NEUSTAR REGISTRAR REFERENCE GUIDE

GENERAL AVAILABILITY CLAIMS SUPPLEMENT

October 11, 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

Table of Figures and Tables

1 Introduction and Audience

This supplement to the *Registry Registrar Reference Guide* aids registrars in interfacing with the Registry Provider during General Availability (GA). A more general reference to Registry Operations for registrars may be found in the *Registry Registrar Reference Guide*.

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 a the initial period of General Availability (GA) which requires claims processing. In GA, domains are awarded in a First Come First Served (FCFS) method. However, if the domain has a claim filed in the TMCH, a claims acknowledgement must be presented with the domain create during the claims period. 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 GA Claims phase, 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-11.txt
- Mark and Signed Mark Objects Mapping draft-lozano-tmch-smd describes the format of the mark and digitally signed mark required for domain creates in Sunrise. http://tools.ietf.org/html/draft-lozano-tmch-smd-03
- TMCH functional specfactions describes both registry and registrar interfaces with the TMCH. http://tools.ietf.org/html/draft-lozano-tmch-func-spec-08

1.4 First Come, First Served (FCFS) Award Method

During GA, the first domain name that is not already allocated and/or not on the reserved list for the Registry is awarded. GA is a registration-based as opposed to an application-based period, where multiple applications for the same domain name are accepted and the winners are determined by a defined process, such as an auction.

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. Deloitte and IBM operate the TMCH via an agreement with ICANN. By performing trademark validation services, the TMCH 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 Claims

Claims provides notice to prospective domain name registrants that the particular domain name they wish to registrer has a trademark claim against it. The prospective registrant must then choose to proceed with the registration or not. If the registrant proceeds with the registration, the rightful owner of the trademark with a claim against the name is notified of the registration by the TMCH. The rightful owner may then take legal action against the registrant, if desired. ICANN’s Rights Protection Mechanisms (RPMs) requirements have stipulated all domain name registrations in the the first 90 days of GA be subject to the claims process. In addition, any domain name being released off of the reserved name list for the first time must be subject to the claims process.

Table 1 provides descriptions of the participants and roles during GA.

Table 1 Registry GA Participants and Roles

Participant	Role(s)
Applicants / Registrants	• Submit requests for registrations • Accept claims notices if applicable
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 Registry based on ICANN requirements
Registry Service Provider (RSP)	• Acts on behalf of the Registry Operator during GA to implement business rules • Accepts Registration Requests via EPP • During the Claims period, checks to see if there is a claim against the requested domain name and informs the registrar • Maintains the authoritative database for all Valid Registration Requests in a FCFS manner • Invoices Registrars • Registers names into the production SRS • Provides Registrar reports
Trademark Clearinghouse (TMCH)	• Performs trademark validation services • Holds claims information for domain names • Manages the appeals process

1.7 GA Process Description

The registrant for a domain name may be an individual, reseller, or some other agent. All Registration requests for which the domain name is not on the reserved list or already taken, and for which the registrant accepts the claims notice, if applicable, is accepted and registered in the production SRS. All successful Registrations incur a Registration Fee when the domain is registered.

2 Billing

2.1 Billing

During GA, the single billing event to the Registrar is the *Registration Fee*.

The *Registration Fee* amount is in the form of a new registration of a term also decided by the Registry Operator.

3 Reporting

3.1.1 Registrar Reports

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

4 Toolkit API Classes

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

4.1 EppLaunchPhase

The launch phase must be set for claims commands to let the server know the target phase of the command. The launch phase is “claims” for the GA Claims period and the type is “registration”.

5 Operations

This section details domain operations for GA Claims where they differ from normal general availability operations.

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

5.1 Object Management Operations

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

Table 2 Object Management Operations

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

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

5.2 Domain Management Operations

5.2.1 Create Domain Name [EppCommandCreate, EppCommandCreateLaunchRegistration, EppResponseDataCreateLaunchRegistration]

This operation enters the Domain Name into the system.

