

Internet Engineering Task Force  
Internet-Draft  
Intended status: Informational  
Expires: January 24, 2016

G. Lozano  
ICANN  
Jul 23, 2015

**TMCH functional specifications  
draft-lozano-tmch-func-spec-10**

Abstract

This document describes the requirements, the architecture and the interfaces between the Trademark Clearing House (TMCH) and Domain Name Registries as well as between the TMCH and Domain Name Registrars for the provisioning and management of domain names during Sunrise and Trademark Claims Periods.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 24, 2016.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

|                          |  |                    |
|--------------------------|--|--------------------|
| <a href="#">1.</a>       | <a href="#">Introduction . . . . .</a>                                   | <a href="#">4</a>  |
| <a href="#">2.</a>       | <a href="#">Terminology . . . . .</a>                                    | <a href="#">4</a>  |
| <a href="#">3.</a>       | <a href="#">Glossary . . . . .</a>                                       | <a href="#">4</a>  |
| <a href="#">4.</a>       | <a href="#">Architecture . . . . .</a>                                   | <a href="#">9</a>  |
| <a href="#">4.1.</a>     | <a href="#">Sunrise Period . . . . .</a>                                 | <a href="#">9</a>  |
| <a href="#">4.2.</a>     | <a href="#">Trademark Claims Period . . . . .</a>                        | <a href="#">10</a> |
| <a href="#">4.3.</a>     | <a href="#">Interfaces . . . . .</a>                                     | <a href="#">11</a> |
| <a href="#">4.3.1.</a>   | <a href="#">hv . . . . .</a>   | <a href="#">11</a> |
| <a href="#">4.3.2.</a>   | <a href="#">vd . . . . .</a>   | <a href="#">11</a> |
| <a href="#">4.3.3.</a>   | <a href="#">dy . . . . .</a>   | <a href="#">11</a> |
| <a href="#">4.3.4.</a>   | <a href="#">tr . . . . .</a>   | <a href="#">11</a> |
| <a href="#">4.3.5.</a>   | <a href="#">ry . . . . .</a>   | <a href="#">11</a> |
| <a href="#">4.3.6.</a>   | <a href="#">dr . . . . .</a>   | <a href="#">12</a> |
| <a href="#">4.3.7.</a>   | <a href="#">yd . . . . .</a>   | <a href="#">12</a> |
| <a href="#">4.3.8.</a>   | <a href="#">dv . . . . .</a>   | <a href="#">12</a> |
| <a href="#">4.3.9.</a>   | <a href="#">vh . . . . .</a>   | <a href="#">12</a> |
| <a href="#">4.3.10.</a>  | <a href="#">vs . . . . .</a>   | <a href="#">12</a> |
| <a href="#">4.3.11.</a>  | <a href="#">sy . . . . .</a>   | <a href="#">13</a> |
| <a href="#">4.3.12.</a>  | <a href="#">sr . . . . .</a>   | <a href="#">13</a> |
| <a href="#">4.3.13.</a>  | <a href="#">vc . . . . .</a>   | <a href="#">13</a> |
| <a href="#">4.3.14.</a>  | <a href="#">cy . . . . .</a>   | <a href="#">13</a> |
| <a href="#">4.3.15.</a>  | <a href="#">cr . . . . .</a>   | <a href="#">13</a> |
| <a href="#">5.</a>       | <a href="#">Process Descriptions . . . . .</a>                           | <a href="#">14</a> |
| <a href="#">5.1.</a>     | <a href="#">Bootstrapping . . . . .</a>                                  | <a href="#">14</a> |
| <a href="#">5.1.1.</a>   | <a href="#">Bootstrapping for Registries . . . . .</a>                   | <a href="#">14</a> |
| <a href="#">5.1.1.1.</a> | <a href="#">Credentials . . . . .</a>                                    | <a href="#">14</a> |
| <a href="#">5.1.1.2.</a> | <a href="#">IP Addresses for Access Control . . . . .</a>                | <a href="#">14</a> |
| <a href="#">5.1.1.3.</a> | <a href="#">TMCH Trust Anchor . . . . .</a>                              | <a href="#">14</a> |
| <a href="#">5.1.1.4.</a> | <a href="#">TMDB/SMDM PGP Key . . . . .</a>                              | <a href="#">15</a> |
| <a href="#">5.1.2.</a>   | <a href="#">Bootstrapping for Registrars . . . . .</a>                   | <a href="#">16</a> |
| <a href="#">5.1.2.1.</a> | <a href="#">Credentials . . . . .</a>                                    | <a href="#">16</a> |
| <a href="#">5.1.2.2.</a> | <a href="#">IP Addresses for Access Control . . . . .</a>                | <a href="#">16</a> |
| <a href="#">5.1.2.3.</a> | <a href="#">TMCH Trust Anchor . . . . .</a>                              | <a href="#">16</a> |
| <a href="#">5.1.2.4.</a> | <a href="#">TMDB PGP Key . . . . .</a>                                   | <a href="#">16</a> |
| <a href="#">5.2.</a>     | <a href="#">Sunrise Period . . . . .</a>                                 | <a href="#">17</a> |
| <a href="#">5.2.1.</a>   | <a href="#">Domain Name Registration . . . . .</a>                       | <a href="#">17</a> |
| <a href="#">5.2.2.</a>   | <a href="#">Sunrise DN Registration by Registries . . . . .</a>          | <a href="#">18</a> |
| <a href="#">5.2.3.</a>   | <a href="#">TMDB Sunrise Services for Registries . . . . .</a>           | <a href="#">19</a> |
| <a href="#">5.2.3.1.</a> | <a href="#">SMD Revocation List . . . . .</a>                            | <a href="#">19</a> |
| <a href="#">5.2.3.2.</a> | <a href="#">Certificate Revocation List . . . . .</a>                    | <a href="#">20</a> |
| <a href="#">5.2.3.3.</a> | <a href="#">Notice of Registered Domain Names . . . . .</a>              | <a href="#">21</a> |
| <a href="#">5.2.4.</a>   | <a href="#">Sunrise DN registration by Registrars . . . . .</a>          | <a href="#">23</a> |
| <a href="#">5.2.5.</a>   | <a href="#">TMDB Sunrise Services for Registrars . . . . .</a>           | <a href="#">24</a> |
| <a href="#">5.3.</a>     | <a href="#">Trademark Claims Period . . . . .</a>                        | <a href="#">25</a> |
| <a href="#">5.3.1.</a>   | <a href="#">Domain Registration . . . . .</a>                            | <a href="#">25</a> |
| <a href="#">5.3.2.</a>   | <a href="#">Trademark Claims DN registration by Registries . . . . .</a> | <a href="#">27</a> |

