

NEUSTAR REGISTRAR REFERENCE GUIDE



LANDRUSH SUPPLEMENT

October 11, 2013

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.

© 2013 Neustar, Inc. All rights reserved.

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

<b>1</b>	<b>Introduction .....</b>	<b>1</b>
1.1	Purpose .....	1
1.2	Intended Audience & Reading Suggestions .....	1
1.3	Reference Documents .....	1
1.4	Multi-application Award Method .....	1
1.5	The Trademark Clearinghouse.....	2
1.6	Signed Mark Data .....	2
1.7	Claims.....	2
1.8	Landrush Process Description .....	3
1.9	Landrush Process Flow.....	5
<b>2</b>	<b>Landrush Billing .....</b>	<b>6</b>
2.1	Billing.....	6
<b>3</b>	<b>Reporting .....</b>	<b>6</b>
3.1	Registrar Landrush Reports .....	6
3.2	Daily Activity Report.....	6
3.3	Daily Phase Summary Report.....	6
3.4	Daily Phase Activity Report.....	7
<b>4</b>	<b>Landrush Business Rules.....</b>	<b>7</b>
4.1	Business Rules for Registrars .....	7
4.2	Landrush Statuses .....	8
<b>5</b>	<b>Toolkit API Classes .....</b>	<b>9</b>
5.1	EppCommandCreateLaunchRegistration.....	9
5.2	EppCommandCheckLaunchRegistration .....	9
5.3	EppCommandInfoLaunchRegistration.....	9
5.4	EppStatus .....	9

5.5	EppLaunchPhase.....	9
5.6	EppResponseDataCreateLaunchRegistration.....	9
5.7	EppResponseDataInfoLaunchRegistration.....	9
<b>6</b>	<b>Operations .....</b>	<b>10</b>
6.1	Object Management Operations .....	10
6.2	Domain Management Operations .....	10
<b>7</b>	<b>Support .....</b>	<b>17</b>

## **Table of Figures and Tables**

Table 1	Registry Landrush Phase Participants and Roles .....	3
Figure 1	Landrush Process Flow .....	5
Table 2	Landrush Reporting Launch Phase Status .....	6
Table 3	Launch Phase Status and Corresponding EPP Status During Landrush .....	8
Table 1	Object Management Operations .....	10
Table 2	EPP and Launch Status Values.....	16

# 1 Introduction

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

The Landrush phase provides anyone wishing to register a name access to the namespace to potentially more valuable names prior to the Top Level Domain (TLD) entering General Availability. All registrations during Landrush are subject to the Trademark Claims process.

## 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 Landrush phase. In Landrush, multiple applications are accepted for a domain. If multiple applications exist for the same domain, the domain is awarded by auction or other third party decision. Landrush differs from a MULTI Sunrise phase in that no trademark data is required, but the application must go through that claims process and present a claim acknowledgement if a claim exists in the TMCH for the domain. 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 Landrush 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 launchphase 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 creation in Sunrise. <http://tools.ietf.org/html/draft-lozano-tmch-smd-03>
- TMCH functional specifications describes both registry and registrar interfaces with the TMCH. <http://tools.ietf.org/html/draft-lozano-tmch-func-spec-08>

## 1.4 Multi-application Award Method

Landrush is always operated using the Multi-application Award Method. The term *Application* refers to a phase where multiple requests for the same domain are accepted, and the winners

are determined by a defined process, such as an auction. During the Landrush phase, all applications for domain names are subject to the Trademark Claims process—hereafter referred to as *Claims*. Requests for the same name are accepted during this period. At the end of the Landrush 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. 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 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 Claims

Claims provides notice to prospective domain name registrants that the particular domain name they wish to register has a trademark claim against it. The claims notice is provided to the prospective registrant prior to the defined process for multiple applications for the same domain name. 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.

Table 1 provides descriptions of the participants and roles during the Registry Landrush phase.

