



# .CAM Registry

Technical description

Version 1.10

17<sup>th</sup> February 2017



## Table of contents

1	Change log .....	5
2	Scope .....	6
3	Initial Setup.....	7
4	Extensible Provisioning Protocol (EPP).....	8
4.1	RFC Relevance to KSregistry (KSR) .....	8
4.1.1	RFC 5730 .....	8
4.1.2	RFC 5731 .....	8
4.1.2.1	EPP poll example for successful domain transfer .....	9
4.1.2.2	EPP poll example for failed domain transfer.....	9
4.1.3	RFC 5732 .....	10
4.1.4	RFC 5733 .....	11
4.1.5	RFC 5734 .....	11
4.1.6	RFC 5735 .....	11
4.1.7	RFC 3915 .....	11
4.1.8	RFC 5910 .....	12
4.2	Extensions used by KSR and Related Internet Drafts .....	13
4.2.1	Draft-tan-epp-launchphase-12 (Launch Phase Mapping for the EPP).....	13
4.2.2	EPP launchphase examples .....	14
4.2.2.1	Check claims in the claims phase (sunrise, landrush, LRP, open) .....	14
4.2.2.2	Create domain in the sunrise phase .....	15
4.2.2.3	Create domain in the landrush phase .....	16
4.2.2.4	Create domain in the landrush phase (with claim) .....	18
4.2.2.5	Create domain in the LRP phase .....	19
4.2.2.6	Create domain in the LRP phase (with claim).....	20
4.2.2.7	Create IDN domain in the LRP phase .....	22
4.2.2.8	Create IDN domain in the LRP phase (with claim).....	23
4.2.2.9	Create domain in the claims phase .....	24
4.2.2.10	Info domain in the sunrise phase.....	25
4.2.2.11	Delete domain in the sunrise phase .....	27
4.2.2.12	Poll message for successful Landrush Application.....	28
4.2.2.13	Poll message for failed Landrush Application .....	28
4.2.2.14	Poll message for successful LRP Application .....	29
4.2.2.15	Poll message for failed LRP Application.....	30
4.2.3	Draft-obispo-epp-idn-04 (Internationalized Domain Name Mapping Extension for the EPP) 31	
4.2.3.1	Create IDN domain.....	31
4.2.3.2	Info IDN domain.....	32
4.2.4	Draft-brown-epp-fees-04 (Registry Fee Extension for EPP).....	33
4.2.4.1	Check domain .....	34
4.2.4.2	Create domain .....	36
4.2.4.3	Create domain application .....	37
4.2.4.4	Update domain.....	38
4.2.4.5	Info domain.....	38
4.2.4.6	Renew domain .....	39
4.2.4.7	Delete domain .....	40
4.2.4.8	Restore domain .....	40



4.2.4.9	Transfer domain.....	41
4.2.5	EPP Balance commands.....	42
4.2.5.1	Balance:info.....	43
4.2.5.2	Low balance poll event.....	44
4.3	Supported EPP commands.....	45
4.3.1	Supported EPP commands in general availability.....	45
4.3.2	Launchphase details.....	45
4.4	EPP object relationship.....	46
5	WHOIS (RDDS) service.....	47
5.1	Abuse Protections.....	47
5.2	Searchable RDDS (WHOIS).....	47
5.3	Whois Output Fields.....	48
5.3.1	Query on domain name.....	48
5.3.2	Query on name server name.....	49
5.3.3	Query on registrar.....	50
6	The domain Life Cycle.....	52
6.1	Registration Life Cycle Periods.....	52
6.1.1	Available.....	52
6.1.2	Add-Grace Period (AGP).....	52
6.1.3	Registered.....	53
6.1.4	Expired.....	53
6.1.5	Auto-Renew Grace Period (ARGP).....	53
6.1.6	Redemption Grace Period (RGP).....	53
6.1.7	Pending Delete.....	54
6.1.8	Released.....	54
6.2	Domain Name Operations.....	54
6.2.1	Create Domain.....	54
6.2.2	Delete Domain.....	54
6.2.3	Transfer Domain.....	55
6.2.4	Update Domain.....	55
6.2.5	Renew Domain.....	55
6.2.6	Restore Domain.....	56
6.3	Domain Name Statuses.....	56
6.3.1	ok.....	57
6.3.2	inactive.....	57
6.3.3	clientHold.....	57
6.3.4	clientUpdateProhibited.....	57
6.3.5	clientTransferProhibited.....	57
6.3.6	clientDeleteProhibited.....	57
6.3.7	clientRenewProhibited.....	58
6.3.8	serverHold.....	58
6.3.9	serverUpdateProhibited.....	58
6.3.10	serverTransferProhibited.....	58
6.3.11	serverDeleteProhibited.....	58
6.3.12	serverRenewProhibited.....	58
6.3.13	redemptionPeriod.....	59
6.3.14	pendingRestore.....	59
6.3.15	pendingDelete.....	59
6.3.16	pendingTransfer.....	59



6.3.17	addPeriod.....	59
6.3.18	autoRenewPeriod.....	59
6.4	Reserved Premium Domain Names .....	59
7	Internationalised Domain Names (IDN).....	61
7.1	IETF Standards .....	61
7.2	General rules .....	61
7.2.1	Protocol validity .....	61
7.2.2	NON-LDH (Letter Digit Hyphen) Check .....	61
7.2.3	Hyphen check.....	61
7.2.4	Protocol enforcement .....	61
7.2.5	Length check .....	61
7.3	Language tag definition .....	61
7.3.1	Allowed code points.....	62
7.4	Example of EPP extension block.....	62
8	SFTP download area for registrars .....	63
8.1	Folder invoice .....	63
8.2	Folder reports .....	63
8.3	Folder statements .....	63
9	Appendix A.....	64
10	Appendix B.....	67



## 1 Change log

Change date	Chapter	Description
2014-07-04	4.1.2	Corrected description of allowed/required authcode values
2014-07-04	4.1.3	Corrected description of IP limitation for host-object
2014-07-04	4.1.4	Added authcode description for contacts
2014-07-07	4.1.2.1	Added EPP poll examples for successful transfer
2014-07-07	4.1.2.2	Added EPP poll examples for failed transfer
2014-07-07	4.2.2.8	Added Poll message for successful Landrush Application
2014-07-07	4.2.2.9	Poll message for failed Landrush Application
2015-05-28	4.2	Added examples for LRP phase (modified structure of chapter 4.2)
2015-05-28	4.3.2	Added LRP parameters
2015-10-08	4.2.3, 7.4	Reworked the IDN description
2015-10-09	4.2.4	Added the Fee extension
2016-09-06	8	Added new Chapter for SFTP area
2016-09-06	9	Moved Chapter 8 to Chapter 9
2016-28-06	4.2.4	Added description for activating the Fee extension
2016-28-06	4.2.5	Added EPP balance commands
2017-02-17	7.3.1	Updated description where to find the IDN tables
2017-02-17	7.3.2	Removed Listing of "Allowed code points"



## 2 Scope

This document describes all technical aspects registrars need in order to connect to the KSregistry system (KSR) and to manage their domains. The KSR is in full compliance with the ICANN requirements defined in the gTLD Application Guidebook for new TLDs. The following aspects will be covered:

- Initial Setup
- Extensible Provisioning Protocol (EPP)
- WHOIS (RDDS) service
- The domain Life Cycle
- Internationalised Domain Names (IDN)

Besides the production system KSR grants each registrar access to the OT&E system. In the OT&E system there will be 2 accounts available for transfer testing.



### 3 Initial Setup

Each registrar in the SRS environment will be entitled to up to five sessions from two different IP addresses. The registrar will be forced to update the registry password for the EPP servers and registrar extranet (see below) at least once every six months.

In order to signup with the registry please send fill out the registrar signup form and send an email to [info@nic.cam](mailto:info@nic.cam).

See Appendix A for a template for the signup form.

The KSR system can be reached by the following URLs:

#### **Production System:**

Registrar Frontend:	<a href="https://login.ksregistry.net/">https://login.ksregistry.net/</a>
EPP: epp.ksregistry.net	Port 700 (SSL)
SSL Client Certificate required	No
WHOIS (Port 43 and web based):	whois.nic.cam
SFTP (Port 2222):	sftp.ksregistry.net

#### **OT&E System:**

Registrar Frontend:	<a href="https://login-ote.ksregistry.net/">https://login-ote.ksregistry.net/</a>
EPP: epp-ote.ksregistry.net	Port 700 (SSL)
SSL Client Certificate required	No
WHOIS (Port 43 and web based):	whois-ote.ksregistry.net
SFTP (Port 2222):	sftp-ote.ksregistry.net



## 4 Extensible Provisioning Protocol (EPP)

The EPP server is set up as a cluster to guarantee a high availability solution. The KSR EPP API is offered over TCP on port 700 with mandatory SSL session enforcement for registrars for automated interaction with username and password.

To increase security, a registrar IP address limitation is in place for the EPP servers (both production and OT&E). This API also supports a secure web-based (over https) EPP client for registrars' manual use only. The web-based graphical interface interacts with the EPP server through standard EPP XML queries. The EPP XML responses are in turn displayed in the web interface. This allows the registrars to perform registry transactions through the web-based interface.

KSR does not offer a registrar tool kit or Java EPP client.

### 4.1 RFC Relevance to KSRregistry (KSR)

#### 4.1.1 RFC 5730

This RFC is a base protocol document for EPP. EPP is an XML-text object based client-server protocol, atomic in its transactions, and developed to support multiple transports and lower level security protocols. There are no partial failures; all commands either succeed or fail definitively. Object-to-object associations are standard with limited application of parent-child relationships where delegate relationships are necessary for affected functionality, such as internal host data and its relationship to domain objects. The KSR registry fully implements the service discovery, commands, responses, and the extension framework described.

<http://tools.ietf.org/html/rfc5730>

#### 4.1.2 RFC 5731

This RFC explains the mapping of the primary EPP registry object, the domain object. It reviews associated attributes and states of the domain object as well as child object relationships (hosts). It also details associations with other contact objects. KSR complies with the full XML examples and descriptions and applies flexibility where permitted. For example, 5731 allows operators to implement the info command with different responses for a “sponsoring registrar” and a “non-sponsoring registrar” in regards to most domain object attributes. KSR implements this as a base protocol document for EPP.

<http://tools.ietf.org/html/rfc5731>

Additional information:

Q: What is the minimum and maximum length of the AuthInfo for each object?

A: Minimum 6, Maximum 16

Q: What specific characters are allowed or required?

A: Allowed 1234567890A-Za-z@%\$.:;,!





A: Required 2 special characters and one number

#### 4.1.2.1 *EPP poll example for successful domain transfer*

```
<?xml version="1.0" encoding="UTF-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1301">
      <msg>Command completed successfully; ack to dequeue</msg>
    </result>
    <msgQ count="1" id="2">
      <qDate>2014-05-02T10:05:28.0Z</qDate>
      <msg>DOMAIN_TRANSFER_SUCCESSFUL; TRANSFER registrar1 to
registrar2</msg>
    </msgQ>
    <resData>
      <domain:trnData xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>domain.tld</domain:name>
        <domain:trStatus>clientApproved</domain:trStatus>
        <domain:reID>registrar2</domain:reID>
        <domain:reDate>2014-05-02T09:00:41.0Z</domain:reDate>
        <domain:acID>registrar1</domain:acID>
        <domain:acDate>2014-05-02T10:05:28.0Z</domain:acDate>
      </domain:trnData>
    </resData>
    <trID>
      <cI TRID>123</cI TRID>
      <sv TRID>123</sv TRID>
    </trID>
  </response>
</epp>
```

#### 4.1.2.2 *EPP poll example for failed domain transfer*

```
<?xml version="1.0" encoding="UTF-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1301">
      <msg>Command completed successfully; ack to dequeue</msg>
    </result>
    <msgQ count="1" id="3">
      <qDate>2014-05-05T08:23:17.0Z</qDate>
      <msg>DOMAIN_TRANSFER_FAILED; TRANSFER registrar1 to registrar2</msg>
    </msgQ>
    <resData>
      <domain:trnData xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>domain.tld</domain:name>
      </domain:trnData>
    </resData>
  </response>
</epp>
```



```
<domain:trStatus>clientRejected</domain:trStatus>
<domain:reID>registrar2</domain:reID>
<domain:reDate>2014-05-05T08:21:59.0Z</domain:reDate>
<domain:acID>registrar1</domain:acID>
<domain:acDate>2014-05-05T08:23:17.0Z</domain:acDate>
</domain:trnData>
</resData>
<trID>
  <c|TRID>123</c|TRID>
  <svTRID>123</svTRID>
</trID>
</response>
</epp>
```

### 4.1.3 RFC 5732

KSR implements this as a base protocol document for EPP. KSR notes this RFC describes the mapping of relationships to host objects, which are by definition subordinate to the superordinate domain name object. Host objects that are defined as internal or in the namespace of the registry must be related to a superordinate domain object to be created. Internal hosts, as full child objects, face restrictions associated with the management of their superordinate domain object. External hosts are hosts belonging to another domain namespace and as such are not subordinate in the present namespace. Internal hosts can have a glue or an A record associated with them, external hosts refer to another namespace or zone for the associated A record.

