Extensible Provisioning Protocol (EPP) v1.0
Registrar Acceptance Criteria

Version 1.0.3

EXTENSIBLE PROVISIONING PROTOCOL OT&E
REGISTRAR ACCEPTANCE CRITERIA
VERSION 1.0

MAY 7, 2007

techsupport@afilias-grs.info / +1.416.646.3309

http://www.afilias-grs.info

Contents

1. Introduction

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) and interactions with the Registry. Registrars should have a detailed knowledge of the following internet RFCs before attempting the test:

Extensible Provisioning Protocol RFC 3070[1]
Extensible Provisioning Protocol Domain Name Mapping RFC 3731[2]
Extensible Provisioning Protocol Contact Mapping RFC 3733[3]
Extensible Provisioning Protocol Host Mapping RFC 3732[4]

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.

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: ***foo.AG***

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>2004-02-11T22:07:28.0Z</domain:exDate>

[1] Hollenbeck; Extensible Provisioning Protocol, IETF RFC, March 2004
[2] Hollenbeck; Extensible Provisioning Protocol Domain Name Mapping, IETF RFC, March 2004
[3] Hollenbeck; Extensible Provisioning Protocol Contact Mapping, IETF RFC, March 2004
[4] Hollenbeck; Extensible Provisioning Protocol Host Mapping, IETF RFC, March 2004

1.3 Accounts

To perform the tests, the applying Registrar is given an account, named ClientX. When scheduling a test, the Registry Technical Support Group will provide hostnames and port numbers for the Registrar's client to connect to.

1.4 Additional Requirements

Prior to taking this Test, the 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 Afilias technical support.

1.6 Passing the Test

The Registrar must complete the test perfectly (with no typos or breaking the sequence of operations) from start to finish, including the session keep alive portion in section 2.6, within the allotted time.

1.7 Contact and Name Server Policy Requirements

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

A minimum of 3 contacts (including 1 Registrant and at least 1 of each Admin 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 1 assigned name server.

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 cryptograpy. Clients must communicate with the EPP server using a commercial or open source implementation of TLS, such as OpenSSL. Additional information cornerning mapping EPP over TCP is available in the IETF RFC 3734 'Extensible Provisioning Protocol Transport Over TCP'[5]*. Addtional information concerning the TLS protocol may be found in RFC 2246*[6]*.*

[5] Hollenbeck; Extensible Provisioning Protocol Transport Over TCP, IETF RFC, March 2004.
[6] Dierks, Allen; The TLS ProtocolVersion 1.0, RFC 2246, January, 1999.

2.1 Session Management

2.1.1 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.

2.1.2 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>

2.1.3 Change Password

To change a Registrar's password, an additional field is required in the Login command. At this point, the client must logout (keep the connection open), then log back in, and pass the following information to the Login command.

> *Client ID:* ***ClientX***
> *Password:* ***foo-BAR2***
> *New Password:* ***bar-FOO2***

Verify that the following response is received:

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

2.2 Domain Name Operations

The following tests exercises EPP commands that revolve around Domain Name creation and manipulation.

2.2.1 Create Domain with 4 Contacts and 2 Name Servers

Create a new Domain and associate 2 Name Servers and 3 Contacts to it by supplying the following elements to the Create command.

> *Domain Name:* ***valid1.ag***
> *Domain Server:* ***ns1.valid.ag***
> *Domain Server:* ***ns2.valid.ag***
> *Domain registrant contact ID :* ***OTNE-C2***
> *Domain Admin contact ID :* ***OTNE-C3***
> *Domain Technical contact ID:* ***OTNE-C5***
> *Domain Billing Contact ID:* ***OTNE-C4***

Auth info: ***my secret***

Verify that the following response is received:

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

2.2.2 Create Domain with Minimum Registration Period (.AG)

Enter the following data to the Create command.

*Domain Name: 1**year.ag***
Domain Server: ***ns1.valid.ag***
Domain Server: ***ns2.valid.ag***
Domain registrant contact ID: ***OTNE-C2***
Domain Admin contact ID: ***OTNE-C3***
Domain Technical contact ID: ***OTNE-C5***
Domain Billing Contact ID: OTNE-C4
Domain Period (Years): ***1***
Auth info: ***my secret***

Verify that the following response is received:

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

<domain:exDate>2007-02-11T22:07:28.0Z</domain:exDate>

2.2.3 Create Domain with Minimum Registration Period (.MN)

Enter the following data to the Create command.