Lozano

Expires January 24, 2016

[Page 2]

|                             |  |    |
|-----------------------------|--|----|
| 5.3.3.                      | TMBD Claims Services for Registries . . . . .  | 29 |
| 5.3.3.1.                    | Domain Name Label (DNL) List . . . . .   | 29 |
| 5.3.3.2.                    | Notice of Registered Domain Names . . . . .  | 30 |
| 5.3.4.                      | Trademark Claims DN Registration by Registrars . . . . .   | 31 |
| 5.3.5.                      | TMBD Claims Services for Registrars . . . . .  | 33 |
| 5.3.5.1.                    | Claims Notice Information Service . . . . .  | 33 |
| 5.4.                        | Qualified Launch Program Period . . . . .  | 34 |
| 5.4.1.                      | Domain Registration . . . . .  | 34 |
| 5.4.2.                      | TMBD QLP Services for Registries . . . . .   | 36 |
| 5.4.2.1.                    | Sunrise List (SURL) . . . . .  | 36 |
| 6.                          | Data Format Descriptions . . . . .   | 37 |
| 6.1.                        | DNL List file . . . . .  | 37 |
| 6.2.                        | SMD Revocation List . . . . .  | 39 |
| 6.3.                        | LORDN File . . . . .   | 41 |
| 6.3.1.                      | LORDN Log File . . . . .   | 46 |
| 6.3.1.1.                    | LORDN Log Result Codes . . . . .   | 49 |
| 6.4.                        | SMD File . . . . .   | 53 |
| 6.5.                        | Trademark Claims Notice . . . . .  | 55 |
| 6.6.                        | Sunrise List File . . . . .  | 63 |
| 7.                          | Formal Syntax . . . . .  | 65 |
| 7.1.                        | Trademark Claims Notice . . . . .  | 65 |
| 8.                          | Acknowledgements . . . . .   | 69 |
| 9.                          | Change History . . . . .   | 69 |
| 9.1.                        | Changes from <a href="#">draft-lozano-tmch-func-spec-06</a> to<br><a href="#">draft-lozano-tmch-func-spec-07</a> . . . . . | 69 |
| 9.2.                        | Changes from <a href="#">draft-lozano-tmch-func-spec-07</a> to<br><a href="#">draft-lozano-tmch-func-spec-08</a> . . . . . | 69 |
| 9.3.                        | Changes from <a href="#">draft-lozano-tmch-func-spec-08</a> to<br><a href="#">draft-lozano-tmch-func-spec-09</a> . . . . . | 69 |
| 9.4.                        | Changes from <a href="#">draft-lozano-tmch-func-spec-09</a> to<br><a href="#">draft-lozano-tmch-func-spec-10</a> . . . . . | 70 |
| 10.                         | IANA Considerations . . . . .  | 70 |
| 11.                         | Security Considerations . . . . .  | 70 |
| 12.                         | References . . . . .   | 70 |
| 12.1.                       | Normative References . . . . .   | 70 |
| 12.2.                       | Informative References . . . . .   | 71 |
| <a href="#">Appendix A.</a> | Document Changelog . . . . .   | 73 |
| Author's Address            | . . . . .  | 73 |



## **1. Introduction**

Domain Name Registries may operate in special modes within certain periods of time to facilitate registration of domain names.

Along with the upcoming introduction of new generic Top Level Domains (gTLD), two special modes will come into effect:

- o Sunrise Period
- o Trademark Claims Period

The Sunrise and Trademark Claims Periods are defined in the gTLD Applicant Guidebook [[ICANN-GTLD-AGB-20120604](#)].

This document describes the requirements, the architecture and the interfaces between the Trademark Clearing House (TMCH) and Domain Name Registries (called Registries in the rest of the document) as well as between the TMCH and Domain Name Registrars (called Registrars in the rest of the document) for the provisioning and management of domain names during the Sunrise and Trademark Claims Periods.

For any date and/or time indications, Coordinated Universal Time (UTC) applies.

## **2. Terminology**

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [[RFC2119](#)].

XML is case sensitive. Unless stated otherwise, XML specifications and examples provided in this document MUST be interpreted in the character case presented in order to develop a conforming implementation.

"tmNotice-1.0" is used as an abbreviation for "urn:ietf:params:xml:ns:tmNotice-1.0". The XML namespace prefix "tmNotice" is used, but implementations MUST NOT depend on it and instead employ a proper namespace-aware XML parser and serializer to interpret and output the XML documents.

