NEUSTAR REGISTRAR REFERENCE GUIDE

SUNRISE SUPPLEMENT: MULTI-APPLICATION AWARD METHOD

December 20, 2013

neustar®
Real Intelligence. Better Decisions.℠

This document is for informational purposes only. NEUSTAR MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Neustar.

Neustar may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Neustar, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

Neustar Ultra Services and UltraCare are Neustar’s trademarks and any use of these or any other Neustar mark without Neustar’s express written consent is prohibited. All other trademarks and/or service marks identified or referenced are the property of their respective owners and subject to their usage requirements.

Table of Contents

Contents

1 Introduction

This supplement to the *Registry Registrar Reference Guide* aids registrars in interfacing with the Registry Service Provider during the Sunrise phase. A more general reference to Registry Operations for registrars may be found in the *Registry Registrar Reference Guide*.

The Sunrise phase is designed to provide legitimate trademark holders with the opportunity to apply for 2nd level domains within the Registry space.

1.1 Purpose

This document explains how to use Neustar's Shared Registry System (SRS) Registrars Toolkit to interface with new generic TLD domain registrations during Sunrise phase that accepts multiple applications for the same domain name (MULTI). In a MULTI Sunrise, if multiple applications exist for the same domain, the domain is award by auction or other third party decision processes. This document supplements the Developer's Guide for new generic TLDs.

1.2 Intended Audience & Reading Suggestions

This document is targeted toward those desiring an understanding of the EPP requirements for a MULTI Sunrise, especially those involved in the design, development, integration, and testing.

1.3 Reference Documents

The Launch Phase Mapping for the Extensible Provisioning Protocol (EPP) draft-tan-epp launchpahse document describes the EPP extension mapping for the provisioning and management of domain name registrations and applications during the launch of a domain name registry. http://www.ietf.org/id/draft-tan-epp-launchphase-12.txt

Mark and Signed Mark Objects Mapping http://tools.ietf.org/html/draft-lozano-tmch-smd-03 describes the format of the mark and digitally signed mark required for domain creates in Sunrise http://tools.ietf.org/html/draft-lozano-tmch-func-spec-08.

1.4 Multi-application Award Method

The Registry operates a Sunrise Period using the Multi-application Award Method. The term *Application* refers to a Sunrise where multiple requestsfor the same domain are accepted, and the winners are determined by a defined process, such as an auction. During the Sunrise phase, applications for domain names that meet the Trademark Clearinghouse (TMCH) criteria are stored, provided the domain name is not on the reserved list for the Registry. The Registry accepts requests for the same name during the Sunrise period. At the end of the Sunrise period, the Registry Operator identifies domain names for which there is more than one application. Where multiple applications for the same domain name exist, the Registry Operator selects the winning application or contracts out the selection process to an auction provider, or other third party. The registrant of a domain name for which there is only one application is deemed the winner.

1.5 The Trademark Clearinghouse

The Trademark Clearinghouse (TMCH) is a global database and repository for trademark data supporting rights protection mechanisms offered by new gTLD registries. The TMCH is operated by Deloitte and IBM via an agreement with ICANN. By performing trademark validation services, it replaces the need for trademark holders to register their marks in many different databases as new gTLDs are launched. The clearinghouse:

- Is a centralized database;
- Verifies trademark information from rights holders all over the world;
- Provides this information to registries and registrars during TLD startup phases

1.6 Signed Mark Data

Signed Mark Data (SMD) is an electronic document or file that contains a digital signature from the TMCH. A trademark holder with an SMD can be assured their trademark has been verified by the TMCH.

1.7 Particpants and Roles

The table below provides descriptions of the participants and roles during the Registry Sunrise phase.