Domain Name: ***1year.mn***
Domain Server: ***ns1.valid.ag***
Domain Server: ***ns2.valid.ag***
Domain registrant contact ID: ***OTNE-C2***
Domain Admin contact ID: ***OTNE-C3***
Domain Technical contact ID: ***OTNE-C5***
Domain Period (Years): ***1***
Auth info: ***my secret***

Verify that the following response is received:

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

<domain:exDate>2006-02-11T22:07:28.0Z</domain:exDate>

2.2.4 Create Domain with Maximum Registration Period

Enter the following data to the Create command.

Domain Name: ***10years.ag***
Domain Server: ***ns1.valid.ag***

*Domain Server: **ns2.valid.ag***
*Domain registrant contact ID: **OTNE-C2***
*Domain Admin contact ID: **OTNE-C3***
Domain Technical contact ID: OTNE-C5
Domain Billing Contact ID: OTNE-C4
*Domain Period (Years): **10***
*Auth info: **my secret***

Verify that the following response is received:

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

<domain:exDate>2015-02-11T22:07:28.0Z</domain:exDate>

2.2.5 Create Domain with Maximum Number of Name Servers

Using the Create command, add the following Domain along with the thirteen Name Servers.

*Domain Name: **maxhost.ag***
*Domain registrant contact ID: **OTNE-C2***
*Domain Admin contact ID: **OTNE-C3***
*Domain Technical contact ID: **OTNE-C5***
Domain Billing Contact ID: OTNE-C4
*Name Server: **ns1.valid.ag***
*Name Server: **ns2.valid.ag***
*Name Server: **ns3.valid.ag***
*Name Server: **ns4.valid.ag***
*Name Server: **ns5.valid.ag***
*Name Server: **ns6.valid.ag***
*Name Server: **ns7.valid.ag***
*Name Server: **ns8.valid.ag***
*Name Server: **ns9.valid.ag***
*Name Server: **ns10.valid.ag***
*Name Server: **ns11.valid.ag***
*Name Server: **ns12.valid.ag***
*Name Server: **ns13.valid.ag***
*Auth info: **my secret***

Verify that the following response is received:

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

2.2.6Create Domain with Minimum Length Domain Name (2 Chars + .AG)

Using the Create command, pass the following data.

*Domain Name: **ab.ag***
*Domain Server: **ns1.valid.ag***
*Domain Server: **ns2.valid.ag***
*Domain registrant contact ID: **OTNE-C2***
*Domain Admin contact ID: **OTNE-C3***

Domain Technical contact ID: ***OTNE-C5***
Domain Billing Contact ID: OTNE-C4
Auth info: ***my secret***

Verify that the following response is received:

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

2.2.7 Create Domain with Maximum Length Domain Name (63 Chars + .AG)

Using the Create command, enter the following data.

Domain Name:
abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijk.ag
Domain Server: ***ns1.valid.ag***
Domain Server: ***ns2.valid.ag***
Domain registrant contact ID: ***OTNE-C2***
Domain Admin contact ID: ***OTNE-C3***
Domain Technical contact ID: ***OTNE-C5***
Domain Billing Contact ID: OTNE-C4
Auth info: ***my secret***

Verify that the following response is received:

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

2.2.8 Check Domain (Domain Not Available)

Use the Check command with the following data to determine that the domain is not available:

Domain Name: ***valid.ag***

Verify that the following response is received:

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

domain:name avail='0'

2.2.9 Check Domain (Domain Available)

Use the Check command with the following data to determine that the domain is available:

Domain Name: ***available.ag***

Verify that the following response is received:

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

domain:name avail='1'

2.2.10 Check Domain with Maximum Length Domain Name (Domain Not Available)

Provide the following information to the Check command.

> *Domain Name:*
> ***abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijk.ag***

Verify that the following response is received:

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

> *domain:name avail='0'*

2.2.11 Query Domain

The info command is used to retrieve information associated with a Domain. Enter the following information to the info command.