<http://tools.ietf.org/html/rfc5732>

Additional information:

Fact: Host Attributes are not supported (only Host objects)

Q: What is the minimum and maximum length of a host?

A: Minimum 1, Maximum 223

A: How many IPs can be listed with a single host object:

Q: 1 -13

Q: Can duplicate IPs be listed between different host objects for the same domain?

A: Yes

Q: Can duplicate IPs be listed between different host objects for the same domain?

A: No



#### **4.1.4 RFC 5733**

Another base RFC implemented in the KSR server, this RFC describes the contact object mappings in EPP. Contact objects are used to contain related data surrounding the standardized contacts types in TLD registries including attributes such as contact type, country, telephone numbers, email addresses, etc. As a standalone object, a contact object can be created and associated with no domain objects or with any number of domain objects available in the registry. This is used commonly by registrars to update common contact information associated across large numbers of domains in a single transaction. Like the domain object, it can be secured with a passphrase or “authinfo” code. Contact object data represents the definitive data source for authoritative RDDS (WHOIS) in new TLDs.

<http://tools.ietf.org/html/rfc5733>

Additional information:

Q: What is the minimum and maximum length of the AuthInfo for each object?

A: Minimum 6, Maximum 16

Q: What specific characters are allowed or required?

A: Allowed 1234567890A-Za-z@%\$.:;,!

A: Required 2 special characters and one number

#### **4.1.5 RFC 5734**

KSR will implement this RFC as the preferred industry transport and in compliance with ICANN's requirements. Early implementations of EPP were considered over BEEP. This RFC describes a standard implementation of TCP incorporating TLS. As mentioned earlier, EPP can be implemented over multiple transports. The transport of choice for the EPP registry community has been TCP. Implementers are encouraged to take precautions against denial of service attacks through the use of standard technologies such as firewall and border router filters. IANA awarded port 700 as the dedicated port for the server side. There is no dedicated port assignment for the client side.

<http://tools.ietf.org/html/rfc5734>

#### **4.1.6 RFC 5735**

KSR will implement this RFC as applicable to any extensions it utilizes as this RFC provides specific and detailed guidance on EPP extensions. An important principle in creating extensions to, as opposed to modifying, the EPP protocol was to fully preserve the integrity of the existing protocol schema. Additionally, a valid extension itself should be extensible. Another important requirement in the RFC is to include announcements of all available extensions in the EPP server greeting element before establishing an interactive client session.

<http://tools.ietf.org/html/rfc5735>

#### **4.1.7 RFC 3915**

KSR will support this extension since this TLD implements the grace period implementation known as the Redemption Grace Period or “RGP”. When RGP is in use, domains are deleted into the RGP



where Registrars may request a restoration of the domain. This is a billable event and requires a three-step process: placement of the domain into a pending restore state, submission of a restore report explaining why the domain is being restored, and finally the restoration of the domain. The RFC extends the domain update command, adds related domain statuses, such as "redemptionPeriod" and "pendingRestore," and extends the responses of domain info and other details. The RFC provides a lifecycle description of the RGP and defines the format and content for client to server submission of the associated restore reports.

<http://tools.ietf.org/html/rfc3915>

Additional information:

Fact: A restore does not require a restore report and will be completed immediately.

#### **4.1.8 RFC 5910**

KSR will support DNSSEC from the initiation of this TLD and therefore will also support this extension from initiation of the registration process. DNSSEC is a mechanism for cryptographically verifying that each delegate zone in the DNS hierarchy has been referred to or is referring to its genuine parent or child zone respectively. Since TLD zone files are generated from authoritative registry data, this extension specifically provides the ability to add elements to the domain-create and domain-update functions and to the domain-info responses, allowing registrars to submit associated delegated signer information of the child zone indicating it is digitally signed and that the parent zone recognizes the indicated key as a valid zone key for the child zone.

<http://tools.ietf.org/html/rfc5910>

Additional information:

Fact: DNSSEC zone file configuration of the domain is not required prior to sending DNSSEC EPP

Q: What version of DNSSEC is being implemented?

A: 1.1

Q: Will the Registry require DS Data for DNSSEC?

A: Yes (Both Key Data and DS Data is supported)

Q: What are the valid algorithm values?

A: 3,4,5,6,7,8,10,12

Q: Is the maxsiglife attribute enabled?

A: No



## **4.2 Extensions used by KSR and Related Internet Drafts**

### **4.2.1 Draft-tan-epp-launchphase-12 (Launch Phase Mapping for the EPP)**

KSR uses this EPP internet draft to facilitate Sunrise phases during the initiation of this TLD registry. This internet draft proposes an extension mechanism that supports the organization of Sunrise related domain applications

<http://tools.ietf.org/html/draft-tan-epp-launchphase-12>

Please note the following important information:

Only base64 encoded signed mark allowed. The maximum line length are 76 characters.



## 4.2.2 EPP launchphase examples

### 4.2.2.1 Check claims in the claims phase (sunrise, landrush, LRP, open)

Request:

```
<?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">

  <command>
    <check>
      <domain:check 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>example1.tld</domain:name>
        <domain:name>example2.tld</domain:name>
        <domain:name>example3.tld</domain:name>
      </domain:check>
    </check>
    <extension>
      <launch:check xmlns:launch="urn:ietf:params:xml:ns:launch-1.0"
xsi:schemaLocation="urn:ietf:params:xml:ns:launch-1.0 launch-1.0.xsd"
type="claims">
        <launch:phase>claims</launch:phase>
      </launch:check>
    </extension>
    <clTRID>C6A6B9E1-1AF3-44BB-81F0-41CE758FC1CC</clTRID>
  </command>
</epp>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <extension>
      <launch:chkData xmlns:launch="urn:ietf:params:xml:ns:launch-1.0">
        <launch:phase>claims</launch:phase>
        <launch:cd>
          <launch:name exists="0">example1.tld</launch:name>
        </launch:cd>
        <launch:cd>

```



```
<launch:name exists="1">example2.tld</launch:name>
```

```
<launch:claimKey>/G4yEcl0t/RaMgER0rjkxx16xLVEW9e/WkVj7gx+T306STwj1n3yNGcZKIEV9ZLOP inOb1S5WWJQ
vJ7W+3SunA== </launch:claimKey>
  </launch:cd>
  <launch:cd>
    <launch:name exists="0">example3.tld</launch:name>
  </launch:cd>
</launch:chkData>
</extension>
<trID>
  <cITRID>C6A6B9E1-1AF3-44BB-81F0-41CE758FC1CC</cITRID>
  <svTRID>2595c13c-5019-4a46-88e1-434997ae81e7</svTRID>
</trID>
</response>
</epp>
```

#### 4.2.2.2 Create domain in the sunrise phase

Request:

**Important Info: Only base64 encoded signed mark allowed. The maximum line length are 76 characters.**

```
<?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>example.tld</domain:name>
        <domain:registrar>EXAMPLE-0001</domain:registrar>
        <domain:contact type="admin">EXAMPLE-0001</domain:contact>
        <domain:contact type="tech">EXAMPLE-0001</domain:contact>
        <domain:contact type="billing">EXAMPLE-0001</domain:contact>
        <domain:authInfo>
          <domain:pw>2fooBAR!</domain:pw>
        </domain:authInfo>
      </domain:create>
    </create>
    <extension>
      <launch:create
        xmlns:launch="urn:ietf:params:xml:ns:launch-1.0">
        <launch:phase>sunrise</launch:phase>
        <smd:encodedSignedMark
          xmlns:smd="urn:ietf:params:xml:ns:signedMark-1.0">
```

Marks: Example

smdID: 00000012345678901234567-65535

U-labels: example

notBefore: 2014-01-01 00:00:00



```
notAfter: 2018-01-01 00:00:00
-----BEGIN ENCODED SMD-----
[base64 encoded signed mark]
-----END ENCODED SMD-----
  </smd:encodedSignedMark>
  </launch:create>
</extension>
<cITRID>C6A6B9E1-1AF3-44BB-81F0-41CE758FC1CD</cITRID>
</command>
</epp>
```

#### Response:

```
<?xml version="1.0" encoding="UTF-8"?>

<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1001">
      <msg>Command completed successfully; action pending</msg>
    </result>
    <resData>
      <domain:creData xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>example.tld</domain:name>
        <domain:crDate>2014-02-11T11:19:52.0Z</domain:crDate>
      </domain:creData>
    </resData>
    <extension>
      <launch:creData xmlns:launch="urn:ietf:params:xml:ns:launch-1.0">
        <launch:phase>sunrise</launch:phase>
        <launch:applicationID>123</launch:applicationID>
      </launch:creData>
    </extension>
    <trID>
      <cITRID>C6A6B9E1-1AF3-44BB-81F0-41CE758FC1CD</cITRID>
      <svTRID>25c95d76-2d30-4c2d-b029-17a6a9946da5</svTRID>
    </trID>
  </response>
</epp>
```

### 4.2.2.3 Create domain in the landrush phase

#### Request:

```
<?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>example.tld</domain:name>
  <domain:registrar>EXAMPLE-0001</domain:registrar>
  <domain:contact type="admin">EXAMPLE-0001</domain:contact>
  <domain:contact type="tech">EXAMPLE-0001</domain:contact>
  <domain:contact type="billing">EXAMPLE-0001</domain:contact>
  <domain:authInfo>
    <domain:pw>2fooBAR!</domain:pw>
  </domain:authInfo>
</domain:create>
</create>
<extension>
  <launch:create
    xmlns:launch="urn:ietf:params:xml:ns:launch-1.0">
    <launch:phase>landrush</launch:phase>
  </launch:create>
</extension>
<cITRID>C6A6B9E1-1AF3-44BB-81F0-41CE758FC1C1</cITRID>
</command>
</epp>
```

## Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1001">
      <msg>Command completed successfully; action pending</msg>
    </result>
    <resData>
      <domain:creData xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>keysys-test-domain-006.technology</domain:name>
        <domain:crDate>2014-02-11T16:35:54.0Z</domain:crDate>
      </domain:creData>
    </resData>
    <extension>
      <launch:creData xmlns:launch="urn:ietf:params:xml:ns:launch-1.0">
        <launch:phase>landrush</launch:phase>
        <launch:applicationID>10000499</launch:applicationID>
      </launch:creData>
    </extension>
    <trID>
      <cITRID>C6A6B9E1-1AF3-44BB-81F0-41CE758FC1C1</cITRID>
      <svTRID>1b82cb58-1d97-4813-9718-16b3baa39528</svTRID>
    </trID>
  </response>
</epp>
```



#### 4.2.2.4 Create domain in the landrush phase (with claim)

Request:

```
<?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>example.tld</domain:name>
        <domain:registrar>EXAMPLE-0001</domain:registrar>
        <domain:contact type="admin">EXAMPLE-0001</domain:contact>
        <domain:contact type="tech">EXAMPLE-0001</domain:contact>
        <domain:contact type="billing">EXAMPLE-0001</domain:contact>
        <domain:authInfo>
          <domain:pw>2fooBAR!</domain:pw>
        </domain:authInfo>
      </domain:create>
    </create>
    <extension>
      <launch:create
        xmlns:launch="urn:ietf:params:xml:ns:launch-1.0">
        <launch:phase name="landrush">claims</launch:phase>
        <launch:notice>
          <launch:noticeID validatorID="tmch">ec0ea0ce000000000000000012345</launch:noticeID>
          <launch:notAfter>2014-01-01T00:00:00.0Z</launch:notAfter>
          <launch:acceptedDate>2014-01-24T00:00:00.0Z</launch:acceptedDate>
        </launch:notice>
      </launch:create>
    </extension>
    <clTRID>C6A6B9E1-1AF3-44BB-81F0-41CE758FC1C1</clTRID>
  </command>
</epp>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>

<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
```



```
<response>
  <result code="1001">
    <msg>Command completed successfully; action pending</msg>
  </result>
  <resData>
    <domain:creData xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
      <domain:name>keysys-test-domain-006.technology</domain:name>
      <domain:crDate>2014-02-11T16:35:54.0Z</domain:crDate>
    </domain:creData>
  </resData>
  <extension>
    <launch:creData xmlns:launch="urn:ietf:params:xml:ns:launch-1.0">
      <launch:phase name="landrush">claims</launch:phase>
      <launch:applicationID>10000499</launch:applicationID>
    </launch:creData>
  </extension>
  <trID>
    <cITRID>C6A6B9E1-1AF3-44BB-81F0-41CE758FC1C1</cITRID>
    <svTRID>1b82cb58-1d97-4813-9718-16b3baa39528</svTRID>
  </trID>
</response>
</epp>
```

#### 4.2.2.5 Create domain in the LRP phase

Request:

```
<?xml version="1.0"?>
<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">
  <command>
    <create>
      <domain:create 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>example.tld</domain:name>
        <domain:period unit="y">2</domain:period>
        <domain:ns>
          <domain:hostObj>ns1.ksregistry.tld</domain:hostObj>
        </domain:ns>
        <domain:registrant>P-INA22</domain:registrant>
        <domain:contact type="admin">P-INA22</domain:contact>
        <domain:contact type="tech">P-INA22</domain:contact>
        <domain:contact type="billing">P-INA22</domain:contact>
        <domain:authInfo>
          <domain:pw>fdg.Z546$</domain:pw>
        </domain:authInfo>
      </domain:create>
    </create>
  </command>
</epp>
```



```
<launch:create xmlns:launch="urn:ietf:params:xml:ns:launch-1.0">
  <launch:phase name="lrp">custom</launch:phase>
</launch:create>
</extension>
<cITRID>F381F161-F477-48AC-B088-52CF54BEF723</cITRID>
</command>
</epp>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <resData>
      <domain:creData xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>example.tld</domain:name>
        <domain:crDate>2015-05-20T05:42:21.0Z</domain:crDate>
      </domain:creData>
    </resData>
    <extension>
      <launch:creData xmlns:launch="urn:ietf:params:xml:ns:launch-1.0">
        <launch:phase name="lrp">custom</launch:phase>
        <launch:applicationID>13</launch:applicationID>
      </launch:creData>
    </extension>
    <trID>
      <cITRID>F381F161-F477-48AC-B088-52CF54BEF723</cITRID>
      <svTRID>906ddd10-56f7-4be7-91bc-9665a82ad2ea</svTRID>
    </trID>
  </response>
</epp>
```

#### 4.2.2.6 Create domain in the LRP phase (with claim)

Request:

```
<?xml version="1.0"?>
<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">
  <command>
    <create>
      <domain:create 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>example.tld</domain:name>
        <domain:period unit="y">2</domain:period>
        <domain:ns>
          <domain:hostObj>ns1.ksregistry.tld</domain:hostObj>
        </domain:ns>
      </domain:create>
    </create>
  </command>
</epp>
```



```
<domain:registrar>P-INA22</domain:registrar>
<domain:contact type="admin">P-INA22</domain:contact>
<domain:contact type="tech">P-INA22</domain:contact>
<domain:contact type="billing">P-INA22</domain:contact>
<domain:authInfo>
  <domain:pw>fdg,Z546$</domain:pw>
</domain:authInfo>
</domain:create>
</create>
<extension>
  <launch:create xmlns:launch="urn:ietf:params:xml:ns:launch-1.0">
    <launch:phase name="lrp">custom</launch:phase>
    <launch:notice>
      <launch:noticeID>ec0ea0ce000000000000000012345</launch:noticeID>
      <launch:notAfter>2015-05-17T00:00:00.0Z</launch:notAfter>
      <launch:acceptedDate>2015-06-16T00:00:00.0Z</launch:acceptedDate>
    </launch:notice>
  </launch:create>
</extension>
<clTRID>F381F161-F477-48AC-B088-52CF54BEF723</clTRID>
</command>
</epp>
```

## Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <resData>
      <domain:creData xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>example.tld</domain:name>
        <domain:crDate>2015-05-20T05:43:22.0Z</domain:crDate>
      </domain:creData>
    </resData>
    <extension>
      <launch:creData xmlns:launch="urn:ietf:params:xml:ns:launch-1.0">
        <launch:phase name="lrp">custom</launch:phase>
        <launch:applicationID>14</launch:applicationID>
      </launch:creData>
    </extension>
    <trID>
      <clTRID>F381F161-F477-48AC-B088-52CF54BEF723</clTRID>
      <svTRID>d28fc19e-7f75-463b-81f9-d599d9fe2698</svTRID>
    </trID>
  </response>
</epp>
```



#### 4.2.2.7 Create IDN domain in the LRP phase

Request:

```
<?xml version="1.0"?>
<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">
  <command>
    <create>
      <domain:create 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>xn--mcxmller-95a.tld</domain:name>
        <domain:period unit="y">2</domain:period>
        <domain:ns>
          <domain:hostObj>ns1.ksregistry.tld</domain:hostObj>
        </domain:ns>
        <domain:registrar>P-INA22</domain:registrar>
        <domain:contact type="admin">P-INA22</domain:contact>
        <domain:contact type="tech">P-INA22</domain:contact>
        <domain:contact type="billing">P-INA22</domain:contact>
        <domain:authInfo>
          <domain:pw>fdg.Z546$</domain:pw>
        </domain:authInfo>
      </domain:create>
    </create>
    <extension>
      <idn:data xmlns:idn="urn:ietf:params:xml:ns:idn-1.0">
        <idn:table>GER</idn:table>
        <idn:uname>mcxmüller.tld</idn:uname>
      </idn:data>
      <launch:create xmlns:launch="urn:ietf:params:xml:ns:launch-1.0">
        <launch:phase name="lrp">custom</launch:phase>
      </launch:create>
    </extension>
    <cI TRID>F381F161-F477-48AC-B088-52CF54BEF723</cI TRID>
  </command>
</epp>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <resData>
      <domain:creData xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>xn--mcxmller-95a.tld</domain:name>
        <domain:crDate>2015-05-19T09:58:50.0Z</domain:crDate>
      </domain:creData>
    </resData>
  </response>
</epp>
```



```
</resData>
<extension>
  <launch:creData xmlns:launch="urn:ietf:params:xml:ns:launch-1.0">
    <launch:phase name="lrp">custom</launch:phase>
    <launch:applicationID>11</launch:applicationID>
  </launch:creData>
</extension>
<trID>
  <cITRID>F381F161-F477-48AC-B088-52CF54BEF723</cITRID>
  <svTRID>14e1211e-4cc1-4677-b4aa-faf1f9478509</svTRID>
</trID>
</response>
</epp>
```

#### 4.2.2.8 Create IDN domain in the LRP phase (with claim)

Request:

```
<?xml version="1.0"?>
<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">
  <command>
    <create>
      <domain:create 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>xn--ncxmüller-95a.tld</domain:name>
        <domain:period unit="y">2</domain:period>
        <domain:ns>
          <domain:hostObj>ns1.ksregistry.tld</domain:hostObj>
        </domain:ns>
        <domain:registrant>P-INA22</domain:registrant>
        <domain:contact type="admin">P-INA22</domain:contact>
        <domain:contact type="tech">P-INA22</domain:contact>
        <domain:contact type="billing">P-INA22</domain:contact>
        <domain:authInfo>
          <domain:pw>fdg.Z546$</domain:pw>
        </domain:authInfo>
      </domain:create>
    </create>
  <extension>
    <idn:data xmlns:idn="urn:ietf:params:xml:ns:idn-1.0">
      <idn:table>GER</idn:table>
      <idn:uname>ncxmüller.tld</idn:uname>
    </idn:data>
    <launch:create xmlns:launch="urn:ietf:params:xml:ns:launch-1.0">
      <launch:phase name="lrp">custom</launch:phase>
      <launch:notice>
        <launch:noticeID>ec0ea0ce0000000000000000000012345</launch:noticeID>
        <launch:notAfter>2015-05-17T00:00:00.OZ</launch:notAfter>
        <launch:acceptedDate>2015-06-16T00:00:00.OZ</launch:acceptedDate>
      </launch:notice>
    </launch:create>
```



```
</extension>
<cITRID>F381F161-F477-48AC-B088-52CF54BEF723</cITRID>
</command>
</epp>
```

#### Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <resData>
      <domain:creData xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>xn--ncxml1er-95a.tld</domain:name>
        <domain:crDate>2015-05-20T05:43:22.0Z</domain:crDate>
      </domain:creData>
    </resData>
    <extension>
      <launch:creData xmlns:launch="urn:ietf:params:xml:ns:launch-1.0">
        <launch:phase name="lrp">custom</launch:phase>
        <launch:applicationID>14</launch:applicationID>
      </launch:creData>
    </extension>
    <trID>
      <cITRID>F381F161-F477-48AC-B088-52CF54BEF723</cITRID>
      <svTRID>d28fc19e-7f75-463b-81f9-d599d9fe2698</svTRID>
    </trID>
  </response>
</epp>
```

#### 4.2.2.9 Create domain in the claims phase

#### Request:

```
<?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>example.tld</domain:name>
        <domain:registrar>EXAMPLE-0001</domain:registrar>
        <domain:contact type="admin">EXAMPLE-0001</domain:contact>
        <domain:contact type="tech">EXAMPLE-0001</domain:contact>
        <domain:contact type="billing">EXAMPLE-0001</domain:contact>
        <domain:authInfo>
          <domain:pw>2fooBAR!</domain:pw>
        </domain:authInfo>
      </domain:create>
    </create>
  </command>
</epp>
```





```
</domain:authInfo>
</domain:create>
</create>
<extension>
  <launch:create
    xmlns:launch="urn:ietf:params:xml:ns:launch-1.0">
    <launch:phase>claims</launch:phase>
    <launch:notice>
      <launch:noticeID validatorID="tmch">ec0ea0ce000000000000000012345</launch:noticeID>
      <launch:notAfter>2014-01-01T00:00:00.0Z</launch:notAfter>
      <launch:acceptedDate>2014-01-24T00:00:00.0Z</launch:acceptedDate>
    </launch:notice>
  </launch:create>
</extension>
<cITRID>C6A6B9E1-1AF3-44BB-81F0-41CE758FC1C2</cITRID>
</command>
</epp>
```

#### Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <resData>
      <domain:creData xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>example.tld</domain:name>
        <domain:crDate>2014-02-01T00:00:00.0Z</domain:crDate>
        <domain:exDate>2015-02-01T00:00:00.0Z</domain:exDate>
      </domain:creData>
    </resData>
    <trID>
      <cITRID>C6A6B9E1-1AF3-44BB-81F0-41CE758FC1C2</cITRID>
      <svTRID>d09183a2-6276-4a50-acf6-7734408ffc40</svTRID>
    </trID>
  </response>
</epp>
```

#### 4.2.2.10 Info domain in the sunrise phase

Only includeMark="false" is allowed!

