



## **EPP Extensions for the .coop TLD Registrant Verification**

<b>Version:</b>	<b>1.7</b>
<b>Updated:</b>	<b>2010 October 5</b>
<b>Detail:</b>	<b>1.7 Changes to verification process: cooperative references and .Coop Verification Codes (CVC).</b>

### **Midcounties Co-operative Domains Ltd.**

This document describes extensions to the EPP protocol to support the requirements of the .coop domain.

1	The .coop Verification Process .....	1
1.1	Thick Registry .....	1
1.2	Verification of Registrants.....	1
1.3	Verification Process.....	1
1.4	Verification States.....	1
1.5	Active Domains.....	2
1.6	Acceptance of New Domain and Host Provisioning Requests.....	2
1.7	Cooperative references and .Coop Verification Codes (CVCs) .....	2
1.8	Communication With Cooperative References During Verification....	3
1.9	Communication With Registrar/Registrants During Verification.....	3
2	EPP Extensions .....	4
2.1	Relationship to other EPP Documents .....	4
2.2	Background .....	4
2.3	Extensions Namespace .....	4
2.4	Creating a .coop registrant .....	5
2.5	Summary .....	5
2.6	<greeting> .....	6
2.7	Extensions to <contact:create> command.....	6
2.8	Extensions to <contact:update> command.....	7
2.9	Extensions to <contact:info> response.....	8
2.10	Extensions to <domain:create> response.....	9
2.11	Extensions to <domain:update> response.....	10
2.12	Extensions to <poll> response.....	10
2.13	<coop:state>.....	11
3	XML Schema for Extensions.....	12

# 1 The .coop Verification Process

This section summarises the process by which a registrant is verified.

## 1.1 Thick Registry

The .coop registry is a thick registry, and therefore stores significant amounts of contact information. To register a domain, a registrant must be provided. A registrant is a contact that has been specified as the registrant for the provisioning of a new domain. Not all contacts are registrants.

## 1.2 Verification of Registrants

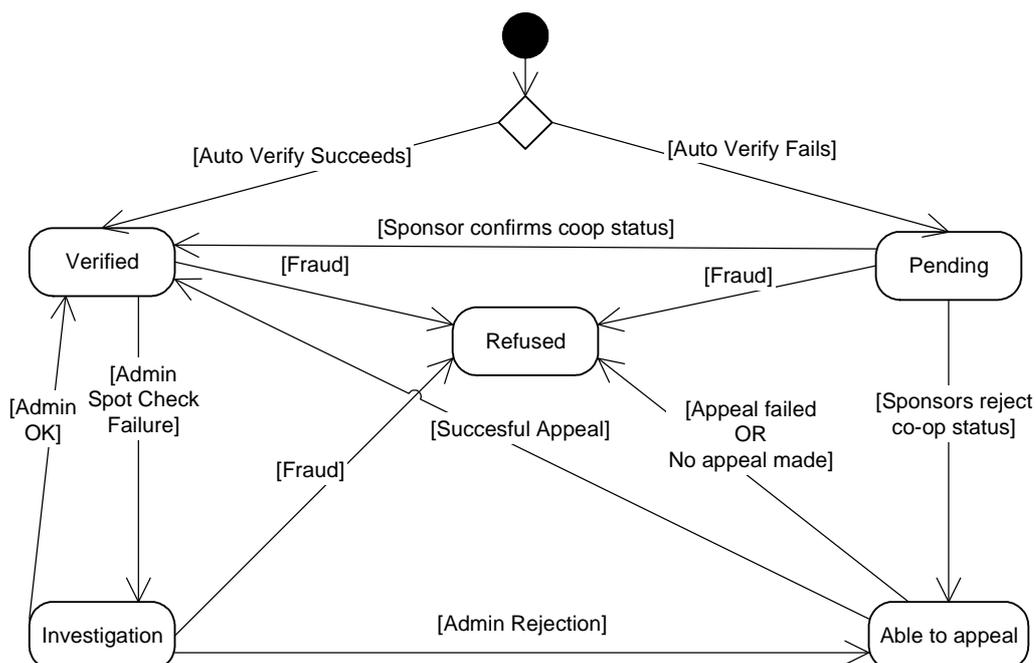
The .coop registry verifies the eligibility of registrants to register .coop domain names, as required by the dotCoop agreement with ICANN. To be successfully verified, a registrant must be a valid co-operative organisation as defined in the Charter. Note that the registry does *not* verify individual domain names. All domain names provisioned by the registry to a single registrant have exactly the same status with respect to verification and that status is determined solely by the verification status of the registrant.

