10.	Change History	32
10.1.	draft-xie-epp-trademark-mapping: Version 00	32
11.	References	32
11.1.	Normative References	32
11.2.	Informative References	33
	Author's Address	33

1. Introduction

This document describes a trademark mapping for version 1.0 of the Extensible Provisioning Protocol (EPP). This document is specified using the Extensible Markup Language (XML) 1.0 as described in [W3C.REC-xml-20040204] and XML Schema notation as described in [W3C.REC-xmlschema-1-20041028] and [W3C.REC-xmlschema-2-20041028].

The EPP core protocol specification [RFC5730] provides a complete description of EPP command and response structures. A thorough understanding of the base protocol specification is necessary to understand the trademark 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.

2. Terminology

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].

"trademark-1.0" in this document is used as an abbreviation for urn:ietf:params:xml:ns:trademark-1.0.

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 are provided only to illustrate element relationships and are not a REQUIRED feature of this specification.

3. Object Attributes

An EPP trademark 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.

3.1. Trademark

Trademark which is a complex object can take 3 forms: Registered Trademark, Court Validated Marks, Marks protected by statute or treaty.

3.2. Domain Label

The domain name without its TLD extension.

3.3. 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].

3.4. Status Values

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

- o clientDeleteProhibited, serverDeleteProhibited. Requests to delete the object MUST be rejected.
- o clientRenewProhibited, serverRenewProhibited. Requests to renew the object MUST be rejected.
- o clientTransferProhibited, serverTransferProhibited. Requests to transfer the object MUST be rejected.
- o clientUpdateProhibited, serverUpdateProhibited. Requests to update the object (other than to remove this status) MUST be rejected.
- o 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.

- o 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.

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.

3.5. 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 using upper case "T" and "Z" characters defined in [W3C.REC-xmlschema-2-20041028] MUST be used to represent date-time values, as XML Schema does not support truncated date-time forms or lower case "t" and "z" characters.

3.6. Validity Periods

A trademark object MAY have a specified validity period. If server policy supports trademark-object validity periods, the validity period is defined when a trademark object is created, and it MAY be extended by the EPP <renew> or <transfer> commands. As a matter of

server policy, this specification does not define actions to be taken upon expiration of a trademark 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 (1). The maximum allowable value is ninety-nine decimal (99). A server MAY support a lower maximum value.

3.7. Authorization Information

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

4. EPP Command Mapping

A detailed description of the EPP syntax and semantics can be found in the EPP core protocol specification [RFC5730]. The command mappings described here are specifically for use in provisioning and managing trademark via EPP.

4.1. EPP Query Commands

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

4.1.1. 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, as object-provisioning requirements are ultimately a matter of server policy.

In addition to the standard EPP command elements, the <check>command MUST contain a <trademark:check> element that identifies the trademark namespace. The <trademakr:check> element contains the following child elements:

- o One or more <trademark:id> element that contains the fully qualified id of the trademark object 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:      <trademark:check
C:        xmlns:trademark="urn:ietf:params:xml:ns:trademark-1.0">
C:          <trademark:id>123456-001</trademark:id>
C:          <trademark:id>123456-002</trademark:id>
C:          <trademark:id>123456-003</trademark:id>
C:        </trademark: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 <trademark:chkData> element that identifies the trademark namespace. The <trademark:chkData> element contains one or more <trademark:cd> elements that contain the following child elements:

- o A <trademark:id> element that contains the fully qualified id of the queried trademark object. This element MUST contain an "avail" attribute whose value indicates object availability (can it be provisioned or not) at the moment the <check> command was completed. A value of "1" or "true" means that the object can be provisioned. A value of "0" or "false" means that the object can not be provisioned.
- o An OPTIONAL <trademark:reason> element that MAY be provided when an object cannot be provisioned. If present, this element contains server-specific text to help explain why the object cannot be provisioned. 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:  <response>
S:    <result code="1000">
S:      <msg>Command completed successfully</msg>
```

