

# EPP mapping

## Contents

1. Introduction .....	3
1.1 Custom extensions .....	3
1.2 Schema namespace .....	3
1.3 Scope of objects .....	3
2. Object Attributes .....	4
2.1 Domain name objects .....	4
2.1.1 Domain Names .....	4
2.1.2 Billing codes .....	4
2.1.3 Purchase Order Field .....	4
2.1.4 Notes fields .....	4
2.2 Accounts .....	4
2.2.1 Account Identifiers .....	4
2.2.2 Account name .....	4
2.2.3 Trading name .....	4
2.2.4 Account type .....	4
2.2.5 Company Number .....	5
2.2.6 Opt Out Flag .....	5
2.2.7 Address type .....	5
2.2.8 Country code identifiers .....	5
2.3 Contacts .....	5
2.3.1 Contact Identifiers .....	5
2.3.2 Contact name .....	5
2.3.3 Contact type .....	5
2.3.4 Email .....	5
2.4 Dates and times .....	5
2.5 Nameserver identifiers .....	6
3. EPP Command Mapping .....	6
3.1 EPP <check> Command .....	6
3.2 EPP <info> Command .....	6
3.2.1 Domain <info> command .....	6
3.2.1.1 Example domain name <info> Command: .....	6
3.2.1.2 Example successful domain name <info> response .....	6
3.2.1.3 <domain:infData> element .....	8
3.2.2 Account <info> command .....	9
3.2.2.1 Example account <info> Command: .....	9
3.2.2.2 Example successful account <info> response: .....	9
3.2.2.3 <account:infData> element .....	10
3.2.3 Contact <info> command .....	11
3.2.3.1 Example contact <info> command: .....	11
3.2.3.2 Example successful contact <info> response: .....	12
3.2.3.3 <contact:infData> element .....	12
3.2.4 Nameserver <info> command .....	12
3.2.4.1 Example successful ns <info> response: .....	13
3.2.4.2 <ns:infData> element .....	13
3.3 EPP <transfer> Query Command .....	14
3.3.1.1 Example EPP <transfer> Command .....	14
3.3.1.2 Example successful <transfer> response .....	14
3.4 EPP <create> Command .....	15
3.4.1 Domain <create> command .....	15
3.4.1.1 <domain:create> element .....	15
3.4.1.2 <account:create> element .....	15
3.4.1.3 <contact:create> element .....	16
3.4.1.4 <contact:refer> element .....	16

- 3.4.1.5 <ns:create> element ..... 16
- 3.4.1.6 Example <domain:create> commands ..... 16
- 3.4.1.7 Example <domain:create> responses ..... 18
- 3.4.1.8 <domain:creData> element ..... 19
- 3.4.1.9 <account:creData> element..... 19
- 3.4.1.10 <contact:creData> element..... 20
- 3.4.1.11 <ns:creData> element..... 20
- 3.5 EPP <delete> Command ..... 20
- 3.5.1 Domain <delete> command..... 20
- 3.5.1.1 Example domain <delete> command ..... 20
- 3.5.1.2 Example successful domain <delete> response ..... 21
- 3.6 EPP <renew> Command ..... 21
- 3.6.1 Domain name <renew> command..... 21
- 3.6.1.1 Example domain <renew> command ..... 21
- 3.6.1.2 Example successful domain <renew> response ..... 21
- 3.7 EPP <update> Command ..... 22
- 3.7.1 Domain <update> command ..... 22
- 3.7.1.1 <domain:update> element ..... 22
- 3.7.1.2 Example Domain <update> Command..... 22
- 3.7.1.3 Example successful <update> response ..... 23
- 3.7.2 Account <update> command..... 23
- 3.7.2.1 <account:update> element ..... 24
- 3.7.2.2 Example account <modify> command:..... 24
- 3.7.2.3 Example account successful modify response..... 25
- 3.7.3 Contact <update> Command..... 25
- 3.7.3.1 <contact:update> element ..... 25
- 3.7.3.2 Example contact <modify> command:..... 26
- 3.7.3.3 Example contact successful modify response:..... 26
- 3.7.4 Host <update> command..... 26
- 3.7.4.1 <ns:update> element ..... 26
- 3.7.4.2 Example nameserver <modify> command: ..... 27
- 3.7.4.3 Example contact successful modify response:..... 27
- 4. Error messages and codes ..... 27
- 4.1 Example error message ..... 27
- 4.2 <domain:failData> element ..... 28
- 4.3 <account:failData> element ..... 28
- 4.4 <contact:failData> element ..... 28
- 4.5 <ns:failData> element ..... 28
- 5. Messages..... 28
- 5.1 Accepted referrals ..... 29
- 5.1.1 Example accepted referrals message ..... 29
- 5.2 Rejected referrals..... 29
- 5.2.1 Example rejected referrals message ..... 30
- 5.3 Tag change notifications ..... 30
- 5.3.1 Example tag change notification message ..... 30
- 5.4 Account object change notifications ..... 32
- 5.4.1 Example account object change notification message ..... 32
- 5.5 Notification of change of account object for a domain name ..... 33
- 5.5.1 Example notification of change of account object for a domain name. .... 33
- 6. Security ..... 34
- 7. Divergences from EPP standard..... 34
- 8. References..... 34

## 1. Introduction

We are engaged in a twofold process of restructuring our data to redevelop the domain name registry in a more relational way and of implementing new technologies. The data restructuring is described in [DRD]. The new technology part includes the development of an EPP server to allow domain name, account, contact and nameserver objects to be modified. This document describes the EPP mapping for these objects. It is recommended that it is read side by side with [DRD].

### 1.1 Custom extensions

EPP was developed largely by those working for gTLD registries, who have a very different data model from us. As a result we have been unable to use the standard definitions for domains, contacts and hosts and have instead implemented our own versions.

Please be clear that all of the extensions we have made conform to the guidelines for extending EPP.

To summarise the differences:

- We have developed our own mapping for domains and contacts to reflect the data we hold.
- We have added a new object, call account, to represent our account structure.
- The RFC has a host object mapping. This does not fit with our model and so we are developing our own nameserver model, named differently to emphasise the differences.
- Nameservers are held as objects rather than as domain name attributes. Nameservers can be non-unique and so existing nameservers must be referred to by an identifier. Multiple tag-holders may hold nameservers with the same name.
- We have used the EPP transfer command to represent our release operation. This was considerably simpler to understand than using the EPP command extension mechanism.

### 1.2 Schema namespace

This document has the schema namespace to be at <http://www.nominet.org.uk/epp/xml>. This may change. For those who are unfamiliar with XML namespaces, do not expect to find any document at that URL.

### 1.3 Scope of objects

Our data model is different from those of gTLD registrars and attention should be paid to the scope (ownership/control) of objects:

- Accounts objects are within the scope of one tag. So if you have two tags, you cannot share an account object between those two tags. If you have a customer with domains on two of your tags then you need to create two accounts objects for that customer, one within the scope of each tag.
- Nameserver objects are within the scope of one tag. There is no restriction on the uniqueness of the host or IP address pointed to by a nameserver object. So within one tag you can have two nameserver objects that point to the same physical nameserver but neither of these can be accessed or used by any domain on another tag.
- Contact objects are within the scope of one account object. So you cannot share a contact object between accounts. When the account object is deleted the contact objects are also deleted.

## **2. Object Attributes**

### **2.1 Domain name objects**

#### **2.1.1 Domain Names**

The syntax for domain and host names described in this document **MUST** conform to [RFC952] as updated by [RFC1123].

Domain names are restricted to the .uk top level domain, and to the .co.uk, .org.uk, .ltd.uk, .plc.uk, .sch.uk, .net.uk and .me.uk second level domains.

#### **2.1.2 Billing codes**

First-bill and recur-bill billing codes must be either 'th' for tag-holder billing or 'bc' for billing-contact billing.

Auto-bill and next-Bill fields must be an integer between 0 and 182, determining the number of days before expiry that a domain name should be automatically renewed.

#### **2.1.3 Purchase Order Field**

This is a free-form text field of up to 200 characters.

#### **2.1.4 Notes fields**

There may be up to 20 notes fields each of up to 200 characters.

## **2.2 Accounts**

An account object contains information about the registrant of a .uk domain name. This information includes account level information, i.e. trading name, type, company number and administrative and billing contact information for the registrant.

### **2.2.1 Account Identifiers**

All accounts are identified by a unique identifier. A account identifier is a numeric identifier.

### **2.2.2 Account name**

The account name is an alphanumeric field and is the legal owner of any domain names associated with the registrant object.

### **2.2.3 Trading name**

The trading name is an alphanumeric field and contains the trading name for the account if any.

### **2.2.4 Account type**

This field denotes the type of organisation the account object refers to, it may be equal to any of the following values:

- LTD – a UK limited company
- PLC – a UK public limited company
- IND – a UK private individual
- FIND – a non-UK private individual

- PTNR – A UK Partnership
- STRA – A UK Sole Trader
- SCH – A UK school
- RCHAR – A UK Registered charity
- GOV – A UK Government body
- CRC – A UK Corporation by Royal Charter
- STAT – A UK Statutory Body
- FCORP – A non-UK Corporation
- LLP – A UK Limited Liability Partnership

### **2.2.5 Company Number**

This field gives the UK company number for the account if there is one

### **2.2.6 Opt Out Flag**

If the account is for a private individual, they can choose not to have their contact information displayed on the Whois. Setting this flag to 'Y' indicates this choice.

### **2.2.7 Address type**

Two addresses may be provided for a registrant account – of type 'admin' or 'billing',

### **2.2.8 Country code identifiers**

Country identifiers are represented using two character identifiers specified in [ISO 3166].