## **3. Glossary**

In the following section, the most common terms are briefly



explained:

- o Effective allocation: A DN is considered effectively allocated when the DN object for the DN has been created in the SRS of the Registry and has been assigned to the effective user. A DN object in status "pendingCreate" or any other status that precedes the first time a DN is assigned to an end-user is not considered an effective allocation. A DN object created internally by the Registry for subsequent delegation to another Registrant is not considered an effective allocation.
- o Backend Registry Operator: Entity that manages (a part of) the technical infrastructure for a Registry Operator. The Registry Operator may also be the Backend Registry Operator.
- o CA: Certificate Authority, see [[RFC5280](#)] and [[RFC6818](#)]
- o CSV: Comma-Separated Values, see [[RFC4180](#)]
- o CNIS, Claims Notice Information Service: This service provides TCNs to Registrars.
- o CRC32, Cyclic Redundancy Check: algorithm used in the ISO 3309 standard and in [section 8.1.1.6.2](#) of ITU-T recommendation V.42.
- o CRL: Certificate Revocation List, see [[RFC5280](#)] and [[RFC6818](#)].
- o Date and time, datetime: Date and time are specified following the standard "Date and Time on the Internet specification", see [[RFC3339](#)].
- o DN: Domain Name, domain name, see [[RFC1034](#)]
- o DN Repository Object Identifier, DNROID: an identifier assigned by the Registry to each DN object that unequivocally identifies said DN object. For example, if a new DN object is created for a name that existed in the past, the DN objects will have different DNROIDs.
- o DNS: Domain Name System, see [[RFC1034](#)]
- o DNL, Domain Name Label: A label as specified in [[RFC1035](#)]. For IDNs the A-Label is used [[RFC5890](#)].
- o DNL List: A list of DNLs that are covered by a PRM.
- o EPP: Extensible Provisioning Protocol, see [[RFC5730](#)].



- o FQDN: Fully Qualified Domain Name (e.g. myname.example.com)
- o HTTP: Hypertext Transfer Protocol, see [[RFC2616](#)]
- o HTTPS: HTTP over TLS (Transport Layer Security), [[RFC5246](#)].
- o ICANN-CA: ICANN's Certificate Authority (CA); Trust Anchor for the SMD PKI model.
- o IDN: Internationalized Domain Name, see [[RFC5890](#)]
- o LORDN, List of Registered Domain Names: This is the list of effectively allocated DNS matching a DNL of a PRM. Registries will upload this list to the TMDB (during the NORDN process).
- o Lookup Key: A random string of up to 51 chars from the set [a-zA-Z0-9/] to be used as the lookup Key by Registrars to obtain the TCN using the CNIS. Lookup Keys are unique and are related to one DNL only.
- o NORDN, Notification of Registered Domain Names: The process by which Registries upload their recent LORDN to the TMDB.
- o PGP: Pretty Good Privacy, see [[RFC4880](#)]
- o PKI: Public Key Infrastructure, see [[RFC5280](#)] and [[RFC6818](#)]
- o PRM, Pre-registered mark: Mark that has been pre-registered with the TMCH.
- o Registrant: Person or Organization registering a DN via a Registrar.
- o Registrar, Domain Name Registrar: Entity that registers DNS with the Registry on behalf of the Registrant.
- o Registry, Registry Operator, Domain Name Registry: Entity that accepts DN registrations from Registrars, maintains the central Database of Registered DNS. A Registry Operator is the contracting party with ICANN for the TLD.
- o Qualified Launch Program period, QLP period: During this OPTIONAL period, a special process applies to DNS matching the Sunrise List and/or the DNL List, to ensure a TMH gets informed of a DN matching his PRM.
- o SMD, Signed Mark Data: A cryptographically signed token issued by the TMV to the TMH to be used in the Sunrise Period to apply for a



DN that matches a DNL of a PRM; see also [[I-D.lozano-tmch-smd](#)]

- o SMD File: A file containing the SMD (see above) and some human readable data. The latter is usually ignored in the processing of the SMD File. See also [Section 6.4](#).
- o SMDM, SMD Manager: A entity managing the SMDs, mainly maintaining lists of revoked SMDs (SMD Revocation List); see also [[I-D.lozano-tmch-smd](#)]
- o SMD Revocation List: The SMD Revocation List is used by Registries (and optionally by Registrars) during the Sunrise Period to ensure that an SMD is still valid (i.e. not revoked). The SMD Revocation List has a similar function as CRLs used in PKI.
- o SRS: Shared Registration System, see also [[ICANN-GTLD-AGB-20120604](#)].
- o Sunrise Period: During this period DNs matching a DNL of a PRM can be exclusively obtained by the respective TMHs. For DNs matching a PRM, a special process applies to ensure a TMH gets informed on the effective allocation of a DN matching his/her PRM.
- o Sunrise List, SURL: The list of DNLs that are covered by a PRM and eligible for Sunrise.
- o TLD: Top Level Domain Name, see [[RFC1591](#)]
- o Trademark, mark: Marks are used to claim exclusive properties of products or services. A mark is typically a name, word, phrase, logo, symbol, design, image, or a combination of these elements. For the scope of this document only textual marks are relevant.
- o Trademark Claims, Claims: Provides information to enhance the understanding of the Trademark rights being claimed by the TMH.
- o Trademark Claims Notice, Claims Notice, Trademark Notice, TCN: A Trademark Claims Notice consist of one or more Trademark Claims and are provided to prospective Registrants of DNs.
- o Trademark Claims Notice Identifier, TCNID: An element of the Trademark Claims Notice (see above), identifying said TCN. The Trademark Claims Notice Identifier is specified in the element <tmNotice:id>.
- o Trademark Claims Period: During this period, a special process applies to DNs matching the DNL List, to ensure a TMH gets informed of a DN matching his PRM. For DNs matching the DNL List,



Registrars show a TCN to prospective Registrants that has to be acknowledged before effective allocation of the DN.

- o TMCH, Trademark Clearinghouse: The Trademark Clearinghouse is an ICANN central repository for information to be authenticated, stored, and disseminated, pertaining to the rights of TMHs. The Trademark Clearinghouse is split into two functions TMV and TMDB (see below). There could be several entities performing the TMV function, but only one entity performing the TMDB function.
- o TMDB, Trademark Clearinghouse Database: Serves as a database of the TMCH to provide information to the new gTLD Registries and Registrars to support Sunrise or Claims services. There is only one TMDB in the TMCH that concentrates the information about the "verified" Trademark records from the TMVs.
- o TMH, Trademark Holder: The person or organization owning rights on a mark.
- o TMV, Trademark Validator, Trademark validation organization: An entity authorized by ICANN to authenticate and validate registrations in the TMDB ensuring the marks qualify as registered or are court-validated marks or marks that are protected by statute or treaty. This entity would also be asked to ensure that proof of use of marks is provided, which can be demonstrated by furnishing a signed declaration and one specimen of current use.
- o UTC: Coordinated Universal Time, as maintained by the Bureau International des Poids et Mesures (BIPM); see also [[RFC3339](#)].