Table 1 Registry Landrush Phase Participants and Roles

Participant	Role(s)
<b>Applicants / Registrants</b>	<ul style="list-style-type: none"> <li>• Submit applications to registrars</li> <li>• Approve [or not] registration of a domain name after being shown a claims notice</li> </ul>
<b>Registrars</b>	<ul style="list-style-type: none"> <li>• Collect Applications from applicants and resellers</li> <li>• Submit them to the Registry via EPP</li> <li>• Communicate with applicants</li> <li>• Display claims notices to applicants</li> <li>• Bills applicants</li> </ul>
<b>Registry Operator</b>	<ul style="list-style-type: none"> <li>• Defines business rules for the Landrush phase based on ICANN requirements</li> </ul>
<b>Registry Service Provider (RSP)</b>	<ul style="list-style-type: none"> <li>• Acts on behalf of the Registry Operator during Landrush to implement business rules</li> <li>• Accepts Registration Requests via EPP</li> <li>• Checks to see if a claims notice exists for a requested domain name and informs the registrar</li> <li>• Maintains the authoritative Landrush database for Valid applicationsInvoices Registrars</li> <li>• Injects winning Registrations into the production SRS</li> <li>• Provides Registrar reports and Poll Messages</li> </ul>
<b>Claim</b>	<ul style="list-style-type: none"> <li>• Is associated with a trademark owner who has rights to a particular domain name</li> </ul>
<b>Trademark Clearinghouse (TMCH)</b>	<ul style="list-style-type: none"> <li>• Performs trademark validation services</li> <li>• Issues Signed Mark Data (SMD)</li> <li>• Retains claims notices</li> <li>• Manages the appeals process</li> </ul>

## 1.8 Landrush Process Description

The Landrush process is detailed below.

The applicant for the domain name may be an individual, reseller, or some other agent. All winning Registrations incur a Registration Fee when the domain is injected (registered) in the production SRS. Detailed billing information is provided below.

The stages of the Landrush phase are as follows:

1. The applicant submits an application via the Registrar.
2. The SRS checks to see if the name is either already taken or on the reserved list. If the name is unavailable, the application is rejected. If the name is available, the registrar must request of the RSP if the name has a claim against it. If the RSP responds that there is a claim against the applied for name, then the registrar must get the claim from the TMCH and display it to the prospective registrant. The prospective registrant must

acknowledge reading the claim before registering the name and/or participating in a multi-application process.

3. If there is no claim against the name, then the multi-application process simply proceeds without a claim notice being shown to the prospective registrant. Once the multi-application process is complete, the winning names are injected into the General Availability SRS. If there were any names that had claims against them when injected, the TMCH notifies the rightful owner that a domain name was registered on which they have a trademark claim.
4. There are no limitations on the length of the Landrush phase, but all names applied for during Landrush must be subject to Claims. Once the Landrush phase ends, the Registry Operator determines the winner of all applications for the same domain name. All winning applications are injected. All single name applications are injected as well. 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;
5. This injection process runs until all applications have been injected into the GA system.
6. 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 be live in the DNS system. The registration date of the domain corresponds to the injection date.