## 1.3 Verification Process

The verification process for a contact starts the first time that the contact is used as the registrant for a domain. Merely creating a contact does *not* trigger verification because a contact is not necessarily a registrant. For example, the technical contact for a .coop domain may well be a hosting service employed by the co-operative, rather than the co-operative itself.

## 1.4 Verification States

A verification state is associated with each registrant. The diagram below shows the summary of the states and transitions between them.



There are two possible initial states – Pending or Verified. The registry system automatically selects a certain percentage of registrations for further verification. Those registrations are set to a Pending verification status. All other registrants are set to a Verified status. Registrants in a Pending status cannot activate their domain names until their eligibility is confirmed by the registry staff.

For registrants placed in the Pending status, a message is generated to the designated dotCoop admin staff that will be able to confirm the eligibility of the registrant either by contacting the cooperative reference contact, provided by the registrant, or by other means. If dotCoop confirms the eligibility of the registrant the registrant is changed to a “Verified” status.

If the dotCoop staff is not able to confirm the eligibility of the registrant, the registrant is placed in the “Able to Appeal” state. The registrant may appeal against this decision. If no appeal is forthcoming within 30 days [or as is defined by current procedures], the registrant’s application is denied and the registrant enters the “Refused” state. When this occurs, any related domain registrations are revoked and deleted from the registry. No refund of registration fees is provided for revoked domains. The mechanisms for communicating during the appeal process are outside the scope of this document. Refer to the dotCoop Verification Policy for these processes.

From time to time the dotCoop staff may perform spot checks on existing Verified registrants. If the registry is satisfied that the registrants selected for checking meet the eligibility criteria, no state changes occur. In cases where the registry has cause to suspect that a registrant does not conform to the registry’s eligibility criteria, the registrant is put into the “Under Investigation” status. After this step, the process proceeds as above with a registrant either being reset to a Verified status or proceeding through the Able to Appeal and Refused status using the same methods.

## **1.5 Active Domains**

The domain(s) registered by a registrant are only made available via DNS when the registrant is in the “Verified” or “Under Investigation” state.

## **1.6 Acceptance of New Domain and Host Provisioning Requests**

New provisioning requests and updates to existing provisioned entities are accepted from all registrants regardless of verification status, except for registrants in the “Refused” state, who are denied access.

## **1.7 Cooperative references and .Coop Verification Codes (CVCs)**

The .coop registry uses external organisations to help verify the co-operative status of a registrant. A cooperative reference is another co-operative organization, co-operative supporting organization or legal representative that can provide a confirmation of the registrant’s eligibility.

Registrants may choose one of the two following methods for their verification by dotCoop at the discretion of the registrar:

1. they can provide a cooperative reference contact upon registering their domain name or
2. they can apply to dotCoop for a .coop verification code (CVC) and then submit this code upon their domain name registration.<sup>1</sup>

In effect both above methods are different ways of achieving the same result whereby an additional contact is required for each registrant contact before the registrant can proceed with the registration of a .coop domain name. Ultimately, a cooperative reference is the same as a CVC, since the latter is mapped to a default cooperative reference contact.

The cooperative reference in the EPP is “regref” and the CVC is “regpvc.”

Although the timeframe for the registrant to appeal regarding their eligibility is set at 30 days, the registry often extends this period to ensure that all adequate contact methods are explored to contact the registrant. The system does not take any automatic steps for completion of the verification process for a registrant.

### ***1.8 Communication With Cooperative References During Verification***

Communications between the dotCoop staff and the cooperative reference(s) are not relevant to the registrars. The registrar is not involved in any of these communications.

### ***1.9 Communication With Registrar/Registrants During Verification***

The dotCoop registry directly informs the registrant of changes of verification status during the process, and may also request supplementary verification information from the registrant at certain stages of the verification process. The dotCoop registry may notify the registrar when a registrant is placed in a Pending or Under Investigation status. Registrars may choose to assist dotCoop in efforts to contact the registrant to provide information on how to update inaccurate information that may be preventing the completion of verification although there is no absolute requirement to do so. However, registrants whose registrations are revoked may submit chargeback requests for credit card charges.