> *Domain Name:* ***infodomain.ag***

Verify that the following response is received:

> *Domain Name: infodomain.**ag***
> *Client ID: ClientX*
> *Domain Status: ok*
> *Domain Contact (Registrant) ID: OTNE-C2*
> *Domain Admin Contact: OTNE-C3*
> *Domain Billing Contact: OTNE-C4*
> *Domain Technical Contact: OTNE-C5*
> *Domain Name Server: ns1.infodomain.**ag***
> *Domain Name Server: ns2.infodomain.**ag***
> *Domain Host Server: ns1.infodomain.**ag***
> *Domain Host Server: ns2.infodomain.**ag***
> *Auth info: my secret*
> *Created By: ClientX*
> *Created Date: 1999-04-03T22:00:00.0Z*
> *Expiration Date: 2005-04-03T22:00:00.0Z*
> *Last Updated By: ClientX*
> .
> .

You will need the Admin Contact ID for section 2.2.21

2.2.12 Delete Domain

Issue the Delete command with the following data:

> *Domain Name:* ***deleteme.ag***

Verify that the following response is received:

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

2.2.13 Renew Domain

First, get the Expiration Date of the Domain by issuing the info command with the following data.

> *Domain Name:* ***renewable.ag***

Examine the Expiration Date returned from the previous command (output should be similar to the following).

> *Domain Expiration Date: 2009-04-03T22:00:00.0Z*

Issue the Renew command with the following data.

> *Domain Name:* ***renewable.ag***
> *Current Expiration Date: 2009-04-03 (returned in the previous info command)*
> *Domain Years Period:* ***5***

Verify the output so that the expected Expiration Date is correct.

> *Domain Name: renewable.**ag***
> *Expiration Date: 2013-04-03T22:00:00.0Z*

2.2.14 Renew Domain to Maximum Registration Period

Enter the following information to the Renew command.

> *Domain Name:* ***renewable.ag***
> *Current Expiration Date: 2013-04-03 (returned in the previous renew command)*
> *Domain Period (Years):* ***3***

Verify the change by issuing the info command and compare the output here.

> *Domain Name: renewable.**ag***
> *Expiration Date: 2017-04-03T22:00:00.0Z*

2.2.15 Transfer a Domain

Use the Transfer command with the op="request" attribute, with the following information to request a transfer of a domain

> *Domain Name:* ***transfer1.ag***
> *Auth info:* ***my secretY***

Verify that the following response is received:

<result code='1001'><msg lang='en-US'>Command completed successfully</msg>

2.2.16 Approve Domain Transfer

Another Registrar has made a transfer request for one of ClientX's domains. This section involves the approval of this transfer request. Check the status of the transfer using the Transfer command with the op="query" attribute and the following information:

Domain Name: ***transfer2.ag***
Auth info: ***my secretX***

Verify that the following response is received:

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

Transfer Status: pending

Approve the transfer by using the Transfer command with the op="approve" attribute and the following information:

Domain Name: ***transfer2.ag***
Auth info: ***my secretX***

Verify the following output:

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

2.2.17 Reject Domain Transfer

Another Registrar has made a transfer request for one of ClientX's domain's. This section involves the rejection of this transfer request. Reject the transfer by using the Transfer command with the op="reject" attribute and the following information:

Domain Name: ***transfer3.ag***
Auth Info: ***my secretX***

Verify that the following response is received:

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

2.2.18 Change Domain Name Servers

Enter the following information to the Update command.

Domain Name: ***infodomain.ag***
Remove Name Server: ***ns1.infodomain.ag***
Remove Name Server: ***ns2.infodomain.ag***
Add Name Server: ***ns1.valid.ag***
Add Name Server: ***ns2.valid.ag***

Verify that the following response is received:

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

2.2.19 Change Domain Contact

Issue the Update command with the following data. Remove the Admin Contact that was shown in section 2.2.11 and add a new Contact (this new contact already exists - it was seeded in the database by the registry operator prior to the test).

Domain Name: ***infodomain.ag***
Remove Admin Contact ID: XXXX-00 (get this ID from section 2.2.11)
Add Admin Contact ID: ***OTNE-C4***

Verify that the following response is received:

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

2.2.20 Change Domain Status

Change the status of a Domain by issuing the Update command with the following values.

