DNSSEC Abridged Test
Extensible Provisioning Protocol (EPP) v1.9.2
.ORG Registrar Acceptance Criteria

Published March 13, 2014

Version 1.9.2

Technical Support:
techsupport@pir.org / +1.416.646.3306
http://www.pir.org

Contents

2.4 End Session

2.5 Completing the Test

Appendix A -Seeded Registry information

This document is made available to the registrars that have entered into Registry-Registrar Agreements with Public Interest Registry (PIR), manager of the registry of .ORG. The contents of this document are proprietary information of Afilias, Limited. This information may be used by recipient only for the purpose for which it was transmitted and shall be returned upon request to Afilias Limited or when no longer needed by recipient.

Afilias has made efforts to ensure the accuracy and completeness of the information in this document. However, Afilias makes no warranties of any kind (express or implied) with respect to the information contained herein. Afilias assumes no liability to any party for any loss or damage (whether direct or indirect) caused by any errors, omissions, or statements of any kind contained in this document. Afilias reserves the right to make changes to any information herein at any time, without further notice.

1. Introduction

1.1 Purpose

This document describes the basic operations that a Registrar's client application must perform to be accepted by the Registry. Each of the following sections describes the actions that the client must perform to demonstrate correct implementation of the Extensible Provisioning Protocol (EPP) v1.0 and interactions with the Registry. Registrars should have a detailed knowledge of the following internet RFCs before attempting the test:

*EPP RFC: 5730 (*http://www.rfc-editor.org/rfc/rfc5730.txt*)*
*EPP Domain Name Mapping RFC: 5731 (*http://www.rfc-editor.org/rfc/rfc5731.txt*)*
*EPP Host Mapping RFC: 5732 (*http://www.rfc-editor.org/rfc/rfc5732.txt*)*
*EPP Contact Mapping RFC: 5733 (*http://www.rfc-editor.org/rfc/rfc5733.txt*)*
*EPP Transport Over TCP RFC: 5734 (*http://www.rfc-editor.org/rfc/rfc5734.txt*)*
*EPP DNS Security Extension Mapping RFC: 5910 (*http://www.rfc-editor.org/rfc/rfc5910.txt*)*

The tests presented herein verify the correct interface with the Registry for standard Registrar operations. They do not cover all possible error and unusual conditions. The Registrar client application is responsible for correctly handling all unusual error conditions.

1.2 Formatting Conventions

Proper completion of the test requires that all commands and data must be entered exactly as given in this document. Any deviations will be considered a failure. The following items show the formatting conventions included in this document for required input and output values and for variable input and output responses.

Regular text in this format represents expected system input and output values that the client system will send to the server and that the server system will display in response to an action or actions provided by the Registrar. The following example illustrates an expected system output.

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

When **bold** text is located in Regular text, this represents a required input value that the Registrar must provide -the Registrar must enter the text exactly as shown. The following example illustrates the format for the required input values.

Domain Name: **dsdomain3.org**

Italicized text in output represents data returned from the server, which may or may not be the exact values represented in this document. It is the responsibility of the client to interpret these values properly and possibly reuse these for subsequent commands.

<domain:exDate>*2011-06-21T22:07:28.0Z*</domain:exDate>

1.3 Accounts

For the duration of the test, the Registrar will use a seeded test account, called ClientX. The Registrar will provide PIR Technical Support Group with a valid email address. Standard registry transfer notifications, processed by the registry during the initial test seeding (**see appendix for details**), will be sent to this e-mail address for Registrar reference. Upon the scheduling of a test, Afilias Technical Support Group will provide hostnames and port numbers for the Registrar's client connection.

1.4 Additional Requirements

Registry Operator will prime the Test Registry with data required to complete this test. Please refer to Appendix A if you wish to review this data. Do not attempt to enter this data into the Test Registry.

1.5 Successful Command & Test Completion

While performing this test, if the response to a command is not exactly as shown, then stop your test and contact PIR Technical support.