## **2.3 Contacts**

A contact object contains details about a person or role and has an email address, telephone and fax number. Addresses are not associated with contact objects but directly with accounts.

### **2.3.1 Contact Identifiers**

All contacts are identified by a unique identifier. A contact identifier is a numeric identifier preceded by 'C'.

### **2.3.2 Contact name**

The individual or organisational name for the contact is represented using a character string.

### **2.3.3 Contact type**

A contact may be of type "admin" or "billing". Other types may be added later.

### **2.3.4 Email**

Email address syntax is defined in [RFC2822].

## **2.4 Dates and times**

Date and time attribute values MUST be represented in Universal Coordinated Time (UTC) using the Gregorian calendar. The extended date-time form using upper case "T" and "Z" characters defined in [RFC3339] 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.

## 2.5 Nameserver identifiers

Nameservers are identified by a unique identifier – a numeric identifier prefixed by 'NS'.

## 3. EPP Command Mapping

### 3.1 EPP <check> Command

The EPP <check> command is used to determine if an object can be provisioned within a repository. We will not be implementing this command, tag holders are referred to the Domain Availability Checker instead.

### 3.2 EPP <info> Command

The EPP <info> command is used to retrieve information associated with an object. It is equivalent to the query automaton function. The <info> command will be created for all object types – domain names, accounts, contacts and nameservers. It will be possible to only query objects on your own tag. Domain <info> command

To obtain information associated with a domain object, the <info> command must contain a <domain:info> element that identifies the domain namespace and location of the domain schema. The <domain:info> element must contain the following child elements:

- A <domain:name> element that contains the fully qualified name of the domain object to be queried.

#### 3.2.1.1 Example domain name <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:       <domain:info
C:         xmlns:domain="http://www.nominet.org.uk/epp/xml/nom-domain-1.0"
C:         xsi:schemaLocation="http://www.nominet.org.uk/epp/xml/nom-domain-1.0
C:           nom-domain-1.0.xsd">
C:           <domain:name>example.co.uk</domain:name>
C:         </domain:info>
C:       </info>
C:     </command>
C:   </epp>
```

In this and other examples. C: denotes a message sent by the EPP client and S: a by the server.

#### 3.2.1.2 Example successful domain name <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:       <domain:infData
S:         xmlns:domain="http://www.nominet.org.uk/epp/xml/nom-domain-1.0"
S:         xsi:schemaLocation="http://www.nominet.org.uk/epp/xml/nom-domain-1.0
S:           nom-domain-1.0.xsd">
S:           <domain:name>example.co.uk</domain:name>
```

```

S: <domain:account>
S: <account:infData
S:   xmlns:account="http://www.nominet.org.uk/epp/xml/nom-account-1.0"
S:   xmlns:contact="http://www.nominet.org.uk/epp/xml/nom-contact-1.0">
S:   <account:roid>S123456</account:roid>
S:   <account:name>Mr R. Strant</account:name>
S:   <account:trade-name>R. S. Industries</account:trade-name>
S:   <account:type>STRA</account:type>
S:   <account:co-no>NI123456</account:co-no>
S:   <account:opt-out>N</account:opt-out>
S:   <account:addr type="admin">
S:     <account:street>2102 High Street</account:street>
S:     <account:locality>Carfax</account:locality>
S:     <account:city>Oxford</account:city>
S:     <account:county>Oxfordshire</account:county>
S:     <account:pc>OX1 1DF</account:pc>
S:     <account:cc>GB</account:cc>
S:   </account:addr>
S:   <account:contact type="admin" order="1">
S:     <contact:infData>
S:       <contact:roid>C12345</contact:roid>
S:       <contact:name>Mr R.Strant</contact:name>
S:       <contact:voice>01865 123456</contact:voice>
S:       <contact:fax>01865 123456</contact:fax>
S:       <contact:email>r.strant@strant.co.uk</contact:email>
S:       <contact:clID>TEST</contact:clID>
S:       <contact:crID>domains@isp.com</contact:crID>
S:       <contact:crDate>1999-04-03T22:00:00.0Z</contact:crDate>
S:       <contact:upID>domains@isp.com</contact:upID>
S:       <contact:upDate>1999-12-03T09:00:00.0Z</contact:upDate>
S:     </contact:infData>
S:   </account:contact>
S:   <account:contact type="admin" order="2">
S:     <contact:infData>
S:       <contact:roid>C23456</contact:roid>
S:       <contact:name>Ms S. Strant</contact:name>
S:       <contact:voice>01865 123457</contact:voice>
S:       <contact:fax>01865 123456</contact:fax>
S:       <contact:email>s.strant@strant.co.uk</contact:email>
S:       <contact:clID>TEST</contact:clID>
S:       <contact:crID>domains@isp.com</contact:crID>
S:       <contact:crDate>1999-04-03T22:00:00.0Z</contact:crDate>
S:       <contact:upID>domains@isp.com</contact:upID>
S:       <contact:upDate>1999-12-03T09:00:00.0Z</contact:upDate>
S:     </contact:infData>
S:   </account:contact>
S:   <account:contact type="billing" order="1">
S:     <contact:infData>
S:       <contact:roid>C12347</contact:roid>
S:       <contact:name>A. Ccountant</contact:name>
S:       <contact:voice>01865 657893</contact:voice>
S:       <contact:email>acc@billing.co.uk</contact:email>
S:       <contact:clID>TEST</contact:clID>
S:       <contact:crID>domains@isp.com</contact:crID>
S:       <contact:crDate>1999-04-03T22:00:00.0Z</contact:crDate>
S:       <contact:upID>domains@isp.com</contact:upID>
S:       <contact:upDate>1999-12-03T09:00:00.0Z</contact:upDate>
S:     </contact:infData>
S:   </account:contact>
S:   <account:clID>TEST</account:clID>
S:   <account:crID>TEST</account:crID>
S:   <account:crDate>1999-04-03T22:00:00.0Z</account:crDate>
S:   <account:upID>domains@isp.com</account:upID>
S:   <account:upDate>1999-12-03T09:00:00.0Z</account:upDate>
S: </account:infData>
S: </domain:account>
S: <domain:ns
S:   xmlns:ns="http://www.nominet.org.uk/epp/xml/nom-ns-1.0">
S:   <ns:infData>

```

```

S:      <ns:roid>NS12345</ns:roid>
S:      <ns:name>ns1.example.co.uk</ns:name>
S:      <ns:addr ip="v4">10.10.10.10</ns:addr>
S:      <ns:clID>TEST</ns:clID>
S:      <ns:crID>domains@isp.com</ns:crID>
S:      <ns:crDate>1999-04-03T22:00:00.0Z</ns:crDate>
S:      <ns:upID>domains@isp.com</ns:upID>
S:      <ns:upDate>1999-12-03T09:00:00.0Z</ns:upDate>
S:      </ns:infData>
S:      <ns:infData>
S:      <ns:roid>NS12346</ns:roid>
S:      <ns:name>ns1.example.com</ns:name>
S:      <ns:clID>TEST</ns:clID>
S:      <ns:crID>domains@isp.com</ns:crID>
S:      <ns:crDate>1999-04-03T22:00:00.0Z</ns:crDate>
S:      <ns:upID>domains@isp.com</ns:upID>
S:      <ns:upDate>1999-12-03T09:00:00.0Z</ns:upDate>
S:      </ns:infData>
S:      </domain:ns>
S:      <domain:clID>TEST</domain:clID>
S:      <domain:crID>TEST</domain:crID>
S:      <domain:crDate>1999-04-03T22:00:00.0Z</domain:crDate>
S:      <domain:upID>domains@isp.com</domain:upID>
S:      <domain:upDate>1999-12-03T09:00:00.0Z</domain:upDate>
S:      <domain:exDate>2007-12-03T09:00:00.0Z</domain:exDate>
S:      </domain:infData>
S:      </resData>
S:      <trID>
S:      <clTRID>ABC-12345</clTRID>
S:      <svTRID>54322-XYZ</svTRID>
S:      </trID>
S:      </response>
S:      </epp>

```

### 3.2.1.3 <domain:infData> element

The <domain:infData> element in the response above contains the following child elements:

- A <domain:name> element that contains the fully qualified name of the domain object.
- A <domain:account> element that contains an <account:infData> element defining the account object associated with the domain name. This is described below in section 3.2.2.3.
- An optional <domain:ns> element. This has 1 to 10 <ns:infData> describing the nameservers for the domain name. The <ns:infData> element is described below in section 3.2.4.2.
- An optional <domain:first-bill> element containing the bill-code for the registration invoice.
- An optional <domain:recur-bill> element containing the bill-code for renewal invoices.
- An optional <domain:auto-bill> or <domain:next-bill> element giving the number of days before expiry that a domain name should be automatically renewed. Only one of these two elements should be present
- An optional <domain:notes> element containing a free-form text field.
- A <domain:clID> containing the TAG of the domain name.
- A <domain:crID> containing the identifier (TAG, email address or identifier of Nominet staff member) of who or what created the domain name. This element is not present if this information is not known (e.g. for some pre-Nominet domain names).
- A <domain:crDate> containing the date and time of domain object creation. This element is not present for pre-Nominet domain names.
- A <domain:exDate> giving the date and time of expiry. This element is not present for .sch.uk domain names.
- A <domain:upID> giving the identifier of who (TAG, email address, contact identifier or identifier of Nominet staff member) or what (automatic process identifier) last updated the domain object fields (not the account fields). This element is not present if the domain has not been modified since registration.
- A <domain:upDate> giving the date and time of the most recent domain object modification. This element is not present if the domain has not been modified since registration.