Table 1 Participants and Roles

Participant	Role(s)
Applicants / Registrants	• Submit trademarks to the TMCH for validation Submit applications to registrars
Registrars	• Collect Registrations from applicants and resellers • Submit them to the Registry via EPP • Communicate with applicants • Bill applicants
Registry Operator	• Defines business rules for the Sunrise phase based on ICANN requirements
Registry Service Provider (RSP)	• Acts on behalf of the Registry during Sunrise to implement business rules • Accepts Registration Requests via EPP • Validates Registration Requests (SMD and domain name) • Maintains the authoritative Sunrise database for Valid applications in a Multi-application Sunrise • Invoices Registrars • Injects winning Registrations into the production SRS • Provides Registrar reports and Poll Messages
Trademark Clearinghouse (TMCH)	• Performs trademark validation services • Issues Signed Mark Data (SMD) • Manages the appeals process

1.8 Sunrise Process Description

The applicant for the domain name may be an individual, reseller, or some other agent. All applications that pass the Signed Mark Data (SMD) criteria as established by ICANN incur a Sunrise Fee charged to the submitting Registrar. In addition, all winning Registrations incur a Registration Fee when the domain is injected (registered) in the production SRS. An Injection Fee may also be assessed at the time of injection. Detailed billing information is provided below.

The stages of the Sunrise phase are as follows:

1. Trademark Validation by the TMCH: Trademark owners submit trademark information to the TMCH and received SMD.

2. The applicant submits an application via the Registrar.

3. The SRS checks to see if the name is either on the reserved list If the name is unavailable and the application is rejected.. If the name is available, the Sunrise system validates the SMD using the TMCH public key. If the SMD is valid, the Sunrise system verifies that the name being applied for is a valid name variant for the trademark. The domain name being requested must exist in the domain list included in the SMD.

4. If the Request is valid according to the rules above, the Registrar is be billed for the Sunrise fee.

6. 5. The Sunrise period must run a minimum of 60-days in a Multi-application Sunrise. In ICANN parlance, a multi-application Sunrise as described in this document is referred to as an End Date Sunrise. Once the TLD is delegated and Sunrise plans are approved, the Registry Operator may start the Sunrise at any time provided that the end date is at least 60 days from the start date. Once the Sunrise period ends, the Registry Operator determines the winner of all applications for the same domain name.All winning Sunrise applications are injected. All single name applications are injected as well. If any domain names are in a pending state for any reason, the name is placed on the reserved list in the GA system.Once a domain name has been injected into the GA system, the RSP issues the following notifications:

   a. To the Registrar via standard reports on the SCP server. See details below;

   b. To the TMCH via the agreed upon practices established in the ICANN specifications;

   c. On the Sunrise Poll Queue via a Poll Message indicating that the domain was allocated.

7. This injection process runs until all applications have been injected into the GA system.

8. Once injected into the GA System, the domain names are live in the WHOIS database for the Registry. Registrars then need to provide nameservers to the RSP so that the domain may go live in the DNS system. The registration date of the domain corresponds to the injection date.

9. Applications that are not selected as winners are marked as rejected and registrar notification takes place via standard reports and poll messages.

Sunrise Registrations are considered non-refundable and Registrars cannot delete registrations. Domain names created during the Sunrise phase may not be deleted during the Sunrise phase. If there is a need to delete a domain name that was created during the Sunrise Phase, then that name may be deleted once General Availability opens.

2 Sunrise Billing

2.1 Billing

For each Sunrise phase, there are three distinct billing events to the Registrar: *Sunrise Fee, Injection Fee, and Registration Fee*.

- The *Sunrise Fee* is billed when the request for domain name is received by the SRS and is valid. The *Sunrise Fee* amount is decided by the Registry Operator.
- The *Registration Fee* amount is in the form of a new registration of a term also decided by the Registry Operator.

The *Injection Fe*e is billed when the application is injected from the Sunrise Environment into the Production environment and is decided by the Registry Operator.

3 Reporting

3.1 Registrar Sunrise Reports

Several reports are provided to the Registrar so that a basic understanding of performance can be ascertained.

Some reports provide Launch Status, which is an indicator of the state of domains as of the time the reports were run.

Table 1 Sunrise Reporting Launch Phase Status

Launch Phase Status	Description
Validated	The domain name was successfully validated in the TMCH and has a valid SMD associated with it.
Allocated	The domain name has been successfully allocated/injected into the GA system.
PendingAllocation	The domain name has been validated and is pending allocation into the GA system.
Rejected	The domain was rejected and not injected into the GA system.

The following reports are generated on a daily basis and deposited in each Registrar's Secure Copy (SCP) account.