---

<sup>1</sup> The detailed process for verification using the CVC method is documented in *.Coop Verification Code Upgrade* and the *dotCoop EPP Servers*.

## **2 EPP Extensions**

### **2.1 Relationship to other EPP Documents**

This specification assumes the existence of, and is dependent on, the following documents

- RFC 3730 - Extensible Provisioning Protocol (EPP)
- RFC 3731 - Extensible Provisioning Protocol (EPP) Domain Name Mapping
- RFC3733 - Extensible Provisioning Protocol (EPP) Contact Mapping

### **2.2 Background**

To support the .coop verification process a small number of extensions to the EPP protocol are required. These extensions allow contacts to possess the following additional properties:

- A list of cooperative references.
- A verification state.
- A language preference.
- A mailing list preference.

These properties warrant a short introduction:

Cooperative references are themselves regular contacts created via EPP. In “standard” EPP, contacts can only be associated with domains (i.e. as a registrant, billing contact, technical contact, etc). When using the .coop extension, contacts can also be associated directly with other contacts, i.e. one contact can act as a cooperative reference for another contact.

The verification state property is described above in section 1.4.

The language preference property allows registrars to specify a language preference on behalf of a contact. If possible, the registry will communicate with contacts in their chosen language.

Finally the mailing list preference allows registrars to pass on the ability for contacts to opt-in to .coop newsletters that the registry may send to registrants from time to time. This flag has no effect for contacts that don't become registrants.

### **2.3 Extensions Namespace**

All of the extension elements for verification will be placed in a namespace of 'http://www.nic.coop/contactCoopExt-1.0'.

In this document, examples will use the 'coop' XML namespace prefix for this namespace.

## 2.4 Creating a .coop registrant

In general, the sequence of EPP commands involved in creating a domain for a completely new registrant is as follows

- A `<contact:create>` command to create the contact that will become the cooperative reference for the registrant contact created in the next step.
- A `<contact:create>` command to create the contact that will become the registrant. The registrant `<contact:create>` command will specify the registrant's related cooperative reference contact / .coop verification code (CVC) in a `<coop:createType>` extension.
- Zero or more `<contact:create>` commands to create other contacts that will become technical, administration or billing contacts for the domain.
- A `<domain:create>` command to provision the domain. The `<domain:create>` command will specify the related registrant and any other contacts (i.e. technical, billing, etc).

Assuming a contact hasn't entered the verification process, cooperative reference information can also be added after its creation by using a `<coop:updateType>` extension with the `<contact:update>` command.

Verification is *not* triggered by a `<contact:create>` command, unless a CVC is entered, because it is not possible at this stage to distinguish contacts that may become registrants from those which are to be used for other purposes (sponsor, technical, admin, billing).

The `<domain:create>` command identifies a particular contact as a registrant through the `<domain:registrant>` element. The verification process starts when a contact is first used as a registrant.

## 2.5 Summary

Registrars *must* supply a `<domain:registrant>` element for `<domain:create>`.

Registrars *must* supply a `<contact:org>` element when creating or updating contacts that act as registrants.

There are server-imposed limitations on the updating of contact data in situations where such updates might compromise the verification process.

Registrars must supply at least one cooperative reference contact identifier or a CVC for contacts that are to become registrants. This can be done either in the `<contact:create>` command, or in a subsequent `<contact:update>` command.

The registry will send notification of state changes through responses to the EPP `<poll>` command.

Extensions to EPP can be added at various levels. The scope of the dotCoop extensions is small and localised and has been implemented using command-response extensions.

Ids and ROIDs for registrants are identical to those of contacts, because a registrant is a special case of a contact.

## 2.6 <greeting>

The <greeting> response from the server (received when an initial connection is made, and on receipt of a <hello> request), will contain the verification namespace.