```
S:    </result>
S:    <resData>
S:      <trademark:chkData
S:        xmlns:trademark="urn:ietf:params:xml:ns:trademark-1.0">
S:          <trademark:cd>
S:            <trademark:id avail="1">123456-001</trademark:id>
S:          </trademark:cd>
S:          <trademark:cd>
S:            <trademark:id avail="0">123456-002</trademark:id>
S:            <trademark:reason>In use</trademark:reason>
S:          </trademark:cd>
S:          <trademark:cd>
S:            <trademark:id avail="1">123456-003</trademark:id>
S:          </trademark:cd>
S:        </trademark:chkData>
S:      </resData>
S:      <trID>
S:        <clTRID>ABC-12345</clTRID>
S:        <svTRID>54322-XYZ</svTRID>
S:      </trID>
S:    </response>
S:  </epp>
```

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

4.1.2. EPP <info> Command

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

- o A <trademark:id> element that contains the fully qualified id of the trademark object.

- o An OPTIONAL <trademark:authInfo> element that contains authorization information associated with the trademark object or authorization information associated with the trademark object's holder or associated contacts. An OPTIONAL "roid" attribute MUST be used to identify the holder or contact object if and only if the given authInfo is associated with a holder or contact object, and not the trademark 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> Response:

```
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:      <trademark:info
C:        xmlns:trademark="urn:ietf:params:xml:ns:trademark-1.0">
C:          <trademark:id>123456-01</trademark:name>
C:        </trademark: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 <trademark:infData> element that identifies the trademark namespace. Elements that are not OPTIONAL MUST be returned; OPTIONAL elements are returned based on client authorization and server policy. The <trademark:infData> element contains the following child elements:

- o A <trademark:id> element that contains the fully qualified id of the trademark object.
- o A <trademark:roid> element that contains the Repository Object Identifier assigned to the trademark object when the object was created.
- o Zero or more OPTIONAL <trademark:status> elements that contain the current status descriptors associated with the trademark.
- o If supported by the server, one OPTIONAL <trademark:holder> element and one or more OPTIONAL <trademark:contact> elements that contain identifiers for the human or organizational social information objects associated with the trademark object.

- o An OPTIONAL element that contains the fully qualified names of the delegated host objects or host attributes (name servers) associated with the domain object. See Section 1.1 for a description of the elements used to specify host objects or host attributes.
- o Zero or more OPTIONAL elements that contain the fully qualified names of the subordinate host objects that exist under this superordinate domain object.
- o A element that contains the identifier of the sponsoring client.
- o An OPTIONAL element that contains the identifier of the client that created the domain object.
- o An OPTIONAL element that contains the date and time of domain object creation.
- o An OPTIONAL element that contains the date and time identifying the end of the domain object's registration period.
- o An OPTIONAL element that contains the identifier of the client that last updated the domain object. This element MUST NOT be present if the domain has never been modified.
- o An OPTIONAL element that contains the date and time of the most recent domain-object modification. This element MUST NOT be present if the domain object has never been modified.
- o An OPTIONAL element that contains the date and time of the most recent successful domain-object transfer. This element MUST NOT be provided if the domain object has never been transferred.
- o An OPTIONAL element that contains authorization information associated with the domain 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:      <trademark:infData
S:      xmlns:trademark="urn:ietf:params:xml:ns:trademark-1.0">
S:      <trademark:id>00000712423-1</trademark:id>
S:      <trademark:roid>000000001-abc</trademark:roid>
S:      <trademark:name>example one</trademark:name>
S:      <trademark:status s="pendingCreate"/>
S:      <trademark:jurisdiction>US</trademark:jurisdiction>
S:      <trademark:class>35</trademark:class>
S:      <trademark:class>36</trademark:class>
S:      <trademark:goodsAndServices>Dirigendas et eiusmodi
S:      featuring infringo in airfare et cartam servicia.
S:      </trademark:goodsAndServices>
S:      <trademark:regNum>234235</trademark:regNum>
S:      <trademark:regDate>2009-08-16T09:00:00.0Z
S:      </trademark:regDate>
S:      <trademark:exDate>2015-08-16T09:00:00.0Z
S:      </trademark:exDate>
S:      <trademark:period unit="y">5</trademark:period>
S:      <trademark:holder>jd1234</trademark:holder>
S:      <trademark:contact type="admin">sh8013</trademark:contact>
S:      <trademark:contact type="tech">sh8013</trademark:contact>
S:      <trademark:authInfo>
S:      <trademark:pw>2fooBAR</trademark:pw>
S:      </trademark:authInfo>
S:      <trademark:document>
S:      <trademark:docType>declProofOfUseOneSample
S:      </trademark:docType>
S:      <trademark:fileName>C:\\ddafs\\file.jpg
S:      </trademark:fileName>
S:      <trademark:fileType>jpg</fileType>
S:      <trademark:fileContent>
S:      9j/4AAQSkZJRgABAQEASABIAAD/2wBDAAUDBAQEAwUEBAQFBQUGBwwI
S:      wCHBw8LCwkMEQ8SEhEPERETFhwXExQaFRERGCCEYGH0dHx8fExcijCie
S:      JBweHx7/wAALCAABAAEBAAREA/8QAFABAAAAAAAAAAAAAAAAAAACP/
S:      ABQQAQAAAAAAAAAAAAAAAAAAD/2gAIAQEAAAD8AZb//2Q==
S:      </trademark:fileContent>
S:      </trademark:document>
S:      <trademark:label>
S:      <trademark:aLabel>exampleone</trademark:aLabel>
S:      <trademark:smdInclusion enable="1"/>
S:      <trademark:claimsNotify enable="1"/>
S:      </trademark:label>
S:      <trademark:label>
S:      <trademark:aLabel>example-one</trademark:aLabel>
S:      <trademark:smdInclusion enable="1"/>
S:      <trademark:claimsNotify enable="1"/>
S:      </trademark:label>
S:      <trademakr:creDate>2009-08-16T09:00:00.0Z</trademakr:creDate>

```

