.MD EPP Server description

Contents

Short Glossary

Registry: A domain name registry is a part of the Domain Name System (DNS) of the Internet which converts domain names to IP addresses. It is an organisation which manages the registration of Domain names within the top-level domains for which it is responsible, controls the policies of domain name allocation, and technically operates its top-level domain.

Registrar: A domain name registrar is an organization or commercial entity, accredited by a country code top-level domain (ccTLD) registry,to register Internet domain names in accordance with the guidelines of the designated domain name registry.

Registrant: The person or entity that owns the domain name according to the records of the Registry (Registrar). Registrant has a freedom of choice and can change the Registrar at any moment by transferring to another Registrar.

EPP: The Extensible Provisioning Protocol (EPP) is a modern protocol designed for communication between domain name registry and registrars. The EPP protocol is based on XML - a structured, text-based format. The underlying network transport is not fixed, although the only currently specified method is over TCP.

1. Introduction

.MD EPP-server supports the Extensible Provisioning Protocol (EPP), the IETF standards-track protocol for interaction between registry and registrar defined in the following RFC's:
Extensible Provisioning Protocol (EPP) RFC 3730
Extensible Provisioning Protocol (EPP) Domain Name Mapping RFC 3731
Extensible Provisioning Protocol (EPP) Contact Mapping RFC 3733
Extensible Provisioning Protocol (EPP) Host Mapping RFC 3732
Guidelines for Extending the Extensible Provisioning Protocol (EPP) RFC 3735
Extensible Provisioning Protocol (EPP) Transport RFC 3734

The purpose of this document is to describe the existing EPP commands of the .MD EPP-server. Readers are assumed to be familiar with the EPP-protocol as we will not cover this in this document. Please refer to the RFC:s listed above for more information on the EPP-protocol.

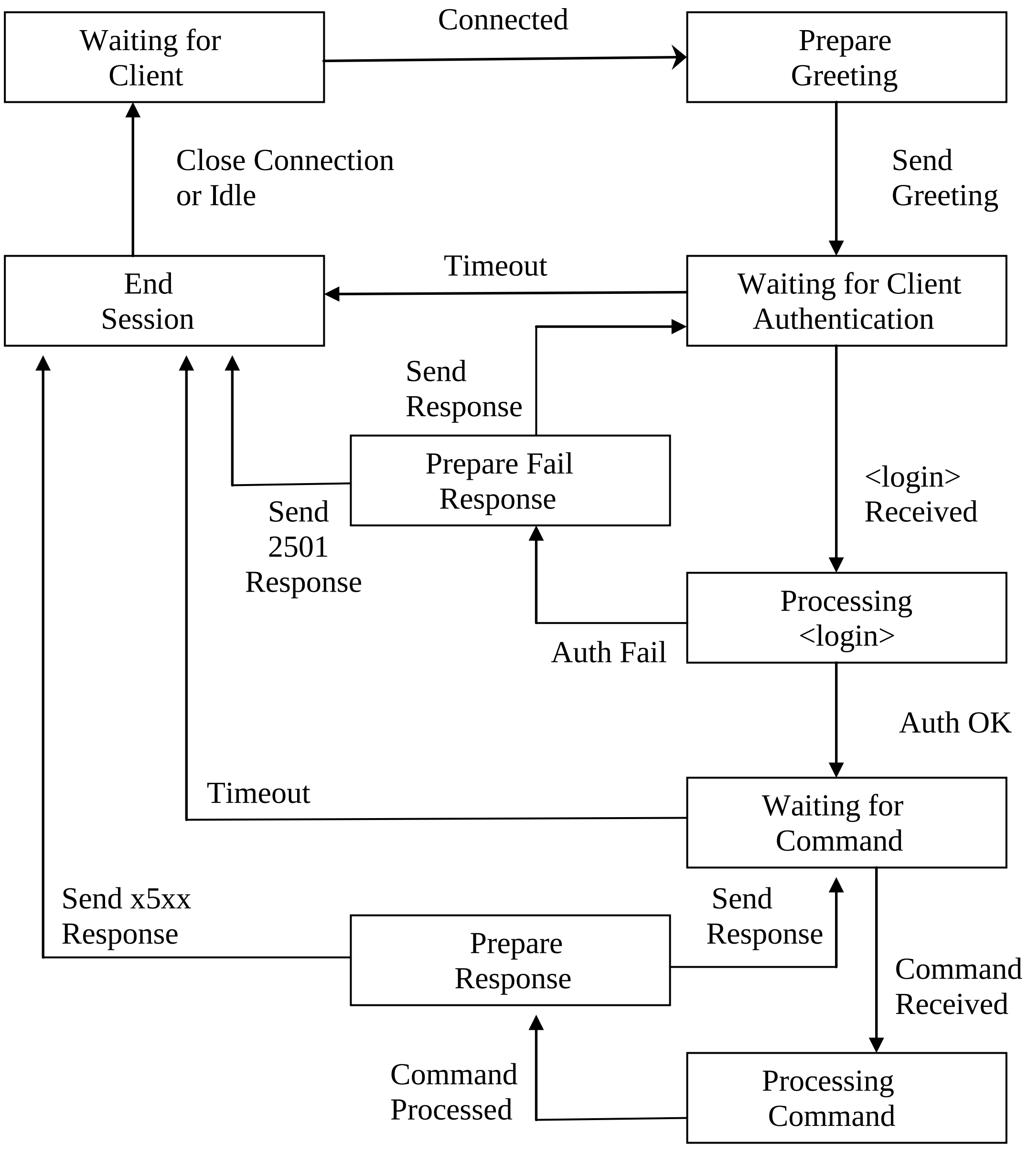
The EPP-commands are, in this document, divided into two main categories, session commands and object commands.