### 3.2.2 Account <info> command

To obtain information associated with a account object, the <info> command must contain a <account:info> command that identifies the account namespace and location of the account schema. The <account:info> element must contain the following child elements:

- A <account:roid> element that contains the identifier of the account object to be queried.

#### 3.2.2.1 Example account <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:       <account:info
C:         xmlns:account="http://www.nominet.org.uk/epp/xml/nom-account-1.0"
C:         xsi:schemaLocation="http://www.nominet.org.uk/epp/xml/nom-account-1.0
C:           nom-account-1.0.xsd">
C:           <account:roid>S1001</account:roid>
C:         </account:info>
C:       </info>
C:     <clTRID>ABC-12345</clTRID>
C:   </command>
C: </epp>
```

#### 3.2.2.2 Example successful account <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:       <account:infData
S:         xmlns:account="http://www.nominet.org.uk/epp/xml/nom-account-1.0"
S:         xmlns:contact="http://www.nominet.org.uk/epp/xml/nom-contact-1.0"
S:         xsi:schemaLocation="http://www.nominet.org.uk/epp/xml/nom-account-1.0
S:           nom-account-1.0.xsd">
S:         <account:roid>123456</account:roid>
S:         <account:name>Mr R. Strant</account:name>
S:         <account:trade-name>R. S. Industries</account:trade-name>
S:         <account:type>STRA</account:type>
S:         <account:co-no>NI123456</account:co-no>
S:         <account:opt-out>N</account:opt-out>
S:         <account:addr>
S:           <account:street>2102 High Street</account:street>
S:           <account:locality>Carfax</account:locality>
S:           <account:city>Oxford</account:city>
S:           <account:county>Oxfordshire</account:county>
S:           <account:pc>OX1 1DF</account:pc>
S:           <account:cc>GB</account:cc>
S:         </account:addr>
S:         <account:contact type="admin" order="1">
S:           <contact:infData>
S:             <contact:roid>C12345</contact:roid>
S:             <contact:name>Mr R. Strant</contact:name>
S:             <contact:voice>01865 123456</contact:voice>
S:             <contact:fax>01865 123456</contact:fax>
S:             <contact:email>r.strant@strant.co.uk</contact:email>
S:             <contact:clID>TEST</contact:clID>
S:             <contact:crID>domains@isp.com</contact:crID>
```

```

S:      <contact:crDate>1999-04-03T22:00:00.0Z</contact:crDate>
S:      <contact:upID>domains@isp.com</contact:upID>
S:      <contact:upDate>1999-12-03T09:00:00.0Z</contact:upDate>
S:      </contact:infData>
S:      </account:contact>
S:      <account:contact type="admin" order="2">
S:      <contact:infData>
S:      <contact:roid>C23456</contact:roid>
S:      <contact:name>Ms S. Strant</contact:name>
S:      <contact:voice>01865 123457</contact:voice>
S:      <contact:fax>01865 123456</contact:fax>
S:      <contact:email>s.strant@strant.co.uk</contact:email>
S:      <contact:clID>TEST</contact:clID>
S:      <contact:crID>domains@isp.com</contact:crID>
S:      <contact:crDate>1999-04-03T22:00:00.0Z</contact:crDate>
S:      <contact:upID>domains@isp.com</contact:upID>
S:      <contact:upDate>1999-12-03T09:00:00.0Z</contact:upDate>
S:      </contact:infData>
S:      </account:contact>
S:      <account:contact type="billing" order="1">
S:      <contact:infData>
S:      <contact:roid>C12347</contact:roid>
S:      <contact:name>A. Ccountant</contact:name>
S:      <contact:voice>01865 657893</contact:voice>
S:      <contact:email>acc@billing.co.uk</contact:email>
S:      <contact:clID>TEST</contact:clID>
S:      <contact:crID>domains@isp.com</contact:crID>
S:      <contact:crDate>1999-04-03T22:00:00.0Z</contact:crDate>
S:      <contact:upID>domains@isp.com</contact:upID>
S:      <contact:upDate>1999-12-03T09:00:00.0Z</contact:upDate>
S:      </contact:infData>
S:      </account:contact>
S:      <account:clID>TEST</account:clID>
S:      <account:crID>TEST</account:crID>
S:      <account:crDate>1999-04-03T22:00:00.0Z</account:crDate>
S:      <account:upID>domains@isp.com</account:upID>
S:      <account:upDate>1999-12-03T09:00:00.0Z</account:upDate>
S:      </account:infData>
S:      </resData>
S:      <trID>
S:      <clTRID>ABC-12345</clTRID>
S:      <svTRID>54322-XYZ</svTRID>
S:      </trID>
S:      </response>
S:      </epp>

```

### 3.2.2.3 <account:infData> element

The <account:infData> in the response above contains the following child elements:

- A <account:roid> element that contains the identifier of the account object.
- A <account:name> element that contains the name of the account and therefore the domain name registrant for all domain names on this account.
- An optional <account:trade-name> element that contains the trading name of the account.
- An optional <account:type> element that gives the account type (IND, LTD etc)
- An optional <account:co-no> element that gives the account's company number
- A <account:opt-out> element that determines if the account holder has opted out of displaying its address on the whois.
- 1 or two <account:addr> elements giving the administrative and (optional) billing addresses for the account. The administrative address is mandatory. This element has the following sub-elements:
  - <account:street> giving the first line of the address.
  - <account:locality> giving the postal village or locality.
  - <account:city> giving the postal tow.
  - <account:county> giving the county or postal administrative area.

- <account:pc> giving the postal code or zip code.
- <account:cc> giving the two letter country code.

The <account:street> and <account:cc> sub-elements are mandatory. <account:pc> is mandatory for British country codes. Either <account:city> must be provided or both of <account:locality> and <account:county>.

- 1-6 <account:contact> elements giving the administrative and billing contacts for the account. This element has two attributes – the first is type which may be “admin” or “billing”. The second is order which may be “1”, “2” or “3” and identifies the contact specifically. The <account:contact> element has a single <contact:infData> element which is defined below in section 3.2.3.3.
- A <account:clID> element giving the sponsoring TAG for the account object.
- A <account:crID> element giving the identifier of who (TAG id, email address or identifier of Nominet staff member) or what (automatic process identifier) created the account object if this information is available (it may not be for data that precedes the data migration process).
- A <account:crDate> element giving the date the account was created.
- A <account:upID> element giving the tag or other identifier of who (TAG id, email address, contact identifier or identifier of Nominet staff member) or what (automatic process identifier) last updated the account object. This element is not present if the account has not been modified since it was created.
- A <account:upDate> giving the date at which the account object was last modified. This element is not present if the account has not been modified since it was created.

### 3.2.3 Contact <info> command

To obtain information associated with a contact object, the <info> command must contain a <contact:info> command that identifies the contact namespace and location of the contact schema. The <contact:info> element must contain the following child elements:

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

#### 3.2.3.1 Example contact <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="http://www.nominet.org.uk/epp/xml/nom-contact-1.0"
C:         xsi:schemaLocation="http://www.nominet.org.uk/epp/xml/nom-contact-1.0
C:           nom-contact-1.0.xsd">
C:           <contact:roid>C12345</contact:roid>
C:         </contact:info>
C:       </info>
C:     <clTRID>ABC-12345</clTRID>
C:   </command>
C: </epp>
```

#### 3.2.3.2 Example successful contact <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="http://www.nominet.org.uk/epp/xml/nom-contact-1.0"
S:      xsi:schemaLocation="http://www.nominet.org.uk/epp/xml/nom-contact-1.0
S:      nom-contact-1.0.xsd">
S:      <contact:roid>R12345</contact:roid>
S:      <contact:name>Mr Contact</contact:name>
S:      <contact:voice>01865 123456</contact:voice>
S:      <contact:fax>01865 123456</contact:fax>
S:      <contact:email>r.strant@strant.co.uk</contact:email>
S:      <contact:clID>TEST</contact:clID>
S:      <contact:crID>TEST</contact:crID>
S:      <contact:crDate>1999-04-03T22:00:00.0Z</contact:crDate>
S:      <contact:upID>TEST</contact:upID>
S:      <contact:upDate>1999-12-03T09:00:00.0Z</contact:upDate>
S:      </contact:infData>
S:      </resData>
S:      <trID>
S:      <clTRID>ABC-12345</clTRID>
S:      <svTRID>54322-XYZ</svTRID>
S:      </trID>
S:      </response>
S: </epp>

```

### 3.2.3.3 <contact:infData> element

The <contact:infData> in the response contains the following child elements:

- A <contact:roid> element that contains the identifier of the contact object.
- A <contact:name> element containing the name of the individual or role represented by the contact.
- An optional <contact:voice> element that contains the contact's voice telephone number.
- An optional <contact:fax> element containing the contact's fax telephone number.
- A <contact:email> element that contains the contact's email address.
- A <contact:clID> element giving the sponsoring tag for the contact object.
- A <contact:crID> element giving the identifier of who (tag, email address, contact identifier or identifier of Nominet staff member) or what (identifier of automatic process) created the contact if this information is available (may not be for data that preceded the data migration process)
- A <contact:crDate> element giving the date the contact was created.
- A <contact:upID> element giving the identifier of who (tag, email address, contact identifier or identifier of Nominet staff member) or what (identifier of automatic process) modified the contact object. This element is not present if the contact has not been modified since it was created.
- A <contact:upDate> giving the date at which the contact object was last modified. This element is not present if the contact has not been modified since it was created.

