

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



May 2008

v1.0

**.ME Support**  
Web: <http://www.registry.me>  
E-mail: [support@registry.me](mailto:support@registry.me)  
Telephone: +1.416.619.3037

## Contents

1.	Introduction .....	4
1.1.	Purpose.....	4
1.2.	Formatting Conventions .....	4
1.3.	Accounts .....	4
1.4.	Additional Requirements .....	5
1.5.	Successful Command & Test Completion.....	5
1.6.	Passing the Test .....	5
1.7.	Contact and Name Server Policy Requirements.....	5
2.	EPP Communications .....	5
2.1.	Starting the Test.....	6
2.2.	Session Management .....	6
2.2.1	Start Session.....	6
2.2.2	Authentication.....	6
2.2.3	Change Password.....	6
2.3.	Contact Operations .....	6
2.3.1	Check Contact (Contact Available) .....	6
2.3.2	Create Contact.....	7
2.3.3	Check Contact (Contact Not Available).....	7
2.3.4	Query Contact.....	7
2.3.5	Transfer Contact.....	8
2.3.6	Query Contact Transfer.....	8
2.3.7	Approve Contact Transfer .....	8
2.3.8	Reject Contact Transfer .....	9
2.3.9	Change Contact (Change Element) .....	9
2.3.10	Change Contact (Remove Element) .....	9
2.3.11	Delete Contact.....	9
2.4.	Name Server Operations .....	10
2.4.1	Check Name Server (Name Server Available).....	10
2.4.2	Create Name Server .....	10
2.4.3	Check Name Server (Name Server Not Available) .....	10
2.4.4	Create Name Server (Foreign Registry).....	10
2.4.5	Query Name Server .....	10
2.4.6	Change Name Server (Add IP Address, Check Status).....	11
2.4.7	Change Name Server (Remove IP Address) .....	11
2.5.	Domain Name Operations.....	11
2.5.1	Check Domain (Domain Available for Registration) .....	11
2.5.2	Create Domain .....	12
2.5.3	Check Domain (Domain Not Available for Registration) .....	12
2.5.4	Query Domain .....	12
2.5.5	Query Domain with Trademark .....	13
2.5.6	Renew Domain.....	13
2.5.7	Change Domain Name Servers .....	14
2.5.8	Change Domain Contact.....	14
2.5.9	Change Domain Status .....	14
2.5.10	Transfer Domain.....	14
2.5.11	Approve Domain Transfer .....	15
2.5.12	Reject Domain Transfer .....	15
2.5.13	Delete Domain.....	15
2.6.	Client Error Handling.....	16

2.6.1	Correctly Handle 2003 Exception.....	16
2.6.2	Correctly Handle 2005 Exception.....	16
2.6.3	Correctly Handle 2306 Exception.....	16
2.6.4	Correctly Handle 2002 Exception.....	17
2.6.5	Correctly Handle 2303 Exception.....	17
2.6.6	Correctly Handle 2305 Exception.....	18
2.6.7	Correctly Handle 2201 Exception.....	18
2.7.	Efficiency of Client Session Management.....	18
2.7.1	Keep Session Alive .....	18
2.7.2	Request Message Queue Information.....	18
2.7.3	Ack Queued Message .....	19
2.8.	End Session .....	19
2.9.	Completing the Test .....	19
	Appendix A - Seeded Registry information .....	20

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

Afiliias has made efforts to ensure the accuracy and completeness of the information in this document. However, Afiliias makes no warranties of any kind (express or implied) with respect to the information contained herein. Afiliias 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. Afiliias 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:

*Extensible Provisioning Protocol (EPP) RFC: 3730*  
*Extensible Provisioning Protocol Domain Name Mapping RFC: 3731*  
*Extensible Provisioning Protocol Host Mapping RFC: 3732*  
*Extensible Provisioning Protocol Contact Mapping RFC: 3733*  
*Extensible Provisioning Protocol Transport over TCP RFC: 3734*

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: **example.me**

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>2007-01-01T22:07:28.0Z</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 .ME Support with a valid email address. Standard registry transfer

notifications, processed by the Registry during the initial test seeding (see Appendix A for details), will be sent to this e-mail address for Registrar reference. Upon the scheduling of a test, .ME Support will provide hostnames and port numbers for the Registrar's client connection.

