



.ngo|.ong EPP Acceptance Criteria

With optional DNSSEC – Version 1.1

Table of Contents

- 2.1 Starting the test 3
- 2.2. Session Management..... 3
 - 2.2.1 Start Session 3
 - 2.2.2 Authentication 3
 - 2.2.3 Domain Create 3
 - 2.2.4 Query Domains 4
 - 2.2.5 Domain Update 4
 - 2.2.6 Query Domains 4
 - 2.2.7 Create Name Server 4
 - 2.2.8 Query Name Servers 5
 - 2.2.9 Domain Create with DS Record..... 5
 - 2.2.10 Query Domains with DS Record..... 6
 - 2.2.11 Change Ownership 6
 - 2.2.12 Request Poll queue message (Status NGO) 6
 - 2.2.13 Ack Queued Message..... 7
 - 2.2.14 Request Poll queue message (Status ONG) 7
 - 2.2.15 Ack Queued Message..... 7
 - 2.2.15 End Session 8
 - 2.2.16 Create Domain with multiple DS Records..... 8
- 2.5.1.5 Update Domain - Adding Single DS Data 9
- 2.5.1.6 Update Domain – Changing DS Data 9
- 2.5.1.7 Update Domain – Adding Multiple DS Records 10
- 2.5.1.8 Update Domain – Remove Multiple DS Records 11
- 2.5.1.9 Update Domain – Remove Single DS Record (Update: Remove) 11
- 2.5.1.10 Update Domain – Adding and Removing Multiple DS Records 12
- 2.5.2 Client Error Handling in DNSSEC 13
 - 2.5.2.1 Correctly Handle 2306 Error Exception 13
 - 2.5.2.2 Correctly Handle 2303 Error Exception (Remove Single DS Record) 14
 - 2.5.2.3 Correctly Handle 2005 Error Exception (Adding Digest with space in between) 15

2. EPP Communications

2.1 Starting the test

Public Interest Registry Technical Support will contact the Registrar by telephone a few minutes before the scheduled start time, to provide final confirmation prior to the registrar commencing the OT&E test.

2.2. Session Management

2.2.1 Start Session

After making an initial connection to the Registry, the server shall reply with a greeting. A registrar must receive the greeting message before attempting authentication and/or other supplementary commands.

2.2.2 Authentication

After the initial greeting the Registrar client shall send the Login command to authenticate itself to the test registry with the following information:

Client ID: **ClientX**
Password: **foo-BAR2**

Verify that the following response is received:

```
<result code='1000'><msg lang='en-US'>Command completed successfully</msg>
```

2.2.3 Domain Create

Supply the following information in the Create Command:

Domain Name: **example.ngo**
Domain Server: **ns1.sample.com**
Domain Server: **ns2.sample.com**
Domain Registrant Contact ID: **OTE-C1**
Domain Admin Contact ID: **OTE-C2**
Domain Billing Contact ID: **OTE-C3**
Domain Technical Contact ID: **OTE-C4**
Domain Period (Years): **5**
Auth Info: **my secret**

Verify that the following response is received:

```
<validation:claimID>claimIDValue</validation:claimID>
```

As part of the NGO/ONG process registrants are required to go through verification. When creating a domain or running the change of ownership command registrars will be provided a Claim ID Value. Registrars are required to provide the Claim ID Value to their Registrants in the following manner.

```
https://BaseURL?d1= claimIDValue
```

2.2.4 Query Domains

Send **two separate domain info commands** for the following:

Domain Name: **example.ngo**

Domain Name: **example.ong**

Verify that the following response is received:

```
<result code='1000'><msg lang='en-US'>Command completed successfully</msg>
```

You will notice when you send the domain create command with either the ngo or ong, the system will create both names with the same information that was supplied during the create command.

2.2.5 Domain Update

Issue the Update command with the following data: Remove the contact, OTE-C1 from the domain example.ngo and add the contact OTE-C2 as the new admin contact.