### 3.2.4 Nameserver <info> command

To obtain information associated with a host object, the <info> command must contact a <ns:info> command that identifies the account namespace and location of the account schema. The <ns:info> element must contain the following child elements:

- A <ns:roid> element that contains the identifier of the nameserver 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:      <ns:info
C:      xmlns:ns="http://www.nominet.org.uk/epp/xml/nom-ns-1.0"
C:      xsi:schemaLocation="http://www.nominet.org.uk/epp/xml/nom-ns-1.0
C:      nom-ns-1.0.xsd">
C:      <ns:roid>NS12345</ns:roid>
C:      </ns:info>
C:    </info>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C: </epp>

```

### 3.2.4.1 Example successful ns <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:       <ns:infData
S:         xmlns:ns="http://www.nominet.org.uk/epp/xml/nom-ns-1.0"
S:         xsi:schemaLocation="http://www.nominet.org.uk/epp/xml/nom-ns-1.0
S:         nom-ns-1.0.xsd">
S:           <ns:roid>NS12346</ns:roid>
S:           <ns:name>ns1.example.co.uk</ns:name>
S:           <ns:addr ip="v4">10.10.10.10</ns:addr>
S:           <ns:clID>TEST</ns:clID>
S:           <ns:crID>TEST</ns:crID>
S:           <ns:crDate>1999-04-03T22:00:00.0Z</ns:crDate>
S:           <ns:upID>TEST</ns:upID>
S:           <ns:upDate>1999-12-03T09:00:00.0Z</ns:upDate>
S:         </ns:infData>
S:       </resData>
S:       <trID>
S:         <clTRID>ABC-12345</clTRID>
S:         <svTRID>54322-XYZ</svTRID>
S:       </trID>
S:     </response>
S: </epp>

```

### 3.2.4.2 <ns:infData> element

The <ns:infData> contains the following child elements

- A <ns:roid> giving the identifier of the host object.
- A <ns:name> giving the name of the nameserver.
- One optional <ns:addr> element with an ip attribute of "v4" giving the IPV4 address of the nameserver.
- One optional <ns:addr> element with an ip attribute of "v6" giving the IPV6 address of the nameserver.
- A <ns:clID> element giving the sponsoring tag of the nameserver object.
- A <ns:crID> element giving the TAG or other identifier of who created the nameserver if this information is available.
- A <ns:crDate> element giving the date the nameserver was created.
- A <ns:upID> element giving the TAG or other identifier of who or what last modified the nameserver object.
- A <ns:upDate> giving the date the nameserver was last modified.

### 3.3 EPP <transfer> Command

We shall be implementing the <transfer> command for domain names only. The effect of a <transfer> command on a domain name is as the automaton release operation and is slightly different from the standard. A domain name <transfer> may only be requested by currently sponsoring tag and the new tag must be included in the request. Initially we will only be implementing the "request" value for the op attribute as there is no handshake process.

The <transfer> command will contain a <domain:transfer> element that will contain the following sub-elements:

- A <domain:name> element containing the domain name to be transferred.
- A <domain:ips-key> element containing the tag to transfer the domain name to.
- An option <domain: account> element containing a <domain:account-id> with the account to move

#### 3.3.1.1 Example EPP <transfer> 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:       <domain:transfer
C:         xmlns:domain="http://www.nominet.org.uk/epp/xml/nom-domain-1.0"
C:         xsi:schemaLocation="http://www.nominet.org.uk/epp/xml/nom-domain-1.0
C:           nom-domain-1.0.xsd">
C:           <domain:name>example.co.uk</domain:name>
C:           <domain:tag>TEST</domain:tag>
C:           <domain:account>
C:             <domain:account-id>
C:             </domain:account-id>
C:           </domain:account>
C:         </domain:transfer>
C:       </transfer>
C:     <clTRID>ABCDEF-12345</clTRID>
C:   </command>
C: </epp>
```

#### 3.3.1.2 Example successful <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:     <trID>
S:       <clTRID>ABC-12345</clTRID>
S:       <svTRID>123456</svTRID>
S:     </trID>
S:   </response>
S: </epp>
```

### 3.4 EPP <create> Command

We shall be implementing the <create> command for domain names only. Other objects can be created only as part of the domain name creation operation. Contacts can also be created as part of the account update operation.

### 3.4.1 Domain <create> command

The EPP <create> allows a client to register a domain name. In addition to the standard EPP command elements, the <create> command MUST contain a <domain:create> element that identifies the domain namespace and the location of the domain schema.

#### 3.4.1.1 <domain:create> element

The <domain:create> element will contain the following child elements:

- A <domain:name> element that contains the fully qualified name of the domain object to be created.
- An optional <domain:period> element that specifies the registration period in years. This may currently only be set to 2.
- A <domain:account> element. This contains either a <domain:account-id> element containing the identifier of the account object to be used for the domain name or an <account:create> object defining the sub-elements of a account object to be created and used. The <account:create> element is defined below in section 3.4.1.2.
- An optional <domain:ns> element containing the nameservers in order for the domain name (if there are any). This has one to ten elements of either of the following types:
  - <ns:create> containing the sub-elements of a nameserver object to be created and used. The <ns:create> element is described below in section 3.4.1.4.
  - <domain:nsObj> containing the identifier of a nameserver object to be used.
- An optional <domain:first-bill> element containing the bill-code for the registration invoice.
- An optional <domain:recur-bill> element containing the bill-code for renewal invoices.
- An optional <domain:auto-bill> element giving the number of days before expiry that a domain name should be automatically renewed.
- An optional <domain:next-bill> element giving the number of days before expiry that a domain name should be automatically renewed. This should not be present if a <domain:auto-bill> element is present and vice-versa. The <domain:next-bill> element is removed after renewal.
- An optional <domain:notes> element containing a free-form text field.

#### 3.4.1.2 <account:create> element

The <account:create> element will have the following sub-elements:

- A <account:name> element giving the name of the account.
- An optional <account:trade-name> element giving the trading name of the account.
- A <account:type> element giving the type of the account. Possible types are listed in section 2.2.4.
- A <account:opt-out> element denoting if the account is opting out of displaying its address on the whois. This may be 'Y' or 'N' and defaults to 'N'.
- One to six <account:contact> elements giving admin and billing contacts to be added to the account. At least one admin contact must be provided. The <account:contact> element has attributes type (of value "admin" or "billing") and order ("1", "2" or "3") and has a single element which may be one of the following types:
  - <contact:create> defining the sub-elements of the contact to be created. The <contact:create> element is described below in section 3.4.1.3.
  - <contact:refer> defining a contact which has been defined earlier in the request. The <contact:refer> element is described below in section 3.4.1.4 and may only contain a <contact:reference> field here.
- One or two <account:addr> elements giving the admin and billing addresses for the account. The <account:addr> has a type attribute which may be of value "admin" or "billing". The admin address is mandatory, the billing address optional. The <account:addr> element has the following sub-elements defining the parts of the address:
  - <account:street>
  - <account:locality>

- o <account:city>
- o <account:county>
- o <account:pc>
- o <account:cc>

#### 3.4.1.3 <contact:create> element

The <contact:create> element will have the following sub-elements:

- An optional clReference giving a client's reference for the contact. This allows the contact to be referred to at other points in the containing request and allow (for example) both the main admin and billing contact to point to the same object.
- A <contact:name> element giving the name of the contact
- An optional <contact:phone> element giving the phone number for the contact.
- An optional <contact:fax> element giving the fax number for the contact.
- An optional <contact:email> element giving the email address for the contact.

#### 3.4.1.4 <contact:refer> element

The <contact:refer> element allows an existing or previously defined contact to be referred to. It contains one of the following two elements:

- A <contact:reference> element defining a contact previously defined in the request. This element can be set to one of a1, a2, a3, b1, b2 or b3
- A <contact:roid> element defining an existing contact.

#### 3.4.1.5 <ns:create> element

The <ns:create> element will have the following sub-elements:

- A <ns:name> element giving the name of the nameserver.
- An optional <ns:addr> element with an ip attribute of "v4" giving the ipv4 address for the nameserver.
- An optional <ns:addr> element with an ip attribute of "v6" giving the ipv6 address for the nameserver.

#### 3.4.1.6 Example <domain:create> commands

Example create command where the account id is not known and a new account object is created. Two existing nameservers are references and a new one is created and then referenced:

```
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:       <domain:create
C:         xmlns:domain="http://www.nominet.org.uk/epp/xml/nom-domain-1.0"
C:         xsi:schemaLocation="http://www.nominet.org.uk/epp/xml/nom-domain-1.0
C:           nom-domain-1.0.xsd">
C:           <domain:name>example.co.uk</domain:name>
C:           <domain:period>2</domain:period>
C:           <domain:account>
C:             <account:create
C:               xmlns:account="http://www.nominet.org.uk/epp/xml/nom-account-1.0"
C:               xmlns:contact="http://www.nominet.org.uk/epp/xml/nom-contact-1.0">
C:               <account:name>Mr R. Strant</account:name>
C:               <account:trade-name>R. S. Industries</account:trade-name>
C:               <account:type>LTD</account:type>
```

```

C: <account:co-no>NI123456</account:co-no>
C: <account:opt-out>N</account:opt-out>
C: <account:addr type="admin">
C: <account:street>2102 High Street</account:street>
C: <account:locality>Carfax</account:locality>
C: <account:city>Oxford</account:city>
C: <account:county>Oxfordshire</account:county>
C: <account:pc>OX1 1DF</account:pc>
C: <account:cc>GB</account:cc>
C: </account:addr>
C: <account:contact type="admin" order="1">
C: <contact:create>
C: <contact:name>Mr R. Strant</contact:name>
C: <contact:voice>01865 123456</contact:voice>
C: <contact:fax>01865 123456</contact:fax>
C: <contact:email>r.strant@strant.co.uk</contact:email>
C: </contact:create>
C: </account:contact>
C: <account:contact type="admin" order="2">
C: <contact:create>
C: <contact:name>Ms S. Strant</contact:name>
C: <contact:voice>01865 123457</contact:voice>
C: <contact:fax>01865 123456</contact:fax>
C: <contact:email>s.strant@strant.co.uk</contact:email>
C: </contact:create>
C: </account:contact>
C: <account:contact type="billing" order="1">
C: <contact:create>
C: <contact:name>A. Ccountant</contact:name>
C: <contact:voice>01865 657893</contact:voice>
C: <contact:email>acc@billing.co.uk</contact:email>
C: </contact:create>
C: </account:contact>
C: </account:create>
C: </domain:account>
C: <domain:ns
C: xmlns:ns="http://www.nominet.org.uk/epp/xml/nom-ns-1.0">
C: <ns:create>
C: <ns:name>ns0.example.co.uk</ns:name>
C: <ns:addr ip="v4">10.10.10.10</ns:addr>
C: </ns:create>
C: <domain:nsObj>NS10001</domain:nsObj>
C: <domain:nsObj>NS10002</domain:nsObj>
C: <ns:create>
C: <ns:name>ns3.example.net</ns:name>
C: </ns:create>
C: </domain:ns>
C: <domain:recur-bill>bc</domain:recur-bill>
C: </domain:create>
C: </create>
C: <clTRID>ABC-12345</clTRID>
C: </command>
C: </epp>