Domain Name: ***infodomain.ag***
Add Domain Status: ***clientUpdateProhibited***

Verify that the following response is received:

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

2.3 Name Server Operations

The following tests exercise EPP commands that revolve around Domain Name Server creation and manipulation.

2.3.1 Create Name Server

Supply the following information to the Create command.

Host: ***ns1.newname.ag***
Host Address: ***192.168.10.11***

Verify that the following response is received:

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

2.3.2 Create Name Server with Maximum Length Host Name (63 Chars + "." + 63 Chars + ".AG")

Supply the following information to the Create command.

Host:
abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijk.abcdefghijkl mnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijk.ag
Host Address: ***192.168.10.12***

Verify that the following response is received:

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

2.3.3 Create Name Server with Maximum Allowable IPs

Supply the following information to the create command:

Host: ***ns2.newname.ag***
Host Address: ***192.168.10.1***
Host Address: ***192.168.10.2***
Host Address: ***192.168.10.3***
Host Address: ***192.168.10.4***
Host Address: ***192.168.10.5***
Host Address: ***192.168.10.6***
Host Address: ***192.168.10.7***
Host Address: ***192.168.10.8***
Host Address: ***192.168.10.9***
Host Address: ***192.168.10.10***
Host Address: ***192.168.10.11***
Host Address: ***192.168.10.12***
Host Address: ***192.168.10.13***

Verify that the following response is received:

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

2.3.4 Create Name Server (Foreign Registry)

Supply the following to the create command:

Host: ***ns1.valid.com***

Verify that the following response is received:

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

2.3.5 Create External Name Server (.GI)

This command will create a name server in the .GI zone and will be owned by ClientX despite the fact that ClientX does not have permission to create objects in the .GI zone. This is known as an external host.

Pass the following information to the create command

Host Name: ns1.externalhost.gi

Verify that the following response was received

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

2.3.6 Query External Name Server (.GI)

This command will show .GI name server owned by ClientY even though ClientY does not have *permissions to create objects in the .GI registry.*

Pass the following information to the info command

> *Host Name: ns1.external2internal.gi*

Verify that the following response was received

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

> *Host Name: ns1.external2internal.gi*
> *Client ID: ClientY*
> *Created By: ClientY*
> *Created Date: 2004-05-07T13:17:59.0Z*
> *Status: Linked*

2.3.7 Convert External Host to Internal Host

This command will create a name server that already exists as an external host and convert it to an internal host thereby changing the ownership.

Pass the following information to the create command

> *Host Name: ns1.external2internal.gi*
> *IP Address: 123.123.123.123*

Verify that the following response was received

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

2.3.8 Query Host

This command will show that the name server created in the previous step is now an internal host and owned by ClientX

Pass the following information to the info command

> *Host Name: ns1.external2internal.gi*

Verify that the following response was received

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

> *Host Name: ns1.external2internal.gi*
> *IP Address: 123.123.123.123*
> *Client ID: ClientX*
> *Created By: ClientY*
> *Created Date: 2004-05-07T13:26:44.0Z*
> *Last Updated: 2004-05-07T13:30:54.0Z*
> *Status: OK, linked*

2.3.9 Delete Internal Host

This command will show that if a host was originally created as an external host and that host is linked to existing domains as a name server, when that host is deleted it reverts back to being an external host owned by the admin user.

Pass the following information to the delete command

Host Name: ns1.external2internal.gi

Verify that the following response was received

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

2.3.10 Query External Host

This command will show that the name server still exists as an external host and is now owned by admin

Host Name: ns1.external2internal.gi

Verify that the following response was received

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

Host Name: ns1.external2internal.gi
Client ID: admin
Created By: ClientY
Created Date: 2004-05-07T14:42:58.0Z
Last Updated: 2004-05-07T14:45:32.0Z
Status: OK, linked

2.3.11 Check Name Server (Name Server Known)

Use the Check command to check for a Name Server.

Host Name: ns1.valid.ag

Verify that the following response is received:

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

<host:name avail='0'>

2.3.12 Check Name Server (Name Server Unknown)

Enter the following data to the Check command.

Host Name: ns1.available.ag

Verify that the following response is received:

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

<host:name avail='1'>

2.3.13 Query Name Server

Supply the following values to the info command.