## 4. Architecture

### 4.1. Sunrise Period

Architecture Sunrise Period

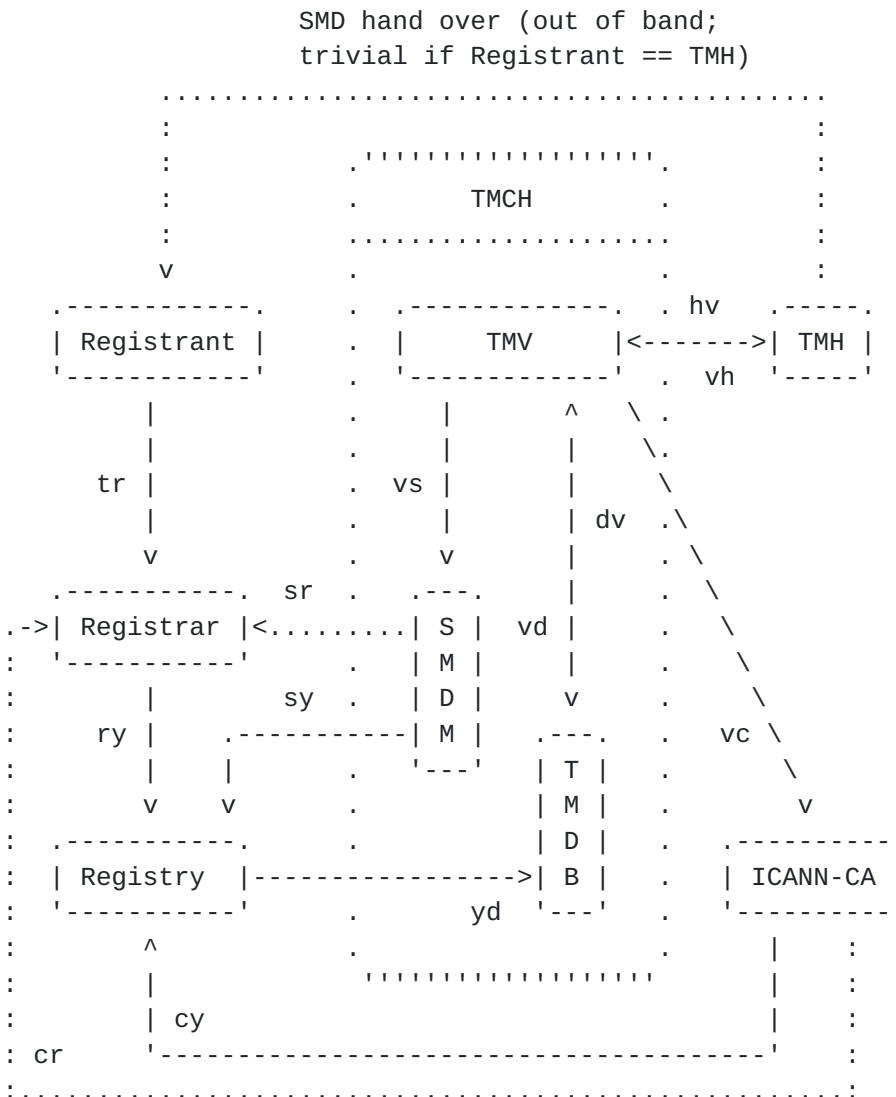


Figure 1



#### 4.2. Trademark Claims Period

##### Architecture Trademark Claims Period

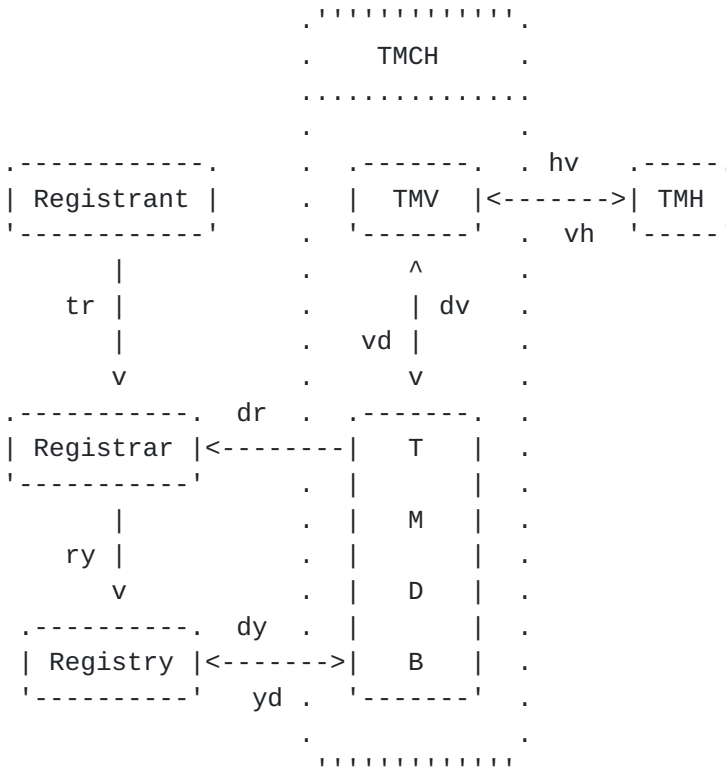


Figure 2



### **[4.3.](#) Interfaces**

In the sub-sections below follows a short description of each interface to provide an overview of the architecture. More detailed descriptions of the relevant interfaces follow further below ([Section 5](#)).