3.2 Daily Activity Report

The Daily Activity Report lists, for each Registrar, activity over the previous day. It includes the following information:

- Domain Name
- Registration timestamp
- Launch Phase Status

3.3 Daily Phase Summary Report

The Daily Phase Summary Report lists the number of registrations created during the phase. The report also has a tally of Registrations in each status, whether it is validated, allocated, or another status.

3.4 Daily Phase Activity Report

The Daily Phase Activity Report provides information on the cumulative, running inventory of activity during the entire phase. It includes the following information:

Domain Name

Registration timestamp

Launch Phase Status

4 Sunrise Business Rules

The Registry Operator sets the Business Rules for how the TLD behaves, what transactions are possible and when those transactions are executed.

4.1 Business Rules for Registrars

During the Sunrise phase, a limited set of business rules are permitted and only these transactions are permitted to Registrars.

During the Contact Create periods before Sunrise opens, registrars may perform the following:

- Contact Create
- Contact Check
- Contact Info

Registrars may perform the following contact operations during open Sunrise Periods:

- Contact Create
- Contact Check
- Contact Info

Registrars may perform the following domain operations during open Sunrise Periods:

- Domain Create
- Domain Check
- Domain Info

Quiet periods are the periods after Sunrise closes and before the next phase opens.

Registrars may perform the following contact operations during quiet periods:

- Contact Check
- Contact Info

Registrars may perform the following domain operations during quiet periods:

- Domain Check
- Domain Info

The table below summarizes the commands available during Sunrise transitions.

	Contact Create Period – Before Sunrise Opens	Sunrise Open	Sunrise Close (between phases)
Contact Create	X	X	
Contact Check	X	X	X
Contact Info	X	X	X
Domain Create		X	
Domain Check		X	X
Domain Info		X	X

Note: As Sunrise transitions from contact create period, to open, to quiet periods, registrars may experience momentary outages.

4.2 Sunrise Statuses

During normal GA, registry operations domain names may be put into various EPP statuses after operations are performed. In Sunrise, names are being created, so the applicable EPP statuses are pendingCreate and inactive:

pendingCreate: This status indicates that a request to create a domain name has been received and is being processed.

Inactive: This status indicates *that* delegation information (DNS or nameservers) has not been associated with the domain.

In addition to EPP status, during Sunrise there are also Launch Phase Statuses. Table 3 below depicts the Launch Phase Status along with the EPP status and description.

Table 2 Launch Phase Status and Corresponding EPP Status During Sunrise

Launch Phase Status	EPP Status	Description
Validated	pendingCreate	The domain name was successfully validated in the TMCH and has a valid SMD associated with it.
Allocated	inactive	The domain name has been successfully allocated/injected into the GA system.
PendingAllocation	pendingCreate	The domain name has been validated and is pending allocation into the GA system.
Rejected	inactive	The domain was rejected and not injected into the GA system.

5 Toolkit API Classes

This section describes the classes provided with the Toolkit API that specifically address Sunrise functionality.

5.1 EppCommandCreateLaunchRegistration

This class is the required extension for a MULTI Application object and specifies the phase, registration type and trademark data required for Sunrise phases. For a MULTI Sunrise, the type is “application”.

5.2 EppEncodedSignedMarkData

This class holds the Base 64 encoded version of the signed mark data. The registrant may use encoded object received from the TMCH instead of the signed mark data.

5.3 EppSignedMarkData

This is the unencoded version of the trademark data received from the TMCH. Either the encoded or unencoded mark data may be used when creating a Sunrise application.

5.4 EppCommandCheckLaunchRegistration

This class is the required command extension to do an availability check on a domain while a MULTI Sunrise is operational.

5.5 EppCommandInfoLaunchRegistration

This class is the required command extension to perform a query, or info, on a Sunrise Application object. The application ID must be specified. Optionally, the mark data can be returned with the response if specified.

5.6 EppStatus

This class contains the status information of an EPP Object.

5.7 EppMarkMark, EppTrademarkData, EppCourtData, EppTreatyorStatuteData

These classes contain the varieties of trademark information that can be associated associated to a Sunrise application. This information is supplied in the signed mark data (or encoded signed mark data) by the TMCH.