Host Name: ***ns1.valid.ag***

Verify that the following response is received:

Host Name: ns1.valid.ag
Client ID: ClientX
Host IP Address: 192.168.10.11
Created By: ClientX
Created Date: 1999-04-03T22:00:00.0Z
Client Trans ID: 11AA
Server Trans ID: 22BB
Status: ok

2.3.14 Delete Name Server

Use the Delete command with the following values.

Host Name: ***ns1.delns.ag***

Verify that the following response is received.

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

2.3.15 Change Name Server (Add IP Address)

Supply the following information to the Update command.

Host Name: ***ns12.valid.ag***
Add IP Address: ***192.1.2.3***

Verify that the following response is received.

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

2.3.16 Change Name Server (Remove IP Address)

Supply the following information to the Update command.

Host Name: ***ns12.valid.ag***
Remove IP Address: ***192.1.2.3***

Verify that the following response is received.

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

2.4 Contact Operations

The following tests exercises EPP commands that revolve around Contact creation and manipulation.

2.4.1 Create Contact

Supply the following information to the Create command.

> *Contact ID:* ***OTNE-C6***
> *Contact Name:* ***John Doe***
> *Contact Organization:* ***Example Corp. Inc***
> *Contact Address Street1:* ***123 Example St.***
> *Contact Address Street2:* ***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@valid.ag***
> *Auth info:* ***my secret***

Verify that the following response is received:

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

The contact id submitted with this create command will be necessary for the completion of sections 2.4.2, 2.4.4, 2.4.9 and 2.4.10.

2.4.2 Check Contact (Contact Known)

Use the Check command with the following arguments.

> *ID: XXXX-00 (use the Contact ID from section 2.4.1)*

Verify that the following response is received:

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

> *<contact:id avail='0'>*

2.4.3 Check Contact (Contact Unknown)

Use the Check command with the following arguments.

> *ID:* ***OTNE-C8***

Verify that the following response is received:

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

<contact:id avail='1'>

2.4.4 Query Contact

Supply the following information to the info command. Use the ID created in section 2.4.1.

ID: XXXX-00 (use the ContactID from the contact created in 2.4.1)

Verify that the following response is received:

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

ContactID: XXXX-00
Contact Name: John Doe
Contact Organization: Example Corp, Inc
Contact Address Street1: 123 Example St.
Contact Address Street2: 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@valid.ag
Auth info: my secret
Status: ok

2.4.5 Transfer Contact

This section tests the client's ability to request the transfer of a contact owned by another registrar. Please note that this contact was seeded in the database by Afilias prior to the start of the test. Supply the following information to the Transfer command with the op="request" attribute and the following information.

ID: ***OTNE-C7***
Auth info: ***my secret***

Verify that the following response is received:

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

2.4.6 Query Contact Transfer Status

Use the Transfer command's op="query" attribute, along with the following information.

ID: ***OTNE-C7***

Auth info: ***my secret***

Verify that the following response is received:

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

Contact Transfer Status: pending

2.4.7 Approve Contact Transfer

Another Registrar has an outstanding Transfer Request of one of ClientX's Contacts. This section involves the approval of the transfer request. Supply the following information to the Transfer command with the op="approve" attribute.

ID: ***OTNE-C1-approve***
Auth info: ***my secret***

Verify that the following response is received:

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

2.4.8 Reject Contact Transfer

Another Registrar has an outstanding Transfer Request of one of ClientX's Contacts. This section involves the rejection of the transfer request. Supply the following information to the Transfer command with the op="reject" attribute.

ID: ***OTNE-C1-reject***
Auth info: ***my secret***

Verify that the following response is received:

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

2.4.9 Change Contact (Change Element)

Supply the following information to the Update command. Use the ID created in section 2.4.1.

ID: XXXX-00 (use the ID from the contact created in 2.4.1)
Contact Name: ***Mr. Contact***

Verify that the following response is received:

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

2.4.10 Change Contact (Remove Element)

Supply the following information to the Update command. To remove a value, overwrite it as a NULL value. Use the ID created in section 2.4.1.