#### **1.4. Additional Requirements**

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 .ME 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 ME implementation of EPP:

A minimum of four (4) contacts - including one (1) Registrant and at least one (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 two (2) name servers. Registrars may, however, when working with the "live" registry, create domains with fewer than two (2) name servers, though DNS resolution depends upon a minimum of two (2) assigned name servers.

### **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 Registrar's 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 3734 (Extensible Provisioning Protocol Transport over TCP). Additional information concerning the TLS may be found in RFC 2246.

## 2.1. Starting the Test

.ME 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 Change Password

To change a Registrar's password, an additional field is required in the Login command. At this point, the client must log out, then log in again, 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>
```

## 2.3. Contact Operations

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

### 2.3.1 Check Contact (Contact Available)

Use the Check command with the following argument.

ID: **OTE-C5**

Verify that the following response is received:

```
<result code='1000'><msg lang='en-US'>Command completed successfully</msg>  
<contact.id avail='1'>
```

### 2.3.2 Create Contact

Supply the following information to the Create command.

```
Contact ID: OTE-C5  
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@test.me  
Auth Info: my secret
```

Verify that the following response is received:

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

### 2.3.3 Check Contact (Contact Not Available)

Use the Check command with the following argument.

```
ID: OTE-C5
```

Verify that the following response is received:

```
<result code='1000'><msg lang='en-US'>Command completed successfully</msg>  
<contact.id avail='0'>
```

### 2.3.4 Query Contact

Supply the following information to the Info command.

```
ID: OTE-C5
```

Verify that the following response is received:

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

Contact ID: OTE-C5  
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@test.me  
Auth Info: my secret  
Status: ok

### 2.3.5 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 Test Registry by .ME Support 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: **OTE-C6**  
Auth Info: **my secret**

Verify that the following response is received:

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

### 2.3.6 Query Contact Transfer

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

ID: **OTE-C6**  
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>
```

### 2.3.7 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: **OTE-C7**  
Auth Info: **my secret**

Verify that the following response is received:

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

### 2.3.8 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: **OTE-C8**  
Auth Info: **my secret**

Verify that the following response is received:

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

### 2.3.9 Change Contact (Change Element)

Supply the following information to the Update command.

ID: **OTE-C5**  
Contact Name: **Jane Smith**

Verify that the following response is received:

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

### 2.3.10 Change Contact (Remove Element)

Supply the following information to the Update command. To remove a value, overwrite it as a NULL value.

ID: **OTE-C5**  
Contact Fax:

Verify that the following response is received:

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

### 2.3.11 Delete Contact

Delete the Contact created in Section 2.3.2 by supplying the following information to the Contact Delete command.

ID: **OTE-C5**

Verify that the following response is received:

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

## 2.4. Name Server Operations

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

### 2.4.1 Check Name Server (Name Server Available)

Supply the following data to the Check command.

Host Name: **ns5.example.me**

Verify that the following response is received:

```
<result code='1000'><msg lang='en-US'>Command completed successfully</msg>  
<host:name avail='1'>
```

### 2.4.2 Create Name Server

Supply the following information to the Create command.

Host Name: **ns5.example.me**  
Host Address: **192.168.10.11**

Verify that the following response is received:

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

### 2.4.3 Check Name Server (Name Server Not Available)

Use the Check command to check for the Name Server.

Host Name: **ns5.example.me**

Verify that the following response is received:

```
<result code='1000'><msg lang='en-US'>Command completed successfully</msg>  
<host:name avail='0'>
```

### 2.4.4 Create Name Server (Foreign Registry)

Supply the following to the Create command:

Host Name: **ns1.example.info**

Verify that the following response is received:

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

### 2.4.5 Query Name Server

Supply the following to the Info command.

Host Name: **ns5.example.me**

Verify that the following response is received:

Host Name: ns5.example.me  
Client ID: ClientX  
Host IP Address: 192.168.10.11  
Created By: ClientX  
Created Date: 2007-04-03T22:00:00.0Z  
Client Trans ID: 11AA  
Server Trans ID: 22BB  
Status: ok

#### **2.4.6 Change Name Server (Add IP Address, Check Status)**

Supply the following information to the Update command.

Host Name: **ns5.example.me**  
Add IP Address: **192.168.12.13**

Verify that the following response is received.