#### Request:



```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <info>
      <domain:info xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>example.tld</domain:name>
      </domain:info>
    </info>
    <extension>
      <launch:info xmlns:launch="urn:ietf:params:xml:ns:launch-1.0"
        includeMark="false">
        <launch:phase>sunrise</launch:phase>
        <launch:applicationID>123</launch:applicationID>
      </launch:info>
    </extension>
    <clTRID>C6A6B9E1-1AF3-44BB-81F0-41CE758FC1CF</clTRID>
  </command>
</epp>
```

#### Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <resData>
      <domain:infData xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>example.tld</domain:name>
        <domain:roid>1234567890_DOMAPP-KEYSYS</domain:roid>
        <domain:status s="pendingCreate"/>
        <domain:registrar>EXAMPLE-0001</domain:registrar>
        <domain:contact type="admin">EXAMPLE-0001</domain:contact>
        <domain:contact type="tech">EXAMPLE-0001</domain:contact>
        <domain:contact type="billing">EXAMPLE-0001</domain:contact>
        <domain:clID>1234</domain:clID>
        <domain:crDate>2014-02-01T00:00:00.0Z</domain:crDate>
      </domain:infData>
    </resData>
    <extension>
      <launch:infData xmlns:launch="urn:ietf:params:xml:ns:launch-1.0">
        <launch:phase>sunrise</launch:phase>
      </launch:infData>
    </extension>
  </response>
</epp>
```



```
<launch:applicationID>123</launch:applicationID>
<launch:status s="pendingValidation"/>
</launch:infData>
</extension>
<trID>
  <cITRID>C6A6B9E1-1AF3-44BB-81F0-41CE758FC1CF</cITRID>
  <svTRID>51172b78-f383-4874-8b3d-61617c8139af</svTRID>
</trID>
</response>
</epp>
```

#### 4.2.2.11 Delete domain in the sunrise phase

Request:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <delete>
      <domain:delete
        xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>example.tld</domain:name>
      </domain:delete>
    </delete>
    <extension>
      <launch:delete
        xmlns:launch="urn:ietf:params:xml:ns:launch-1.0">
        <launch:phase>sunrise</launch:phase>
        <launch:applicationID>123</launch:applicationID>
      </launch:delete>
    </extension>
    <cITRID>62CB2558-A467-495E-89BF-3ACF9F1EAF9B</cITRID>
  </command>
</epp>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <trID>
      <cITRID>62CB2558-A467-495E-89BF-3ACF9F1EAF9B</cITRID>
    </trID>
  </response>
</epp>
```



```
<svTRID>0ee118e6-155e-4ffe-9fb0-1db13d24c5f2</svTRID>
</trID>
</response>
</epp>
```

#### 4.2.2.12 Poll message for successful Landrush Application

Poll message:

```
<?xml version="1.0" encoding="UTF-8"?>

<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1301">
      <msg>Command completed successfully; ack to dequeue</msg>
    </result>
    <msgQ count="7" id="12">
      <qDate>2014-06-12T05:33:22.0Z</qDate>
      <msg>DOMAIN_APPLICATION_SUCCESSFUL</msg>
    </msgQ>
    <resData>
      <domain:panData xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name paResult="1">domain.tld</domain:name>
        <domain:paTRID>
          <svTRID>UNDEF</svTRID>
        </domain:paTRID>
        <domain:paDate>2014-06-12T05:33:22.0Z</domain:paDate>
      </domain:panData>
    </resData>
    <extension>
      <launch:infData xmlns:launch="urn:ietf:params:xml:ns:launch-1.0">
        <launch:phase>landrush</launch:phase>
        <launch:applicationID>12</launch:applicationID>
        <launch:status s="allocated"/>
      </launch:infData>
    </extension>
  </trID>
  <c|TRID>123</c|TRID>
  <svTRID>123</svTRID>
</trID>
</response>
</epp>
```

#### 4.2.2.13 Poll message for failed Landrush Application

Poll message:



```
<?xml version="1.0" encoding="UTF-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <resData>
      <domain:infData xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>domain.tld</domain:name>
        <domain:roid>XXX</domain:roid>
        <domain:status s="pendingCreate"/>
        <domain:registrant>XXX</domain:registrant>
        <domain:contact type="admin">XXX</domain:contact>
        <domain:contact type="tech">XXX</domain:contact>
        <domain:contact type="billing">XXX</domain:contact>
        <domain:ns>
          <domain:hostObj>ns1.domain.com</domain:hostObj>
        </domain:ns>
        <domain:clID>123</domain:clID>
        <domain:crDate>2014-06-06T21:52:48.0Z</domain:crDate>
      </domain:infData>
    </resData>
    <extension>
      <launch:infData xmlns:launch="urn:ietf:params:xml:ns:launch-1.0">
        <launch:phase>landrush</launch:phase>
        <launch:applicationID>14</launch:applicationID>
        <launch:status s="rejected"/>
      </launch:infData>
    </extension>
    <trID>
      <clTRID>123</clTRID>
      <svTRID>123</svTRID>
    </trID>
  </response>
</epp>
```

#### 4.2.2.14 Poll message for successful LRP Application

```
<?xml version="1.0" encoding="UTF-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1301">
      <msg>Command completed successfully; ack to dequeue</msg>
    </result>
    <msgQ count="1" id="2">
      <qDate>2015-05-28T06:41:08.0Z</qDate>
      <msg>DOMAIN_APPLICATION_SUCCESSFUL</msg>
    </msgQ>
    <resData>
      <domain:panData xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name paResult="1">example.tld</domain:name>
      </domain:panData>
    </resData>
  </response>
</epp>
```



```
<domain:paTRID>
  <svTRID>UNDEF</svTRID>
</domain:paTRID>
<domain:paDate>2015-05-28T06:41:08.0Z</domain:paDate>
</domain:panData>
</resData>
<extension>
  <launch:infData xmlns:launch="urn:ietf:params:xml:ns:launch-1.0">
    <launch:phase name="lrp">custom</launch:phase>
    <launch:applicationID>10</launch:applicationID>
    <launch:status s="allocated"/>
  </launch:infData>
</extension>
<trID>
  <cITRID>8F564950-181C-4B9B-8677-6CA6189E5ED1</cITRID>
  <svTRID>b98c2e1a-d526-4e01-bda7-240043cff373</svTRID>
</trID>
</response>
</epp>
```

#### 4.2.2.15 Poll message for failed LRP Application

```
<?xml version="1.0" encoding="UTF-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1301">
      <msg>Command completed successfully; ack to dequeue</msg>
    </result>
    <msgQ count="1" id="5">
      <qDate>2015-05-28T10:32:47.0Z</qDate>
      <msg>DOMAIN_APPLICATION_FAILED; Data was incorrect</msg>
    </msgQ>
    <resData>
      <domain:panData xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name paResult="0">example.tld</domain:name>
        <domain:paTRID>
          <svTRID>UNDEF</svTRID>
        </domain:paTRID>
        <domain:paDate>2015-05-28T10:32:47.0Z</domain:paDate>
      </domain:panData>
    </resData>
    <extension>
      <launch:infData xmlns:launch="urn:ietf:params:xml:ns:launch-1.0">
        <launch:phase name="lrp">custom</launch:phase>
        <launch:applicationID>11</launch:applicationID>
        <launch:status s="rejected"/>
      </launch:infData>
    </extension>
    <trID>
      <cITRID>8F564950-181C-4B9B-8677-6CA6189E5ED1</cITRID>
      <svTRID>aadd753b-fa84-4240-a975-7ed220626ca1</svTRID>
    </trID>
  </response>
</epp>
```



```
</trID>  
</response>  
</epp>
```

### 4.2.3 Draft-obispo-epp-idn-04 (Internationalized Domain Name Mapping Extension for the EPP)

KSR uses this EPP internet draft to facilitate the usage of provisioning Internationalized Domain Names (IDNs). This internet draft extends the EPP domain name mapping to provide additional features that are required to implement registrations of domain names in character sets other than ASCII.

<https://tools.ietf.org/html/draft-obispo-epp-idn-04>

#### 4.2.3.1 Create IDN domain

If the domain name is an IDN, the EPP command MUST contain an

<extension> element, which MUST contain a child <idn:data> element with the following child elements:

- A <idn:table> element that contains the IDN table identifier as provided by the server.
- An optional <idn:uname> element that contains the domain name to be registered in Unicode NFC.

Request:

```
<?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>xn--exmple2-6wa.tld</domain:name>  
        <domain:period unit="y">2</domain:period>  
        <domain:ns>  
          <domain:hostObj>ns1.example.net</domain:hostObj>  
          <domain:hostObj>ns2.example.net</domain:hostObj>  
          <domain:hostObj>ns3.example.net</domain:hostObj>  
        </domain:ns>  
        <domain:registrant>P-MKM22</domain:registrant>  
        <domain:contact type="admin">P-MKM22</domain:contact>  
        <domain:contact type="tech">P-MKM22</domain:contact>  
        <domain:contact type="billing">P-MKM22</domain:contact>  
        <domain:authInfo>  
          <domain:pw>!$D2fooBAR</domain:pw>  
        </domain:authInfo>  
      </domain:create>  
    </create>  
    <extension>
```



```
<idn:data xmlns:idn="urn:ietf:params:xml:ns:idn-1.0">
  <idn:table>GER</idn:table>
  <idn:uname>example2.tld</idn:uname>
</idn:data>
</extension>
<clTRID>KSregistry-12345</clTRID>
</command>
</epp>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <resData>
      <domain:creData xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>xn--example2-6wa.tld</domain:name>
        <domain:crDate>2015-10-08T08:48:39.0Z</domain:crDate>
        <domain:exDate>2017-10-08T08:48:39.0Z</domain:exDate>
      </domain:creData>
    </resData>
    <trID>
      <clTRID>KSregistry-12345</clTRID>
      <svTRID>79fcbfe6-26ad-48aa-803b-2991a1a63914</svTRID>
    </trID>
  </response>
</epp>
```

#### 4.2.3.2 Info IDN domain

This extension does not add any elements to the EPP <info> command, but does include elements in the response, when the extension has been selected during a <login> command.

Request:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <info>
      <domain:info xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>xn--example-cua.tld</domain:name>
      </domain:info>
    </info>
    <clTRID>KSregistry-12345</clTRID>
  </command>
</epp>
```





Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <resData>
      <domain:infData xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>xn--exmple-cua.tld</domain:name>
        <domain:roid>1158915893_DOMAIN-TLD</domain:roid>
        <domain:status s="serverTransferProhibited"/>
        <domain:registrar>P-MKM22</domain:registrar>
        <domain:contact type="admin">P-MKM22</domain:contact>
        <domain:contact type="tech">P-MKM22</domain:contact>
        <domain:contact type="billing">P-MKM22</domain:contact>
        <domain:ns>
          <domain:hostObj>NS3.EXAMPLE.NET</domain:hostObj>
          <domain:hostObj>NS1.EXAMPLE.NET</domain:hostObj>
          <domain:hostObj>NS2.EXAMPLE.NET</domain:hostObj>
        </domain:ns>
        <domain:clID>tld-ksr</domain:clID>
        <domain:crID>tld-ksr</domain:crID>
        <domain:crDate>2015-10-08T08:38:41.0Z</domain:crDate>
        <domain:upID>tld-ksr</domain:upID>
        <domain:upDate>2015-10-08T08:38:41.0Z</domain:upDate>
        <domain:exDate>2017-10-08T08:38:41.0Z</domain:exDate>
        <domain:authInfo>
          <domain:pw>!As2$2fooBAR</domain:pw>
        </domain:authInfo>
      </domain:infData>
    </resData>
    <extension>
      <idn:data xmlns:idn="urn:ietf:params:xml:ns:idn-1.0">
        <idn:table>GER</idn:table>
        <idn:uname>exämple.tld</idn:uname>
      </idn:data>
    </extension>
    <trID>
      <clTRID>KSregistry-12345</clTRID>
      <svTRID>22660179-2c20-49c4-ad1f-05e880a1e05a</svTRID>
    </trID>
  </response>
</epp>
```

#### 4.2.4 Draft-brown-epp-fees-04 (Registry Fee Extension for EPP)

Given the present expansion of the DNS namespace, and the proliferation of novel business models,



it is now desirable to provide a method for EPP clients to query EPP servers for the fees and credits associated with certain commands and specific objects.