ID: XXXX-00 (use the ID from the contact created in 2.4.1)

Contact Fax:

Verify that the following response is received:

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

2.4.11 Delete Contact

First, create a simple Contact by supplying the following information to the Create command:

Contact ID: ***OTNE-C8***
Contact Name: ***Delete Me***
Contact Organization: ***Example Corp. Inc***
Contact Address Street1: ***123 Example St.***
Contact Address Street2: ***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@valid.ag***
Auth info: ***my secret***

Verify that the following response is received:

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

Now delete the Contact by supplying the following information to the Delete command.

ID: ***OTNE-C8***

Verify that the following response is received:

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

2.5 Client Error Handing

The following section exercises the client's ability to correctly handle common EPP exceptions. The client should remain connected to the Test Registry despite the receipt of exceptions. A definition of each exception code is provided.

2.5.1 Correctly Handle 2003 Exception

2003 "Required parameter missing" This response code MUST be returned when a server receives a command for which a required parameter value has not been provided.

Submit the following using the create command (do NOT submit the auth info):

Domain Name: ***exception.ag***
Domain Server: ***ns1.valid.ag***
Domain Server: ***ns2.valid.ag***
Domain registrant contact ID: ***OTNE-C2***
Domain Admin contact ID: ***OTNE-C3***
Domain Technical contact ID: ***OTNE-C5***

Verify that the following response is received:

<result code='2003'><msg lang='en-US'>Required parameter missing</msg>

2.5.2 Correctly Handle 2005 Exception

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 an element in the EPP response.

Submit the following using the create command:

Domain Name: ***-*iaminvalid.ag***
Domain Server: ***ns1.valid.ag***
Domain Server: ***ns2.valid.ag***
Domain registrant contact ID: ***OTNE-C2***
Domain Admin contact ID: ***OTNE-C3***
Domain Technical contact ID: ***OTNE-C5***
Auth info: ***my secret***

Verify that the following response is received:

<result code='2005'><msg lang='en-US'>Parameter value syntax error</msg>

2.5.3 Correctly Handle 2306 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 using the create command:

Domain Name: ***exception.ag***
Domain Server: ***ns1.valid.ag***
Domain Server: ***ns2.valid.ag***
Domain registrant contact ID: ***OTNE-C2***
Domain Admin contact ID: ***OTNE-C3***
Domain Technical contact ID: ***OTNE-C5***
Domain Billing Contact ID: ***OTNE-C6***
Domain Period (Years): ***99***
Auth info: ***my secret***

Verify that the following response is received:

<result code='2306'><msg lang='en-US'>Parameter value policy error</msg>

2.5.4 Correctly Handle 2002 Exception

2002 "Object COMMAND USE ERROR" This response code MUST be returned when a server receives a command that is properly formed, but can not be executed due to a sequencing or context error.

Submit the following using the renew command:

Domain Name: ***infodomain.ag***
Expiration Date: ***2011-04-03***

Verify that the following response is received:

<result code='2002'><msg lang='en-US'>Command use error</msg>

2.5.5 Correctly Handle 2303 Exception

2303 "Object does not exist" This response code MUST be returned when a server receives a command to transform an object that does not exist in the repository.

Submit the following using the create command:

Domain Name: ***exception.ag***
Domain Server: ***ns1.valid.ag***
Domain Server: ***ns2.valid.ag***
Domain registrant contact ID: ***OTNE-C99***
Domain Admin contact ID: ***OTNE-C3***
Domain Technical contact ID: ***OTNE-C5***
Domain Billing Contact ID: ***OTNE-C6***
Domain Period (Years): ***2***
Auth info: ***my secret***

Verify that the following response is received:

<result code='2303'><msg lang='en-US'>Object does not exist</msg>

2.5.6 Correctly Handle 2305 Exception

2305 "Object association prohibits operation" This response code MUST be returned when a server receives a command to transform an object that can not be completed due to dependencies on other objects that are associated with the target object. For example, a server MAY disallow commands while an object has active associations with other objects.

Submit the following to the delete command:

Contact ID: ***OTNE-C2***

Verify that the following response is received:

<result code='2305'><msg lang='en-US'>Object association prohibits operation</msg>

2.5.7 Correctly Handle 2201 Exception

2201 "Authorization error" This response code MUST be returned when a server notes a client authorization error when executing a command. This error is used to note that a client lacks privileges to execute the requested command.