Example greeting:

```
<?xml version="1.0" encoding="utf-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <greeting>
    <svID>dotCoop EPP Server</svID>
    <svDate>2004-11-09T18:11:00.6Z</svDate>
    <svcMenu>
      <version>1.0</version>
      <lang>en</lang>
      <objURI>urn:ietf:params:xml:ns:contact-1.0</objURI>
      <objURI>urn:ietf:params:xml:ns:host-1.0</objURI>
      <objURI>urn:ietf:params:xml:ns:domain-1.0</objURI>
      <svcExtension>
        <extURI>http://www.nic.coop/contactCoopExt-1.0</extURI>
      </svcExtension>
    </svcMenu>
    <dcP>
      <access>
        <all />
      </access>
      <statement>
        <purpose>
          <admin />
          <prov />
        </purpose>
        <recipient>
          <ours />
          <public />
        </recipient>
        <retention>
          <stated />
        </retention>
      </statement>
    </dcP>
  </greeting>
</epp>
```

## 2.7 Extensions to <contact:create> command

The .coop extensions to <contact:create> allow you to assign a cooperative reference, a language preference, and a mailing list preference to a contact. This is done by adding a <coop: createType> child to the <extension> element.

Cooperative references or a CVC can be added to a <coop:createType> by adding <coop:sponsor><sup>2</sup> values unless a CVC has already been successful applied to a contact. These values are contact ids for contacts that have previously been created for the purpose of acting as cooperative references.

---

<sup>2</sup> The value for this node is either the cooperative reference contact, the CVC, or the CId of a contact.

A language preference can be added to a `<coop:createType>` by adding a `<coop:langPref>` value.

Finally a mailing list preference flag can be added to a `<coop:createType>` by adding a `<coop:mailingListPref>` value.

Example command:

```
<?xml version="1.0" encoding="utf-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <create>
      <contact:create xmlns:contact="urn:ietf:params:xml:ns:contact-1.0">
        <contact:id>thldomainTest</contact:id>
        <contact:postalInfo type="loc">
          <contact:name>Kermit The Frog</contact:name>
          <contact:org>The Muppet Show</contact:org>
          <contact:addr>
            <contact:city>Chicago</contact:city>
            <contact:cc>US</contact:cc>
          </contact:addr>
        </contact:postalInfo>
        <contact:email>k.frog@example.tld</contact:email>
        <contact:authInfo>
          <contact:pw>Match Sticks</contact:pw>
        </contact:authInfo>
      </contact:create>
    </create>
    <extension>
      <coop:create xmlns:coop="http://www.nic.coop/contactCoopExt-1.0">
        <coop:sponsor>regref1</coop:sponsor>
      </coop:create>
    </extension>
  </command>
</epp>
```

## **2.8 Extensions to `<contact:update>` command**

The .coop extensions to `<contact:update>` allow registrars to update a contact's cooperative reference, language preference, and mailing list preference. This is done by adding a `<coop:updateType>` child to the extension element.

Please note that a contact's cooperative reference may only be updated up until the point that the contact enters the verification process (i.e. when they are first used as a registrant or a CVC has been applied). This is for business reasons. If the cooperative reference information is volatile during the verification process, the process of consultation with organisations acting as cooperative references could be seriously compromised. After the verification process has been completed, there is no business value in changing the cooperative reference information.

Cooperative references can be modified in a `<coop:updateType>` by adding `<coop:sponsor>` values to the `<coop: addRemType>` element.

A contact's language preference and mailing list preference can be updated in a `<coop:updateType>` by adding a `<coop: chgType>` value.

## 2.9 Extensions to <contact:info> response

The response of <contact:info> command can be used to view the verification status and the cooperative reference of a contact and to check the language and mailing list preferences.

Example response:

```
<?xml version="1.0" encoding="utf-16"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <resData>
      <contact:infData xmlns:contact="urn:ietf:params:xml:ns:contact-
1.0">
        <contact:id>thldomainTest</contact:id>
        <contact:roid>62273C-COOP</contact:roid>
        <contact:status s="ok">ok</contact:status>
        <contact:postalInfo type="loc">
          <contact:name>Kermit The Frog</contact:name>
          <contact:org>The Muppet Show</contact:org>
          <contact:addr>
            <contact:city>Chicago</contact:city>
            <contact:cc>US</contact:cc>
          </contact:addr>
        </contact:postalInfo>
        <contact:email>k.frog@example.tld</contact:email>
        <contact:clID>TestHarness1</contact:clID>
        <contact:crID>TestHarness1</contact:crID>
        <contact:crDate>2004-10-29T12:29:02.6Z</contact:crDate>
        <contact:authInfo>
          <contact:pw>Match Sticks</contact:pw>
        </contact:authInfo>
      </contact:infData>
    </resData>
    <extension>
      <coop:infData xmlns:coop="http://www.nic.coop/contactCoopExt-
1.0">
        <coop:state code="verified">Verified</coop:state>
        <coop:sponsor>regref1</coop:sponsor>      </coop:infData>
      </extension>
    <trID>
      <svTRID>00000000000000554898</svTRID>
    </trID>
  </response>
</epp>
```

The <coop:state> element describes the contact's verification state. If the contact is not in use as a registrant for a domain, the <coop:state> element will not be provided, because in such cases the contact has no meaningful verification status.