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

#### **2.4.7 Change Name Server (Remove IP Address)**

Supply the following information to the Update command.

Host Name: **ns5.example.me**  
Remove IP Address: **192.168.12.13**

Verify that the following response is received.

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

### **2.5. Domain Name Operations**

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

#### **2.5.1 Check Domain (Domain Available for Registration)**

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

Domain Name: **domain.me**

Verify that the following response is received:

```
<result code='1000'><msg lang='en-US'>Command completed successfully</msg>  
<domain:name avail='1'>
```

## 2.5.2 Create Domain

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: **domain.me**  
Domain Server: **ns1.example.me**  
Domain Server: **ns2.example.me**  
Domain Registrant Contact ID: **OTE-C1**  
Domain Admin Contact ID: **OTE-C2**  
Domain Billing Contact ID: **OTE-C3**  
Domain Technical Contact ID: **OTE-C4**  
Auth Info: **my secret**

Verify that the following response is received:

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

## 2.5.3 Check Domain (Domain Not Available for Registration)

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

Domain Name: **domain.me**

Verify that the following response is received:

```
<result code='1000'><msg lang='en-US'>Command completed successfully</msg>  
<domain:name avail='0'>
```

## 2.5.4 Query Domain

Supply the following information to the Info command.

Domain Name: **domain.me**

Verify that the following response is received:

Domain Name: domain.me  
Client ID: ClientX  
Domain Status: ok  
Domain Contact (Registrant) ID: OTE-C1  
Domain Admin Contact: OTE-C2  
Domain Billing Contact: OTE-C3  
Domain Technical Contact: OTE-C4  
Domain Name Server: ns1.example.me  
Domain Name Server: ns2.example.me  
Auth Info: my secret  
Created By: ClientX  
Created Date: 2007-04-03T22:00:00.OZ  
Expiration Date: 2012-04-03T22:00:00.OZ

Last Updated By: ClientX

### 2.5.5 Query Domain with Trademark

Supply the following information to the Info command.

Domain Name: **trademark.me**

Verify that the following response is received:

Domain Name: trademark.me  
Client ID: ClientX  
Domain Status: ok  
Domain Contact (Registrant) ID: OTE-C1  
Domain Admin Contact: OTE-C2  
Domain Billing Contact: OTE-C3  
Domain Technical Contact: OTE-C4  
Domain Name Server: ns1.example.me  
Domain Name Server: ns2.example.me  
Auth Info: my secret  
Created By: ClientX  
Created Date: 2007-04-03T22:00:00.0Z  
Expiration Date: 2012-04-03T22:00:00.0Z  
Last Updated By: ClientX  
IPR Name: Test Trademark  
IPR Applied: 2000-01-01  
IPR Registered: 2002-01-01  
IPR Number: 998877  
IPR CC Locality: CA

### 2.5.6 Renew Domain

First, obtain the Expiration Date of the domain by issuing the Info command with the following data.

Domain Name: **renew.me**

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

Domain Expiration Date: 2007-04-03T22:00:00.0Z

Issue the Renew command with the following data.

Domain Name: **renew.me**  
Current Expiration Date: **2007-04-03** (returned in the previous Info command)  
Domain Years Period: **5**

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

Domain Name: **renew.me**  
Expiration Date: *2012-04-03T22:00:00.0Z*

### 2.5.7 Change Domain Name Servers

Enter the following information to the Update command.

Domain Name: **domain.me**  
Remove Name Server: **ns1.example.me**  
Remove Name Server: **ns2.example.me**  
Add Name Server: **ns3.example.me**  
Add Name Server: **ns4.example.me**

Verify that the following response is received:

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

### 2.5.8 Change Domain Contact

Issue the Update command with the following data: Remove the Admin Contact 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: **domain.me**  
Remove Admin Contact ID: **OTE-C2**  
Add Admin Contact ID: **OTE-C9**

Verify that the following response is received:

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

### 2.5.9 Change Domain Status

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

Domain Name: **domain.me**  
Add Domain Status: **clientUpdateProhibited**

Verify that the following response is received:

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

### 2.5.10 Transfer Domain

Use the Transfer command with the op='request' attribute, with the following information to request a transfer of a domain from ClientY.