Session commands are used for session management. These include a command to initiate a session (login), and the end a session (logout).

Object commands are used to query and transform objects at the registry. There are three kinds of objects used by the .MD EPP server: domain objects, contact objects and host (nameservers) objects. Domain objects contain all data necessary to represent internet domain names, contact objects contains all necessary data to represent persons or companies and host objects contains all data necessary to represent name servers in the registry.

The server promptly respond to each EPP command with a coordinated response that describes the results of processing the command. The following server state machine diagram illustrates the message exchange process between registry (EPP-server) and registrar (EPP-client):

EPP Server State Machine



EPP COMMANDS

2. SESSION COMMANDS

2.1 LOGIN

The EPP <login> command is used to establish a session with an EPP server in response to a greeting issued by the server. A <login> command must be sent to a server before any other EPP command to establish an ongoing session.

In addition to the standard EPP command elements, the <login> command contains the following child elements:
<clID> element that contains the client identifier assigned to the client by the server.
<pw> element that contains the client's plain text password.
The value of this element is case sensitive.

Client (registrar) <login> command:

```
C:     <?xml version="1.0" encoding="UTF-8"?>
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 epp-1.0.xsd">
C:      <command>
C:          <login>
C:               <clID>test</clID>
C:                <pw>test123</pw>
C:                     <options/>
C:                        <svcs/>
C:          </login>
C:               <clTRID/>
C:        </command>
C:        </epp>
```

Server (registry) <login> 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  epp-1.0.xsd">
S:         <response>
S:             <result code="1000">
S:                 <msg lang="en">User vas was authenticated. Welcome.</msg>
S:             </result>
S:                 <trID>
S:                     <clTRID/>
S:                 </trID>
S:          </response>
S:       </epp>
```

2.2. LOGOUT

The EPP <logout> command is used to end a session with an EPP server. The <logout> command must be represented as an empty element with no child elements.

Client <logout> command:

```
C:     <?xml version="1.0" encoding="UTF-8"?>
C:     <epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
              xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
C:                xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp1.0.xsd">
C:        <command>
C:           <logout/>
C:               <clTRID/>
C:         </command>
C:       </epp>
```

Server <logout> 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 epp-1.0.xsd">
S:            <response>
S:                <result code="1500">
S:                    <msg lang="en">User  logged out. Closing Connection.</msg>
S:                </result>
S:                    <trID>
S:                       <clTRID></clTRID>
S:                    </trID>
S:              </response>
S:        </epp>
```

3. OBJECT COMMANDS (Query Commands)

3.1. Domain Objects

Domain objects contain all data necessary to represent an internet domain name.