The <coop:langPref> element contains the contact's language preference.

The <coop:mailingListPref> element contains the contact's mailing list preference.

Cooperative reference contact IDs for contacts will be provided in <coop:sponsor> elements.

If the registrar making the query is not the current registrar for the contact, the <coop:infData> element will only be provided if a valid authInfo was submitted with the <contact:info> command.

## 2.10 Extensions to <domain:create> response

When creating the domain, the <domain:create> command follows the EPP specification with the following exceptions:

- The <domain:registrant> element is mandatory.
- The <contact:org> field for registrants is mandatory (this is enforced at the time of domain:create)

Note that a <domain:create> does not fail if the registrant is selected for verification by the automatic verification procedure. It will, however, fail if the registrant is in the Refused status. It will also fail if a <domain:registrant> is not provided or is invalid.

The response for the domain:create command will contain an <extension> element containing the current registrant's status.

```
<?xml version="1.0" encoding="utf-16"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <resData>
      <domain:creData xmlns:domain="urn:ietf:params:xml:ns:domain-
1.0">
        <domain:name>thldomainltest.coop</domain:name>
        <domain:crDate>2004-12-06T11:32:39.0Z</domain:crDate>
        <domain:exDate>2006-12-06T11:32:39.0Z</domain:exDate>
      </domain:creData>
    </resData>
    <extension>
      <coop:stateChange
xmlns:coop="http://www.nic.coop/contactCoopExt-1.0">
        <coop:id>thlcontactlTest</coop:id>
        <coop:state code="verified">verified</coop:state>
      </coop:stateChange>
    </extension>
    <trID>
      <svTRID>00000000000000575881</svTRID>
    </trID>
  </response>
</epp>
```

Note that the initial state of the registrant following the <domain:create> command is returned as part of the <domain:create> response. Further changes of state are sent via responses to the <poll> command.

## 2.11 Extensions to <domain:update> response

If a <domain:update> command changes the registrant of the domain the response will contain the same extension element as a <domain:create> command.

## 2.12 Extensions to <poll> response

Any verification action that results in a change of a registrant's verification state after it enters the verification process will cause a service message to be placed on the registry's outgoing message queue for the appropriate registrar. These messages can be retrieved using the <poll> command.

A sample <poll> response is shown here.

```
<?xml version="1.0" encoding="utf-8" ?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1301">
      <msg>Command completed successfully; ack to dequeue</msg>
    </result>
    <msgQ count="1" id="11082">
      <qDate>2004-12-21T13:46:06.8Z</qDate>
      <msg>Registrant verification state changed</msg>
    </msgQ>
    <extension>
      <coop:stateChange xmlns:coop="http://www.nic.coop/contactCoopExt-
1.0">
        <coop:id>th1Test2</coop:id>
        <coop:state code="verified">verified</coop:state>
      </coop:stateChange>
    </extension>
    <trID>
      <svTRID>000000000000000032212</svTRID>
    </trID>
  </response>
</epp>
```

The <coop: stateChangeType> element contains:

- <coop:id> element that describes the id of the contact.
- <coop:state> element that defines the current verification state.

## 2.13 <coop:state>

The <coop:state> element contains the verification state of a contact. Its value is one of the values of the <coop:stateCodeType> enumerated type.

The values of <coop:stateCodeType> are as follows:

<b>Value</b>	<b>Meaning</b>
verified	Registrant has been verified. Domains assigned to this registrant will appear in the DNS.
pendingVerification	Initial verification is pending for the registrant. Domains do not yet appear in the DNS.
ableToAppeal	Registrant has been rejected after investigation, but may appeal. Domains do not appear in the DNS.
underInvestigation	Registrant is under investigation from registry administrators. Domains continue to appear in the DNS.
refused	Registrant appeal has failed or timed out. No further provisioning requests will be accepted.