```

S:      <trademakr:upDate>2010-08-16T09:00:00.0Z</trademakr:upDate>
S:      <trademakr:trnDate>2010-10-16T09:00:00.0Z</trademakr:trnDate>
S:      </trademark:info>
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 cannot be processed for any reason.

4.1.3. EPP <transfer> Query Command

The EPP <transfer> command provides a query operation that allows a client to determine the 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 <trademark:transfer> element that identifies the trademark namespace. The <trademark:transfer> element contains the following child elements:

- o A <trademark:id> element that contains the fully qualified id of the trademark object.
- o An OPTIONAL <trademark:authInfo> element that contains authorization information associated with the trademark object or authorization information associated with the trademark object's holder or associated contacts. An OPTIONAL "roid" attribute MUST be used to identify the holder or contact object if and only if the given authInfo is associated with a holder or contact object, and not the trademark 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:      <trademark:transfer
C:        xmlns:trademark="urn:ietf:params:xml:ns:trademark-1.0">
C:        <trademark:id>123456-001</trademark:id>

```

```

C:      <trademark:authInfo>
C:      <trademark:pw>2fooBAR</trademark:pw>
C:      </trademark:authInfo>
C:      </trademark: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 <trademakr:trnData> element that identifies the trademark namespace. The <trademark:trnData> element contains the following child elements:

- o A <trademark:id> element that contains the fully qualified id of the trademark object.
- o A <trademark:trStatus> element that contains the state of the most recent transfer request.
- o A <trademark:reID> element that contains the identifier of the client that requested the object transfer.
- o A <trademark:reDate> element that contains the date and time that the transfer was requested.
- o A <trademark:acID> element that contains the identifier of the client that SHOULD act upon a PENDING transfer request. For all other status types, the value identifies the client that took the indicated action.
- o A <trademark: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.
- o An OPTIONAL <trademark:exDate> element that contains the end of the trademark object's validity period if the <transfer> command caused or causes a change in the validity period.