Tagname/attribute	Representation
<domain:account>	An email address - the client (registrar) Login ID assigned to the client by the server (registry) for manage the domains in his account.
<domain:account_pw>	Account password. 6-16 characters a-z,A-Z,0-9
<domain:name>	Domain name (without “www”and “.md”). 2-63 characters a-z, 0-9, - A domain name cannot start or end with a hyphen (-).
<domain:name years="n">	“n” in “years” attribute specify a period of activity for the domain
<domain:org_name>	The organization name (registrant)
<domain:org_str>	The organization street.
<domain:org_city>	The organization city
<domain:org_postc>	The organization postal code
<domain:org_country>	The organization country. (2 characters – country code)
<domain:org_state>	The organization state (province)
<domain:adm_firstname>	Administrative Contact firat name
<domain:adm_lastname>	Administrative Contact last name
<domain:adm_str>	Administrative Contact street
<domain:adm_city>	Administrative Contact city
<domain:teh_postc>	Administrative Contact postal code
<domain:adm_country>	Administrative Contact country (2 characters – country code)
<domain:adm_phone>	Administrative Contact phone
<domain:adm_fax>	Administrative Contact fax
<domain:adm_email>	Administrative Contact email address
<domain:adm_state>	Administrative Contact state (province)
<domain:teh_firstname>	Technical Contact first name
<domain:teh_lastname>	Technical Contact last name
<domain:teh_str>	Technical Contact street
<domain:teh_city>	Technical Contact city
<domain:teh_postc>	Technical Contact postal code
<domain:teh_country>	Technical Contact country (2 characters – country code)
<domain:teh_phone>	Technical Contact phone
<domain:teh_fax>	Technical Contact fax
<domain:teh_email>	Technical Contact email address
<domain:teh_state>	Technical Contact state (province)
<domain:bil_firstname>	Billing Contact first name
<domain:bil_lastname>	Billing Contact last name
<domain:bil_str>	Billing Contact street
<domain:bil_city>	Billing Contact city
<domain:bil_postc>	Billing Contact postal code
<domain:bil_country>	Billing Contact country (2 characters – country code)
<domain:bil_phone>	Billing Contact phone
<domain:bil_fax>	Billing Contact fax
<domain:bil_email>	Billing Contact email address
<domain:bil_state>	Billing Contact state (province)

<domain:primNSname>	Primary Name Server hostname
<domain: primNSip4>	Primary Name Server IP-v4 address
<domain:secNSname>	Secondary Name Server hostname
<domain:secNSip4>	Secondary Name Server IP-v4 address
<domain: primNSip6>	Primary Name Server IP-v6 address
<domain:secNSip6>	Secondary Name Server IP-v6 address
<clTRID>	It is a client transaction identifier holder. The clTRID, if supplied, is returned along with the svTRID (server transaction identifier) in the response to an EPP command.

3.2. EPP <check> command

The EPP <check> command allow the registrar to check if a domain name is available before actually submitting a registration for it.
This command actually allows a list of domains to be checked.
The command is as follows:

Client <check> command:

```
C:  <?xml version="1.0" encoding="UTF-8"?>
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 epp-1.0.xsd">
C:   <command>
C:     <check>
C:       <domain:check xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
C:           xsi:schemaLocation="urn:ietf:params:xml:ns:domain-1.0 domain-.0.xsd">
C:       <domain:name>domain-1</domain:name>
C:       <domain:name>nic</domain:name>
C:       <domain:name>domain-n</domain:name>
C:    </domain:check>
C:    </check>
C:       <clTRID>20101126093102</clTRID>
C:   </command>
C: </epp>
```