### 3 XML Schema for Extensions

File contactCoopExt-1.0.xsd listing follows:

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

  <schema targetNamespace="http://www.nic.coop/contactCoopExt-1.0"
    xmlns:contactCoopExt="http://www.nic.coop/contactCoopExt-
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" />

    <annotation>
      <documentation>
        Extensible Provisioning Protocol v1.0
        dotCoop extension to contact provisioning schema.
      </documentation>
    </annotation>

  <!--
  Child elements found in EPP command extensions.
  -->
    <element name="create" type="contactCoopExt:createType" />
    <element name="update" type="contactCoopExt:updateType" />

  <!--
  Child elements of the <createType> command extension.

  This extension is used with the <contact:create> command. It
  allows the addition of a cooperative reference to a contact so that
  the contact
  may enter the dotCoop verification process and be used as a
  registrant. It also allows a language preference to be specified
  for a contact which enables the dotCoop registry to attempt to
  communicate in an appropriate language with the registrant
  during the verification process.
  -->
    <complexType name="createType">
      <sequence>
        <element name="langPref" type="language" minOccurs="0" />
        <element name="mailingListPref" type="boolean" minOccurs="0"
/>
        <element name="sponsor" type="eppcom:clIDType" minOccurs="0"
maxOccurs="unbounded"/>
      </sequence>
    </complexType>

  <!--
  Child elements of the <updateType> command extension.

  This extension is used with the <contact:update> command to add and
  remove a cooperative reference, and change the language preference of
  a contact
```

after it has been created. Note that server policy may prevent cooperative reference modifications after a contact has entered the verification process.

```
-->
  <complexType name="updateType">
    <sequence>
      <element name="add" type="contactCoopExt:addRemType"
minOccurs="0" />
      <element name="rem" type="contactCoopExt:addRemType"
minOccurs="0" />
      <element name="chg" type="contactCoopExt:chgType"
minOccurs="0" />
    </sequence>
  </complexType>

<!--
Data elements that can be added or removed.
-->
  <complexType name="addRemType">
    <sequence>
      <element name="sponsor" type="eppcom:clIDType"
maxOccurs="unbounded"/>
    </sequence>
  </complexType>

<!--
Data elements that can be changed.
-->
  <complexType name="chgType">
    <sequence>
      <element name="langPref" type="language" minOccurs="0" />
      <element name="mailingListPref" type="boolean" minOccurs="0"
/>
    </sequence>
  </complexType>

<!--
Child response extension elements.
-->
  <element name="infData" type="contactCoopExt:infDataType" />
  <element name="stateChange" type="contactCoopExt:stateChangeType"
/>

<!--
Response utility types.
-->
  <complexType name="stateType">
    <simpleContent>
      <extension base="string">
        <attribute name="code" type="contactCoopExt:stateCodeType"
use="required" />
      </extension>
    </simpleContent>
  </complexType>

  <simpleType name="stateCodeType">
    <restriction base="token">
      <enumeration value="ableToAppeal" />
      <enumeration value="underInvestigation" />
      <enumeration value="refused" />
    </restriction>
  </simpleType>

```

```

        <enumeration value="pendingVerification" />
        <enumeration value="verified" />
    </restriction>
</simpleType>

<!--
<contact:info> response extension elements.
-->
    <complexType name="infDataType">
        <sequence>
            <element name="state" type="contactCoopExt:stateType"
minOccurs="0" />
            <element name="langPref" type="language" minOccurs="0" />
            <element name="mailingListPref" type="boolean" minOccurs="0"
/>
            <element name="sponsor" type="eppcom:clIDType" minOccurs="0"
maxOccurs="unbounded" />
        </sequence>
    </complexType>

<!--
<poll>, <domain:create>, and <domain:update> response extension
element.

This extension will only be present in a <poll> response if the
Service message being retrieved is a "registrant verification state
changed" message.

This extension will only be present in <domain:create> and
<domain:update> responses if the domain's registrant contact
(identified by the id element) has entered the dotCoop verification
process as a result of the command. This happens the first time a
contact is used as a registrant.
-->
    <complexType name="stateChangeType">
        <sequence>
            <element name="id" type="eppcom:clIDType" />
            <element name="state" type="contactCoopExt:stateType" />
        </sequence>
    </complexType>

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

```