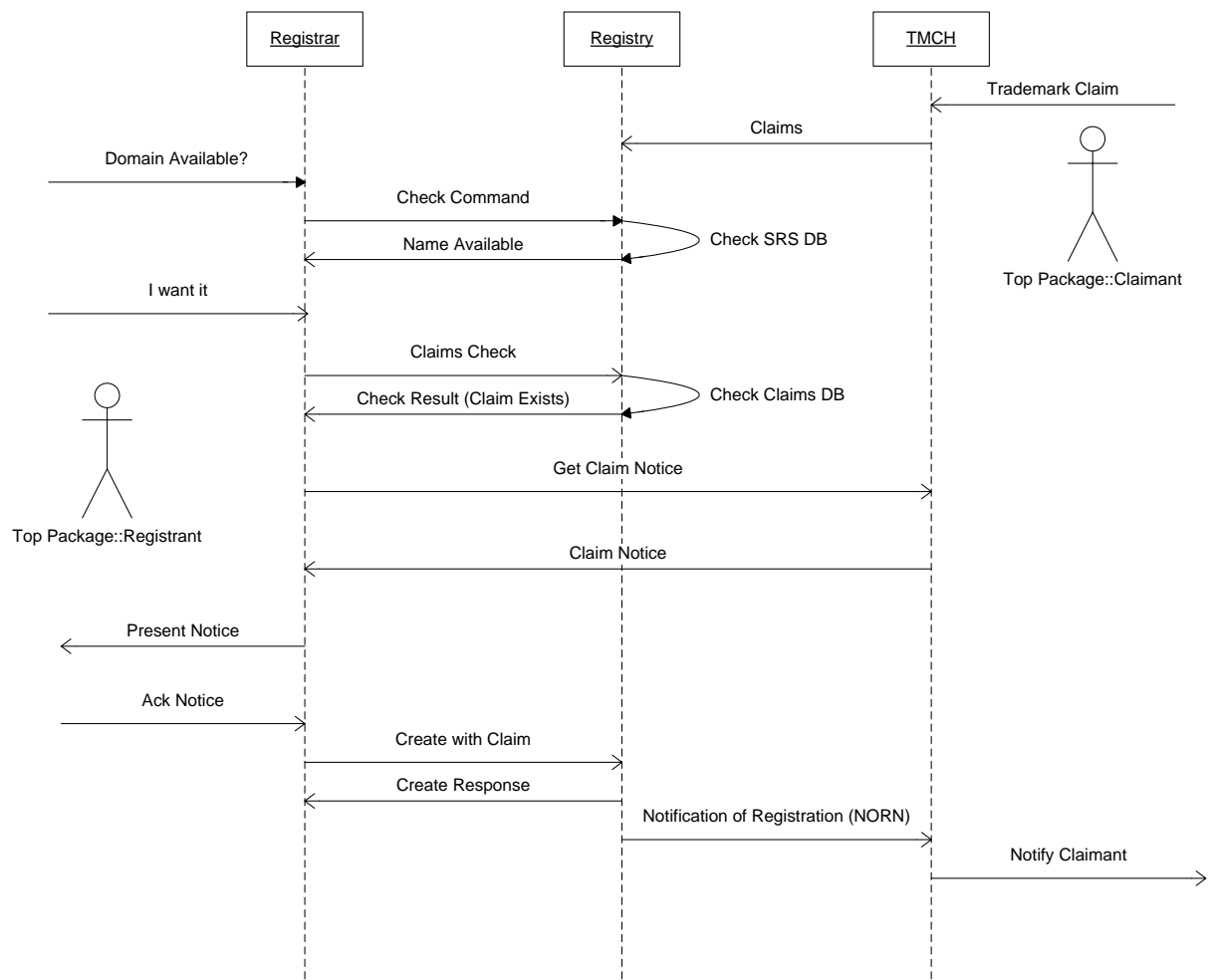


Landrush Registrations are considered non-refundable and Registrars are not permitted to delete registrations. Domain names created during the Landrush phase may not be deleted during the Landrush phase. If there is a need to delete a domain name that was created during the Landrush Phase, then that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



## 1.9 Landrush Process Flow

Figure 1 Landrush Process Flow



## 2 Landrush Billing

### 2.1 Billing

For each Landrush phase, the single billing event to the Registrar is the *Registration Fee*, which is determined by the Registry Operator.

## 3 Reporting

### 3.1 Registrar Landrush Reports

Several performance reports are provided to the Registrar.

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

Table 2 Landrush Reporting Launch Phase Status

Launch Phase Status	Description
<b>Validated</b>	Any claim against the domain name was successfully accepted by at least one prospective the registrant
<b>Allocated</b>	The domain name has been successfully allocated/injected into the GA system.
<b>Rejected</b>	The domain was rejected and not injected into the GA system.
<b>Invalid</b>	The domain name was invalid and was not validated by Claims

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.

An example of the Daily Phase Summary Report is depicted below.

## 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 Landrush 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 Landrush phase, a limited set of business rules are permitted and only these transactions are permitted to Registrars. During the Contact Create periods before Landrush opens:

- Contact Create
- Contact Check
- Contact Info

Registrars may perform the following contact operations during Landrush Periods:

- Contact Create
- Contact Check
- Contact Info

Registrars may perform the following domain operations during Landrush Periods:

- Domain Create
- Domain Check

Quiet periods are the periods after Landrush close 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

## 4.2 Landrush Statuses

During normal GA registry operations, domain names may be put into various EPP statuses after operations are performed. In Landrush, 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 Landrush there are also Launch Phase Statuses. Table 3 below depicts the Launch Phase Status along with the EPP status and description.

**Table 3 Launch Phase Status and Corresponding EPP Status During Landrush**

<b>Launch Phase Status</b>	<b>EPP Status</b>	<b>Description</b>
<b>Validated</b>	pendingCreate	Any claim against the domain name was successfully accepted by at least one prospective the registrant.
<b>Allocated</b>	Inactive	The domain name has been successfully allocated/injected into the GA system.
<b>Rejected</b>	Inactive	The domain was rejected and not injected into the GA system.
<b>Invalid</b>	inactive	The domain name was invalid and/or no claims notices were accepted by prospective registrants.

## 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 Landrush Application object and specifies the phase, registration type and claims data required for Landrush phases. For Landrush, the type is “application”.

### 5.2 EppCommandCheckLaunchRegistration

This class is the required command extension to do an availability check on a domain while Landrush is operational. This class also supports the claims check command. Claim checking is required during Landrush. If a domain is found to have a claim in the TMCH, the claim acknowledgement must be provided with the domain create.

### 5.3 EppCommandInfoLaunchRegistration

This class is the required command extension to perform a query, or info, on a Landrush Application object. The application ID must be specified.

### 5.4 EppStatus

This class contains the status information of an EPP Object.