5.8 EppLaunchPhase

The launch phase (e.g “sunrise”) must be set for Sunrise commands to let the server know the target phase of the command.

5.9 EppResponseDataCreateLaunchRegistration

This class contains the launch extension for the Create command response. The application ID is in this extension and will be required for subsequent domain operations. The combination of domain name and application id uniquely identifies a MULTI sunrise application.

5.10 EppResponseDataInfoLaunchRegistration

This class contains the launch extension for an Info command response including the mark data, status, and application id.

6 Operations

This section details domain operations for MULTI Sunrise where they differ from normal general availability operations.

In the Registry OT&E environment for a MULTI Sunrise, the tld “MULTI” may be used in EPP commands, as well as a valid new gTLD that is scheduled to be launched, such as “UNO”.

The primary operations supported by the toolkit are classified into following sets:

6.1 Object Management Operations

Registry Toolkit allows the user to perform the following operations on domain, host and contact objects supported in Sunrise MULTI.

Table 3 Object Management Operations

Operations	Domain	Contact	Host
Create	✓	✓	
Modify			
Query	✓	✓	
Delete			
Check	✓	✓	
Renew			
Transfer			

The following three sections describe the Domain Operations in detail. Contact Operations do not differ from normal general availability, so examples are not provided.

6.2 Domain Management Operations

6.2.1 Create Domain Name [EppCommandCreate, EppCommandCreateLaunchRegistration, EppResponseDataCreateLaunchRegistration]

This operation enters the Domain Name into the system. By registry policy, only 1 Signed Mark or Encoded Signed Mark will be accepted by the registry on a Create Domain Command.

6.2.1.1 Required Attributes

- Domain Name
- Admin Contact ID
- Technical Contact ID
- Billing Contact ID
- Registrant ID
- Launch Phase = sunrise
- Signed Mark or Encoded Signed Mark
- Type=application

6.2.1.2 Optional Attributes

- Registration Period
- Registrar name/value pairs

6.2.1.3 Implicit Attributes

- Status
- Expiration Date
- Registration Date
- Sponsoring Registrar
- Created By
- Created Date
- Updated By
- Updated Date

6.2.1.4 Data Returned

- Application ID
- Domain Name
- Expiration Date

6.2.1.5 Result Codes

- 1000 Command completed successfully
- 2001 Command syntax error
- 2002 Command use error
- 2003 Required Parameter missing
- 2004 Parameter value range error
- 2005 Parameter value syntax error
- 2104 Billing failure
- 2302 Object exists
- 2306 Parameter value policy error – Launch Phase does not match
- 2400 Command failed

6.2.1.6 Usage

```
EppDomain domain = new EppDomain("example1.MULTI");
// each contact is set by giving the contact id of an existing
// contact object as well as the contact type for this domain,
// more than one contact of any given type is possible.
domain.addContact("JL34" , EppDomain.CONTACT_TYPE_ADMIN);
domain.addContact("BN156", EppDomain.CONTACT_TYPE_TECH);
domain.addContact("JL34" , EppDomain.CONTACT_TYPE_BILLING);
domain.addContact("LS2"  , EppDomain.CONTACT_TYPE_BILLING);
// set the registration period to one year
domain.setPeriod(new EppPeriod(1, EppPeriod.UNIT_YEAR));
// set the registrant of the domain to an existing contact
domain.setRegistrant("TK450");
// set authorization information for the domain that will be used
// for domain transfer request operations.
domain.setAuthInfo(new EppAuthInfo(EppAuthInfo.TYPE_PW, "SecretPW"));
// generate an actual create domain command using the domain
EppCommandCreate cmd = new EppCommandCreate(domain);

// set transaction id, if needed

cmd.setClientTransactionId("MY-DOMAIN-CREATE-TRANSACTION-ID");

// create the launch extension
// set the registration type, phase, and signed mark data
EppCommandCreateLaunchRegistration launchExt =
    new EppCommandCreateLaunchRegistration();
launchExt.type("application");
EppLaunchPhase _phase = new EppLaunchPhase();
_phase.phase("sunrise");
launchExt.phase(_phase);
EppEncodedSignedMarkData _mark =
    new EppEncodedSignedMarkData();
_mark.rawXml(_encMark);
launchExt.encodedSignedMark(_mark);

// add the extension to the command
cmd.addExtension(launchExt);

// send the command to the server and receive a response
EppResponse res = channel.send(cmd);
if( res == null || ! res.success() )
{
    // error processing ...
}
// get the response data for the domain

EppExtension extension = null;
String applicationId = null;
EppResponseDataCreateLaunchRegistration launchResponseExt = null;

EppResponse res = channel.send(cmd);
```