Example <transfer> 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:   <trademark:trnData
S:     xmlns:trademark="urn:ietf:params:xml:ns:trademark-1.0">
S:     <trademark:id>123456-001</trademark:id>
S:     <trademark:trStatus>pending</trademark:trStatus>
S:     <trademark:reID>ClientX</trademark:reID>
S:     <trademark:reDate>2010-06-06T22:00:00.0Z</trademark:reDate>
S:     <trademark:acID>ClientY</trademark:acID>
S:     <trademark:acDate>2011-06-11T22:00:00.0Z</trademark:acDate>
S:     <trademark:exDate>2012-09-08T22:00:00.0Z</trademark:exDate>
S:   </trademark:trnData>
S: </resData>
S: <trID>
S:   <clTRID>ABC-12345</clTRID>
S:   <svTRID>54322-XYZ</svTRID>
S: </trID>
S: </response>
S:</epp>

```

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

4.2. EPP Transform Commands

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

4.2.1. EPP <create> Command

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

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:   <trademark:create
C:     xmlns:trademark="urn:ietf:params:xml:ns:trademark-1.0">
C:       <trademark:jurisdiction>US</trademark:jurisdiction>
C:       <trademark:class>35</trademark:class>
C:       <trademark:class>36</trademark:class>
C:       <trademark:goodsAndServices>Dirigendas et eiusmodi
C:         featuring infringo in airfare et cartam servicia.
C:     </trademark:goodsAndServices>
C:     <trademark:regNum>234235</trademark:regNum>
C:     <trademark:regDate>2009-08-16T09:00:00.0Z</trademark:regDate>
C:     <trademark:exDate>2015-08-16T09:00:00.0Z</trademark:exDate>
C:     <trademark:period unit="y">5</trademark:period>
C:     <trademark:holder>jd1234</trademark:holder>
C:     <trademark:contact type="admin">sh8013</trademark:contact>
C:     <trademark:contact type="tech">sh8013</trademark:contact>
C:     <trademark:authInfo>
C:       <trademark:pw>2fooBAR</trademark:pw>
C:     </trademark:authInfo>
C:     <trademark:document>
C:       <trademark:docType>declProofOfUseOneSample
C:     </trademark:docType>
C:       <trademark:fileName>C:\\ddafs\\file.jpg</trademark:fileName>
C:       <trademark:fileType>jpg</fileType>
C:       <trademark:fileContent>
C:         /9j/4AAQSkZJRgABAQEASABIAAD/2wBDAAUDBAQEAwUEBAQFBQUGBwwIB
C:         cHBw8LCwkMEQ8SEhEPPERETFhwXExQaFRERGCeYGH0dHx8fExcIJCIEJB
C:         eHx7/wAALCAABAAEREA/8QAFABAAAAAAAAAAAAAAAAAAAAACP/EABQQA
C:         AAAAAAAAAAAAAAAAAAAD/2gAIAQEAAAD8AZb//2Q==
C:       </trademark:fileContent>
C:     </trademark:document>
C:     <trademark:label>
C:       <trademark:aLabel>exampleone</trademark:aLabel>
C:       <trademark:smdInclusion enable="1"/>
C:       <trademark:claimsNotify enable="1"/>
C:     </trademark:label>
C:     <trademark:label>
C:       <trademark:aLabel>example-one</trademark:aLabel>
C:       <trademark:smdInclusion enable="1"/>
C:       <trademark:claimsNotify enable="1"/>
C:     </trademark:label>
C:   </trademark: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 <trademark:creData> element that identifies the trademark namespace. The <trademark:creData> element contains the following child elements:

- o A <trademark:id> element that contains the fully qualified id of the trademark object.
- o A <trademark:crDate> element that contains the date and time of trademark object creation.
- o An OPTIONAL <trademark:exDate> element that contains the date and time identifying the end of the trademark 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:      <trademark:creData
S:        xmlns:trademark="urn:ietf:params:xml:ns:trademark-1.0">
S:        <trademark:id>00000712423-1</trademark:id>
S:        <trademark:crDate>1999-04-03T22:00:00.0Z</trademark:crDate>
S:        <trademark:exDate>2001-04-03T22:00:00.0Z</trademark:exDate>
S:      </trademark:creData>
S:    </resData>
S:    <trID>
S:      <clTRID>ABC-12345</clTRID>
S:      <svTRID>54321-XYZ</svTRID>
S:    </trID>
S:  </response>
S:</epp>
```

4.2.2. EPP <delete> Command

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

- o A <trademark:id> element that contains the fully qualified id of the trademark 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:      <trademark:delete
C:        xmlns:trademark="urn:ietf:params:xml:ns:trademark-1.0">
C:          <trademark:id>123456-001</trademark:id>
C:        </trademark: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:  <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 cannot be processed for any reason.