```

Example create command where the account identifier is known:

```

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: <domain:create
C: xmlns:domain="http://www.nominet.org.uk/epp/xml/nom-domain-1.0"
C: xsi:schemaLocation="http://www.nominet.org.uk/epp/xml/nom-domain-1.0
C: nom-domain-1.0.xsd">
C: <domain:name>example1.co.uk</domain:name>
C: <domain:account>

```

```

C:      <domain:account-id>1000</domain:account-id>
C:      </domain:account>
C:      <domain:ns>
C:        <domain:nsObj>NS1001</domain:nsObj>
C:        <domain:nsObj>NS1002</domain:nsObj>
C:      </domain:ns>
C:      <domain:recur-bill>bc</domain:recur-bill>
C:    </domain:create>
C:  </create>
C:  <clTRID>ABC-12345</clTRID>
C: </command>
C: </epp>

```

### 3.4.1.7 Example <domain:create> responses

Example successful create response where a account and nameserver object were created as part of the operation:

```

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:      <domain:creData
S:        xmlns:domain="http://www.nominet.org.uk/epp/xml/nom-domain-1.0"
S:        xmlns:account="http://www.nominet.org.uk/epp/xml/nom-account-1.0"
S:        xmlns:contact="http://www.nominet.org.uk/epp/xml/nom-contact-1.0"
S:        xsi:schemaLocation="http://www.nominet.org.uk/epp/xml/nom-domain-1.0
S:          nom-domain-1.0.xsd">
S:        <domain:name>example.co.uk</domain:name>
S:        <domain:account>
S:          <account:creData>
S:            <account:roid>100029</account:roid>
S:            <account:name>Mr R. Strant</account:name>
S:            <account:contact type="admin" order="1">
S:              <contact:creData>
S:                <contact:roid>C100081</contact:roid>
S:                <contact:name>Mr R. Strant</contact:name>
S:              </contact:creData>
S:            </account:contact>
S:            <account:contact type="billing" order="1">
S:              <contact:creData>
S:                <contact:roid>C100082</contact:roid>
S:                <contact:name>A. Ccountant</contact:name>
S:              </contact:creData>
S:            </account:contact>
S:            <account:contact type="admin" order="2">
S:              <contact:creData>
S:                <contact:roid>C100083</contact:roid>
S:                <contact:name>Ms S. Strant</contact:name>
S:              </contact:creData>
S:            </account:contact>
S:          </account:creData>
S:        </domain:account>
S:        <domain:crDate>2005-10-14T13:40:50</domain:crDate>
S:        <domain:exDate>2007-10-14T13:40:50</domain:exDate>
S:      </domain:creData>
S:    </resData>
S:    <trID>
S:      <clTRID>ABC-12345</clTRID>
S:      <svTRID>102497</svTRID>
S:    </trID>
S:  </response>

```

```
S:</epp>
```

The identifiers of all objects created as part of the operation are returned in the response.

Example successful create response where no objects (other than the domain name) were created as part of the operation:

```
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:       <domain:creData
S:         xmlns:domain="http://www.nominet.org.uk/epp/xml/nom-domain-1.0"
S:         xsi:schemaLocation="http://www.nominet.org.uk/epp/xml/nom-domain-1.0
S:           nom-domain-1.0.xsd">
S:         <domain:name>example1.co.uk</domain:name>
S:         <domain:crDate>2005-10-14T13:30:48</domain:crDate>
S:         <domain:exDate>2007-10-14T13:30:48</domain:exDate>
S:       </domain:creData>
S:     </resData>
S:     <trID>
S:       <clTRID>ABC-12345</clTRID>
S:       <svTRID>102493</svTRID>
S:     </trID>
S:   </response>
S:</epp>
```

#### 3.4.1.8 <domain:creData> element

The <domain:creData> element describes a domain name that has been created and contains the following sub-elements

- A <domain:name> element giving the domain name that was created.
- If an account object was created for the registrant, a <domain:account> field containing an <account:creData> sub-element giving details of the created account object. The <account:creData> element is described below in section 3.4.1.9.
- If any nameserver objects were created, a <domain:ns> element. This contains <ns:creData> sub-elements describing the nameserver objects that were created. The <ns:creData> element is described below in section 3.4.1.11.
- A <domain:crDate> element giving the creation date of the domain name.
- A <domain:exDate> element giving the expiry date of the domain name

#### 3.4.1.9 <account:creData> element

The <account:creData> element describes an account object that has been created and contains the following sub-elements.

- An <account:roid> element giving the repository object identifier for the domain name.
- An <account:name> element giving the name of the account holder (registrant of the domain name)
- If any contact objects were created, an <account:contact> element with type and main attributes for each one. This contains a single <contact:creData> element giving the details of the created contact object. This sub-element is described below in section 3.4.1.10.

### 3.4.1.10 <contact:creData> element

The <contact:creData> element describes a contact object that has been created and contains the following sub-elements.

- A <contact:roid> element giving the repository object identifier for the contact object.
- A <contact:name> element giving the name of the contact object.

### 3.4.1.11 <ns:creData> element

The <ns:creData> element describes a contact object that has been created and contains the following sub-elements.

- A <ns:roid> element giving the repository object identifier for the nameserver object.
- A <ns:name> element giving the name of the nameserver.

## 3.5 EPP <delete> Command

The EPP <delete> command will be implemented only for domain names. Contacts, accounts and ns objects cannot be removed directly from the system.

### 3.5.1 Domain <delete> command

This command allows a client to cancel a domain name. Such requests will only be accepted under the following circumstances:

- The requesting client is the current tag-holder for the domain name
- The domain name has not been retagged since registration.
- An invoice has not been issued for the domain name

The <domain:delete> element contains the following child elements:

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

#### 3.5.1.1 Example domain <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:       <domain:delete
C:         xmlns:domain="http://www.nominet.org.uk/epp/xml/nom-domain-1.0"
C:         xsi:schemaLocation="http://www.nominet.org.uk/epp/xml/nom-domain-1.0
C:           nom-domain-1.0.xsd">
C:           <domain:name>example.co.uk</domain:name>
C:         </domain:delete>
C:       </delete>
C:     </command>
C:   </epp>
```

#### 3.5.1.2 Example successful domain <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>123456</svTRID>
S: </trID>
S: </response>
S: </epp>

```

## 3.6 EPP <renew> Command

The EPP <renew> command will be implemented only for domain names. It has no meaning for other object types.

### 3.6.1 Domain name <renew> command.

The EPP <renew> command allows a client to renew a domain name and extend its validity period by two years. <renew> commands will only be processed if the expiry date of the domain name is within 6 months. The <domain:renew> element contains the following child elements:

- A <domain:name> element that contains the fully qualified name of the domain object whose validity period is to be extended.
- An optional <domain:period> element that specifies the registration period in years (y) or months (m). This may currently only be set to 2y or 24m.

#### 3.6.1.1 Example domain <renew> command

```

C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C: <epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
C:   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
C:   xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0
C:     epp-1.0.xsd">
C:   <command>
C:     <renew>
C:       <domain:renew
C:         xmlns:domain="http://www.nominet.org.uk/epp/xml/nom-domain-1.0"
C:         xsi:schemaLocation="http://www.nominet.org.uk/epp/xml/nom-domain-1.0
C:           nom-domain-1.0.xsd">
C:           <domain:name>example.co.uk</domain:name>
C:           <domain:period unit="y">2</domain:period>
C:         </domain:renew>
C:       </renew>
C:     <clTRID>ABC-12345</clTRID>
C:   </command>
C: </epp>

```

#### 3.6.1.2 Example successful domain <renew> response

```

S:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
S: <epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
S:   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
S:   xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0
S:     epp-1.0.xsd">
S:   <response>
S:     <result code="1000">
S:       <msg>Command completed successfully</msg>
S:     </result>
S:     <resData>
S:       <domain:renData
S:         xmlns:domain="http://www.nominet.org.uk/epp/xml/nom-domain-1.0"
S:         xsi:schemaLocation="http://www.nominet.org.uk/epp/xml/nom-domain-1.0
S:           nom-domain-1.0.xsd">
S:         <domain:name>example.co.uk</domain:name>
S:         <domain:exDate>2007-04-03T22:00:00.0Z</domain:exDate>

```

```

S:      </domain:renData>
S:      </resData>
S:      <trID>
S:      <clTRID>ABC-12345</clTRID>
S:      <svTRID>265768</svTRID>
S:      </trID>
S:      </response>
S:      </epp>

```

## 3.7 EPP <update> Command

The EPP <update> command allows a client to modify the attributes of a domain, account, host or contact object.