Continued on next page

```
Vector extensions = (Vector)res.getExtension();

        for( int i = 0; i < extensions.size(); i++ )
        {
           extension = (EppExtension) extensions.get(i);
           if (extension instanceof EppResponseDataCreateLaunchRegistration)
            {
               launchResponseExt =
               (EppResponseDataCreateLaunchRegistration) extension;
               applicationId = launchResponseExt.applicationId();
           }
        }
```

6.2.1.7 Example XML

```xml
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <create>
      <domain:create xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>testandvalidate.NEUSTAR</domain:name>
        <domain:period unit="y">2</domain:period>
        <domain:registrant>ABHICONTACT07_10</domain:registrant>
        <domain:contact type="admin">ABHICONTACT07_10</domain:contact>
        <domain:contact type="tech">ABHICONTACT07_10</domain:contact>
        <domain:contact type="billing">ABHICONTACT07_10</domain:contact>
        <domain:authInfo>
          <domain:pw>1234</domain:pw>
        </domain:authInfo>
      </domain:create>
    </create>
    <extension>
        <idn:data xmlns:idn="urn:ietf:params:xml:ns:idn-1.0">
          <idn:table>en</idn:table>
         </idn:data>
        <launch:create xmlns:launch="urn:ietf:params:xml:ns:launch-1.0" type="application">
                <launch:phase>sunrise</launch:phase>
<smd:encodedSignedMark xmlns:smd="urn:ietf:params:xml:ns:signedMark-1.0">
PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz4KPHNtZDpzaWduZWRNYXJrIHht
bG5zOnNtZD0idXJuOmlldGY6cGFyYW1zOnhtbDpuczpzaWduZWRNYXJrLTEuMCIgaWQ9Il84M2Jl
ZDk5ZC0yMDE2LTQxNTItYTY3My05ZGQzZDAxNzhlNjMiPgogIDxzbWQ6aWQ+MDAwMDAwMTcyMTM3…
MDlEYXRhPjwvZHM6S2V5SW5mbz48L2RzOlNpZ25hdHVyZT48L3NtZDpzaWduZWRNYXJrPg==
</smd:encodedSignedMark>
        </launch:create>
    </extension>
    <clTRID>ABC-12345</clTRID>
  </command>
</epp>
```

6.2.2 Check Domain Name [EppCommandCheck, EppCommandCheckLaunchRegistration]

The check command is used to determine whether or not a domain is available to be provisioned. For MULTI Sunrise, the command checks both Sunrise and GA environments if applicable.

6.2.2.1 Required Attributes

- Domain Name
- Extension to check command
    - Check type – avail
    - Phase – Sunrise

6.2.2.2 Data Returned

Returns true if the Domain Name is available or false if unavailable.

6.2.2.3 Usage

```
EppCommandCheckDomain cmd = new EppCommandCheckDomain();

EppCommandCheckLaunchRegistration launchExt = new
    EppCommandCheckLaunchRegistration();

// Create the extension to set the type and phase
// Set the phase and type
launchExt.type("avail");
EppLaunchPhase _phase = new EppLaunchPhase();
_phase.phase("sunrise");
launchExt.phase(_phase);
cmd.addExtension(launchExt);
// add the domains to be checked
cmd.add("example1.MULTI");
cmd.add("example2.MULTI");
// send the command to the server and receive a response object
EppResponse res = channel.send(cmd);
if (res == null || ! res.success())
{
    // error processing ...
}
// get response data
EppResponseDataCheck data;
data = (EppResponseDataCheck) res.getResponseData();
if( data.isAvailable("example1.EXAMPLE") )
{
    // example1.MULTI is available
}
else
{
    // example1.MULTI is not available
    if( data.getReason("example1.MULTI") != null )
    {
        // display reason text, if any
    }
}
```

6.2.3 Info Domain Name [EppCommandInfo, EppCommandInfoLaunchRegistration, EppResponseDataInfoLaunchRegistration]

The info command returns domain attributes and optionally, mark attributes.