#### **[4.3.1.](#) hv**

The TMH registers a mark with a TMV via the hv interface.

After the successful registration of the mark, the TMV makes available a SMD (Signed Mark Data) file (see also [Section 6.4](#)) to the TMH to be used during the Sunrise Period.

The specifics of the hv interface are beyond the scope of this document.

#### **[4.3.2.](#) vd**

After successful mark registration, the TMV ensures the TMDB inserts the corresponding DNLs and mark information into the database via the vd interface.

The specifics of the vd interface are beyond the scope of this document.

#### **[4.3.3.](#) dy**

Not used during the Sunrise Period.

During the Trademark Claims Period the Registry fetches the latest DNL List from the TMDB via the dy interface in regular intervals. The protocol used on the dy interface is HTTPS.

#### **[4.3.4.](#) tr**

The Registrant communicates with the Registrar via the tr interface.

The specifics of the tr interface are beyond the scope of this document.

#### **[4.3.5.](#) ry**

The Registrar communicates with the Registry via the ry interface. The ry interfaces is typically implemented in EPP [[RFC5730](#)].



#### **4.3.6. dr**

Not used during the Sunrise Period.

During the Trademark Claims Period, the Registrar fetches the TCN from the TMDB (to be displayed to the Registrant via the tr interface) via the dr interface. The protocol used for fetching the TCN is HTTPS [[RFC2818](#)].

#### **4.3.7. yd**

During the Sunrise period the Registry notifies the TMDB via the yd interface of all DNS effectively allocated.

During the Trademark Claims period, the Registry notifies the TMDB via the yd interface of all DNS effectively allocated that matched an entry in the Registry previously downloaded DNL List during the creation of the DN.

The protocol used on the yd interface is HTTPS.

#### **4.3.8. dv**

The TMDB notifies via the dv interface to the TMV of all DNS effectively allocated that match a mark registered by that TMV.

The specifics of the dv interface are beyond the scope of this document.

#### **4.3.9. vh**

The TMV notifies the TMH via the vh interface after a DN has been effectively allocated that matches a PRM of this TMH.

The specifics of the vh interface are beyond the scope of this document.

#### **4.3.10. vs**

The TMV requests to add a revoked SMD to the SMD Revocation List at the SMDM.

The specifics of the vs interface are beyond the scope of this document.

Not relevant during the Trademark Claims Period.



#### [4.3.11.](#) sy

During the Sunrise Period the Registry fetches the most recent SMD Revocation List from the SMDM via the sy interface in regular intervals. The protocol used on the sy interface is HTTPS.

Not relevant during the Trademark Claims Period.

#### [4.3.12.](#) sr

During the Sunrise Period the Registrar may fetch the most recent SMD Revocation List from the SMDM via the sr interface. The protocol used on the sr interface is the same as on the sy interface (s. above), i.e. HTTPS.

Not relevant during the Trademark Claims Period.

#### [4.3.13.](#) vc

The TMV requests to add a revoked TMV certificate to the CRL at the ICANN-CA via the vc interface.

The specifics of the vc interface are beyond the scope of this document.

Not relevant during the Trademark Claims Period.

#### [4.3.14.](#) cy

During the Sunrise Period the Registry fetches the most recent CRL from the ICANN-CA via the cy interface in regular intervals. The CRL is mainly used for validation of TMV certificates. The protocol used on the cy interface is HTTPS [[RFC2818](#)].

Not relevant during the Trademark Claims Period.

#### [4.3.15.](#) cr

During the Sunrise Period the Registrar may fetch the most recent CRL from the ICANN-CA via the cr interface. The protocol used on the cr interface is the same as on the cy interface.

Not relevant during the Trademark Claims Period.



## **5. Process Descriptions**

### **5.1. Bootstrapping**

#### **5.1.1. Bootstrapping for Registries**

##### **5.1.1.1. Credentials**

Each Registry Operator will receive authentication credentials from the TMDB/SMDM to be used:

- o During the Sunrise Period to fetch the SMD Revocation List from the SMDM via the sy interface ([Section 4.3.11](#)).
- o During Trademark Claims Period to fetch the DNL List from the TMDB via the dy interface ([Section 4.3.3](#)).
- o During the NORDN process to notify the LORDN to the TMDB via the yd interface ([Section 4.3.7](#)).

Note: credentials are created per TLD and provided to the Registry Operator.

##### **5.1.1.2. IP Addresses for Access Control**

Each Registry Operator MUST provide to the TMDB all IP addresses that will be used to:

- o Fetch the SMD Revocation List via the sy interface ([Section 4.3.11](#)).
- o Fetch the DNL List from the TMDB via the dy interface ([Section 4.3.3](#)).
- o Upload the LORDN to the TMDB via the yd interface ([Section 4.3.7](#)).

This access restriction MAY be applied by the TMDB/SMDM in addition to HTTP Basic access authentication (for credentials to be used, see [Section 5.1.1.1](#)).

The TMDB/SMDM MAY limit the number of IP addresses to be accepted per Registry Operator.

##### **5.1.1.3. TMCH Trust Anchor**

Each Registry Operator MUST fetch the X.509 certificate ([[RFC5280](#)] / [[RFC6818](#)]) of the ICANN-CA (Trust Anchor) from < <https://ca.icann.org/tmch.crt> > to be used:



- o During the Sunrise Period to validate the TMV certificates and the CRL of TMV certificates.

#### **5.1.1.4. TMDB/SMDM PGP Key**

The TMDB MUST provide each Registry Operator with the public portion of the PGP Key used by TMDB and SMDM, to be used:

- o During the Sunrise Period to perform integrity checking of the SMD Revocation List fetched from the SMDM via the sy interface ([Section 4.3.11](#)).
- o During Trademark Claims Period to perform integrity checking of the DNL List fetched from the TMDB via the dy interface ([Section 4.3.3](#)).