4.2.3. EPP <renew> Command

The EPP <renew> command provides a transform operation that allows a client to extend the validity period of a trademark object. In addition to the standard EPP command elements, the <renew> command MUST contain a <trademark:renew> element that identifies the trademark namespace. The <trademark:renew> element contains the following child elements:

- o A <trademark:id> element that contains the fully qualified id of the trademark object whose validity period is to be extended.
- o A <trademark: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.
- o An OPTIONAL <trademark:period> element that contains the number of units to be added to the registration period of the trademark object. The number of units available MAY be subject to limits imposed by the server.

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:      <trademark:renew
C:        xmlns:trademark="urn:ietf:params:xml:ns:trademark-1.0">
C:          <trademark:id>123456-001</trademark:id>
C:          <trademark:curExpDate>2000-04-03</trademark:curExpDate>
C:          <trademark:period unit="y">5</trademark:period>
C:        </trademark: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 <trademark:renData> element that identifies the trademark namespace. The <trademark:renData> element contains the following child elements:

- o A <trademark:id> element that contains the fully qualified id of the trademark object.

- o An OPTIONAL `<trademark:exDate>` element that contains the date and time identifying the end of the trademark 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:      <trademark:renData
S:        xmlns:trademark="urn:ietf:params:xml:ns:trademark-1.0">
S:        <trademark:id>123456-001</trademark:id>
S:        <trademark:exDate>2012-04-03T22:00:00.0Z</trademark:exDate>
S:      </trademark: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 cannot be processed for any reason.

4.2.4. EPP `<transfer>` Command

The EPP `<transfer>` command provides a transform operation that allows a client to manage requests to transfer the sponsorship of a trademark object. In addition to the standard EPP command elements, the `<transfer>` command MUST contain a `<trademark:transfer>` element that identifies the trademark namespace. The `<trademark:transfer>` element contains the following child elements:

- o A `<trademark:id>` element that contains the fully qualified id of the trademark object for which a transfer request is to be created, approved, rejected, or cancelled.

- o An OPTIONAL `<trademark:period>` element that contains the number of units to be added to the registration period of the trademark object at completion of the transfer process. This element can only be used when a transfer is requested, and it MUST be ignored if used otherwise. The number of units available MAY be subject to limits imposed by the server.
- o A `<trademark:authInfo>` element that contains authorization information associated with the trademark object or authorization information associated with the trademark object's holder or associated contacts. An OPTIONAL "roid" attribute MUST be used to identify the holder or contact object if and only if the given authInfo is associated with a holder or contact object, and not the trademark 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:      <trademark:transfer
C:        xmlns:trademark="urn:ietf:params:xml:ns:trademark-1.0">
C:        <trademark:id>123456-001</trademark:id>
C:        <trademark:period unit="y">1</trademark:period>
C:        <trademark:authInfo>
C:          <trademark:pw roid="JD1234-REP">2fooBAR</trademark:pw>
C:        </trademark:authInfo>
C:      </trademark: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 `<trademark:trnData>` element that identifies the trademark namespace. The `<trademark:trnData>` element contains the same child elements defined for a transfer query response.

Example `<transfer>` 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:      <trademark:trnData
S:        xmlns:trademark="urn:ietf:params:xml:ns:trademark-1.0">
S:        <trademark:id>123456-001</trademark:id>
S:        <trademark:trStatus>pending</trademark:trStatus>
S:        <trademark:reID>ClientX</trademark:reID>
S:        <trademark:reDate>2010-06-06T22:00:00.0Z</trademark:reDate>
S:        <trademark:acID>ClientY</trademark:acID>
S:        <trademark:acDate>2011-06-11T22:00:00.0Z</trademark:acDate>
S:        <trademark:exDate>2012-09-08T22:00:00.0Z</trademark:exDate>
S:      </trademark:trnData>
S:    </resData>
S:    <trID>
S:      <clTRID>ABC-12345</clTRID>
S:      <svTRID>54322-XYZ</svTRID>
S:    </trID>
S:  </response>
S:</epp>

```

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

4.2.5. EPP <update> Command

TBD

5. Formal Syntax

An EPP Trademark 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:trademark-1.0"
  xmlns:trademark="urn:ietf:params:xml:ns:trademark-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
    Trademark Mapping Schema v1.0
  </documentation>
</annotation>

<!--
Child elements found in EPP commands.
-->
<element name="check" type="trademark:mNameType"/>
<element name="create" type="trademark:createType"/>
<element name="delete" type="trademark:sNameType"/>
<element name="info" type="trademark:infoType"/>
<element name="renew" type="trademark:renewType"/>
<element name="transfer" type="trademark:transferType"/>
<element name="update" type="trademark:updateType"/>
<!--
Child elements of the <create> command.
-->
<complexType name="createType">
  <sequence>
    <element name="name" type="eppcom:labelType"/>
    <element name="period" type="trademark:periodType"
      minOccurs="0"/>
    <element name="ns" type="trademark:nsType"
      minOccurs="0"/>
    <element name="holder" type="eppcom:clIDType"
      minOccurs="0"/>
    <element name="contact" type="trademark:contactType"
      minOccurs="0" maxOccurs="unbounded"/>
    <element name="authInfo" type="trademark:authInfoType"/>
  </sequence>
</complexType>

<complexType name="periodType">
  <simpleContent>
    <extension base="trademark:pLimitType">