6.2.3.1 Required Attributes

- Domain Name

6.2.3.2 Optional Attributes

- Mark Data

6.2.3.3 Usage

```
EppCommandInfoDomain cmd =
new EppCommandInfoDomain("example1.MULTI");

// create the launch extension
EppCommandInfoLaunchRegistration launchExt = new
EppCommandInfoLaunchRegistration();
// set up transaction id, if needed
cmd.setClientTransactionId("MY-DOMAIN-QUERY-TRANSACTION-ID");

// Set the phase indicate whether to return mark data

launchExt.includeMark(true);
EppLaunchPhase _phase = new EppLaunchPhase();
_phase.phase("sunrise");
launchExt.phase(_phase);
launchExt.applicationId(applicationID);
EppResponse res = channel.send(cmd);
if (res == null || ! res.success())
{
    // error processing ...
}

EppExtension extension = null;
String status = null;
EppMarkMark mark = null;
String applicationId = null;
EppResponseDataInfoLaunchRegistration launchResponseExt = null;

Vector extensions = (Vector)res.getExtension();

        for( int i = 0; i < extensions.size(); i++ )
        {
            extension = (EppExtension) extensions.get(i);
            if (extension instanceof EppResponseDataInfoLaunchRegistration)
            {
                launchResponseExt =
                    (EppResponseDataInfoLaunchRegistration) extension;
                status = launchResponseExt.status();
                mark = launchResponseExt.mark();
                status = launchResponseExt.applicationId();
            }
        }
```

6.3 Poll Messages

When a Sunrise MULTI Application is transitioned from validated to pendingAllocation, a poll message is put on the queue to signal the change of status. A second poll message is put on the queue when the the Application transitions from pendingAllocation to allocated or rejected. The mechanism to retrieve the poll message from the queue is the same as steady state GA. The message contains an extension indicating the status.

6.3.1 Pending Allocation Poll Response

The following is an example poll message response for an "pendingAllocation" Sunrise Application. This state would generally mean that the application is awaiting auction, or other third party decision.

```
<?xml version="1.0" encoding="UTF-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
 <response>
  <result code="1301">
   <msg lang="en-US">Command completed successfully; ack to dequeue</msg>
   <value>
    <text>SRS Major Code: 2000</text>
   </value>
   <value>
    <text>SRS Minor Code: 20024</text>
   </value>
   <value>
    <text>--QUEUE_SUCCESSFULLY_POLLED</text>
   </value>
   <value>
    <text>Registration pending allocation.</text>
   </value>
  </result>
  <msgQ id="3370" count="314">
   <qDate>2013-10-23T09:06:37.0Z</qDate>
   <msg>Registration pending allocation.</msg>
  </msgQ>
  <resData>
   <domain:panData xmlns="urn:ietf:params:xml:ns:domain-1.0"
xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
xsi:schemaLocation="urn:ietf:params:xml:ns:domain-1.0 domain-1.0.xsd">
    <domain:name paResult="1">MULTIPHASE0201A.NEUSTAR</domain:name>
    <domain:paTRID>
     <epp:clTRID xmlns:epp="urn:ietf:params:xml:ns:epp-1.0"
xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">strecdvvap15-52678FA8-32636-
1001</epp:clTRID>
     <epp:svTRID xmlns:epp="urn:ietf:params:xml:ns:epp-1.0"
xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">20131023090426200-1091131201-
2</epp:svTRID>
    </domain:paTRID>
    <domain:paDate>2013-10-23T09:06:37.0Z</domain:paDate>
   </domain:panData>
  </resData>
  <extension>
   <launch:infData xmlns:launch="urn:ietf:params:xml:ns:launch-1.0"
xsi:schemaLocation="urn:ietf:params:xml:ns:launch-1.0 launch-1.0.xsd">
    <launch:phase>sunrise</launch:phase>
    <launch:applicationID>D8125-NEUSTAR</launch:applicationID>
    <launch:status s="pendingAllocation"/>
   </launch:infData>
  </extension>
  <trID>
   <clTRID>strecdvvap15-52679155-1067-1002</clTRID>
   <svTRID>20131023090728920-1091131201-2</svTRID>
  </trID>
 </response>
</epp>
```