1.6 Passing the Test

The Registrar must complete the test perfectly (with no typographical errors and without breaking the sequence of operations) from start to finish within the allotted time.

1.7 Contact and Name Server Policy Requirements

There are certain policies that are enforced in the .ORG implementation of EPP:

A minimum of 4 contacts (including 1 Registrant and at least 1 of each Admin, Billing and Technical contacts) must be provided during the create domain transaction.

For the purpose of this test, all domains must be created with at least 2 name servers. Registrars may, however, when working with the "live" registry, create domains with fewer than 2 name servers, though DNS resolution depends upon a minimum of one (1) assigned name server. The use of at least two (2) valid nameservers is highly recommended.

2. EPP Communications

Registrar to Registry communications utilize the Extensible Provisioning Protocol (EPP) mapped over TCP (Transport Control Protocol). EPP commands are formulated using the Extensible Markup Language (XML). The Registrars' application client must utilize XML to send commands to the Registry and utilize an XML parser to interpret the server's responses. EPP itself relies exclusively upon user authentication for security. Additional security is provided by the use of Transport Layer Security (TLS), for session cryptography. Clients must communicate with the EPP server using a commercial or open source implementation of TLS, such as OpenSSL. Additional information concerning mapping EPP over TCP is available in 'RFC 5734 - Extensible Provisioning Protocol Transport Over TCP RFC'. Additional information concerning the TLS may be found in RFC 5246.

2.1 Starting the Test

PIR 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>
```

2.3 DNSSEC EPP Acceptance Criteria

2.3.1 Creation of objects and their updates

2.3.1.1 Create Domain with DS Record

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

Domain Name: **dsdomain1.org**
Domain Server: **ns1.sample.com**
Domain Server: **ns2.sample.com**
Domain Registrant Contact ID: **OTE-C5**
Domain Admin Contact ID: OTE-C5
Domain Billing Contact ID: **OTE-C5**

Domain Technical Contact ID: **OTE-C5**
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>

2.3.1.2 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 elements to the Create command.

Domain Name: **dsdomain2.org**
Domain Server: **ns1.sample.com**
Domain Server: **ns2.sample.com**
Domain Registrant Contact ID: **OTE-C5**
Domain Admin Contact ID: **OTE-C5**
Domain Billing Contact ID: **OTE-C5**
Domain Technical Contact ID: **OTE-C5**
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>

2.3.1.3 Create Domain without DS records

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

Domain Name: **dsdomain3.org**
Domain Server: **ns1.sample.com**
Domain Server: **ns2.sample.com**
Domain Registrant Contact ID: **OTE-C5**
Domain Admin Contact ID: **OTE-C5**
Domain Billing Contact ID: **OTE-C5**
Domain Technical Contact ID: **OTE-C5**
Domain Period (Years): **5**
Auth Info: **my secret**

Verify that the following response is received:

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

2.3.1.4 Query domain that has DS Data

Supply the following information to the Info command.

> Domain Name: **dsdomain1.org**

Verify that the following response is received:

> Domain Name: **dsdomain1.org**
> Client ID: **ClientX**
> Domain Status: **ok**
> Domain Contact (Registrant) ID: **OTE-C5**
> Domain Admin Contact: **OTE-C5**
> Domain Billing Contact: **OTE-C5**
> Domain Technical Contact: **OTE-C5**
> Domain Name Server: **ns1.sample.com**
> Domain Name Server: ns2.sample.com
> Auth Info: **my secret**
> Created By: **ClientX**
> Created Date: **2010-06-22T22:00:00.0Z**
> Expiration Date: **2015-06-22T22:00:00.0Z**
> Last Updated By: **ClientX**
> Key Tag: **12345**
> Algorithm: **3**
> Digest Type: **1**
> Digest: **49FD46E6C4B45C55D4AC49FD46E6C4B45C55D4AC**
> MaxSigLife: **3456000**

2.3.1.5 Update Domain- Adding Single DS Data

Enter the following information to the Update command.

> Domain Name: **dsdomain3.org**
> 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>

2.3.1.6 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: **dsdomain3.org**
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>

2.3.1.7 Update Domain – Adding Multiple DS Records

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

Domain Name: **dsdomain3.org**
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>

2.3.1.8 Update Domain – Remove Multiple DS Records

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

Domain Name: **dsdomain3.org**
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

Verify that the following response is received:

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

Note: Update:Remove command can be used to remove multiple DS records from a domain. In order to uniquely identify DS records for removal, all 4 child elements, Key Data, Algorithm, Digest Type and Digest associated with a DS record must now be sent with <secDNS:rem> command to remove that DS record.

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

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

Domain Name: **dsdomain1.org**
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.

2.3.1.10 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: **dsdomain3.org**
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.

2.3.1.11 Update Domain – Remove All DS Records

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

Domain Name: **dsdomain3.org**

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.

2.3.2 Client Error Handing in DNSSEC

2.3.2.1 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: **dsdomain3.org**
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:afilias: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.

2.3.2.2 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: **dsdomain3.org**
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:afilias: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.

2.3.2.3 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: **dsdomain3.org**
Add DS Data:
Key Tag: **12355**
Algorithm: **4**
Digest Type: **2**
Digest: **C06D93103F046E056033CA1D47CCD31F60DC7CE8E1BF C381A1252879C98752EE**

A space

Verify that the following response is received:

```
<result code='2005'><msg lang='en-US'>Parameter value syntax error</msg><value
xmlns:oxrs='urn:afilias: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.

2.4 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>
```

2.5 Completing the Test

At this point, contact PIR Technical Support at techsupport@pir.org or call +1.4166463306 and inform them that you have completed this test.

Appendix A - Seeded Registry information

The OT&E test requires the creation and manipulation of several EPP objects prior to the client's initial connection. Afilias Technical Support will perform the necessary operations before the client's initial connection. The data within this Appendix is included for informational purposes only.

***** Registrar: Do not attempt to enter this data into the Test Registry. *****

User

Registrar: **ClientX**
Password: **foo-BAR2**

Contacts

The Contact ID values for each of the seeded contacts are as follows:

Object	**Owned By**	**Notes**
OTE-C5	**ClientY**	
OTE-C6	**ClientX**	Auth Info: my secret ** This contact has pending transfer status, initiated by ClientY**
OTE-C7	**ClientX**	Auth Info: my secret ** This contact has pending transfer status, initiated by ClientY**

The seeded contacts use the following common values:

Contact Name: **Test Contact**
Contact Organization: **Example Corp. Inc**
Contact Address Street: **123 Example St.**
Contact Address Street: **Suite 100**
Contact Address City: **Anytown**
Contact Address State/Province: **Any Prov**
Contact Address Postal Code: **A1A1A1**
Contact Address Country: **CA**
Contact Voice: **+1.4165555555**
Contact Voice Extension: **1111**
Contact Fax: **+1.4165555556**
Contact Email: **jdoe@test.test**

The seeded hosts are as follows:

Object	**Owned By**	**Notes**
ns1.sample.com	**ClientX**	
ns2.sample.com	**ClientX**	

Seeded Domains:

Object	Owned By	Notes
dsdomain3.org	**ClientY**	Auth Info: **my secretY**, **OTE-C5** for all contact types
transfer1.org	**ClientX**	Auth Info: **my secretX**, **OTE-C6** for all contact types ** This domain has pending transfer status, initiated by ClientY**
transfer2.org	**ClientX**	Auth Info: **my secretX**, **OTE-C6** for all contact types ** This domain has pending transfer status, initiated by ClientY**

All seeded domains above use seeded name server values: **ns1.sample.com** and **ns2.sample.com**.