### 3.7.1 Domain <update> command

In order to modify the attributes of a domain name, the <update> command must contain a <domain:update> element that identifies the domain namespace and location of the domain schema.

#### 3.7.1.1 <domain:update> element

The <domain:update> element contains the following child elements:

- A <domain:name> element that contains the fully qualified name of the domain object to be updated.
- An optional <domain:account> element containing an <account:update> element with account attributes to be modified. This is described below in section 3.7.2. Note that modifying account attributes will affect any other domain names that are associated with that account.
- An optional <domain:ns> element which contains the nameservers in order for the domain name after the modification. This contains zero to ten elements which may be either of the following:
  - <ns:create> containing the sub-elements of a nameserver object to be created and used. This is described in section 3.4.1.4.
  - <domain:nsObj> containing the identifier of an existing nameserver object to be used. If an empty <domain:ns> element is given then all nameservers will be removed from the domain name. The individual elements (name, addr) of a nameserver object cannot be modified via a domain <update> command.
- An optional <domain:first-bill> element containing the bill-code for the registration invoice. If a blank element is provided then this field is removed (returned to the default of 'th').
- An optional <domain:recur-bill> element containing the bill-code for renewal invoices. If a blank element is provided then this field is removed (returned to the default of 'th').
- An optional <domain:auto-bill> or <domain:next-bill> element giving the number of days before expiry that a domain name should be automatically renewed. Only one of these two elements should be present. If a blank element is provided for either field then the field is removed.
- An optional <domain:purchase-order> element containing a free-form text field. If this is blank then the field is removed.
- An optional <domain:notes> element containing a free-form text field. If this is blank then the field is removed.

#### 3.7.1.2 Example Domain <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:       <domain:update
C:         xmlns:domain="http://www.nominet.org.uk/epp/xml/nom-domain-1.0"
C:         xsi:schemaLocation="http://www.nominet.org.uk/epp/xml/nom-domain-1.0

```

```

C:      nom-domain-1.0.xsd">
C:      <domain:name>example.co.uk</domain:name>
C:      <domain:account>
C:      <account:update
C:      xmlns:account="http://www.nominet.org.uk/epp/xml/nom-account-1.0"
C:      xmlns:contact="http://www.nominet.org.uk/epp/xml/nom-contact-1.0">
C:      <account:contact type="admin" order="1">
C:      <contact:update>
C:      <contact:email>admin@strant.co.uk</contact:email>
C:      </contact:update>
C:      </account:contact>
C:      </account:update>
C:      </domain:account>
C:      <domain:ns
C:      xmlns:ns="http://www.nominet.org.uk/epp/xml/nom-ns-1.0">
C:      <ns:create>
C:      <ns:name>ns2.example1.co.uk</ns:name>
C:      </ns:create>
C:      <domain:nsObj>NS1001</domain:nsObj>
C:      </domain:ns>
C:      <domain:auto-bill/>
C:      <domain:next-bill>5</domain:next-bill>
C:      </domain:update>
C:      </update>
C:      <clTRID>ABC-12345</clTRID>
C:      </command>
C:      </epp>

```

### 3.7.1.3 Example successful <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>123456</svTRID>
S:     </trID>
S:   </response>
S: </epp>

```

## 3.7.2 Account <update> command

The EPP <update> command allows a client to modify the attributes of a account object. A account object may be modified in one of the following ways:

- Modifying one the account fields (trade-name, type, co-no, opt-out). This includes modifying a fields that was previously empty and deleting a field.
- Modifying an existing address. When this is done all address fields that are to be modified must be provided including those that are to be deleted.
- Removing the billing address.
- Adding a billing address where one did not exist.
- Modifying one of the main contacts.
- Modifying one of the non-main contacts. In order to do this, the contact identifier must be provided.
- Removing a contact (not the main admin contact)
- Adding a contact (not the main admin contact)

The <update> command must contain a <account:update> element that identifies the account namespace and location of the account schema.

### 3.7.2.1 <account:update> element

The <account:update> element contains the following child elements:

- A <account:roid> element to contain the identifier of the account object to be modified. (Note this may be omitted when the <account:update> element sits inside a <domain:update> element).
- An optional <account:trade-name> element giving the trading name of the account. If a blank element is provided then the trade-name is removed.
- An optional <account:type> element giving the type of account.
- An optional <account:co-no> element giving the company number or other identifying number for the account.
- An optional <account:opt-out> element denoting if the account is opting out of displaying its address on the whois.
- Up to six <account:contact> elements. This element has two attributes – type which may be “admin” or “billing” and order which may be “1”, “2” or “3” The <account:contact> contains one of the following sub-elements.
  - <contact:delete>. The contact is to be removed from the account. This has a single <contact:roid> sub-element which defines the identifier of the contact to be deleted.
  - <contact:create>. The contact given here is added to the account. The <contact:create> element is described in section 3.4.1.3.
  - <contact:update>. The contact fields given by the sub-elements are modified. The <contact:update> element is described below in section 3.7.3.1.
  - <contact:refer>. An existing contact is referred to by the account. The <contact:refer> element is described in section 3.4.1.4 . The contact referred to must be a contact already on the account and allows for (e.g) an existing billing contact to be used as an admin contact.
- Up to two <account:addr> elements with a type attribute of “admin” or “billing”. The billing address is removed if a blank contact is provided. If a billing address is provided and one does not already exist then a new address is added. Otherwise fields are modified. The <account:addr> element will have the following sub-elements
  - <account:street> giving the first line of the address.
  - <account:locality> giving the postal village or locality.
  - <account:city> giving the postal tow.
  - <account:county> giving the county or postal administrative area.
  - <account:pc> giving the postal code or zip code.
  - <account:cc> giving the two letter country code.

The <account:street> and <account:cc> sub-elements are mandatory. <account:pc> is mandatory for British country codes. Either <account:city> must be provided or both of <account:locality> and <account:county>.

### 3.7.2.2 Example account <modify> command:

This modifies the trade-name, type and co-no of a account object, and adds an admin contact. A billing contact is removed from the account object and the billing address is also removed. The first admin contact is modified.

```
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:       <account:update
C:         xmlns:account="http://www.nominet.org.uk/epp/xml/nom-account-1.0"
```

```

C:   xmlns:contact="http://www.nominet.org.uk/epp/xml/nom-contact-1.0"
C:   xsi:schemaLocation="http://www.nominet.org.uk/epp/xml/nom-account-1.0
C:     nom-account-1.0.xsd">
C:     <account:roid>286467</account:roid>
C:     <account:trade-name>R. S. Industries</account:trade-name>
C:     <account:type>STRA</account:type>
C:     <account:co-no>NI123456</account:co-no>
C:     <account:addr type="billing"/>
C:     <account:contact type="admin">
C:       <contact:create>
C:         <contact:name>Ms S. Strant</contact:name>
C:         <contact:voice>01865 123457</contact:voice>
C:         <contact:fax>01865 123456</contact:fax>
C:         <contact:email>s.strant@strant.co.uk</contact:email>
C:       </contact:create>
C:     </account:contact>
C:     <account:contact type="billing">
C:       <contact:delete>
C:         <contact:roid>C123543</contact:roid>
C:       </contact:delete>
C:     </account:contact>
C:     <account:contact type="admin">
C:       <contact:update>
C:         <contact:voice>01865 232564</contact:voice>
C:       </contact:update>
C:     </account:contact>
C:   </account:update>
C: </update>
C: <clTRID>ABC-12345</clTRID>
C: </command>
C: </epp>

```

### 3.7.2.3 Example account successful modify 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>65789465</svTRID>
S:     </trID>
S:   </response>
S: </epp>

```

## 3.7.3 Contact <update> Command

In order to modify the attributes of a contact object, the <update> command must contain a <contact:update> element that identifies the contact namespace and location of the contact schema.

### 3.7.3.1 <contact:update> element

The <contact:update> element contains the following child elements:

- A <contact:roid> element to contain the identifier of the contact object to be modified.
- A <contact:name> element that contains the name of the individual or role represented by the contact.
- 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.

### 3.7.3.2 Example contact <modify> 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="http://www.nominet.org.uk/epp/xml/nom-contact-1.0"
C:         xsi:schemaLocation="http://www.nominet.org.uk/epp/xml/nom-contact-1.0
C:           nom-contact-1.0.xsd">
C:         <contact:roid>C11001</contact:roid>
C:         <contact:fax/>
C:         <contact:email>contact@example.co.uk</contact:email>
C:       </contact:update>
C:     </update>
C:   <clTRID>ABC-12345</clTRID>
C: </command>
C: </epp>
```

### 3.7.3.3 Example contact successful modify 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>65789465</svTRID>
S:     </trID>
S:   </response>
S: </epp>
```

## 3.7.4 Host <update> command

In order to modify the attributes of a nameserver object, the <update> command must contain a <ns:update> element that identifies the nameserver namespace and location of the nameserver schema.

Note that modifying a nameserver object will automatically modify all domain names associated with it.

### 3.7.4.1 <ns:update> element

The <ns:update> element contains the following sub-elements

- A <ns:roid> element containing the identifier of the nameserver object to be modified.
- An optional <ns:name> element giving the new name for the nameserver object.
- An optional <ns:addr> element with ip attribute of "v4" giving the new ipv4 address for the nameserver object.
- An optional <ns:addr> element with ip attribute of "v6" giving the new ipv6 address for the nameserver object.

### 3.7.4.2 Example nameserver <modify> 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:      <ns:update
C:      xmlns:ns="http://www.nominet.org.uk/epp/xml/nom-ns-1.0"
C:      xsi:schemaLocation="http://www.nominet.org.uk/epp/xml/nom-ns-1.0
C:      nom-ns-1.0.xsd">
C:      <ns:roid>NS1001</ns:roid>
C:      <ns:name>ns0.example2.co.uk</ns:name>
C:      </ns:update>
C:      </update>
C:      <clTRID>ABC-12345</clTRID>
C:      </command>
C:      </epp>