6.3.2 Allocated Poll Response

The following is an example poll message response for an "allocated" Sunrise Application. This state would generally means that the application has been injected into the general availability environment.

```
<?xml version="1.0" encoding="UTF-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
 <response>
  <result code="1301">
   <msg lang="en-US">Command completed successfully; ack to dequeue</msg>
   <value>
    <text>SRS Major Code: 2000</text>
   </value>
   <value>
    <text>SRS Minor Code: 20024</text>
   </value>
   <value>
    <text>--QUEUE_SUCCESSFULLY_POLLED</text>
   </value>
   <value>
    <text>Registration successfully allocated</text>
   </value>
  </result>
  <msgQ id="3373" count="314">
   <qDate>2013-10-23T09:31:52.0Z</qDate>
   <msg>Registration successfully allocated</msg>
  </msgQ>
  <resData>
   <domain:panData xmlns="urn:ietf:params:xml:ns:domain-1.0"
xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
xsi:schemaLocation="urn:ietf:params:xml:ns:domain-1.0 domain-1.0.xsd">
    <domain:name paResult="1">MULTIPHASE0201A.NEUSTAR</domain:name>
    <domain:paTRID>
     <epp:clTRID xmlns:epp="urn:ietf:params:xml:ns:epp-1.0"
xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">strecdvvap15-52678FA8-32636-
1001</epp:clTRID>
     <epp:svTRID xmlns:epp="urn:ietf:params:xml:ns:epp-1.0"
xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">20131023090426200-1091131201-
2</epp:svTRID>
    </domain:paTRID>
    <domain:paDate>2013-10-23T09:31:52.0Z</domain:paDate>
   </domain:panData>
  </resData>
  <extension>
   <launch:infData xmlns:launch="urn:ietf:params:xml:ns:launch-1.0"
xsi:schemaLocation="urn:ietf:params:xml:ns:launch-1.0 launch-1.0.xsd">
    <launch:phase>sunrise</launch:phase>
    <launch:applicationID>D8125-NEUSTAR</launch:applicationID>
    <launch:status s="allocated"/>
   </launch:infData>
  </extension>
  <trID>
   <clTRID>strecdvvap15-526797A8-3917-1003</clTRID>
   <svTRID>20131023093252912-1091131201-1</svTRID>
  </trID>
 </response>
</epp>
```

6.3.3 Rejected Poll Response

The following is an example poll message response for an "rejected" Sunrise Application. This state would generally mean that the application did not win the auction, or was not selected by another third party provider.

```
<?xml version="1.0" encoding="UTF-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
 <response>
  <result code="1301">
   <msg lang="en-US">Command completed successfully; ack to dequeue</msg>
   <value>
    <text>SRS Major Code: 2000</text>
   </value>
   <value>
    <text>SRS Minor Code: 20024</text>
   </value>
   <value>
    <text>--QUEUE_SUCCESSFULLY_POLLED</text>
   </value>
   <value>
    <text>Application rejected.</text>
   </value>
  </result>
  <msgQ id="3372" count="19">
   <qDate>2013-10-23T09:29:18.0Z</qDate>
   <msg>Application rejected.</msg>
  </msgQ>
  <resData>
   <domain:panData xmlns="urn:ietf:params:xml:ns:domain-1.0"
xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
xsi:schemaLocation="urn:ietf:params:xml:ns:domain-1.0 domain-1.0.xsd">
    <domain:name paResult="1">MULTIPHASE0201A.NEUSTAR</domain:name>
    <domain:paTRID>
     <epp:clTRID xmlns:epp="urn:ietf:params:xml:ns:epp-1.0"
xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">strecdvvap15-52678FA8-32636-
1002</epp:clTRID>
     <epp:svTRID xmlns:epp="urn:ietf:params:xml:ns:epp-1.0"
xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">20131023090459203-1091131201-
2</epp:svTRID>
    </domain:paTRID>
    <domain:paDate>2013-10-23T09:29:18.0Z</domain:paDate>
   </domain:panData>
  </resData>
  <extension>
   <launch:infData xmlns:launch="urn:ietf:params:xml:ns:launch-1.0"
xsi:schemaLocation="urn:ietf:params:xml:ns:launch-1.0 launch-1.0.xsd">
    <launch:phase>sunrise</launch:phase>
    <launch:applicationID>D8126-NEUSTAR</launch:applicationID>
    <launch:status s="rejected"/>
   </launch:infData>
  </extension>
  <trID>
   <clTRID>strecdvvap15-52679703-3345-1001</clTRID>
   <svTRID>20131023092953844-1091131201-1</svTRID>
  </trID>
 </response>
</epp>
```