Submit the following to the delete command:

Domain Name: ***transfer2.ag***

Verify that the following response is received:

<result code='2201'><msg lang='en-US'>Authorization error</msg>

2.6 Empty Element Commands

This section exercise the client's ability to utilize commands that must represented as empty elements, with no child objects.

2.6.1 Keep Session Alive

For this test, the client must keep the current session open to the Registry for at least 30 minutes before proceeding to Section 2.6.2. Use the Hello command at intervals under 10 minutes to maintain client connectivity.

2.6.2 Request Message Queue Information

Clients may use the poll command to retrieve messages queued by the server. Issue the poll command with the op=request attribute to retrieve queue information, and the first message within the queue.

Verify that the following response is received:

<response><result code='1301'><msg lang='en-US'>Command completed successfully; ack to dequeue</msg></result><msgQ count='48' id='43'><msg lang='en-US'>Transfer Requested.</msg>

Note: the value returned for 'id' will be necessary for section 2.6.3

2.6.3 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><msgQ count='47' id='45'/>

2.7 End Session

For a client Registrar 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>

At this point, contact Afilias Technical Support at techsupport@afilias-grs.net or +1.4166463309 and inform them 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 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. ****

Registrar: ClientX
Password: foo-BAR2

Contacts: The seeded contacts use the following common values:

Contact ID: XXXX-00
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@valid.AG
Auth info: my secret

The unique contact ID values for each of the seven seeded contacts are as follows:

ID: OTNE-C7
Owned by ClientY

ID: OTNE-C1-approve
Owned by ClientX
This contact has pending transfer status, initiated by ClientY

ID: OTNE-C1-reject
Owned by ClientX
This contact has pending transfer status, initiated by ClientY

ID: OTNE-C2
Owned by ClientX

ID: OTNE-C3
Owned by ClientX

ID:OTNE-C4
Owned by ClientX

	ID:OTNE-C5
	Owned by ClientX
	ID:OTNE-C10
	Owned by ClientX
Domain:	delns.ag
	Owned by ClientX
Host:	ns1.delns.ag
	Owned by ClientX
Domain:	valid.ag
	Owned by ClientX
Host:	
	ns1.valid.AG 192.169.2.11 Owned by ClientX
	ns2.valid.AG 192.169.2 12 Owned by ClientX
	ns3.valid.AG 192.169.2.13 Owned by ClientX
	ns4.valid.AG 192.169.2.14 Owned by ClientX
	ns5.valid.AG 192.169.2.15 Owned by ClientX
	ns6.valid.AG 192.169.2.16 Owned by ClientX
	ns7.valid.AG 192.169.2.17 Owned by ClientX
	ns8.valid.AG 192.169.2.18 Owned by ClientX
	ns9.valid.AG 192.169.2.19 Owned by ClientX
	ns10.valid.AG 192.169.2.20 Owned by ClientX
	ns11.valid.AG 192.169.2.21 Owned by ClientX
	ns12.valid.AG 192.169.2.22 Owned by ClientX
	ns13.valid.AG 192.169.2.23 Owned by ClientX
Domain:	infodomain.ag
	Owned by ClientX
Hosts:	ns1.infodomain.ag
	ns2.infodomain.ag
Domain:	external2internal.ag
	Owned by ClientX
Host:	ns1.external2internal.ag
	Owned by ClientY
Domain:	externalhost.ag
	Owned by ClientY
	Name server ns1.external2internal.AG
Domain:	authinfo.ag
	Owned by ClientX2

Domain Contact (Registrant) ID: OTNE-C1
Domain Admin Contact: OTNE-C2
Domain Technical Contact: OTNE-C3
Domain Billing Contact: OTNE-C4

Domain: deleteme.ag
Owned by ClientX

Domain: renewable.ag
Owned by ClientX

Domain: newname.ag
Owned by ClientX

Domain: transfer1.ag
Auth info: my secretY
** Owned by ClientY **

Domain: transfer2.ag
** This domain has pending transfer status, initiated by ClientY**
Auth Info: my secretX
Owned by ClientX

Domain: transfer3.ag
This domain has pending transfer status, initiated by ClientY
Auth ID: my secretX
Owned by ClientX