```

### 3.7.4.3 Example contact successful modify 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>65789465</svTRID>
S:     </trID>
S:   </response>
S: </epp>

```

## 4. Error messages and codes

When an operation fails, an error response is given by EPP. This includes an EPP error code and a text message. The <response> element will contain two elements:

- A <result> element containing the EPP error code.
- A <resData> element containing a <namespace:failData> element. The <failData> elements are described below:

### 4.1 Example error message

```

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="2004">
S:       <msg>Parameter value range error</msg>
S:     </result>
S:     <resData>
S:       <domain:failData
S:         xmlns:domain="http://www.nominet.org.uk/epp/xml/nom-domain-1.0"
S:         xsi:schemaLocation="http://www.nominet.org.uk/epp/xml/nom-domain-1.0
S:         nom-domain-1.0.xsd">
S:         <domain:name>example.co.uk</domain:name>
S:         <domain:reason>V017 bad value for 'first-bill' field (must be either
S:         'th' or 'bc')</domain:reason>
S:       </domain:failData>
S:     </resData>
S:     <trID>
S:       <clTRID>ABC-12345</clTRID>

```

```
S: <svTRID>123456</svTRID>
S: </trID>
S: </response>
S: </epp>
```

## 4.2 <domain:failData> element

The <domain:failData> element contains the following two sub-elements:

- A <domain:name> element giving the domain name concerned.
- A <domain:reason> element giving the reason for rejection of the request. This contains an automaton error message and contains an automaton error code (different to the EPP error code).

## 4.3 <account:failData> element

The <account:failData> element will be returned when <account:info> and <account:update> requests fail and will contain the following two sub-elements:

- An <account:roid> element giving the account concerned.
- An <account:reason> element giving the reason for rejection of the request. This contains an automaton error message and contains an automaton error code (different to the EPP error code).

## 4.4 <contact:failData> element

The <contact:failData> element will be returned when <contact:info> and <contact:update> requests fail and will contain the following two sub-elements:

- A <contact:roid> element giving the contact concerned.
- A <contact:reason> element giving the reason for rejection of the request. This contains an automaton error message and contains an automaton error code (different to the EPP error code).

## 4.5 <ns:failData> element

The <ns:failData> element will be returned when <ns:info> and <ns:update> requests fail and will contain the following two sub-elements:

- A <ns:roid> element giving the contact concerned.
- A <ns:reason> element giving the reason for rejection of the request. This contains an automaton error message and contains an automaton error code (different to the EPP error code).

## 5. Messages

EPP has a mechanism whereby messages to the tag-holder are queued up and the tag-holder can connect to EPP and send poll commands to receive the messages. We shall be extending the structure used in the RFC to allow for the message types that we need to send.

There are five types of messages that may be sent by the EPP system. They are:

- Accepted referrals. This is where a request for a .ltd.uk, .plc.uk, .net.uk or .sch.uk domain name has been processed manually and the domain name has been accepted. This message will be generated when the original request has been received via EPP rather than the automaton.
- Rejected referrals. This is where a request for a .ltd.uk, .plc.uk, .net.uk or .sch.uk domain name has been processed manually and the domain name has been rejected. This message will be generated when the original request has been received via EPP rather than the automaton.
- Tag change notification. This is generated when a domain name is transferred to a tag that has opted to receive these messages via EPP rather than the automaton

- Account object (registrant) change notification. This is generated when the details of an account sponsored by the tag are changed.
- Notification of change of account object for domain name. This is generated when a domain name is associated with a different account object during a transfer. This would not be generated when a new account object is generated during tag-changes or during account merging operations.

The details of these messages are described in more detail below.

## 5.1 Accepted referrals

This is essentially a message to say that a domain name has been accepted. The <msg> element will contain a <domain:creData> sub-element as described in section 3.4.1.8.

### 5.1.1 Example accepted referrals message

```
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="1301">
S:       <msg>Command completed successfully; ack to dequeue</msg>
S:     </result>
S:     <msgQ count="4" id="123456">
S:       <qDate>2005-10-06T10:29:30Z</qDate>
S:       <msg>Notification of referral result</msg>
S:     </msgQ>
S:     <resData>
S:       <domain:creData
S:         xmlns:domain="http://www.nominet.org.uk/epp/xml/nom-domain-1.0"
S:         xsi:schemaLocation="http://www.nominet.org.uk/epp/xml/nom-domain-1.0
S:           nom-domain-1.0.xsd">
S:         <domain:name>example.ltd.uk</domain:name>
S:         <domain:crDate>2005-01-03T22:00:00.0Z</domain:crDate>
S:         <domain:exDate>2007-01-03T22:00:00.0Z</domain:exDate>
S:       </domain:creData>
S:     </resData>
S:     <extension>
S:       <domain:notification type="referral" result="accepted"
S:         xmlns:domain="http://www.nominet.org.uk/epp/xml/nom-domain-1.0"/>
S:     </extension>
S:     <trID>
S:       <clTRID>ABC-12345</clTRID>
S:       <svTRID>123456</svTRID>
S:     </trID>
S:   </response>
S: </epp>
```

## 5.2 Rejected referrals

This message will simply contain some text describing why the request failed. The <msg> element will contain a <result> element as for an error message as in section 4.

### 5.2.1 Example rejected referrals message

```
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="1301">
S:       <msg>Command completed successfully; ack to dequeue</msg>
```

```

S: </result>
S: <msgQ count="4" id="123454">
S: <qDate>2005-10-06T10:29:30Z</qDate>
S: <msg>Notification of referral result</msg>
S: </msgQ>
S: <resData>
S: <domain:failData
S: <xmlns:domain="http://www.nominet.org.uk/epp/xml/nom-domain-1.0"
S: <xsi:schemaLocation="http://www.nominet.org.uk/epp/xml/nom-domain-1.0
S: <nom-domain-1.0.xsd">
S: <domain:name>example.ltd.uk</domain:name>
S: <domain:reason>V205 Registrant does not match domain name
S: </domain:reason>
S: </domain:failData>
S: </resData>
S: <extension>
S: <domain:notification type="referral" result="rejected"
S: <xmlns:domain="http://www.nominet.org.uk/epp/xml/nom-domain-1.0"/>
S: </extension>
S: <trID>
S: <clTRID>ABC-12345</clTRID>
S: <svTRID>123456</svTRID>
S: </trID>
S: </response>
S: </epp>

```

## 5.3 Tag change notifications

This message will contain some text informing the tag-holder that a tag change has taken place and a `<domain:infData>` sub-element describing the domain name.

### 5.3.1 Example tag change notification message

```

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="1301">
S: <msg>Command completed successfully; ack to dequeue</msg>
S: </result>
S: <msgQ count="4" id="123456">
S: <qDate>2005-10-06T10:29:30Z</qDate>
S: <msg>Notification of tag change</msg>
S: </msgQ>
S: <resData>
S: <domain:infData
S: <xmlns:domain="http://www.nominet.org.uk/epp/xml/nom-domain-1.0"
S: <xsi:schemaLocation="http://www.nominet.org.uk/epp/xml/nom-domain-1.0
S: <nom-domain-1.0.xsd">
S: <domain:name>example.co.uk</domain:name>
S: <domain:account>
S: <account:infData
S: <xmlns:account="http://www.nominet.org.uk/epp/xml/nom-account-1.0"
S: <xmlns:contact="http://www.nominet.org.uk/epp/xml/nom-contact-1.0">
S: <account:roid>S123456</account:roid>
S: <account:name>Mr R. Strant</account:name>
S: <account:trade-name>R. S. Industries</account:trade-name>
S: <account:type>STRA</account:type>
S: <account:co-no>NI123456</account:co-no>
S: <account:opt-out>N</account:opt-out>
S: <account:addr>
S: <account:street>2102 High Street</account:street>
S: <account:locality>Carfax</account:locality>
S: <account:city>Oxford</account:city>
S: <account:county>Oxfordshire</account:county>

```