The public Internet-Draft can be found at: <https://tools.ietf.org/html/draft-brown-epp-fees-04>

The following optional settings from draft have **not** been implemented:

- no grace periods have been implemented:

```
<fee:fee grace-period="P5D">5.00</fee:fee>
```

-no fee:balance has been implemented:

```
<fee:credit description="AGP Credit">-5.00</fee:credit>  
<fee:balance>1005.00</fee:balance>
```

Before the Fee extension can be used by registrars through EPP the premium handling needs to be activated first. The activation has to be made through the registrar web interface at:

Account->Settings : Tab Premium domains

If the Premium handling is set to “inactive” the Fee extension can not be used by EPP and domain:Check commands will always mark the domain as not available!

Premium domains can always be registered by using the web interface even if the Premium handling is set to “inactive”.

#### 4.2.4.1 Check domain

Request:

```
<?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">  
  <command>  
    <check>  
      <domain:check 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>00a.tld</domain:name>  
      </domain:check>  
    </check>  
    <extension>  
      <fee:check xmlns:fee="urn:ietf:params:xml:ns:fee-0.7"  
xsi:schemaLocation="urn:ietf:params:xml:ns:fee-0.7 fee-0.7.xsd">  
        <fee:domain>  
          <fee:name>00a.tld</fee:name>  
          <fee:command>create</fee:command>  
          <fee:period unit="y">1</fee:period>  
        </fee:domain>  
        <fee:domain>  
          <fee:name>00a.tld</fee:name>
```



```
<fee:command>renew</fee:command>
<fee:period unit="y">5</fee:period>
</fee:domain>
<fee:domain>
  <fee:name>00a.tld</fee:name>
  <fee:command>transfer</fee:command>
  <fee:period unit="y">1</fee:period>
</fee:domain>
<fee:domain>
  <fee:name>00a.tld</fee:name>
  <fee:command>restore</fee:command>
  <fee:period unit="y">1</fee:period>
</fee:domain>
</fee:check>
</extension>
<clTRID>KSregistry-12345</clTRID>
</command>
</epp>
```

#### Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <resData xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
      <domain:chkData>
        <domain:cd>
          <domain:name avail="1">00a.tld</domain:name>
        </domain:cd>
      </domain:chkData>
    </resData>
    <extension>
      <fee:chkData xmlns:fee="urn:ietf:params:xml:ns:fee-0.7">
        <fee:cd>
          <fee:name>00a.tld</fee:name>
          <fee:currency>EUR</fee:currency>
          <fee:command>create</fee:command>
          <fee:period unit="y">1</fee:period>
          <fee:fee refundable="1" description="Registration fee">237.0000</fee:fee>
          <fee:class>premium</fee:class>
        </fee:cd>
        <fee:cd>
          <fee:name>00a.tld</fee:name>
          <fee:currency>EUR</fee:currency>
          <fee:command>renew</fee:command>
          <fee:period unit="y">5</fee:period>
          <fee:fee refundable="1" description="Renewal fee">10.0000</fee:fee>
          <fee:class>premium</fee:class>
        </fee:cd>
      </fee:chkData>
    </extension>
  </response>
</epp>
```



```
<fee:cd>
  <fee:name>00a.tld</fee:name>
  <fee:currency>EUR</fee:currency>
  <fee:command>transfer</fee:command>
  <fee:period unit="y">1</fee:period>
  <fee:fee refundable="1" description="Transfer fee">2.0000</fee:fee>
  <fee:class>premium</fee:class>
</fee:cd>
<fee:cd>
  <fee:name>00a.tld</fee:name>
  <fee:currency>EUR</fee:currency>
  <fee:command>restore</fee:command>
  <fee:period unit="y">1</fee:period>
  <fee:fee refundable="1" description="Restore fee">100.0000</fee:fee>
  <fee:class>premium</fee:class>
</fee:cd>
</fee:chkData>
</extension>
<trID>
```

#### 4.2.4.2 Create domain

Request:

```
<?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>00a.tld</domain:name>
        <domain:period unit="y">1</domain:period>
        <domain:ns>
          <domain:hostObj>ns1.example.net</domain:hostObj>
          <domain:hostObj>ns2.example.net</domain:hostObj>
          <domain:hostObj>ns3.example.net</domain:hostObj>
        </domain:ns>
        <domain:registrant>P-MKM22</domain:registrant>
        <domain:contact type="admin">P-MKM22</domain:contact>
        <domain:contact type="tech">P-MKM22</domain:contact>
        <domain:contact type="billing">P-MKM22</domain:contact>
        <domain:authInfo>
          <domain:pw>!$d2!foo%BAR</domain:pw>
        </domain:authInfo>
      </domain:create>
    </create>
  </extension>
  <fee:create xmlns:fee="urn:ietf:params:xml:ns:fee-0.7">
    <fee:currency>EUR</fee:currency>
    <fee:fee>237.0000</fee:fee>
  </fee:create>
</extension>
```



```
<clTRID>KSregistry-12345</clTRID>
</command>
</epp>
```

#### Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <resData>
      <domain:creData xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>00a.tld</domain:name>
        <domain:crDate>2015-10-08T10:48:36.OZ</domain:crDate>
        <domain:exDate>2016-10-08T10:48:36.OZ</domain:exDate>
      </domain:creData>
    </resData>
    <trID>
      <clTRID>KSregistry-12345</clTRID>
      <svTRID>f29bd5af-224d-4382-ada8-9572205e81f6</svTRID>
    </trID>
  </response>
</epp>
```

#### 4.2.4.3 Create domain application

#### Request:

```
<?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>xn--ncxmller-95a.tld</domain:name>
        <domain:period unit="y">2</domain:period>
        <domain:ns>
          <domain:hostObj>ns1.example.net</domain:hostObj>
        </domain:ns>
        <domain:registrant>P-IUA24</domain:registrant>
        <domain:contact type="admin">P-IUA24</domain:contact>
        <domain:contact type="tech">P-IUA24</domain:contact>
        <domain:contact type="billing">P-IUA24</domain:contact>
        <domain:authInfo>
          <domain:pw>2f!oo$BAR</domain:pw>
        </domain:authInfo>
      </domain:create>
    </create>
    <extension>
      <launch:create xmlns:launch="urn:ietf:params:xml:ns:launch-1.0">
        <launch:phase name="landrush">claims</launch:phase>
      </launch:create>
    </extension>
  </command>
</epp>
```



```
<fee:create xmlns:fee="urn:ietf:params:xml:ns:fee-0.7">
  <fee:currency>EUR</fee:currency>
  <fee:fee>1195.5104</fee:fee>
</fee:create>
</extension>
<clTRID>KSregistry-12345</clTRID>
</command>
</epp>
```

Response:

#### 4.2.4.4 Update domain

The EPP update command does not support the fee extension in the request and response. The optional elements `<fee:currency>` and `<fee:fee>` are not supported.

#### 4.2.4.5 Info domain

The command MAY contain an `<extension>` element which MAY contain a `<fee:info>` element with the following child elements:

- An OPTIONAL `<fee:currency>` element;
- `<fee:command>` element;
- An OPTIONAL `<fee:period>` element.

Request:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <info>
      <domain:info
        xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>00a.tld</domain:name>
      </domain:info>
    </info>
    <extension>
      <fee:info xmlns:fee="urn:ietf:params:xml:ns:fee-0.7">
        <fee:currency>EUR</fee:currency>
        <fee:command>create</fee:command>
        <fee:period unit="y">1</fee:period>
      </fee:info>
    </extension>
    <clTRID>KSregistry-12345</clTRID>
  </command>
</epp>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
```



```
<msg>Command completed successfully</msg>
</result>
<resData>
  <domain:infData xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
    <domain:name>00a.tld</domain:name>
    <domain:roid>1159815893_DOMAIN-TLD</domain:roid>
    <domain:status s="serverTransferProhibited"/>
    <domain:registrar>P-MKM22</domain:registrar>
    <domain:contact type="admin">P-MKM22</domain:contact>
    <domain:contact type="tech">P-MKM22</domain:contact>
    <domain:contact type="billing">P-MKM22</domain:contact>
    <domain:ns>
      <domain:hostObj>NS1.EXAMPLE.NET</domain:hostObj>
      <domain:hostObj>NS2.EXAMPLE.NET</domain:hostObj>
      <domain:hostObj>NS3.EXAMPLE.NET</domain:hostObj>
    </domain:ns>
    <domain:clID>tld-registrar</domain:clID>
    <domain:crID>tld-registrar</domain:crID>
    <domain:crDate>2015-10-08T10:48:36.0Z</domain:crDate>
    <domain:upID>tld-registrar</domain:upID>
    <domain:upDate>2015-10-08T10:48:36.0Z</domain:upDate>
    <domain:exDate>2016-10-08T10:48:36.0Z</domain:exDate>
    <domain:authInfo>
      <domain:pw>!$d2!foo%BAR</domain:pw>
    </domain:authInfo>
  </domain:infData>
</resData>
<extension>
  <rgp:infData xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0">
    <rgp:rgpStatus s="addPeriod">endDate=2015-10-13T10:48:36.0Z</rgp:rgpStatus>
  </rgp:infData>
  <fee:infData xmlns:fee="urn:ietf:params:xml:ns:fee-0.7">
    <fee:currency>EUR</fee:currency>
    <fee:command>create</fee:command>
    <fee:period unit="y">1</fee:period>
    <fee:fee refundable="1" description="Registration fee">237.0000</fee:fee>
    <fee:class>premium</fee:class>
  </fee:infData>
</extension>
<trID>
  <clTRID>KSregistry-12345</clTRID>
  <svTRID>ab26bf20-9a72-4429-a859-73de7e5a3772</svTRID>
</trID>
</response>
</epp>
```

#### 4.2.4.6 Renew domain

The server does **not** include in the <extension> section of the EPP response a <fee:renData>

Request:



```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <renew>
      <domain:renew
        xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>00a.tld</domain:name>
        <domain:curExpDate>2016-10-08</domain:curExpDate>
        <domain:period unit="y">1</domain:period>
      </domain:renew>
    </renew>
    <extension>
      <fee:renew xmlns:fee="urn:ietf:params:xml:ns:fee-0.7">
        <fee:currency>EUR</fee:currency>
        <fee:fee>2.0000</fee:fee>
      </fee:renew>
    </extension>
    <clTRID>KSregistry-12345</clTRID>
  </command>
</epp>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <resData>
      <domain:renData xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>00a.tld</domain:name>
        <domain:exDate>2017-10-08T10:48:36.0Z</domain:exDate>
      </domain:renData>
    </resData>
    <trID>
      <clTRID>KSregistry-12345</clTRID>
      <svTRID>27bd0d63-1b4a-40f7-8e92-2b4461e3a7e5</svTRID>
    </trID>
  </response>
</epp>
```

#### 4.2.4.7 Delete domain

The EPP delete command does **not** support the fee extension in the request and response. The optional element <fee:delData> is **not** supported.

#### 4.2.4.8 Restore domain

Request:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
```





```
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <update>
      <domain:update xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>00a.tld</domain:name>
        <domain:chg/>
      </domain:update>
    </update>
    <extension>
      <rgp:update xmlns:rgp="urn:ietf:params:xml:ns:rgp-1.0">
        <rgp:restore op="request"/>
      </rgp:update>
      <fee:update xmlns:fee="urn:ietf:params:xml:ns:fee-0.7">
        <fee:currency>EUR</fee:currency>
        <fee:fee>100.0000</fee:fee>
      </fee:update>
    </extension>
    <clTRID>KSregistry-12345</clTRID>
  </command>
</epp>
```

Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1000">
      <msg>Command completed successfully</msg>
    </result>
    <trID>
      <clTRID>KSregistry-12345</clTRID>
      <svTRID>4c2c6f2c-6b60-4229-82d9-14f4b1f67f6d</svTRID>
    </trID>
  </response>
</epp>
```

#### 4.2.4.9 *Transfer domain*

The Transfer Request and Transfer Query Response do **not** contain an extension with a <fee:trnData> element.

Request:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <transfer op="request">
      <domain:transfer
        xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>3g.tld</domain:name>
        <domain:period unit="y">1</domain:period>
        <domain:authInfo>
          <domain:pw>!$hfjhhfd818kL</domain:pw>
        </domain:authInfo>
      </domain:transfer>
    </transfer>
  </command>
</epp>
```



```
</domain:authInfo>
</domain:transfer>
</transfer>
<extension>
  <fee:transfer xmlns:fee="urn:ietf:params:xml:ns:fee-0.7">
    <fee:currency>EUR</fee:currency>
    <fee:fee>2.0000</fee:fee>
  </fee:transfer>
</extension>
<clTRID>KSregistry-12345</clTRID>
</command>
</epp>
```

#### Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1001">
      <msg>Command completed successfully; action pending</msg>
    </result>
    <resData>
      <domain:trnData xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>3g.tld</domain:name>
        <domain:trStatus>pending</domain:trStatus>
        <domain:reID>tld-registrar</domain:reID>
        <domain:reDate>2015-10-08T11:31:34.0Z</domain:reDate>
        <domain:acID>tld-registrar1</domain:acID>
        <domain:acDate>2015-10-13T11:31:33.0Z</domain:acDate>
      </domain:trnData>
    </resData>
    <trID>
      <clTRID>KSregistry-12345</clTRID>
      <svTRID>7f5eeae9-31b1-4264-b6a2-6f2f178f223e</svTRID>
    </trID>
  </response>
</epp>
```

## 4.2.5 EPP Balance commands

In order to provide the account balance information also through the EPP interface we integrated two EPP extensions which have been drafted by Verisign.

### Low Balance Mapping (provided through EPP poll queue):

[http://www.verisign.com/assets/epp-sdk/verisign\\_epp-extension\\_low-balance\\_v00.html](http://www.verisign.com/assets/epp-sdk/verisign_epp-extension_low-balance_v00.html)

The poll message provides the same information which is send by mail through KSR whenever the account balance drops under the “Low-balance warning” setting. If no “Low-balance warning is



configured no mail and no poll message are send.

### **Balance Mapping (provided by balance:info command):**

[http://www.verisign.com/assets/epp-sdk/verisign\\_epp-extension\\_balance\\_v00.html](http://www.verisign.com/assets/epp-sdk/verisign_epp-extension_balance_v00.html)

EPP provides the <info> command that is used to retrieve client balance and other financial information.

#### **4.2.5.1 Balance:info**

Request:

```
<?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">
  <command>
    <info>
      <balance:info xmlns:balance="http://www.verisign.com/epp/balance-1.0"
xsi:schemaLocation="http://www.verisign.com/epp/balance-1.0 balance-1.0.xsd"/>
    </info>
    <clTRID>KSregistry-12345</clTRID>
  </command>
</epp>
```

Response:

```
<?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="1000">
      <msg>Command completed successfully</msg>
    </result>
    <resData>
      <balance:infData xmlns:balance="http://www.verisign.com/epp/balance-1.0"
xsi:schemaLocation="http://www.verisign.com/epp/balance-1.0 balance-1.0.xsd">
        <balance:creditLimit>500.00</balance:creditLimit>
        <balance:balance>8351.07</balance:balance>
        <balance:currency>EUR</balance:currency>
        <balance:availableCredit>8851.07</balance:availableCredit>
        <balance:creditThreshold>
          <balance:fixed>0.00</balance:fixed>
        </balance:creditThreshold>
      </balance:infData>
    </resData>
    <trID>
      <clTRID>KSregistry-12345</clTRID>
    </trID>
  </response>
</epp>
```



```
<svTRID>261fa5f6-d1f8-499a-97c3-6cfed6762daf</svTRID>
</trID>
</response>
</epp>
```

#### 4.2.5.2 Low balance poll event

EPP poll message:

```
<?xml version="1.0" encoding="UTF-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1301">
      <msg>Command completed successfully; ack to dequeue</msg>
    </result>
    <msgQ count="1" id="2">
      <qDate>2016-06-08T09:44:26.0Z</qDate>
      <msg>LOWBALANCE-POLL_BALANCE_NOTIFY; Low Account Balance.</msg>
    </msgQ>
    <resData>
      <lowbalance-poll:pollData xmlns:lowbalance-poll="http://www.verisign.com/epp/lowbalance-
poll-1.0">
        <lowbalance-poll:registrarName>testregistrar</lowbalance-poll:registrarName>
        <lowbalance-poll:creditLimit>1000.0000</lowbalance-poll:creditLimit>
        <lowbalance-poll:creditThreshold type="FIXED">50.0000</lowbalance-poll:creditThreshold>
        <lowbalance-poll:availableCredit>-3344.8909</lowbalance-poll:availableCredit>
        <lowbalance-poll:currency>USD</lowbalance-poll:currency>
      </lowbalance-poll:pollData>
    </resData>
  </trID>
  <c|TRID>KSregistry-12345</c|TRID>
  <svTRID>71fe77eb-dacc-4c7e-bf57-ff70977d2112</svTRID>
</trID>
</response>
</epp>
```



## 4.3 Supported EPP commands

### 4.3.1 Supported EPP commands in general availability

Command	Domain	Contact	Host
Create	Yes	Yes	Yes
Update	Yes	Yes	Yes
Info	Yes	Yes	Yes
Delete	Yes	Yes	Yes
Check	Yes	Yes	Yes
Renew	Yes	No	No
Transfer	Yes	Yes	No

### 4.3.2 Launchphase details

Command\Phase	Sunrise	Landrush	LRP	GeneralAvailability
Availability Check Form	no extensions required			
Claims Check Form	type: claims, phase:claims			
Sunrise Create Form	type: application phase: sunrise SMD required	n.a.	n.a.	n.a.
Claims Create Form	n.a.	type: application phase: claims name: landrush Notice ID/Expiry Date/Acceptance Date	type: application phase: custom name: lrp Notice ID/Expiry Date/Acceptance Date	optional: type: registration phase: claims name: open Notice ID/Expiry Date/Acceptance Date
General Create Form	n.a.	type: application phase: landrush	type: application phase: custom name: lrp	no extension required
Update/Transfer/Renew	n.a	n.a.	n.a	no extension required



Delete	phase: sunrise applicatio n ID	phase: landrush application ID	phase: custom application ID	no extension required
Info	Phase: sunrise applicatio n ID	phase: landrush application ID	phase: custom application ID	no extension required

#### **4.4 EPP object relationship**

<b>Object Type</b>	<b>Mandatory</b>	<b>Min occurrence</b>	<b>Max occurrence</b>
Domain	Yes	1	1
Registrant	Yes	1	1
Admin Contact	Yes	1	3
Tech Contact	Yes	1	3
Billing Contact	Yes	1	3
Nameserver	No	0	13



## **5 WHOIS (RDDS) service**

The KSregistry (KSR) provides both web and command line (port 43) publicly accessible RDDS (WHOIS) which offers a central location for all authoritative TLD related information when registering or modifying a domain name. The RDDS (WHOIS) information is reflected in real-time to the public.

The RDDS (WHOIS) service is a public service for interested stakeholders such as registries, registrars, individuals, law enforcement, and trademark owners that require detailed information on one of the following categories of information:

- Domain name including status, creation, and expiration date
- Information on domain registrant, administration, technical and billing contact
- Name server and IP address
- Registrar information

This information will provide the public with the ability to get in touch with the domain holder for any reason that requires action to be taken (e.g. trademark issues, violations with registry policies, offensive content, etc.). In addition to the search capabilities, the service has methods of limiting abuse.

### **5.1 Abuse Protections**

In order to prevent system abuse of the website whois, a Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA) will be used. Each IP address will be entitled to up to six lookups per minute and up to 360 lookups per hour. Each subnet will be entitled to 12 lookups per minute and up to 720 lookups per hour.

### **5.2 Searchable RDDS (WHOIS)**

The KSR RDDS (WHOIS) includes a web-based searchable service for registrars only which reveals more detailed information. For security reasons and legal restrictions some search capabilities are only available in the registrar web-based RDDS (WHOIS). This includes partial match capabilities regarding the registrant's postal address.

The searchable whois is located in the registrar webinterface (<https://login.ksregistry.net>) at Tools->Searchable Whois.



## 5.3 Whois Output Fields

### 5.3.1 Query on domain name

Query on domain name data displays the following information:

Domain Name  
Domain ID  
WHOIS Server  
Referral URL  
Domain Last Updated Date  
Domain Registration Date  
Domain Expiration Date  
Sponsoring registrar  
Sponsoring registrar IANA ID  
Domain Status  
Registrant, Administrative, Technical and Billing Contact  
Information including  
Contact ID  
Contact Name  
Contact Organization  
Contact Address, City, State/Province, Country  
Contact Postal Code  
Contact Phone, Fax, E-mail  
Name Servers associated with this domain  
DNSSEC information

**Example Request:** Query for domain “example.string”

**Example Response:**

Domain Name: EXAMPLE.STRING  
Domain ID: 213232132-TLD  
WHOIS Server: WHOIS.example.string  
Referral URL: http://www.example.string  
Updated Date: 2011-07-22T01:44:02Z  
Creation Date: 2011-06-01T23:45:33Z  
Registry Expiry Date: 2012-06-01T23:59:59Z  
Sponsoring Registrar: EXAMPLE REGISTRAR  
Sponsoring Registrar IANA ID: 1234567890  
Domain Status: clientTransferProhibited  
Registrant ID: 123456-STR  
Registrant Name: EXAMPLE REGISTRANT  
Registrant Organization: EXAMPLE ORGANIZATION  
Registrant Street: 123 EXAMPLE STREET  
Registrant City: SOMEWHERE  
Registrant State/Province: AP  
Registrant Postal Code: 12345  
Registrant Country: EX  
Registrant Phone: +1.5555522222  
Registrant Fax: +1.5555544444  
Registrant Email: EMAIL@EXAMPLE.STRING





Admin ID: 392839283-STR  
Admin Name: EXAMPLE REGISTRANT ADMINISTRATIVE  
Admin Organization: EXAMPLE REGISTRANT ORGANIZATION  
Admin Street: 123 EXAMPLE STREET  
Admin City: SOMEWHERE  
Admin State/Province: AP  
Admin Postal Code: 12345  
Admin Country: EX  
Admin Phone: +1.5555551212  
Admin Phone Ext: 1234  
Admin Fax: +1.5555551213  
Admin Fax Ext:  
Admin Email: EMAIL@EXAMPLE.STRING  
Tech ID: 392811183-STR  
Tech Name: EXAMPLE REGISTRAR TECHNICAL  
Tech Organization: EXAMPLE REGISTRAR LLC  
Tech Street: 123 EXAMPLE STREET  
Tech City: SOMEWHERE  
Tech State/Province: AP  
Tech Postal Code: 12345  
Tech Country: EX  
Tech Phone: +1.1235551234  
Tech Phone Ext: 1234  
Tech Fax: +1.5555551213  
Tech Fax Ext: 93  
Tech Email: EMAIL@EXAMPLE.STRING  
Billing ID: 112811183-STR  
Billing Name: EXAMPLE REGISTRAR BILLING  
Billing Organization: EXAMPLE REGISTRAR LLC  
Billing Street: 123 EXAMPLE STREET  
Billing City: SOMEWHERE  
Billing State/Province: AP  
Billing Postal Code: 12345  
Billing Country: EX  
Billing Phone: +1.1235551234  
Billing Phone Ext: 1234  
Billing Fax: +1.5555551213  
Billing Fax Ext: 93  
Billing Email: EMAIL@EXAMPLE.STRING  
Name Server: NS01.EXAMPLEREGISTRAR.STRING  
Name Server: NS02.EXAMPLEREGISTRAR.STRING  
DNSSEC: signedDelegation  
DNSSEC: unsigned0

### **5.3.2 Query on name server name**

Query on name server name data displays the following information:



Name Server Host Name  
Name Server IP Addresses if applicable  
Sponsoring registrar  
Name Server Creation Date  
Name Server Last Updated Date

**Example Request:** Query for name server ns1.example.string

**Example Response:**

Server Name: NS1.EXAMPLE.STRING  
IP Address: 192.0.3.123  
IP Address: 2001:0DB8::1  
Registrar: Example Registrar, Inc.  
Creation Date: 2011-06-01T23:45:33Z  
Updated Date: 2011-07-22T01:44:02Z

### 5.3.3 Query on registrar

Query on registrar displays the following information:

Registrar ID  
Registrar Name  
Registrar Status  
Registrar Address, City, State/Province, Country  
Registrar Postal Code  
Registrar Phone, Fax, E-mail  
Registrar Creation Date  
Registrar Last Updated Date  
Administrative, Technical Contact  
Information including  
Contact Phone, Fax, E-mail

**Example request:** Query for registrar Example Registrar, Inc.

**Example response:**

Registrar Name: Example Registrar, Inc.  
Street: 4231 King Street  
City: London  
State/Province: XY  
Postal Code: 123445  
Country: XR  
Phone Number: +1.3105559999  
Fax Number: +1.3105559911  
Email: registrar@example.string  
Admin Contact: Pete Registrar  
Phone Number: +1.3105551213  
Fax Number: +1.3105551213  
Email: pete@example-registrar.string  
Technical Contact: Karlo Schlapp  
Phone Number: +1.5610555222



Fax Number: +1.33105551111

Email: [karlo@example-registrar.string](mailto:karlo@example-registrar.string)

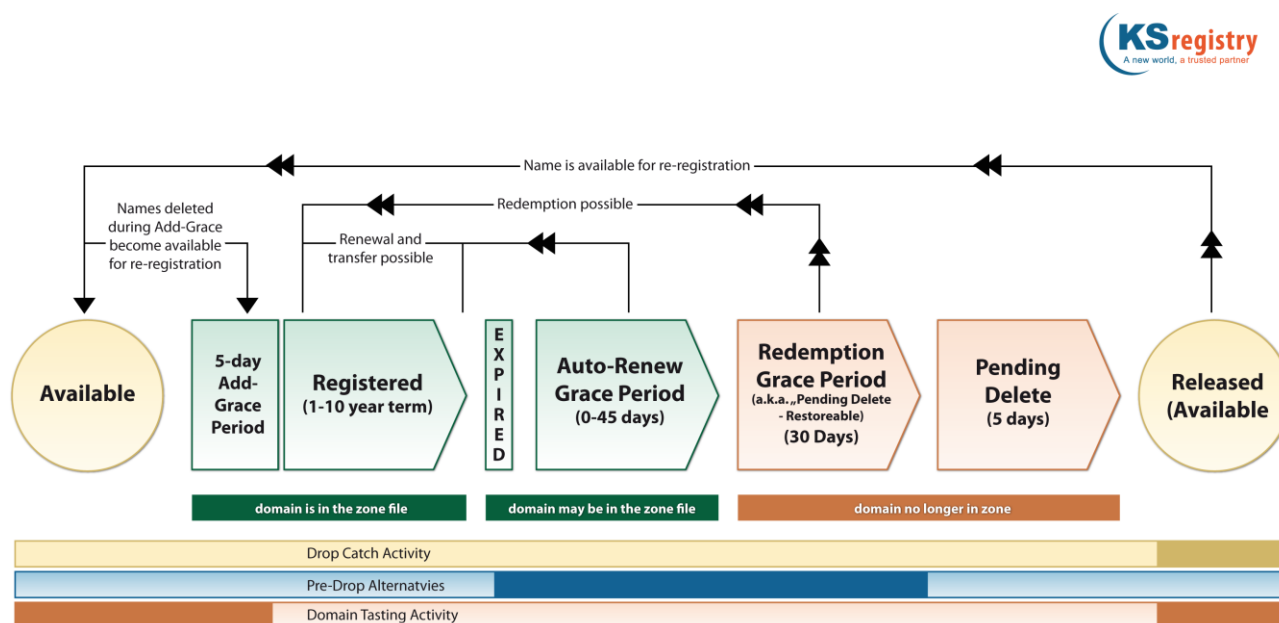


## 6 The domain Life Cycle

This TLD operates with an open registrant policy and only offers one year registration and renewal periods. The KSR platform, however, is also supporting the handling of multiple year periods, as it is capable of managing multiple registrars, as described in the following.

### 6.1 Registration Life Cycle Periods

The registration life cycle of a domain name includes the following periods: available, add-grace period, registered, expired, auto-renew grace period, redemption grace period, pending delete, and released. This life cycle offers the possibility for different business cases, such as domain auctioning, drop catching, and domain tasting. As this TLD operates with an open registrant policy, auctioning, drop catching, and domain tasting may occur. It also includes the domain deletion excess fee in order to avoid massive abuse of the add-grace period. The complete registration life cycle is shown in the figure below.



#### 6.1.1 Available

The domain name is available for registration. Each registrar can register the domain name using a create domain command. The first registrar which submits the domain registration order and has sufficient funding will receive the domain name. After the successful registration of the domain name, the add-grace period will begin. An available domain name cannot be transferred, deleted, updated, or restored and will not be included in the zone file.

#### 6.1.2 Add-Grace Period (AGP)



The AGP starts after the successful registration of the domain name and lasts for five days. During this period the registrar can submit a delete domain command to delete the domain name and will receive a complete refund of the registration fee. This refund does not include non-refundable fees of the registration process. However, the AGP follows the Internet Corporation for Assigned Names and Numbers (ICANN) AGP consensus policy to charge the registrar for excessive deletion activity during this period. During the AGP, the domain name is considered registered and therefore offers the same possibilities available during the registered period described below. A domain name in the AGP is included in the zone file and may not be transferred. The registrar license and agreement prohibits a domain name holder from changing registrars within the first 60 days of the initial registration, enforced by the SRS. If a domain name is renewed during this period and then subsequently deleted, the owning registrar will only receive a refund for the initial registration fee.

### **6.1.3 Registered**

A domain name can be registered for a period of one to ten years. During this period the domain name can be transferred to another registrar using the transfer domain command if the requesting registrar has sufficient funding to pay the transfer domain fee. The transfer of a domain name will extend the registration period for one year, but cannot exceed ten years total registration period. The domain name can be renewed any time by the owning registrar using the renew domain command if the owning registrar has enough funding for the domain renewal fee. However, the maximum registration period of ten years can never be exceeded. A registered domain name is included in the zone file if associated with any host objects.

### **6.1.4 Expired**

If the domain name passes its expiration date without being renewed by the owning registrar, the domain name will enter the auto-renew grace period. An expired domain name is included in the zone file if associated with any host objects.

### **6.1.5 Auto-Renew Grace Period (ARGP)**

During this period the owning registrar has the possibility to delete the domain name using the delete domain command and receive a refund of the domain renewal fee. The ARGP lasts for 45 days. A transfer to another registrar is possible if the requesting registrar has enough funding for the domain transfer fee. A transfer process will extend the domain registration period for one year, but cannot exceed ten years total registration period. If the registration period is already ten years, the transfer will not extend the registration period. After 45 days without a renewal, the domain name will enter the redemption-grace period. During the ARGP the domain name will still be in the zone file if associated with any host objects.

### **6.1.6 Redemption Grace Period (RGP)**

If a domain name enters the RGP it will be excluded from the zone file and the domain name will no longer resolve. During this period no renewal, transfer, or new registration of the domain name



is possible. The RGP lasts for 30 days. During this time the owning registrar can restore the domain name. The registrar can submit an extended update domain command (restore) to reactivate the domain name if the registrar has enough funding for the restore domain fee. If a domain is restored it will enter the registered status with a registration period of one year. It will be included into the zone file with the associated host objects that were used before entering the redemption grace period. If a domain name passes the 30 days of redemption grace period (75 days total after expiration) the domain name will enter the pending delete period and can no longer be restored.

### **6.1.7 Pending Delete**

In the pending delete period the domain name is set for deletion. The period lasts for five days and prohibits any process from occurring regarding the domain name. It will not be included in the zone file. After five days the domain name will be released and available for new registration through a valid domain registration order.

### **6.1.8 Released**

The domain name is deleted from the SRS and enters the available period. The domain name will be immediately available for registration to all registrars.

## **6.2 Domain Name Operations**

### **6.2.1 Create Domain**

A registrar uses the create command to register a domain name. Before a domain name can be created the registrar should use the check command to determine if the domain name is available. The domain name will be registered for the period specified by the registrar. This period may be from one to ten years (the default is one year). Upon registration of a domain name the registrar's credit is immediately debited by the registration fee multiplied by the number of years requested. The registry operator may also add an initial setup fee for a new registration. To be included into the zone file, the domain name must have at least two but no more than thirteen name servers. The registrant may add an authentication code for the domain during the creation or a randomly generated authentication code will be set for the domain name.

### **6.2.2 Delete Domain**

The delete domain command allows the owning registrar of the domain name to delete it. A request to delete a domain name will cause all child name servers of the domain name to also be deleted. A domain must not be deleted if it has child name servers hosting other domains. When a domain name is deleted outside of the AGP it goes into the redemption-grace period status for 30 days. When a domain name is deleted within the AGP it is deleted immediately from the SRS and the



zone file and will be available for a new registration.

### **6.2.3 Transfer Domain**

The transfer process begins when a registrar initiates a transfer with a transfer domain command, the correct authentication code, and sufficient funding in his account for the transfer domain fee. The domain will be flagged in the SRS as being requested for transfer (“pendingTransfer” status). The current registrar has five calendar days to approve or reject the transfer request. If the losing registrar explicitly approves the request the domain is transferred and one year is added to the expiration date. However, the registration period cannot exceed ten years total and will be capped at a maximum of ten years. If the losing registrar explicitly denies the request, then the transfer is immediately canceled and the requesting registrar will receive a refund of the transfer fee. If the gaining registrar mistakenly sends a transfer request, they may cancel the request as long as the transfer is pending. In this case the requesting registrar receives a refund of the transfer fee. Once one of these three actions is complete the SRS creates a poll message to all participating registrars for the domain name. If no action is taken within five days, the request is automatically approved by the SRS batch system. Once a transfer is requested the losing registrar has the response options to reject, approve, or do nothing (auto approve). After the successful transfer of a domain name the old authentication code will be replaced with a new randomly generated authentication code.

### **6.2.4 Update Domain**

The update command enables the owning registrar of the domain name to perform four different update operations on the domain name: Update the name servers, the authentication code, the associated contacts, and the statuses of a domain name. Possible statuses that can be updated include “clientHold”, “clientDeleteProhibited”, “clientUpdateProhibited”, “clientTransferProhibited”, and “clientRenewProhibited”. If an update command removes all name servers of a domain name, it will no longer be included in the zone file and will receive the status “inactive”.

### **6.2.5 Renew Domain**

The renew domain command allows the registrar of the domain name to extend the registration period if they have enough funds for the renew domain fee multiplied by the number of years the registration period will be extended. For this TLD only one year renewals will be offered. The request for a renewal should contain the period to identify the number of years to be added to the registration period. If not provided, the SRS uses a default value of one year. The renewal request should contain the current expiration date to ensure that the domain name will not be renewed multiple times if the request was submitted multiple times due to connection problems between the SRS and the registrar. If no expiration date is given, the SRS will automatically use the current expiration date as the default value. The SRS will renew the domain name for the period specified by the registrar and returns the new registration expiration date.



## 6.2.6 Restore Domain

The restore domain command enables a registrar to restore a deleted domain name after the AGP. In order to successfully restore a domain, the registrar must submit a restore domain command and have sufficient funding for the restore domain fee. The restore operation adds the “pendingRestore” status to the domain name until completion of the request. A successful restore will extend the registration period of the domain name by one year but is capped at a total registration period of ten years. After the restore operation, the domain name will be added back into the zone file as long as there is at least one name server and no “clientHold” or “serverHold” status associated. If the restore operation fails, the domain will stay in the RGP until it enters the pending delete period and the registrar will receive a refund of the restore domain fee.

## 6.3 Domain Name Statuses

To ensure the registration domain life cycle there are several domain statuses that can be seen by everyone using the RDDP (WHOIS) or by any registrar using the extensible provisioning protocol (EPP). These statuses are very important in identifying the current period a domain name is in, or identifying the problems a registrar can encounter while managing a domain name (e.g. a failing transfer request caused by a “serverTransferProhibited” status). A foreign registrar may also be interested in the statuses of a domain name to identify when a specific domain name will be available for registration again (drop catching). Domain name statuses include EPP and RGP domain name statuses as referenced in RFC 3915. Depending on the life cycle period of a domain name, it can have an EPP and a RGP status at the same time. In some cases these two statuses can be different.

These domain statuses are compliant with the RFCs 3915, 5730-5734, and 5910. The domain statuses are:

- addPeriod
- autoRenewPeriod
- inactive
- ok
- pendingRestore
- pendingDelete
- pendingTransfer
- redemptionPeriod
- serverDeleteProhibited
- serverHold
- serverRenewProhibited
- serverTransferProhibited
- serverUpdateProhibited

All server statuses are always set by the SRS. However, the owning registrar also has the ability to assign statuses to a domain name, offering the same functionality as the server statuses. These statuses are:

- clientDeleteProhibited





- clientHold
- clientRenewProhibited
- clientTransferProhibited

A domain name may have more than one status at a time, but must have at least one status. Some statuses prohibit other statuses on the same domain name.

### **6.3.1 ok**

This is the default status of a domain name that has no operations or prohibitions. This value is set and removed by the SRS system as other status values are added or removed and cannot be combined with any other status. The SRS sets this status upon initial creation. A domain name with this status may be updated with any “client” statuses and will be included in the zone file if there is at least one name server associated with it.

### **6.3.2 inactive**

If the delegation information has not been associated with the domain name, this status is applied. This is the default status when a domain name has no associated host objects for the DNS delegation. This status will be set by the SRS when all host-object associations are removed.

### **6.3.3 clientHold**

The owning registrar may set the domain name to this status to prevent the domain name from being included in the zone file.

### **6.3.4 clientUpdateProhibited**

If a domain name status is “clientUpdateProhibited” it cannot be updated using an update domain command. The name servers, authentication code, contacts, and other statuses of the domain name cannot be updated until this status is removed.

### **6.3.5 clientTransferProhibited**

The owning registrar can set this status to a domain name to prevent any other registrar from successfully requesting a transfer for this domain name.

### **6.3.6 clientDeleteProhibited**

If a domain name status is “clientDeleteProhibited”, it cannot be deleted from the SRS using the delete domain command. The domain can still expire after the registration period has passed.



### **6.3.7 clientRenewProhibited**

If a domain name status is “clientRenewProhibited”, it cannot be renewed explicitly by the registrar using the renew domain command. It can still be automatically renewed by the SRS batch system if the owning registrar has set the renewal mode of the domain name to auto renew.

### **6.3.8 serverHold**

The SRS administrator may set the domain name to this status to exclude it from the zone file.

### **6.3.9 serverUpdateProhibited**

The SRS may set the domain name to this status to prevent any updates using the update domain command. The name servers, authentication code, contacts, and the domain name statuses cannot be updated until this status is removed.

### **6.3.10 serverTransferProhibited**

The SRS may set the domain name to this status to prevent any registrar from successfully requesting a transfer for this domain name.

### **6.3.11 serverDeleteProhibited**

The SRS may set the domain name to this status. If a domain name status is “serverDeleteProhibited” it cannot be deleted from the SRS using the delete domain command. This status is slightly different from the “clientDeleteProhibited” as the domain will not even be deleted after the redemption-grace period.

### **6.3.12 serverRenewProhibited**

The SRS may set the domain name to this status. If a domain name status is “serverRenewProhibited”, it cannot be explicitly renewed by the owning registrar using the renew domain command. It can still be automatically renewed by the SRS batch system.

In addition, the SRS batch system may set the RGP pending period statuses as listed below. In EPP, the RGP pending period statuses are represented as substatures of the EPP statuses.



### **6.3.13 redemptionPeriod**

The SRS sets the domain name to this status when a domain is deleted after the AGP. Only the restore domain operation is allowed on a domain with the “redemptionPeriod” status.

### **6.3.14 pendingRestore**

The SRS sets the domain name to this status when a restore is requested. If a domain name status is “pendingRestore”, then no additional restore request can be successfully submitted.

### **6.3.15 pendingDelete**

The SRS sets the domain name to this status once it has been in “redemptionPeriod” for 30 days. A domain name remains on “pendingDelete” status for five days before it is finally deleted from the SRS.

### **6.3.16 pendingTransfer**

The “pendingTransfer” status is automatically set when a domain transfer is requested by a registrar. A domain name remains in “pendingTransfer” status until the transfer is approved, automatically approved through the SRS batch system, rejected, or canceled by the requesting registrar.

### **6.3.17 addPeriod**

This period is entered after the initial registration of a domain name. If the domain name is deleted by the owning registrar during this period he will receive a refund of the domain registration fee in compliance with ICANN's AGP consensus policy.

### **6.3.18 autoRenewPeriod**

This period is set after a domain name registration period expires and is renewed automatically by the SRS. If the domain name is deleted by the owning registrar during this period he will receive a refund of the domain renewal fee.

## **6.4 Reserved Premium Domain Names**

The registry operator reserves the option to define specific domain names as reserved or premium domain names. These domain names will be in compliance with the described registration life cycle



with the addition of a manual registration process through the registry operator support and legal team. Special fees may be accounted for those domain names to reflect manual processing.



## **7 Internationalised Domain Names (IDN)**

.CAM has developed a policy for Internationalized Domain Names (IDN). The policy is implemented as follows.

### **7.1 IETF Standards**

IDNA2008 is a standard which specifies algorithms and rules. .CAM is fully compliant with the IDNA2008 standard by implementing the following RFCs and guidelines:

RFC 5890 - Definitions and Document Framework - <http://tools.ietf.org/html/rfc5890>

RFC 5891 - Protocol - <http://tools.ietf.org/html/rfc5891>

RFC 5892 - The unicode code points - <http://tools.ietf.org/html/rfc5892>

RFC 5893 - Right-to-Left Scripts - <http://tools.ietf.org/html/rfc5893>

ICANN IDN Guidelines - <http://www.icann.org/en/topics/idn/idn-guidelines-02sep11-en.htm>

### **7.2 General rules**

.CAM checks each domain registration against the following rules. If one rule fails, the registration is rejected.

#### **7.2.1 Protocol validity**

Hyphens at the third and fourth position of the punycode decoded domain name are not allowed.

#### **7.2.2 NON-LDH (Letter Digit Hyphen) Check**

At least one code point of the punycode decoded domain name must not be a letter, a digit or a hyphen.

#### **7.2.3 Hyphen check**

The domain name must not begin or end with a hyphen.

#### **7.2.4 Protocol enforcement**

The punycode encoded and redecoded domain name must match the requested domain name.

#### **7.2.5 Length check**

The maximal length is restricted to 63 characters due to limitations implied by DNS. Domain names with the length of two characters are not allowed.

### **7.3 Language tag definition**

All IDN registrations require a language tag. For each language tag, .CAM has a list of assigned / associated code points. The requested IDN must consist only of characters which are defined in that language specific list. Registrations of domain names with code points, which are not defined in that list, are rejected.

The following language tags are permitted:



The allowed language tag(s) for this TLD will be provided by Registry Operator.

Commingling of different scripts in a single domain is prohibited.

### **7.3.1 Allowed code points**

The defined and assigned / associated code points and additionally conducted rules per selected language tag can be found at the IANA Repository of IDN Practices at <https://www.iana.org/domains/idn-tables>.

You can find a complete list of supported IDN tables in the Appendix B.

### **7.4 Example of EPP extension block**