```

```

        <attribute name="unit" type="trademark: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="nsType">
  <choice>
    <element name="hostObj" type="eppcom:labelType"
      maxOccurs="unbounded"/>
    <element name="hostAttr" type="trademark:hostAttrType"
      maxOccurs="unbounded"/>
  </choice>
</complexType>
<!--
Name servers are either host objects or attributes.
-->

<complexType name="hostAttrType">
  <sequence>
    <element name="hostName" type="eppcom:labelType"/>
    <element name="hostAddr" type="host:addrType"
      minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
</complexType>
<!--
If attributes, addresses are optional and follow the
structure defined in the host mapping.
-->

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

```

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

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



```
<complexType name="infoNameType">
  <simpleContent>
    <extension base = "eppcom:labelType">
      <attribute name="hosts" type="trademark:hostsType"
        default="all"/>
    </extension>
  </simpleContent>
</complexType>
```

```
<simpleType name="hostsType">
  <restriction base="token">
    <enumeration value="all"/>
    <enumeration value="del"/>
    <enumeration value="none"/>
    <enumeration value="sub"/>
  </restriction>
</simpleType>
```

```
<!--
Child elements of the <renew> command.
-->
```

```
<complexType name="renewType">
  <sequence>
    <element name="name" type="eppcom:labelType"/>
    <element name="curExpDate" type="date"/>
    <element name="period" type="trademark:periodType"
      minOccurs="0"/>
  </sequence>
</complexType>
```

```
<!--
Child elements of the <transfer> command.
-->
```

```
<complexType name="transferType">
  <sequence>
    <element name="name" type="eppcom:labelType"/>
    <element name="period" type="trademark:periodType"
      minOccurs="0"/>
    <element name="authInfo" type="trademark:authInfoType"
      minOccurs="0"/>
  </sequence>
</complexType>
```

```
<!--
Child elements of the <update> command.
-->
```

```
<complexType name="updateType">
  <sequence>
```

```
<element name="name" type="eppcom:labelType"/>
<element name="add" type="trademark:addRemType"
  minOccurs="0"/>
<element name="rem" type="trademark:addRemType"
  minOccurs="0"/>
<element name="chg" type="trademark:chgType"
  minOccurs="0"/>
</sequence>
</complexType>

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

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

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

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

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

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

```
<!--
<create> response elements.
-->
<complexType name="creDataType">
  <sequence>
    <element name="name" type="eppcom:labelType"/>
    <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="eppcom:labelType"/>
    <element name="roid" type="eppcom:roidType"/>
    <element name="status" type="trademark:statusType"
      minOccurs="0" maxOccurs="11"/>
    <element name="holder" type="eppcom:clIDType"
      minOccurs="0"/>
    <element name="contact" type="trademark:contactType"
      minOccurs="0" maxOccurs="unbounded"/>
    <element name="ns" type="trademark:nsType"
      minOccurs="0"/>
    <element name="host" type="eppcom:labelType"
      minOccurs="0" maxOccurs="unbounded"/>
    <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="trademark: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="trademark: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="inactive"/>
    <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>