Domain name: **example.ngo**

Remove Admin Contact ID: **OTE-C1**

Add Admin Contact ID: **OTE-C2**

Verify that the following response was received:

```
<result code='1000'><msg lang='en-US'>Command completed successfully</msg>
```

2.2.6 Query Domains

Send **two separate domain info commands** for the following:

Domain Name: **example.ngo**

Domain Name: **example.ong**

As shown any updates that you make to one domain will affect the other domain, aside from the DS Record and child hosts, which are unique objects.

2.2.7 Create Name Server

Supply the following to the host create command:

Host Name: **ns1.example.ngo**
IP Address: **192.168.10.11**

Verify that the following response was received:

```
<result code='1000'><msg lang='en-US'>Command completed successfully</msg>
```

2.2.8 Query Name Servers

Send **two separate host info commands** for the following:

Name Server: **ns1.example.ngo**
Name Server: **ns1.example.org**

Verify that the following responses are received:

```
<result code='1000'><msg lang='en-US'>Command completed successfully</msg>  
<result code='1000'><msg lang='en-US'>Object does not exist</msg>
```

You will notice that when you run the host info command for **ns1.host.org** you will receive an error stating that the host does not exist and this is because child hosts are unique objects for the domain name they are created for.

2.2.9 Domain Create with DS Record

Supply the following information in the create command:

Domain Name: **dsdomain1.ngo**
Domain Server: **ns1.sample.com**
Domain Server: **ns2.sample.com**
Domain Registrant Contact ID: **OTE-C1**
Domain Admin Contact ID: **OTE-C2**
Domain Billing Contact ID: **OTE-C3**
Domain Technical Contact ID: **OTE-C4**
Domain Period (Years): **5**
Auth Info: **my secret**
DS Data -
 Key Tag: **12345**
 Algorithm: **3**
 Digest Type: **1**
 Digest: **49FD46E6C4B45C55D4AC49FD46E6C4B45C55D4AC**

Verify that the following response is received:

```
<result code='1000'><msg lang='en-US'>Command completed successfully</msg>
```

2.2.10 Query Domains with DS Record

Send **two separate domain info commands** for the following:

Domain name: **dsdomain1.ngo**

Domain name: **dsdomain1.org**

Verify that the following response is received:

```
<result code='1000'><msg lang='en-US'>Command completed successfully</msg>
```

```
<result code='1000'><msg lang='en-US'>Command completed successfully</msg>
```

You will notice that the DS record does not carry over for the second name. The DS record only gets created for the domain that was part of the create command. You will need to supply a Domain Update with DS extension command to associate a DS record.

2.2.11 Change Ownership

Supply the following domain name to the Domain Update command to obtain a new Claim ID Value:

Domain Name: **example.ngo**

Verify that the following was provided:

```
<validation:claimID>claimIDValue</validation:claimID>
```

If a registrant sells their domain name to another registrant, they are required to verify themselves. Therefore, the change ownership command should be run and a new Claim ID Value shall be given to the new registrant.

2.2.12 Request Poll queue message (Status NGO)

Issue the poll command with the **op='request'** attribute to retrieve messages queued by the server.

Verify that the following response is received:

```
<response><result code="1301"><msg>Command completed successfully; ack to
dequeue</msg></result><msgQ count="5"
id="12345"><qDate>2000-06-08T22:00:00.0Z</qDate><msg>{"changeType":"update","na
me":"pollmsg.ngo","addedStatuses":["Ok"],"removedStatuses":["serverHold"],"authInfoU
pdated":false}</msg>
</msgQ><trID><clTRID>ABC-12345</clTRID><svTRID>54321-XYZ</svTRID></trI
D></response>
```

2.2.13 Ack Queued Message

Issue the poll command with the **op='ack'** attribute to acknowledge receipt of the first message, and remove it from the queue.

Verify that the following response is received:

```
<result code='1000'><msg lang='en-US'>Command completed
successfully</msg></result>
```

2.2.14 Request Poll queue message (Status ONG)

Issue the poll command with the **op='request'** attribute to retrieve messages queued by the server.