```
<extension>
  <idn:data xmlns:idn="urn:ietf:params:xml:ns:idn-1.0">
    <idn:table>ger</idn:table>
    <idn:uname>möller.tld</idn:uname>
  </idn:data>
</extension>
```



## 8 SFTP download area for registrars

KSregistry provides an SFTP area which can be accessed through the registrar webinterface (Account->FTP) or per direct sftp access. URL and Port are defined in Chapter 3.

Below you find the default folders and reports provided by KSregistry. Registry Operator may add additional reports or folders which are not listed here.

### 8.1 Folder invoice

- <TLD>-registry\_Invoice\_2016\_06.pdf (Invoice Summary as .PDF)
- <TLD>-registry\_Invoice\_2016\_06.txt (detailed transaction listing)

Both reports are created at the first day of the month for the previous month.

### 8.2 Folder reports

- accountings.csv.zip (daily generated list of all accountings until file creation)
- domainlist.csv.zip (daily generated list domainlist for the registrar)
- domain\_premiumlist\_<tld>.csv.zip (daily generated premium domain list if premium domains are supported by the registry)

Both reports are created at the first day of the month for the previous month.

### 8.3 Folder statements

- statement\_<tld>-<registrar>\_2016-06.pdf

Shows the statements which have been done for this registrar. Some Registries do not provide this information.



## 9 Appendix A

Draft text for required registrar information in order to create an KSR account.

### **Reseller Details**

Registrar ID:

IANAID (must be a number):

Email:

Urgent email:

Country:

Language:

### **Accounting**

Low-balance warning:

V.A.T.:

Currency:

### **Owner contact:**

Title:

First name:

Middle name:

Last name:

Organization:

Street:

City:

Zip:

Country:

State:

Phone:

Fax:

Email:

### **Admin contact:**

Title:

First name:

Middle name:





Last name:

Organization:

Street:

City:

Zip:

Country:

State:

Phone:

Fax:

Email:

**Tech contact:**

Title:

First name:

Middle name:

Last name:

Organization:

Street:

City:

Zip:

Country:

State:

Phone:

Fax:

Email:

**Billing contact:**

Title:

First name:

Middle name:

Last name:

Organization:

Street:

City:

Zip:



Country:

State:

Phone:

Fax:

Email:

### **Whois**

WHOIS (URL of own registrar WHOIS server):

URL of registrar web side:

IP addresses for Production access:

IP1:

IP2:

IP addresses for OT&E access:

IP1:

IP2:



## 10 Appendix B

### Supported IDN languages by .CAM

Language	Script/Language tag	Has variants?	Has contextual rules?
Albanian	AL	no	no
Armenian	AM	no	no
Arabic	AR	no	yes
Azerbaijani	AZ	no	no
Bosnian latin	BS	no	no
Bulgarian	BG	no	no
Catalan	CAT	no	yes
Croatian-SCR	SCR	no	no
Czech	CZ	no	no
Danish	DA	no	no
Dutch Flemish	NL	no	no
Estonian	ET	no	no
Finnish	FI	no	no
French	FR	no	no
Georgian	KA	no	no
German-GER (without sharp S)	GER	no	no
Greek	GR	no	no
Hungarian	HU	no	no
Icelandic	IS	no	no
Italian	IT	no	no
Japanese	JP	no	yes
Korean	KO	no	no
Latin	LAT	no	no
Lithuanian	LT	no	no
Luxembourgish; Letzeburgesch	LU	no	no
Latvian	LV	no	no
Macedonia	MK	no	no
Moldavian	MD	no	no
Norwegian	NO	no	no
Polish	PL	no	no
Portuguese	PT	no	no
Romanian	RO	no	no
Russian	RU	no	no
Serbian	SR	no	no
Slovak	SK	no	no
Slovenian	SI	no	no
Spanish; Castilian	ES	no	no
Swedish	SE	no	no
Turkish	TR	no	no
Thai	THA	no	no



Language	Script/Language tag	Has variants?	Has contextual rules?
Ukrainian	UKR	no	no