Server <check> response:

```
S: <?xml version="1.0" encoding="UTF-8" standalone="no"?>
S: <epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
S:         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
S:                xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0  epp-1.0.xsd">
S:    <response>
S:        <result code="1000">
S:           <msg>Command completed successfully</msg>
S:        </result>
```

```
S:           <resData>
S: <domain:chkData xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
S:          xsi:schemaLocation="urn:ietf:params:xml:ns:domain-1.0 domain-1.0.xsd">
S:         <domain:name res="1">domain-1</domain:name>
S:         <domain:name res="0">nic</domain:name>
S:         <domain:name res="1">domain-n</domain:name>
S:  </domain:chkData>
S:       </resData>
S:           <clTRID> 20101126093102  </clTRID>
S:      </response>
S: </epp>
```

Note. Variants of response:

attribute	*value*	*tag*	*Means*
res	1	<domain:name res=”1”>	domain is available for registration
res	0	<domain:name res=”0”>	domain already exists

3.3. EPP <create> command

The EPP <create> command is used to create an instance of an object
If a domain is available for registration, then the registrar can continue with the domain create command. The registrar can create a domain only in its account assigned by the server (registry).

Client <create> command:

```
C:<?xml version="1.0" encoding="UTF-8"?>
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 epp-1.0.xsd">
C:<command>
C: <create>
C:    <domain:create xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
C:         xsi:schemaLocation="urn:ietf:params:xml:ns:domain-1.0 domain-1.0.xsd">
C:    <domain:account>epptest@nic.md</domain:account>
C:    <domain:account_pw>test123</domain:account_pw>
C:    <domain:name years="2">domain-1</domain:name>
C:    <domain:name years="1">nic</domain:name>
C:    <domain:name years="5">domain-n</domain:name>
C:  <domain:org_name>MoldData</domain:org_name>
C:  <domain:org_str>Armeneasca 37/1</domain:org_str>
C:  <domain:org_city>Chisinau</domain:org_city>
C:  <domain:org_postc>2012</domain:org_postc>
C:  <domain:org_country>MD</domain:org_country>
C:  <domain:org_state></domain:org_state>
```

```
C:   <domain:adm_firstname>Frunza</domain:adm_firstname>
C:   <domain:adm_lastname>Ion</domain:adm_lastname>
C:    <domain:adm_str>Armeneasca 37/1</domain:adm_str>
C:    <domain:adm_city>Chisinau</domain:adm_city>
C:    <domain:adm_postc>2012</domain:adm_postc>
C:    <domain:adm_country>MD</domain:adm_country>
C:    <domain:adm_phone>545454</domain:adm_phone>
C:   <domain:adm_fax>565656</domain:adm_fax>
C:    <domain:adm_email>hm@nic.md</domain:adm_email>
C:    <domain:adm_state></domain:adm_state>
C:<domain:teh_firstname>Frunza </domain:teh_firstname>
C:<domain:teh_lastname>Ion</domain:teh_lastname>
C:<domain:teh_str>Armeneasca 37/1</domain:teh_str>
C:<domain:teh_city>Chisinau</domain:teh_city>
C:<domain:teh_postc>2012</domain:teh_postc>
C:<domain:teh_country>MD</domain:teh_country>
C:<domain:teh_phone>555555</domain:teh_phone>
C:<domain:teh_fax>555554</domain:teh_fax>
C:<domain:teh_email>hm@nic.md</domain:teh_email>
C:<domain:teh_state></domain:teh_state>
C:     <domain:bil_firstname>Frunza</domain:bil_firstname>
C:     <domain:bil_lastname>Ion</domain:bil_lastname>
C:     <domain:bil_str>Armeneasca 37/1</domain:bil_str>
C:     <domain:bil_city>Chisinau</domain:bil_city>
C:     <domain:bil_postc>2012</domain:bil_postc>
C:     <domain:bil_country>MD</domain:bil_country>
C:     <domain:bil_phone>56789</domain:bil_phone>
C:     <domain:bil_fax>55555</domain:bil_fax>
C:     <domain:bil_email>hm@nic.md</domain:bil_email>
C:     <domain:bil_state></domain:bil_state>
C:<domain:primNSname>nx.dns.md</domain:primNSname>
C:<domain:primNSip4>217.91.81.15</domain:primNSip4>
C:<domain:secNSname>ny.dns.md</domain:secNSname>
C:<domain:secNSip4>217.25.32.2</domain:secNSip4>
C:<domain:primNSip6></domain:primNSip6>
C:<domain:secNSip6></domain:secNSip6>
C:</domain:create>
C:</create><clTRID>20101201150751</clTRID>
C:</command>
C:</epp>
```