Verify that the following response is received:

```
<response><result code="1301"><msg>Command completed successfully; ack to
dequeue</msg></result><msgQ count="5"
id="12346"><qDate>2000-06-08T22:00:00.0Z</qDate><msg>{"changeType":"update","na
me":"pollmsg.ong","addedStatuses":["Ok"],"removedStatuses":["serverHold"],"authInfoU
pdated":false}</msg>
</msgQ><trID><clTRID>ABC-12345</clTRID><svTRID>54321-XYZ</svTRID></trI
D></response>
```

2.2.15 Ack Queued Message

Issue the poll command with the **op='ack'** attribute to acknowledge receipt of the first message, and remove it from the queue.

Verify that the following response is received:

```
<result code='1000'><msg lang='en-US'>Command completed
successfully</msg></result>
```

Poll queue messages now provide messaging of when a Server Status is placed on a domain name or when the registry deletes a domain. Two poll queue messages get created one for .ngo and the other for .ong.

2.2.15 End Session

For a Registrar client to end communications with the Registry, the Logout command is used with no arguments.

If successful, the Registry will send the following response and then end the session.

```
<result code='1500'><msg lang='en-US'>Command completed successfully; ending session</msg>
```

Seeded Objects:

Object	Owned By
Pollmsg.ngo	ClientX
Example.org	ClientX
Ns1.sample.com	ClientX
Ns2.sample.com	ClientX
OTE-C1	ClientX
OTE-C2	ClientX
OTE-C3	ClientX
OTE-C4	ClientX

(OPTIONAL) DNSSEC Acceptance Criteria

This section is for registrars are not yet DNSSEC certified to complete along with the .ngo|.ong criteria. Registrars who complete this section will become DNSSEC certified with PIR.

If you have any questions, please contact Technical Support at techsupport@pir.org

2.2.16 Create Domain with multiple DS Records

Create a new domain and associate two (2) Name Servers and four (4) Contacts to it by supplying the following element to the Create command.

Domain Name: **dsdomain2.ngo**
Domain Server: **ns1.sample.com**
Domain Server: **ns2.sample.com**
Domain Registrar Contact ID: **OTE-C1**
Domain Admin Contact ID: **OTE-C2**
Domain Billing Contact ID: **OTE-C3**
Domain Technical Contact ID: **OTE-C4**
Domain Period (Years): **5**

Auth Info: **my secret**

DS Data -

Key tag: **12346**

Algorithm: **3**

Digest Type: **1**

Digest: **49FD46E6C4B45C55D4AC49FD46E6C4B45C55D4AD**

DS Data -

Key tag: **12344**

Algorithm: **3**

Digest Type: **1**

Digest: **49FC66E6C4B45C56D4AC49FD46E6C4B45C55D4AE**

Verify that the following response is received:

```
<result code='1000'><msg lang='en-US'>Command completed successfully</msg>
```

2.5.1.5 Update Domain - Adding Single DS Data

Enter the following information to the Update command.

Domain Name: **example.ngo**

Add DS Data:

Key Tag: **12348**

Algorithm: **3**

Digest Type: **1**

Digest: **38EC35D5B3A34B44C39B38EC35D5B3A34B44C39B**

Verify that the following response is received:

```
<result code='1000'><msg lang='en-US'>Command completed successfully</msg>
```

2.5.1.6 Update Domain - Changing DS Data

Enter the following information to the Update command (changing Key Tag and Digest).

Update the DS Data in 2.5.1.5

Domain Name: **example.ngo**

Change DS Data:

Key Tag: **12349**

Algorithm: **3**

Digest Type: **1**

Digest: **65EF35D5B3A34B44C39B38EC35D5B3A34B44C39B**

Verify that the following response is received:

```
<result code='1000'><msg lang='en-US'>Command completed successfully</msg>
```

2.5.1.7 Update Domain - Adding Multiple DS Records

Enter the following information to the Update command to add optional key data

Domain Name: **example.ngo**

Add DS Data 2:

Key Tag: **12350**

Algorithm: **4**

Digest Type: **1**

Digest: **38AB35D5B3A34B44C39B38EC35D5B3A34B44C39B**

Add DS Data3:

Key Tag: **12351**

Algorithm: **3**

Digest Type: **1**

Digest: **38AA35D5B3A34B44C39B38EC35D5B3A34B44C39C**

Add DS Data4:

Key Tag: **12352**

Algorithm: **3**

Digest Type: **1**

Digest: **38AC35D5B3A34B44C39B38EC35D5B3A34B44C39D**

Add DS Data5:

Key Tag: **12353**

Algorithm: **4**

Digest Type: **2**

Digest:

651463E06F19D2FCA0215F129F54A2E0A4771EBBA37D8AB1103BCD279F0719E6

After this operation the domain will have effectively 5 sets of DS records as one has already been added to this domain in step 2.5.1.5 and updated in step 2.5.1.6.

Verify that the following response is received:

```
<result code='1000'><msg lang='en-US'>Command completed successfully</msg>
```

2.5.1.8 Update Domain – Remove Multiple DS Records

Enter the following set of additional DS records to the Update command.

Domain Name: **example.ngo**

DS Data:

Key Tag: **12350**

Algorithm: **4**

Digest Type: **1**

Digest: **38AB35D5B3A34B44C39B38EC35D5B3A34B44C39B**

DS Data:

Key Tag: **12351**

Algorithm: **3**

Digest Type: **1**

Digest: **38AA35D5B3A34B44C39B38EC35D5B3A34B44C39C**

This effectively removes above DS records from the domain that now has the following DS records

DS Data:

Key Tag: **12349**

Algorithm: **3**

Digest Type: **1**

Digest: **65EF35D5B3A34B44C39B38EC35D5B3A34B44C39B**

DS Data:

Key Tag: **12352**

Algorithm: **3**

Digest Type: **1**

Digest: **38AC35D5B3A34B44C39B38EC35D5B3A34B44C39D**

DS Data:

Key Tag: **12353**

Algorithm: **4**

Digest Type: **2**

Digest:

651463E06F19D2FCA0215F129F54A2E0A4771EBBA37D8AB1103BCD279F0719E6

2.5.1.9 Update Domain – Remove Single DS Record (Update: Remove)

Enter the following set of DS records information to the Update: Change command

Domain Name: **dsdomain1.ngo**

DS Data:

Key Tag: **12345**

Algorithm: **3**

Digest Type: **1**

Digest: **49FD46E6C4B45C55D4AC49FD46E6C4B45C55D4AC**

Verify that the following response is received:

```
<result code='1000'><msg lang='en-US'>Command completed successfully</msg>
```

Note: In order to uniquely identify DS records for removal, the 4 child elements, Key Data, Algorithm, Digest Type and Digest, must now all be sent with `<secDNS:rem>` command.

2.5.1.10 Update Domain – Adding and Removing Multiple DS Records

Add some DS records and remove some DS records from a domain using one transaction.

Domain Name: **example.ngo**

Add the following DS records to the domain using **Update:Add** command:

DS Data:

Key Tag: **12350**

Algorithm: **4**

Digest Type: **1**

Digest: **38AB35D5B3A34B44C39B38EC35D5B3A34B44C39B**

DS Data:

Key Tag: **12351**

Algorithm: **3**

Digest Type: **1**

Digest: **38AA35D5B3A34B44C39B38EC35D5B3A34B44C39B**

Remove the following DS records from the domain using **Update:Remove** command:

DS Data:

Key Tag: **12352**

Algorithm: **3**

Digest Type: **1**

Digest: **38AC35D5B3A34B44C39B38EC35D5B3A34B44C39D**

DS Data:

Key Tag: **12353**

Algorithm: **4**

Digest Type: **2**

Digest:
651463E06F19D2FCA0215F129F54A2E0A4771EBBA37D8AB1103BCD279F0719E6

So, effectively the domain will now have the following DS records:

DS Data:
Key Tag: **12349**
Algorithm: **3**
Digest Type: **1**
Digest: **65EF35D5B3A34B44C39B38EC35D5B3A34B44C39B**

DS Data:
Key Tag: **12350**
Algorithm: **4**
Digest Type: **1**
Digest: **38AB35D5B3A34B44C39B38EC35D5B3A34B44C39B**

DS Data:
Key Tag: **12351**
Algorithm: **3**
Digest Type: **1**
Digest: **38AA35D5B3A34B44C39B38EC35D5B3A34B44C39B**

Verify that the following response is received:
`<result code='1000'><msg lang='en-US'>Command completed successfully</msg>`

Note: Update command containing both Add and Remove commands must process the Remove first before processing the Add. If the add request fails due to invalid data, then the remove operation cannot be allowed to take place, even though the processing of the remove must actually take place first.