```

S:      <account:pc>OX1 1DF</account:pc>
S:      <account:cc>GB</account:cc>
S:      </account:addr>
S:      <account:contact type="admin" main="Y">
S:      <contact:infData>
S:      <contact:roid>C12345</contact:roid>
S:      <contact:name>Mr R.Strant</contact:name>
S:      <contact:voice>01865 123456</contact:voice>
S:      <contact:fax>01865 123456</contact:fax>
S:      <contact:email>r.strant@strant.co.uk</contact:email>
S:      <contact:clID>TEST</contact:clID>
S:      <contact:crID>domains@isp.com</contact:crID>
S:      <contact:crDate>1999-04-03T22:00:00.0Z</contact:crDate>
S:      <contact:upID>domains@isp.com</contact:upID>
S:      <contact:upDate>1999-12-03T09:00:00.0Z</contact:upDate>
S:      </contact:infData>
S:      </account:contact>
S:      <account:clID>TEST</account:clID>
S:      <account:crID>TEST</account:crID>
S:      <account:crDate>1999-04-03T22:00:00.0Z</account:crDate>
S:      <account:upID>domains@isp.com</account:upID>
S:      <account:upDate>1999-12-03T09:00:00.0Z</account:upDate>
S:      </account:infData>
S:      </domain:account>
S:      <domain:ns
S:      xmlns:ns="http://www.nominet.org.uk/epp/xml/nom-ns-1.0">
S:      <ns:infData>
S:      <ns:roid>NS12345</ns:roid>
S:      <ns:name>ns1.example.co.uk</ns:name>
S:      <ns:addr ip="v4">10.10.10.10</ns:addr>
S:      <ns:clID>TEST</ns:clID>
S:      <ns:crID>domains@isp.com</ns:crID>
S:      <ns:crDate>1999-04-03T22:00:00.0Z</ns:crDate>
S:      <ns:upID>domains@isp.com</ns:upID>
S:      <ns:upDate>1999-12-03T09:00:00.0Z</ns:upDate>
S:      </ns:infData>
S:      <ns:infData>
S:      <ns:roid>NS12346</ns:roid>
S:      <ns:name>ns1.example.com</ns:name>
S:      <ns:clID>TEST</ns:clID>
S:      <ns:crID>domains@isp.com</ns:crID>
S:      <ns:crDate>1999-04-03T22:00:00.0Z</ns:crDate>
S:      <ns:upID>domains@isp.com</ns:upID>
S:      <ns:upDate>1999-12-03T09:00:00.0Z</ns:upDate>
S:      </ns:infData>
S:      </domain:ns>
S:      <domain:clID>TEST</domain:clID>
S:      <domain:crID>TEST</domain:crID>
S:      <domain:crDate>1999-04-03T22:00:00.0Z</domain:crDate>
S:      <domain:upID>domains@isp.com</domain:upID>
S:      <domain:upDate>1999-12-03T09:00:00.0Z</domain:upDate>
S:      </domain:infData>
S:      </resData>
S:      <extension>
S:      <domain:notification type="change" subject="tag"
S:      xmlns:domain="http://www.nominet.org.uk/epp/xml/nom-domain-1.0"/>
S:      </extension>
S:      <trID>
S:      <clTRID>ABC-12345</clTRID>
S:      <svTRID>123456</svTRID>
S:      </trID>
S:      </response>
S:      </epp>

```

Note that the <domain:upID> element shows who made the tag-change.

## 5.4 Account object change notifications

This message will contain some text informing the client that a modification has been made and an <account:infData> object giving the details. Contact objects will be included in this element only modifications have been made to them. Each sub-element will have an extra attribute (not used elsewhere) of changed which will be present and equal to "Y" only when the field has been modified.

### 5.4.1 Example account object change notification message

```
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="1301">
S:       <msg>Command completed successfully; ack to dequeue</msg>
S:     </result>
S:     <msgQ count="4" id="123456">
S:       <qDate>2005-10-06T10:29:30Z</qDate>
S:       <msg>Notification of account change</msg>
S:     </msgQ>
S:     <resData>
S:       <account:infData
S:         xmlns:account="http://www.nominet.org.uk/epp/xml/nom-account-1.0"
S:         xmlns:contact="http://www.nominet.org.uk/epp/xml/nom-contact-1.0"
S:         xsi:schemaLocation="http://www.nominet.org.uk/epp/xml/nom-account-1.0
S:           nom-account-1.0.xsd">
S:         <account:roid>S123456</account:roid>
S:         <account:name>Mr R. Strant</account:name>
S:         <account:type changed="Y">IND</account:type>
S:         <account:opt-out changed="Y">Y</account:opt-out>
S:         <account:addr>
S:           <account:street>2102 High Street</account:street>
S:           <account:locality>Carfax</account:locality>
S:           <account:city>Oxford</account:city>
S:           <account:county changed="Y">Oxfordshire</account:county>
S:           <account:pc>OX1 1DF</account:pc>
S:           <account:cc>GB</account:cc>
S:         </account:addr>
S:         <account:clID>TEST</account:clID>
S:         <account:crID>TEST</account:crID>
S:         <account:crDate>1999-04-03T22:00:00.0Z</account:crDate>
S:         <account:upID>domains@isp.com</account:upID>
S:         <account:upDate>1999-12-03T09:00:00.0Z</account:upDate>
S:       </account:infData>
S:     </resData>
S:     <extension>
S:       <account:notification type="change" subject="account"
S:         xmlns:account="http://www.nominet.org.uk/epp/xml/nom-account-1.0"/>
S:     </extension>
S:     <trID>
S:       <clTRID>ABC-12345</clTRID>
S:       <svTRID>123456</svTRID>
S:     </trID>
S:   </response>
S: </epp>
```

## 5.5 Notification of change of account object for a domain name

This message contains some text indicating that this has happened and a <domain:infData> object showing the domain name that has changed.

### 5.5.1 Example notification of change of account object for a domain name.

```
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="1301">
S:     <msg>Command completed successfully; ack to dequeue</msg>
S:   </result>
S:   <msgQ count="4" id="123456">
S:     <qDate>2005-10-06T10:29:30Z</qDate>
S:     <msg>Notification of change of domain account change</msg>
S:   </msgQ>
S:   <resData>
S:     <domain:infData
S:       xmlns:domain="http://www.nominet.org.uk/epp/xml/nom-domain-1.0"
S:       xsi:schemaLocation="http://www.nominet.org.uk/epp/xml/nom-domain-1.0
S:         nom-domain-1.0.xsd">
S:       <domain:name>example.co.uk</domain:name>
S:       <domain:account>
S:         <account:infData
S:           xmlns:account="http://www.nominet.org.uk/epp/xml/nom-account-1.0"
S:           xmlns:contact="http://www.nominet.org.uk/epp/xml/nom-contact-1.0">
S:           <account:roid>S123454</account:roid>
S:           <account:name>Mr Reg. New</account:name>
S:           <account:type>IND</account:type>
S:           <account:addr>
S:             <account:street>10 High Street</account:street>
S:             <account:city>Big Town</account:city>
S:             <account:pc>BT1 1DF</account:pc>
S:             <account:cc>GB</account:cc>
S:           </account:addr>
S:           <account:contact type="admin" main="Y">
S:             <contact:infData>
S:               <contact:roid>C1542345</contact:roid>
S:               <contact:name>Mr Reg. New</contact:name>
S:               <contact:voice>01455 173456</contact:voice>
S:               <contact:email>reg@new.com</contact:email>
S:               <contact:clID>TEST</contact:clID>
S:               <contact:crID>domains@isp.com</contact:crID>
S:               <contact:crDate>1999-04-03T22:00:00.0Z</contact:crDate>
S:               <contact:upID>domains@isp.com</contact:upID>
S:               <contact:upDate>1999-12-03T09:00:00.0Z</contact:upDate>
S:             </contact:infData>
S:           </account:contact>
S:           <account:clID>TEST</account:clID>
S:           <account:crID>TEST</account:crID>
S:           <account:crDate>1999-04-03T22:00:00.0Z</account:crDate>
S:           <account:upID>domains@isp.com</account:upID>
S:           <account:upDate>1999-12-03T09:00:00.0Z</account:upDate>
S:         </account:infData>
S:       </domain:account>
S:       <domain:clID>TEST</domain:clID>
S:       <domain:crID>TEST</domain:crID>
S:       <domain:crDate>1999-04-03T22:00:00.0Z</domain:crDate>
S:       <domain:upID>domains@isp.com</domain:upID>
S:       <domain:upDate>1999-12-03T09:00:00.0Z</domain:upDate>
S:     </domain:infData>
S:   </resData>
S:   <extension>
S:     <domain:notification type="change" subject="domain account"
S:       xmlns:domain="http://www.nominet.org.uk/epp/xml/nom-domain-1.0"/>
S:   </extension>
S:   <trID>
S:     <clTRID>ABC-12345</clTRID>
S:     <svTRID>123456</svTRID>
S:   </trID>
S: </response>
S: </epp>

```

## 6. Security

The model of tag-holders logging on and giving a password and then being subject to no further checks is less secure than the PGP signatures required by the automaton. Therefore we plan, at some point in the future, to add a new element which contains the PGP signature of some previous part of the message.

## 7. Divergences from EPP standard

- We don't have any status values
- We don't use the <domain:authInfo> element. We require only that the client has successfully logged in as the correct TAG for the domain name.
- We have a different contact and account (registrant) model. One account can be associated with each domain names and contacts are associated with accounts.
- Addresses are associated with accounts only and not with contacts.
- The <transfer> operation is initiated by the currently sponsoring tag and a <tag> sub-element must be provided to indicate the tag to transfer the domain name to.
- The EPP standard uses the field <contact:sp> for the contact's state or province. The equivalent for British addresses is county which has been used here.
- We shall not be implementing the <transfer> command as it is not compatible with our business practises. The <transfer> command would allow for TAGs to request a domain name to be transferred to them, rather than our method of the tag-holder releasing the domain name to another tag. We shall include a <release> command to implement this.
- We have new elements in the domain name object not included in the RFCs – auto-bill, next-bill, first-bill, recur-bill, purchase-order, notes,
- Internal nameserver objects may exist even when the domain name does not. Domain names with subordinate nameserver objects may be deleted.
- We do not need the <domain:curExpDate> in the renew command as domain names can only be renewed. Further attempts at renewal will be rejected.
- We are not using the add, chg, rem mechanism or updates. This is too complex for our needs.

## 8. References

[DRD] Data Restructuring document. <http://www.nominet.org.uk/tag/groups/closed/data>

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

[RFC952] Harrenstien, K., Stahl, M. and E. Feinler, "DOD Internet Host Table Specification, RFC 952, October 1985.

[RFC1123] Braden, R., Ed., "Requirements for Internet Hosts – Application and Support", STD 3, RFC 1123, October 1989.

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

[RFC 3339] Klyne, G. and C. Newman, "Date and Time on the Internet: Timestamps", RFC 3339, July 2002.

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