Server <create> response:

```
S:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
S:    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
S:    xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
S:<response>
S:<result code="1000">
S:<msg>Command completed successfully</msg>
S:</result>
S:<resData>
S:<domain:creData xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
S:   xsi:schemaLocation="urn:ietf:params:xml:ns:domain-1.0 domain-1.0.xsd">
S: <domain:name res=”1”>
                          domain-1=created expiration date yyyy-mm-dd</domain:name>
S:<domain:name res=”0”>nic=busy</domain:name>
S:<domain:name res=”1”>
                           domain-n=created expiration date yyyy-mm-dd</domain:name>
S:</domain:creData>
S:</resData>
S:<clTRID> 20101201150751  </clTRID>
S:</response>
S:</epp>
```

Note. Variants of response:

tag	*value*
<domain:name res=”1”>	“domain name” created expiration date yyyy-mm-dd
<domain:name res=”0”>	“domain name” busy

3.4 EPP <update> Command

The EPP <update> command is used to change information associated with an existing object.

The updatable values for domain objects are the registrant, the contact list, the nameserver list. In the update command, simply specify a new value and the server will replace the old one. If the tag is empty the server will “keep” the existing value. The registrar can update a domain only in its account assigned by the server (registry).

Client <update> command:

```
C:<?xml version="1.0" encoding="UTF-8"?>
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 epp-1.0.xsd">
C:<command>
C: <update>
C:    <domain:update xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
C:          xsi:schemaLocation="urn:ietf:params:xml:ns:domain-1.0 domain-1.0.xsd">
C:    <domain:account>epptest@nic.md</domain:account>
C:    <domain:account_pw>test123</domain:account_pw>
C:    <domain:name>domain</domain:name>
C:  <domain:org_name></domain:org_name>
C:  <domain:org_str></domain:org_str>
C:  <domain:org_city></domain:org_city>
C:  <domain:org_postc></domain:org_postc>
C:  <domain:org_country></domain:org_country>
C:  <domain:org_state></domain:org_state>
C:   <domain:adm_firstname/>
C:   <domain:adm_lastname/>
C:    <domain:adm_str/>
C:    <domain:adm_city/>
C:    <domain:adm_postc/>
C:    <domain:adm_country/>
C:    <domain:adm_phone/>
C:   <domain:adm_fax/>
C:    <domain:adm_email/>
C:    <domain:adm_state/>
C:<domain:teh_firstname/>
C:<domain:teh_lastname/>
C:<domain:teh_str/>
C:<domain:teh_city/>
C:<domain:teh_postc/>
C:<domain:teh_country/>
C:<domain:teh_phone/>
C:<domain:teh_fax/>
C:<domain:teh_email/>
C:<domain:teh_state/>
C:     <domain:bil_firstname/>
C:     <domain:bil_lastname/>
C:     <domain:bil_str/>
C:     <domain:bil_city/>
C:     <domain:bil_postc/>
C:     <domain:bil_country/>
C:     <domain:bil_phone/>
C:     <domain:bil_fax/></domain:bil_fax>
C:     <domain:bil_email>billing@abc.md</domain:bil_email>
C:     <domain:bil_state></domain:bil_state>
```

```
C:<domain:primNSname>ns1.dns.com</domain:primNSname>
C:<domain:primNSip4>113.91.81.15</domain:primNSip4>
C:<domain:secNSname>ns2.dns.com</domain:secNSname>
C:<domain:secNSip4>222.25.32.22</domain:secNSip4>
C:<domain:primNSip6></domain:primNSip6>
C:<domain:secNSip6></domain:secNSip6>
C:</domain:update>
C:</update><clTRID>20101201150751</clTRID>
C:</command>
C:</epp>
```

Note: In this case the server will update only “billing contact” email address and the nameservers.