5.2.1.1 Required Attributes

- Domain Name
- Admin Contact ID
- Technical Contact ID
- Billing Contact ID
- Registrant ID
- Launch Phase = claim
- Launch type = registration

5.2.1.2 Optional Attributes

- Registration Period
- Name Value Pairs

5.2.1.3 Implicit Attributes

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

5.2.1.4 Data Returned

- Application ID
- Domain Name
- Expiration Date

5.2.1.5 Result Codes

1001	Command completed successfully, action pending
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

- 2306 Parameter value policy error – Launch Phase does not match
- 2400 Command failed

5.2.1.6 Usage

```
EppDomain domain = new EppDomain("example1.GACLAIM");
// 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 phase
EppCommandCreateLaunchRegistration launchExt =
    new EppCommandCreateLaunchRegistration();
launchExt.type("registration");
EppLaunchPhase _phase = new EppLaunchPhase();
_phase.phase("claims");
launchExt.phase(_phase);

// if the domain has a claim, specify the claim information here
launchExt.acceptedDate(acceptedDate.toString());
launchExt.noticeID("noticeID1234");
launchExt.notAfter(notAfterDate.toString());

// 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;
EppResponseDataCreateLaunchRegistration launchResponseExt = null;
```

Continued on next page

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

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;
    }
}
```

5.2.2 Check Domain Name [EppCommandCheck, EppCommandCheckLaunchRegistration]

During GA Claims, two types of domain checks are supported. Each is described below.

The availability check command is used to determine whether or not a domain is available to be provisioned. Additionally, the Check command has also been extended to support claim checking during GA Claims. If a claim exists in the TMCH for a domain, the Claim check command provides a claim key to the registrar. The claim key is then used to query the TMCH for the claim notice. Querying for the claim notice is outside the EPP API. Please see this document for further details: http://tools.ietf.org/html/draft-lozano-tmch-func-spec-08

If a domain has a claim in the TMCH, the claim acknowledgement must be presented with the domain create.

5.2.2.1 Required Attributes

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

5.2.2.2 Data Returned

- Availability Check - A true or false based upon whether the Domain Name is available or not.
- Claims Check – A true or false if a claim exists for the domain. If a claim does exist, the claim key is returned as well

5.2.2.3 Usage – Availability Check

```
EppCommandCheckDomain cmd = new EppCommandCheckDomain();

EppCommandCheckLaunchRegistration launchExt = new
    EppCommandCheckLaunchRegistration();

// Create the extension to set the type and phase
// Set the phase and type – for an availability check
launchExt.type("avail");
EppLaunchPhase _phase = new EppLaunchPhase();
_phase.phase("claims");
launchExt.phase(_phase);
cmd.addExtension(launchExt);

// add the domains to be checked
cmd.add("example1.GACLAIM");
cmd.add("example2.GACLAIM");
// 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.GACLAIM") )
{
    // example1.GACLAIM is available
}
else
{
    // example1.GACLAIM is not available
    if( data.getReason("example1.GACLAIM ") != null )
    {
        // display reason text, if any
    }
}
```

5.2.2.4 Usage – Claim Check

```
EppCommandCheckDomain cmd = new EppCommandCheckDomain();
String domainName = null;
String claimKey = null;

// add the domains to be checked to the command
cmd.add("example1.GACLAIM");
cmd.add("example2.GACLAIM");

// add the domains to be checked to a list so that we can check // them when we get a response
ArrayList domains = new ArrayList();
domains.add("example1.GACLAIM");
domains.add("example2.GACLAIM");


EppCommandCheckLaunchRegistration launchExt =
    new EppCommandCheckLaunchRegistration();

// Set the phase and type for claim check
EppLaunchPhase _phase = new EppLaunchPhase();
_phase.phase("claims");
launchExt.phase(_phase);
cmd.addExtension(launchExt);

EppChannel channel = null;
EppExtension extension = null;
HashMap claimsKeys = null;
HashMap exists = null;

EppResponseDataCheckClaims claimsExt = null;

EppResponse res = channel.send(cmd);

EppResponseDataCheck checkResponse =
    (EppResponseDataCheck) res.getResponseData();

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

for( int i = 0; i < extensions.size(); i++ )
{
    extension = (EppExtension) extensions.get(i);
    if (extension instanceof EppResponseDataCheckClaims)
    {
           claimsExt = (EppResponseDataCheckClaims) extension;
           // this hashmap contains the domain name and the claim key
           claimsKeys = claimsExt.getClaimKeyHashMap();
           // spin through the list of domains we checked
           for (int j = 0; j < domains.size(); j++) {
               domainName = (String) domains.get(j);
               // see if a claim was found for the domain
              if (claimsExt.isAvailable(domainName))
              {
              // get the claims key
              claimKey = (String)
              claimsKeys.get(domainName);
              }
           }
    }
}
```

5.2.3 Info Domain Name [EppCommandInfo, EppCommandInfoLaunchRegistration, EppResponseDataInfoLaunchRegistration]

The info command returns domain attributes

5.2.3.1 Required Attributes

- Domain Name

5.2.3.2 Usage

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

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

EppExtension extension = null;
String status = 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();
    }
}
```

6 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.