### **5.1.2. Bootstrapping for Registrars**

#### **5.1.2.1. Credentials**

Each ICANN-accredited Registrar will receive authentication credentials from the TMDB to be used:

- o During the Sunrise Period to (optionally) fetch the SMD Revocation List from the SMDM via the sr interface ([Section 4.3.12](#)).
- o During Trademark Claims Period to fetch TCNs from the TMDB via the dr interface ([Section 4.3.6](#)).

#### **5.1.2.2. IP Addresses for Access Control**

Each Registrar MUST provide to the TMDB all IP addresses, which will be used to:

- o Fetch the SMD Revocation List via the sr interface ([Section 4.3.12](#)).
- o Fetch TCNs via the dr interface ([Section 4.3.6](#)).

This access restriction MAY be applied by the TMDB/SMDM in addition to HTTP Basic access authentication (for credentials to be used, see [Section 5.1.2.1](#)).

The TMDB MAY limit the number of IP addresses to be accepted per Registrar.

#### **5.1.2.3. TMCH Trust Anchor**

Registrars MAY fetch the X.509 certificate ([[RFC5280](#)] / [[RFC6818](#)]) of the ICANN-CA (Trust Anchor) from < <https://ca.icann.org/tmch.crt> > to be used:

- o During the Sunrise Period to (optionally) validate the TMV certificates and the CRL of TMV certificates.

#### **5.1.2.4. TMDB PGP Key**

Registrars MUST receive the public portion of the PGP Key used by TMDB and SMDM from the TMDB administrator to be used:

- o During the Sunrise Period to (optionally) perform integrity checking of the SMD Revocation List fetched from the SMDM via the sr interface ([Section 4.3.12](#)).



## 5.2. Sunrise Period

### 5.2.1. Domain Name Registration

Registration during Sunrise Period

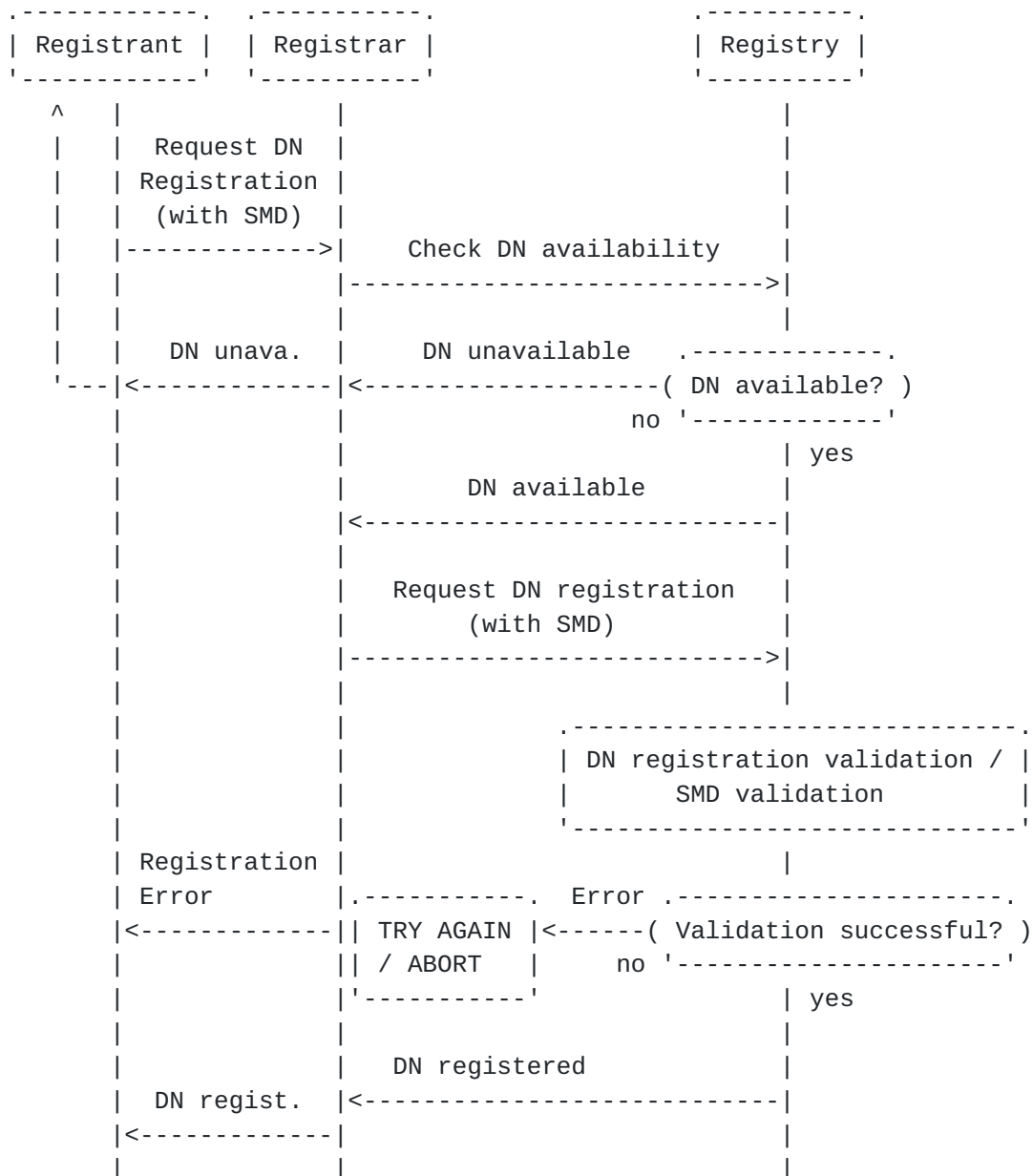


Figure 3

Note: the figure depicted above represents a synchronous DN registration workflow (usually called first come first served).



### **5.2.2. Sunrise DN Registration by Registries**

Registries MUST perform a minimum set of checks for verifying each DN registration during the Sunrise Period upon reception of a registration request over the ry interface ([Section 4.3.5](#)). If any of these checks fails the Registry MUST abort the registration. Each of these checks MUST be performed before the DN is effectively allocated.