Server <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 epp-1.0.xsd">
S:<response>
S:<result code="1000">
S:<msg>Command completed successfully</msg>
S:</result>
S:<resData>
S:<domain:creData xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
S:    xsi:schemaLocation="urn:ietf:params:xml:ns:domain-1.0 domain-1.0.xsd">
S: <domain:name res=”1”>domain-1 updateded</domain:name>
S:<domain:name res=”0”>nic not in account</domain:name>
S:</domain:creData>
S:</resData>
S:<clTRID> 20101201150751  </clTRID>
S:</response>
S:</epp>
```

Note. Variants of response:

tag	*value*
<domain:name res=”1”>	“domain name” updated
<domain:name res=”0”>	“domain name” not in account

3.5 EPP <info> Command

Once an object is in the EPP registry, a registrar may retrieve information regarding the object. This is achieved through the "info" command.

The response to this command may vary depending on the identity of the querying client, use of authorization information, and server policy towards unauthorized clients. If the querying client is the sponsoring registrar, all available information must be returned regarding the domains within your account and partially information will be returned regarding the domains without your account .
The partially information will include: Organization name, city, country; Admin.Contact email address; Name servers; created date; expiry date.

Client <info> command:

```
C: <?xml version="1.0" encoding="UTF-8"?>
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 epp-1.0.xsd">
C:  <command>
C:    <info>
C:      <domain:info xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
C:          xsi:schemaLocation="urn:ietf:params:xml:ns:domain-1.0 domain-1.0.xsd">
C:        <domain:account>epptest@nic.md</domain:account>
C:        <domain:account_pw>test123</domain:account_pw>
C:        <domain:name>nik</domain:name>
C:      </domain:info>
C:    </info>
C:    <clTRID>20101221091834</clTRID>
C:  </command>
C:</epp>
```

Server <info> response *(the domain **within** your account)**:***

```
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 epp-1.0.xsd">
S:<response>
S:<result code="1000">
S:<msg>Command completed successfully</msg>
S:</result>
S:<resData>
S:<domain:infData xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
S:          xsi:schemaLocation="urn:ietf:params:xml:ns:domain-1.0 domain-1.0.xsd">
```

```
S:<domain:name>nik exists</domain:name>
S:<domain:org_name>MoldData</domain:org_name>
S:<domain:org_str>Armeneasca 37/1</domain:org_str>
S:<domain:org_city>Chisinau</domain:org_city>
S:<domain:org_postc>2012</domain:org_postc>
S:<domain:org_country>MD</domain:org_country>
S:<domain:org_state></domain:org_state>
S:<domain:adm_firstname>Admin</domain:adm_firstname>
S:<domain:adm_lastname>Contact</domain:adm_lastname>
S:<domain:adm_str>Armeneasca 37/1</domain:adm_str>
S:<domain:adm_city>Chisinau</domain:adm_city>
S:<domain:adm_postc>2012</domain:adm_postc>
S:<domain:adm_country>MD</domain:adm_country>
S:<domain:adm_phone>545264</domain:adm_phone>
S:<domain:adm_fax></domain:adm_fax>
S:<domain:adm_email>support@nic.md</domain:adm_email>
S:<domain:adm_state></domain:adm_state>
S:<domain:teh_firstname>Tech</domain:teh_firstname>
S:<domain:teh_lastname>Contact</domain:teh_lastname>
S:<domain:teh_str>Armeneasca 37/1</domain:teh_str>
S:<domain:teh_city>Chisinau</domain:teh_city>
S:<domain:teh_postc>2012</domain:teh_postc>
S:<domain:teh_country>MD</domain:teh_country>
S:<domain:teh_phone>545264</domain:teh_phone>
S:<domain:teh_fax></domain:teh_fax>
S:<domain:teh_email>support@nic.md</domain:teh_email>
S:<domain:teh_state></domain:teh_state>
S:<domain:bil_firstname>Billing</domain:bil_firstname>
S:<domain:bil_lastname>Contact</domain:bil_lastname>
S:<domain:bil_str>Armeneasca 37/1</domain:bil_str>
S:<domain:bil_city>Chisinau</domain:bil_city>
S:<domain:bil_postc>2012</domain:bil_postc>
S:<domain:bil_country>MD</domain:bil_country>
S:<domain:bil_phone>545264</domain:bil_phone>
S:<domain:bil_fax></domain:bil_fax>
S:<domain:bil_email>support@nic.md</domain:bil_email>
S:<domain:bil_state>support@nic.md</domain:bil_state>
S:<domain:primNSname>nsa.dns.md</domain:primNSname>
S:<domain:primNSip4>217.26.144.5</domain:primNSip4>
S:<domain:secNSname>nsb.dns.md</domain:secNSname>
S:<domain:secNSip4>217.26.144.15</domain:secNSip4>
S:<domain:primNSip6></domain:primNSip6>
S:<domain:secNSip6></domain:secNSip6>
```