Domain Name: **transfer1.me**  
Auth Info: **my secretY**

Verify that the following response is received:

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

### 2.5.11 Approve Domain Transfer

Another Registrar (ClientY) 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.me**  
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.me**  
Auth Info: **my secretX**

Verify the following output:

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

### 2.5.12 Reject Domain Transfer

Another Registrar (ClientY) has made a transfer request for one of ClientX's domains. 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.me**  
Auth Info: **my secretX**

Verify that the following response is received:

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

### 2.5.13 Delete Domain

Issue the Delete command with the following data:

Domain Name: **domain.me**

Verify that the following response is received:

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

## 2.6. Client Error Handling

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.

### 2.6.1 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 a value for auth info):

```
Domain Name: exception.me  
Domain Server: ns1.example.me  
Domain Server: ns2.example.me  
Domain Registrant Contact ID: OTE-C1  
Domain Admin Contact ID: OTE-C2  
Domain Billing Contact ID: OTE-C3  
Domain Technical Contact ID: OTE-C4
```

Verify that the following response is received:

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

### 2.6.2 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: -*invalid.me  
Domain Server: ns1.example.me  
Domain Server: ns2.example.me  
Domain Registrant Contact ID: OTE-C1  
Domain Admin Contact ID: OTE-C2  
Domain Billing Contact ID: OTE-C3  
Domain Technical Contact ID: OTE-C4  
Auth Info: my secret
```

Verify that the following response is received:

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

### 2.6.3 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.me**  
Domain Server: **ns1.example.me**  
Domain Server: **ns2.example.me**  
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): **99**  
Auth Info: **my secret**

Verify that the following response is received:

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

#### **2.6.4 Correctly Handle 2002 Exception**

2002 "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: **renew.me**  
Expiration Date: **2016-10-01**

Verify that the following response is received:

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

#### **2.6.5 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.me**  
Domain Server: **ns1.example.me**  
Domain Server: **ns2.example.me**  
Domain Registrant Contact ID: **OTE-C99**  
Domain Admin Contact ID: **OTE-C2**  
Domain Billing Contact ID: **OTE-C3**  
Domain Technical Contact ID: **OTE-C4**  
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>
```

### 2.6.6 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: **OTE-C2**

Verify that the following response is received:

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

### 2.6.7 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.me**

Verify that the following response is received:

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

## 2.7. Efficiency of Client Session Management

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

### 2.7.1 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 the next section. Use the Hello command at intervals under 10 minutes to maintain client connectivity.

### 2.7.2 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 2.7.3

### **2.7.3 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'>
```

## **2.8. 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>
```

## **2.9. Completing the Test**

At this point, contact .ME Support at support@registry.me or +1.416.619.3037 to notify them that the OT&E test has been completed.

## 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. .ME 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. \*\*\***

### Users

Registrar: ClientX

Password: foo-BAR2

### Contacts

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

Object	Owned By	Notes
OTE-C1	ClientX	
OTE-C2	ClientX	
OTE-C3	ClientX	
OTE-C4	ClientX	
OTE-C6	ClientY	
OTE-C9	ClientX	
OTE-C7	ClientX	Auth Info: my secret ** This contact has pending transfer status, initiated by ClientY**
OTE-C8	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.me  
Auth Info: my secret

### Hosts

Object	Owned By	Notes
ns1.example.me	ClientX	IP Address: 192.168.0.1
ns2.example.me	ClientX	IP Address: 192.168.2.3
ns3.example.me	ClientX	IP Address: 192.168.4.5
ns4.example.me	ClientX	IP Address: 192.168.6.7

## Domains

<b>Object</b>	<b>Owned By</b>	<b>Notes</b>
trademark.me	ClientX	IPR Name: Test Trademark IPR Applied: 2000-01-01 IPR Registered: 2002-01-01 IPR Number: 998877 IPR CC Locality: CA
renew.me	ClientX	
transfer2.me	ClientX	Auth Info: my secretX ** This domain has pending transfer status, initiated by ClientY**
transfer3.me	ClientX	Auth Info: my secretX ** This domain has pending transfer status, initiated by ClientY**
transfer1.me	ClientY	Auth Info: my secretY

The seeded domains use the following common values:

Hosts: ns1.example.me

Hosts: ns2.example.me