In case of asynchronous registrations (e.g. auctions), the minimum set of checks MAY be performed when creating the intermediate object (e.g. a DN application) used for DN registration. If the minimum set of checks is performed when creating the intermediate object (e.g. a DN application) a Registry MAY effectively allocate the DN without performing the minimum set of checks again.

Performing the minimum set of checks Registries MUST verify that:

1. A SMD has been received from the Registrar along with the DN registration.
2. The certificate of the TMV has been correctly signed by the ICANN-CA. (The certificate of the TMV is contained within the SMD.)
3. The time when the validation is done is within the validity period of the TMV certificate.
4. The certificate of the TMV is not be listed in the CRL file specified in the CRL distribution point of the TMV certificate.
5. The signature of the SMD (signed with the TMV certificate) is valid.
6. The time when the validation is done is within the validity period of the SMD based on <smd:notBefore> and <smd:notAfter> elements.
7. The SMD has not been revoked, i.e., is not contained in the SMD Revocation List.
8. The leftmost DNL (A-label in case of IDNs) of the DN being effectively allocated matches one of the labels (<mark:label>) elements in the SMD.

These procedure apply to all DN effective allocations at the second level as well as to all other levels subordinate to the TLD that the Registry accepts registrations for.



### **5.2.3. TMDB Sunrise Services for Registries**

#### **5.2.3.1. SMD Revocation List**

A new SMD Revocation List MUST be published by the SMDM twice a day, by 00:00:00 and 12:00:00 UTC.

Registries MUST refresh the latest version of the SMD Revocation List at least once every 24 hours.

Note: the SMD Revocation List will be the same regardless of the TLD. If a Backend Registry Operator manages the infrastructure of several TLDs, the Backend Registry Operator could refresh the SMD Revocation List once every 24 hours, the SMD Revocation List could be used for all the TLDs managed by the Backend Registry Operator.

Update SMD Revocation List

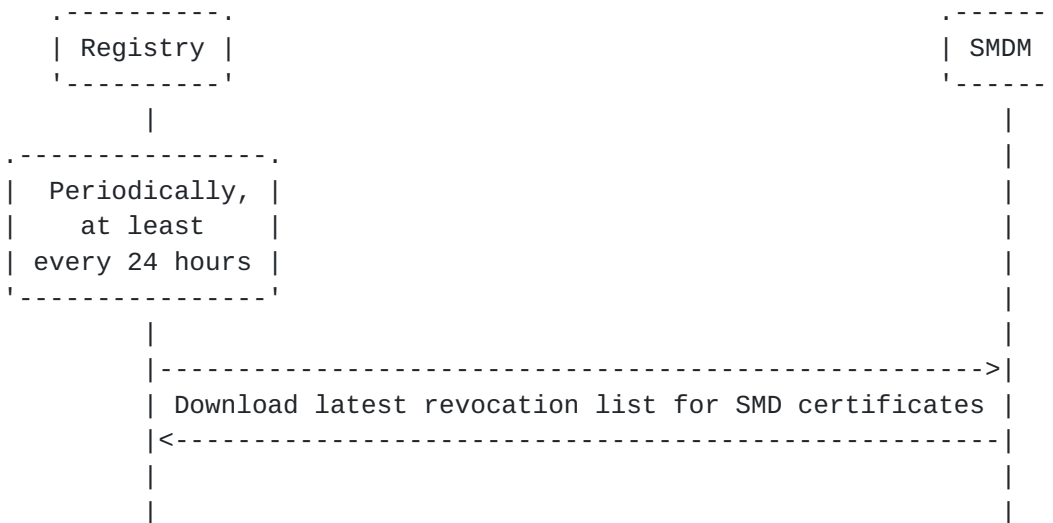


Figure 4



### 5.2.3.2. Certificate Revocation List

Registries MUST refresh their local copy of the CRL at least every 24 hours using the CRL distribution point specified in the TMV certificate.

Operationally, the CRL file and CRL distribution point is the same for all TMVs and (at publication of this document) located at < <http://crl.icann.org/tmch.crl> >.

Note: the CRL file will be the same regardless of the TLD. If a Backend Registry Operator manages the infrastructure of several TLDs, the Backend Registry Operator could refresh the CRL file once every 24 hours, the CRL file could be used for all the TLDs managed by the Backend Registry Operator.

Update CRL for TMV certificates

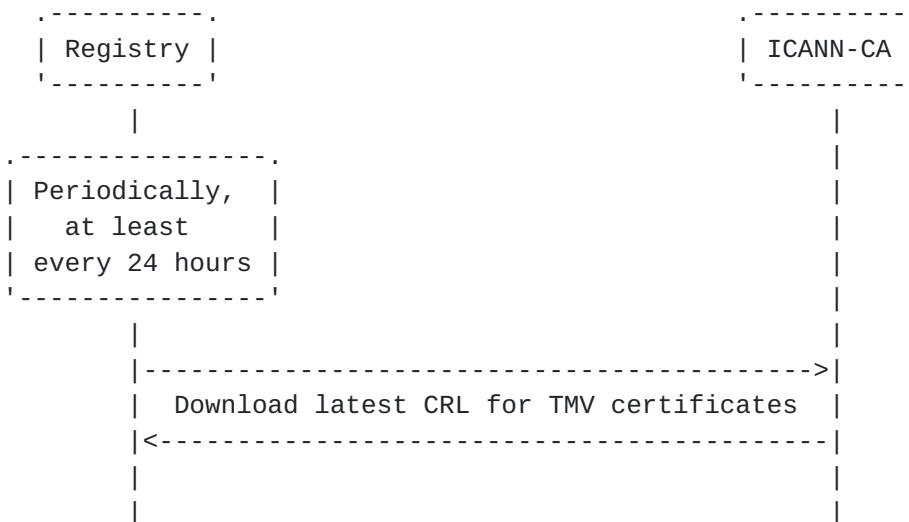


Figure 5



#### **5.2.3.3. Notice of Registered Domain Names**

The Registry MUST send a LORDN file containing DNSs effectively allocated to the TMDB (over the yd interface, [Section 4.3.7](#)).