```
S:<domain:reg_date>2007-05-22</domain:reg_date>
S:<domain:exp_date>2017-05-22</domain:exp_date>
S:</domain:infData>
S:</resData>
S:<clTRID> 20101221091834  </clTRID>
S:</response>
S:</epp>
```

Server <info> response *(the domain **without** your account)**:***

```
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 epp-1.0.xsd">
S:<response>
S:<result code="1000">
S:<msg>Command completed successfully</msg>
S:</result>
S:<resData>
S:<domain:infData xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
S:         xsi:schemaLocation="urn:ietf:params:xml:ns:domain-1.0 domain-1.0.xsd">
S:<domain:name>nik exists</domain:name>
S:<domain:org_name>MoldData</domain:org_name>
S:<domain:org_str></domain:org_str>
S:<domain:org_city>Chisinau</domain:org_city>
S:<domain:org_postc></domain:org_postc>
S:<domain:org_country>MD</domain:org_country>
S:<domain:org_state></domain:org_state>
S:<domain:adm_firstname></domain:adm_firstname>
S:<domain:adm_lastname></domain:adm_lastname>
S:<domain:adm_str></domain:adm_str>
S:<domain:adm_city></domain:adm_city>
S:<domain:adm_postc></domain:adm_postc>
S:<domain:adm_country></domain:adm_country>
S:<domain:adm_phone></domain:adm_phone>
S:<domain:adm_fax></domain:adm_fax>
S:<domain:adm_email>support@nic.md</domain:adm_email>
S:<domain:adm_state></domain:adm_state>
S:<domain:teh_firstname></domain:teh_firstname>
S:<domain:teh_lastname></domain:teh_lastname>
S:<domain:teh_str></domain:teh_str>
S:<domain:teh_city></domain:teh_city>
S:<domain:teh_postc></domain:teh_postc>
S:<domain:teh_country></domain:teh_country>
```

```
S:<domain:teh_phone></domain:teh_phone>
S:<domain:teh_fax></domain:teh_fax>
S:<domain:teh_email></domain:teh_email>
S:<domain:teh_state></domain:teh_state>
S:<domain:bil_firstname></domain:bil_firstname>
S:<domain:bil_lastname></domain:bil_lastname>
S:<domain:bil_str></domain:bil_str>
S:<domain:bil_city></domain:bil_city>
S:<domain:bil_postc></domain:bil_postc>
S:<domain:bil_country></domain:bil_country>
S:<domain:bil_phone></domain:bil_phone>
S:<domain:bil_fax></domain:bil_fax>
S:<domain:bil_email> </domain:bil_email>
S:<domain:bil_state> </domain:bil_state>
S:<domain:primNSname>nsa.dns.md</domain:primNSname>
S:<domain:primNSip4>217.26.144.5</domain:primNSip4>
S:<domain:secNSname>nsb.dns.md</domain:secNSname>
S:<domain:secNSip4>217.26.144.15</domain:secNSip4>
S:<domain:primNSip6></domain:primNSip6>
S:<domain:secNSip6></domain:secNSip6>
S:<domain:reg_date>2007-05-22</domain:reg_date>
S:<domain:exp_date>2017-05-22</domain:exp_date>
S:</domain:infData>
S:</resData>
S:<clTRID> 20101221091834  </clTRID>
S:</response>
S:</epp>
```

Note. Variants of response:

tag	*value*
<domain:name res=”0”>	“domain name” not exists
<domain:name res=”1”>	“domain name” info.
<domain:name res=”2”>	“domain name” not in account

3.6 EPP <renew> Command

When a domain is created, a "period" is usually specified by the registrar. The period, in years, defines the lifetime of the domain object. After that period passes, the domain expires.

Anytime before a domain expires, it may be renewed by the sponsoring registrar. The expiry date of a domain can be found through the domain info command. In the domain renew command, three values can be provided: the domain name (required), the renewal period (years -- required) and the current expiry date (year, month and day -- required). The current expiry is required to provide that the same renew command cannot be processed multiple times.

Client <renew> command:

```
C:<?xml version="1.0" encoding="UTF-8"?>
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 epp-1.0.xsd">
C:<command>
C:<renew>
C:<domain:renew xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
C:      xsi:schemaLocation="urn:ietf:params:xml:ns:domain-1.0 domain-1.0.xsd">
C:<domain:account>epptest@nic.md </domain:account>
C:<domain:account_pw>test123</domain:account_pw>
C:<domain:name curexp=”2011-02-25” years="5">domain-1 </domain:name>
C:<domain:name curexp=”2011-07-23” years="1">nic </domain:name>
C:<domain:name curexp=”2011-01-12” years="2">domain-n </domain:name>
C:</domain:renew>
C:</renew>
C:      <clTRID>20110106081712</clTRID>
C:</command>
C:</epp>
```

Server <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:creData xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
S:        xsi:schemaLocation="urn:ietf:params:xml:ns:domain-1.0 domain-1.0.xsd">
S:<domain:name res=”1”> domain new exp_date year-month-day</domain:name>
S:</domain:creData>
S:</resData>
S:<clTRID> 20110106081712  </clTRID>
S:</response>
```

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

Note. Variants of response:

tag	*value*
<domain:name res=”1”>	“domain name” new exp_date year-month-day
<domain:name res=”0”>	“domain name” not in account
<domain:name res=”2”>	“domain name” FAIL current expiration date

3.7 EPP <delete> Command

Just as you can create objects in a registry, you can delete them too. Of course, you must be the current sponsoring registrar in order to successfully delete an object.

Client <delete> command:

```
C:  <?xml version="1.0" encoding="UTF-8"?>
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 epp-1.0.xsd">
C:   <command>
C:     <delete>
C:       <domain:delete xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
C:           xsi:schemaLocation="urn:ietf:params:xml:ns:domain-1.0 domain-.0.xsd">
C:    <domain:account>epptest@nic.md </domain:account>
C:    <domain:account_pw>test123</domain:account_pw>
C:       <domain:name>domain-1</domain:name>
C:         <domain:name>nic </domain:name>
C:       <domain:name>domain-n</domain:name>
C:    </domain:delete>
C:   </delete>
C:       <clTRID>20101126093102</clTRID>
C:   </command>
C: </epp>
```

Server <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  epp-1.0.xsd">
S:    <response>
S:        <result code="1000">
```

```
S:           <msg>Command completed successfully</msg>
S:         </result>
S:          <resData>
S: <domain:chkData xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
S:          xsi:schemaLocation="urn:ietf:params:xml:ns:domain-1.0 domain-1.0.xsd">
S:         <domain:name res=”1”>domain-1 deleted</domain:name>
S:         <domain:name res=”0”>nic not in account</domain:name>
S:         <domain:name  res=”1”>domain-n deleted</domain:name>
S:  </domain:chkData>
S:       </resData>
S:           <clTRID> 20101126093102  </clTRID>
S:     </response>
S: </epp>
```

Note. Variants of response:

tag	*value*
<domain:name res=”1”>	“domain name” deleted
<domain:name res=”0”>	“domain name” not in account

Appendix A

A short presentation of EPP implementation in Perl
(*Client)*

In order to test the examples EPP implementation in Perl the .md registrar (partner) must:

- ✓ install in your system the modules from CPAN (Comprehensive Perl Archive Network):
  Net::EPP::Client
  Net::EPP::Frame
  XML::LibXML
  XML::XPath

- ✓ ask from NIC support the connect user id, password; account name, password;   and the perl scripts.