Enter the following information to the **Update:Remove (<secDNS:all>)**

Domain Name: **example.ngo**

This will remove all 3 above DS records, associated with this domain, as in section 2.5.1.10.
Note: EPP Server will process this command by deleting all DS records associated with the domain.

2.5.2 Client Error Handling in DNSSEC

2.5.2.1 Correctly Handle 2306 Error Exception

2306 "Parameter value policy error" -This response code must be returned when a server receives a command containing a parameter value that is syntactically valid, but semantically invalid due to local policy. For example, the server may support a subset of a range of valid protocol parameter values. The error value should be returned via an element in the EPP response.

Submit the following **Update:Add** command:

Domain Name: **example.ngo**

Change DS Data:

Key Tag: **12350**

Algorithm: **300**

Digest Type: **1**

Digest: **38AB35D5B3A34B44C39B38EC35D5B3A34B44C39B**

Verify that the following response is received:

```
<result code='2306'><msg lang='en-US'>Parameter value policy error</msg><value xmlns:oxrs='urn:afiliias:params:xml:ns:oxrs-1.0'><oxrs:xcp>2306:Parameter value policy error (alg: value min:0 max:255)</oxrs:xcp></value></result>
```

Note: Algorithm ID should be within a valid range.

2.5.2.2 Correctly Handle 2303 Error Exception (Remove Single DS Record)

"2303" Object does not exist - This response code must be returned when a server receives a command that is trying to Update, delete, renew and transfer commands on an object that is not found in the registry.

Submit the following Update:Remove command to remove DS record:

Domain Name: **example.ngo**

DS Data:

Key Tag: **54321**

Algorithm: **3**

Digest Type: **1**

Digest: **38AA35D5B3A34B44C39B38EC35D5B3A34B44C39B**

Verify that the following response is received:

```
<result code='2303'><msg lang='en-US'>Object does not exist</msg><value xmlns:oxrs='urn:afiliias:params:xml:ns:oxrs-1.0'><oxrs:xcp>2303:Could not find single DS record with keytag 54321. Ensure keytag exists and there is only a single DS Record on the domain</oxrs:xcp></value></result>
```

Note: This error is due to the fact that Update: Remove command is referring to a keytag, 54321, that does not exist in the registry.

2.5.2.3 Correctly Handle 2005 Error Exception (Adding Digest with space in between)

2005 "Parameter value syntax error" -- This response code MUST be returned when a server receives a command containing a parameter whose value is improperly formed. The error value SHOULD be returned via a <value> element in the EPP response.

Add the following DS records to the domain using **Update:Add command**:

Domain Name: **example.ngo**

Add DS Data:


Key Tag: **12355**

Algorithm: **4**

Digest Type: **2**

Digest: **C06D93103F046E056033CA1D47CCD31F60DC7CE8E1BF C381A1252879C98752EE**

Verify that the following response is received:

 A space

```
<result code='2005'><msg lang='en-US'>Parameter value syntax error</msg><value
xmlns:oxrs='urn:afiliat:params:xml:ns:oxrs-1.0'><oxrs:xcp>2005:Parameter value syntax error
(digest:C06D93103F046E056033CA1D47CCD31F60DC7CE8E1BF
C381A1252879C98752EE)</oxrs:xcp></value></result>
```

Note: This error is due to the fact that Digest value has a space, which is not allowed. As per RFC 4509, the format of the SHA-256 digest has been defined to be exactly 32 bytes (64 octets) which does not allow for spaces embedded within the string. Removing that space will allow the DS records to be successfully added to the domain.