<!--
Pending action notification response elements.
-->
<complexType name="panDataType">
  <sequence>
    <element name="name" type="trademark: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="eppcom:labelType"/>
  <element name="exDate" type="dateTime"
    minOccurs="0"/>
</sequence>
</complexType>

<!--
<transfer> response elements.
-->
<complexType name="trnDataType">
<sequence>
  <element name="name" type="eppcom:labelType"/>
  <element name="trStatus" type="eppcom:trStatusType"/>
  <element name="reID" type="eppcom:clIDType"/>
  <element name="reDate" type="dateTime"/>
  <element name="acID" type="eppcom:clIDType"/>
  <element name="acDate" type="dateTime"/>
  <element name="exDate" type="dateTime"
    minOccurs="0"/>
</sequence>
</complexType>

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

6. 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. Conformant XML processors recognize both UTF-8 and UTF-16. Though XML includes provisions to identify and use other character encodings through use of an "encoding" attribute in an `<?xml?>` declaration, use of UTF-8 is RECOMMENDED.

As the EPP Trademark mapping, the elements, element content described in this document MUST inherit the internationalization conventions used to represent higher-layer and core protocol structures present in an XML instance that includes this extension.

7. IANA Considerations

This document uses URNs to describe XML namespaces and XML schemas conforming to a registry mechanism described in [RFC3688]. IANA is requested to assign the following two URI.

Registration request for the trademark namespace:

- o URI: urn:ietf:params:xml:ns:trademark-1.0
- o Registrant Contact: See the "Author's Address" section of this document.
- o XML: None. Namespace URI does not represent an XML specification.

Registration request for the mark XML schema:

- o URI: urn:ietf:params:xml:schema:trademark-1.0
- o Registrant Contact: See the "Author's Address" section of this document.
- o XML: See the "Formal Syntax" section of this document.

8. Security considerations

The object mapping extension described in this document does not provide any other security services or introduce any additional considerations beyond those described by [RFC5730] or those caused by the protocol layers used by EPP.

9. Acknowledgements

The authors especially thank the authors of [RFC5730] and [RFC5731].

10. Change History

TBD

10.1. draft-xie-epp-trademark-mapping: Version 00

- o First draft.

11. References

11.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, January 2004.
- [RFC5730] Hollenbeck, S., "Extensible Provisioning Protocol (EPP)", STD 69, RFC 5730, August 2009.
- [RFC5731] Hollenbeck, S., "Extensible Provisioning Protocol (EPP) Domain Name Mapping", STD 69, RFC 5731, August 2009.
- [RFC5890] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework", RFC 5890, August 2010.
- [RFC5891] Klensin, J., "Internationalized Domain Names in Applications (IDNA): Protocol", RFC 5891, August 2010.
- [RFC5892] Faltstrom, P., "The Unicode Code Points and Internationalized Domain Names for Applications (IDNA)", RFC 5892, August 2010.
- [W3C.REC-xml-20040204]
Bray, T., Paoli, J., Sperberg-McQueen, C., Maler, E., and F. Yergeau, "Extensible Markup Language (XML) 1.0 (Third Edition)", World Wide Web Consortium FirstEdition REC-xml-20040204", February 2004,
<<http://www.w3.org/TR/2004/REC-xml-20040204>>.
- [W3C.REC-xmlschema-1-20041028]
Thompson, H., Beech, D., Maloney, M., and N. Mendelsohn, "XML Schema Part 1: Structures Second Edition", World Wide Web Consortium Recommendation REC-xmlschema-1-20041028", October 2004,
<<http://www.w3.org/TR/2004/REC-xmlschema-1-20041028>>.

[W3C.REC-xmlschema-2-20041028]

Biron, P. and A. Malhotra, "XML Schema Part 2: Datatypes Second Edition", World Wide Web Consortium Recommendation REC-xmlschema-2-20041028", October 2004, <<http://www.w3.org/TR/2004/REC-xmlschema-2-20041028>>.

11.2. Informative References

[RFC4290] Klensin, J., "Suggested Practices for Registration of Internationalized Domain Names (IDN)", RFC 4290, December 2005.

Author's Address

Jiagui XIE
CONAC
Jia 31, North Guangximen, Xibahe, Chaoyang District
Beijing, Beijing 100028
China

Phone: +86 10 5203 5025

Email: xiejg@conac.cn