### 5.5 EppLaunchPhase

The launch phase must be set for Landrush commands to let the server know the target phase of the command. The launch sub-phase is “landrush” and the launch phase is “claims”.

### 5.6 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 Landrush sunrise application.

### 5.7 EppResponseDataInfoLaunchRegistration

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

## 6 Operations

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

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 Landrush.

**Table 4 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.

### 6.2 Domain Management Operations

#### 6.2.1 Create Domain Name [EppCommandCreate, EppCommandCreateLaunchRegistration, EppResponseDataCreateLaunchRegistration]

This operation enters the Domain Name into the system.

##### 6.2.1.1 Required Attributes

Domain Name

Admin Contact ID

Technical Contact ID

Billing Contact ID

Registrant ID

Launch Phase = claim, sub phase = landrush

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

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

### 6.2.1.6 Usage

```
EppDomain domain = new EppDomain("example1.LRCLAIM");
// 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 subphase
EppCommandCreateLaunchRegistration launchExt =
    new EppCommandCreateLaunchRegistration();
launchExt.type("application");
EppLaunchPhase _phase = new EppLaunchPhase();
_phase.phase("claims");
_phase.subPhase("landrush");
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;
String applicationId = 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;  
        applicationId = launchResponseExt.applicationId();  
    }  
}
```

## 6.2.2 Check Domain Name [EppCommandCheck, EppCommandCheckLaunchRegistration]

During Landrush, 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. For Landrush, the command checks both Landrush and GA environments if applicable.

Additionally, the Check command has also been extended to support claim checking during Landrush. 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.

### 6.2.2.1 Required Attributes

Domain Name

Extension to check command

Check type – avail or claims

Phase – claims

Subphase - landrush

### 6.2.2.2 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

### 6.2.2.3 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");
_phase.subPhase("landrush");
launchExt.phase(_phase);
cmd.addExtension(launchExt);

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

### 6.2.2.4 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.EXAMPLE");
cmd.add("example2.EXAMPLE");

// 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.EXAMPLE");
domains.add("example2.EXAMPLE");

EppCommandCheckLaunchRegistration launchExt =
    new EppCommandCheckLaunchRegistration();
```

Continued on next page

```

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

```

### 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 Usage

```

EppCommandInfoDomain cmd =
new EppCommandInfoDomain("example1.LRCLAIM");

// 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

launchExt.includeMark(true);
launchExt.applicationId(applicationID);
EppLaunchPhase _phase = new EppLaunchPhase();
_phase.phase("claims");
_phase.subPhase("landrush");
launchExt.phase(_phase);
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;
        applicationId = launchResponseExt.applicationId();
        status = launchResponseExt.status();
    }
}

```

The table below outlines both the EPP and Launch status values associated with a Landrush application as it progresses through the Landrush process.

**Table 5 EPP and Launch Status Values**

Action	EPP Status	Launch Action
Create Domain	pendingCreate	validated
Send to Auction or Third Party	pendingCreate	pendingAllocation
Inject Winner	inactive	allocated
Mark Loser	Inactive	rejected

When a Landrush Application is transitioned from one Launch status to another as it moves through the lifecycle, a poll message is put on the queue to signal the change of status.

The registrar receives notification of the status change via Poll Messages. 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. Below is an example of a poll message after the domain has been determined to be the winner, and is injected into the GA system.

The following is an example <domain:panData> poll message for an "allocated" Launch Application.

```
S:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S:  <response>
S:    <result code="1301">
S:      <msg>Command completed successfully; ack to dequeue</msg>
S:    </result>
S:    <msgQ count="5" id="12345">
S:      <qDate>2013-04-04T22:01:00.0Z</qDate>
S:      <msg>Registration successfully allocated.</msg>
S:    </msgQ>
S:    <resData>
S:      <domain:panData
S:        xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
S:        <domain:name paResult="1">example1.LRCLAIM</domain:name>
S:        <domain:paTRID>
S:          <clTRID>ABC-12345</clTRID>
S:          <svTRID>54321-XYZ</svTRID>
S:        </domain:paTRID>
S:        <domain:paDate>2013-04-04T22:00:00.0Z</domain:paDate>
S:      </domain:panData>
S:    </resData>
S:    <extension>
S:      <launch:infData
S:        xmlns:launch="urn:ietf:params:xml:ns:launch-1.0">
S:        <launch:phase name="landrush">claims</launch:phase>
S:        <launch:status s="allocated"/>
S:      </launch:infData>
S:    </extension>
S:    <trID>
S:      <clTRID>BCD-23456</clTRID>
S:      <svTRID>65432-WXY</svTRID>
S:    </trID>
S:  </response>
S:</epp>
```

## 7 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](mailto:reg-support@neustar.biz).