7 Simulating Status/Poll Messages in the OT&E Environment

For registrars to view the Sunrise status values (Section 4.2) and retrieve Poll messges that reflect the EPP and Launch Phase status values (Section 6.3), the registry uses a combination of domain naming conventions, a signing utility, and cron jobs to simulate the process of a domain going from validated to either allocated or rejected Launch status.

The following steps must occur in order to simulate the Sunrise MULTI lifecycle.

7.1 Install the xmlSigner utility

Install the provided signing utility. This utility is used to sign an SMD with an internal Neustar certificate. Because the simulation process depends upon a naming convention, the utility allows the registrar to create an SMD for a particular name (see below) and then follow it through the lifecycle. The package contains the following:

1. xmlSigner.jar
2. encryptutil.jar
3. sampleUnsignedSmdForPollMessages.xml
4. signing-good.jks

7.2 Create SMD

Sunrise domains must include a valid signed SMD object. Create an SMD and sign it using the utility in the section above.

In order to use signing utility following command should be used:

```
java -cp XmlSigner.jar:encryptutil.jar com.neustar.srs.test.util.XmlSigner signing-
good.jks <inputXmlFile> <outputXmlFile> <generateEncoded>
```

Four parameters are required by signing utility:

1. Keystore file : signing-good.jks

2. Input XML file - unsigned SMD XML. This must contain mark labels of domain to be created. Mark-labels must match with specific format if poll message needs to be tested. A sample unsigned SMD XML provided in attachment. The domain naming convention should follow the specifics in the next section.

3. Output XML file – Generated signed SMD. This XML would either contain `smd:signedMark` element or `smd:encodedSignedMark` element, based on 4th parameter.

4. “true” or “false”, depending upon whether encoded signed mark data is required or not. “true” is recommended option.

Note:

1. Input XML’s `smd:signedMark` element should have `id="signedMark"`.

7.3 Create Domains according to convention

Create domain names according to the naming convention in the table below and include the Signed SMD generated above.. Execute an EPP create domain for a specific domain name with a valid SMD file included.

Note:

1. XML Prolog in generated SMD should be removed while including it into create domain XML.

7.4 Domain Naming Convention

The following table describes the naming convention used to simulate the Sunrise EPP and Launch statuses for a domain.

Type	**Final Statuses (Launch/EPP)**	**Domain Names (CCID is Common Customer ID for the Registrar)**
MULTI	Pending Allocation/Pending Create (sent to Auction) Allocated/Inactive (Auction Winner) Rejected/Inactive (Auction Loser)	DN-MULT-CCID-PEND1.TLD thru DN-MULT-CCID-PEND20.TLD DN-MULT-CCID-ALLOC1.TLD thru DN-MULT-CCID-ALLOC20.TLD DN-MULT-CCID-REJECT1.TLD thru DN-MULT-CCID-REJECT20.TLD

The domains that have a final status of allocated or rejected will generate two poll messges:

1. Transition from Validated/Pending Create to Pending Allocation/Pending Create
2. Transition from Pending Allocation/Pending Create to Allocated OR Rejected

7.5 Wait

The process that picks up the submitted domains and moves them to the next state run every x on the x. Once the job has run, check for poll messges and/or do EPP Info commands to see the change in status reflected in the domain/application.

8 Support

For EPP support, Organizational Test and Environment Certification (OT&E), or specific domain transaction questions, please contact your registrar relations representative or email reg-support@neustar.biz.