The effective allocation of a DN MUST be reported by the Registry to the TMDB within 26 hours of the effective allocation of such DN.

The Registry MUST create and upload a LORDN file in case there are effective allocations in the SRS, that have not been successfully reported to the TMDB in a previous LORDN file.

Based on the timers used by TMVs and the TMDB, the RECOMMENDED maximum frequency to upload LORDN files from the Registries to the TMDB is every 3 hours.

It is RECOMMENDED that Registries try to upload at least two LORDN files per day to the TMDB with enough time in between, in order to have time to fix problems reported in the LORDN file.

The Registry SHOULD upload a LORDN file only when the previous LORDN file has been processed by the TMDB and the related LORDN Log file has been downloaded and processed by the Registry.

The Registry MUST upload LORDN files for DNSs effectively allocated during the Sunrise or Claims period (same applies to DNSs effectively allocated using applications created during the Sunrise or Claims period in case of using asynchronous registrations).

The yd interface ([Section 4.3.7](#)) MUST support at least 1 and MAY support up to 10 concurrent connections from each IP address registered by a Registry Operator to access the service.

The TMDB MUST process each uploaded LORDN file and make the related log file available for Registry download within 30 minutes of the finalization of the upload.



NORDN

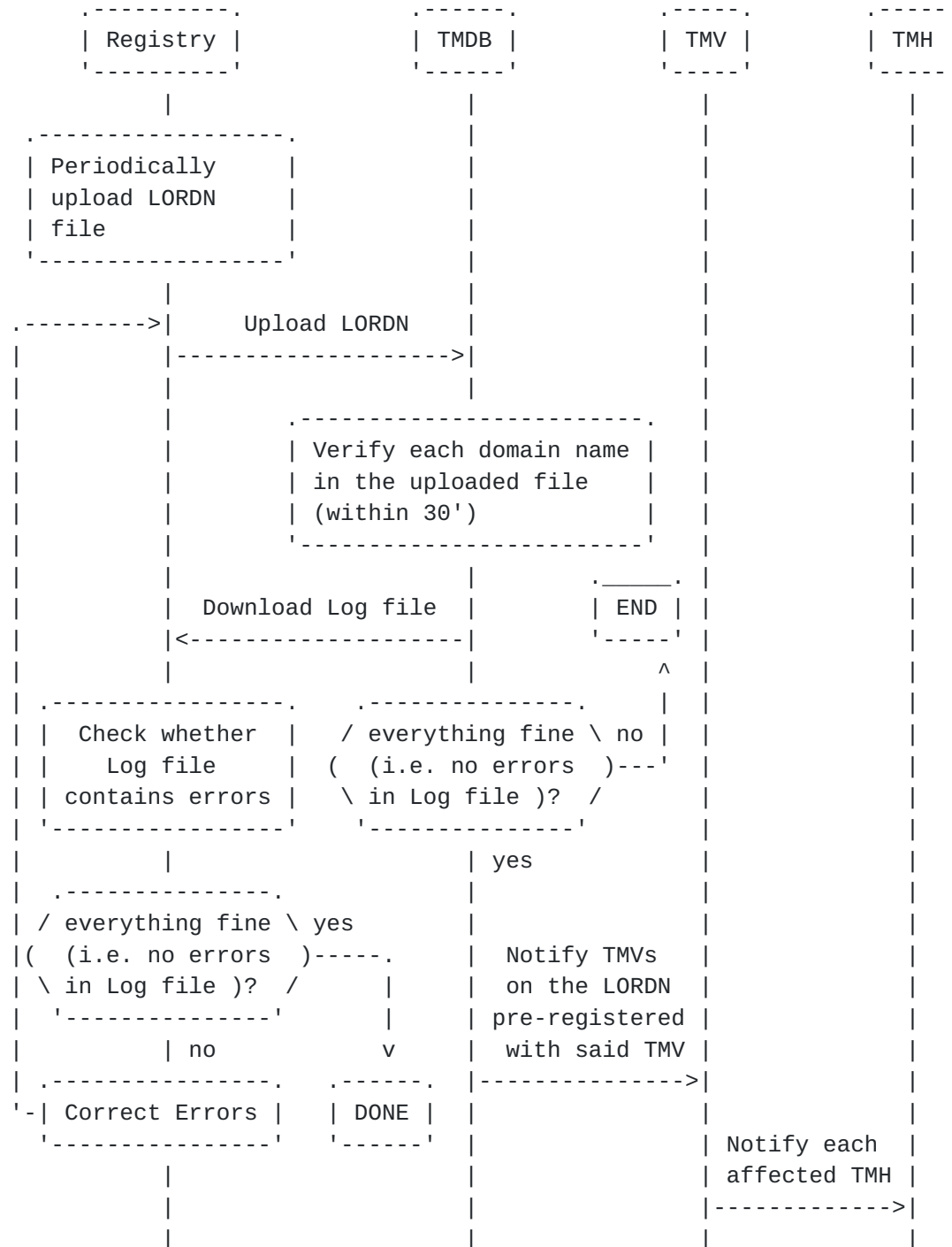


Figure 6

The format used for the LORDN is described in [Section 6.3](#)



#### **5.2.4. Sunrise DN registration by Registrars**

Registrars MAY choose to perform the checks for verifying DN registrations as performed by the Registries (see [Section 5.2.2](#)) before sending the command to register a DN.

#### **5.2.5. TMDB Sunrise Services for Registrars**

The processes described in [Section 5.2.3.1](#) and [Section 5.2.3.2](#) are also available for Registrars to optionally validate the SMDs received.

### 5.3. Trademark Claims Period

#### 5.3.1. Domain Registration

Registration during Trademark Claims Period

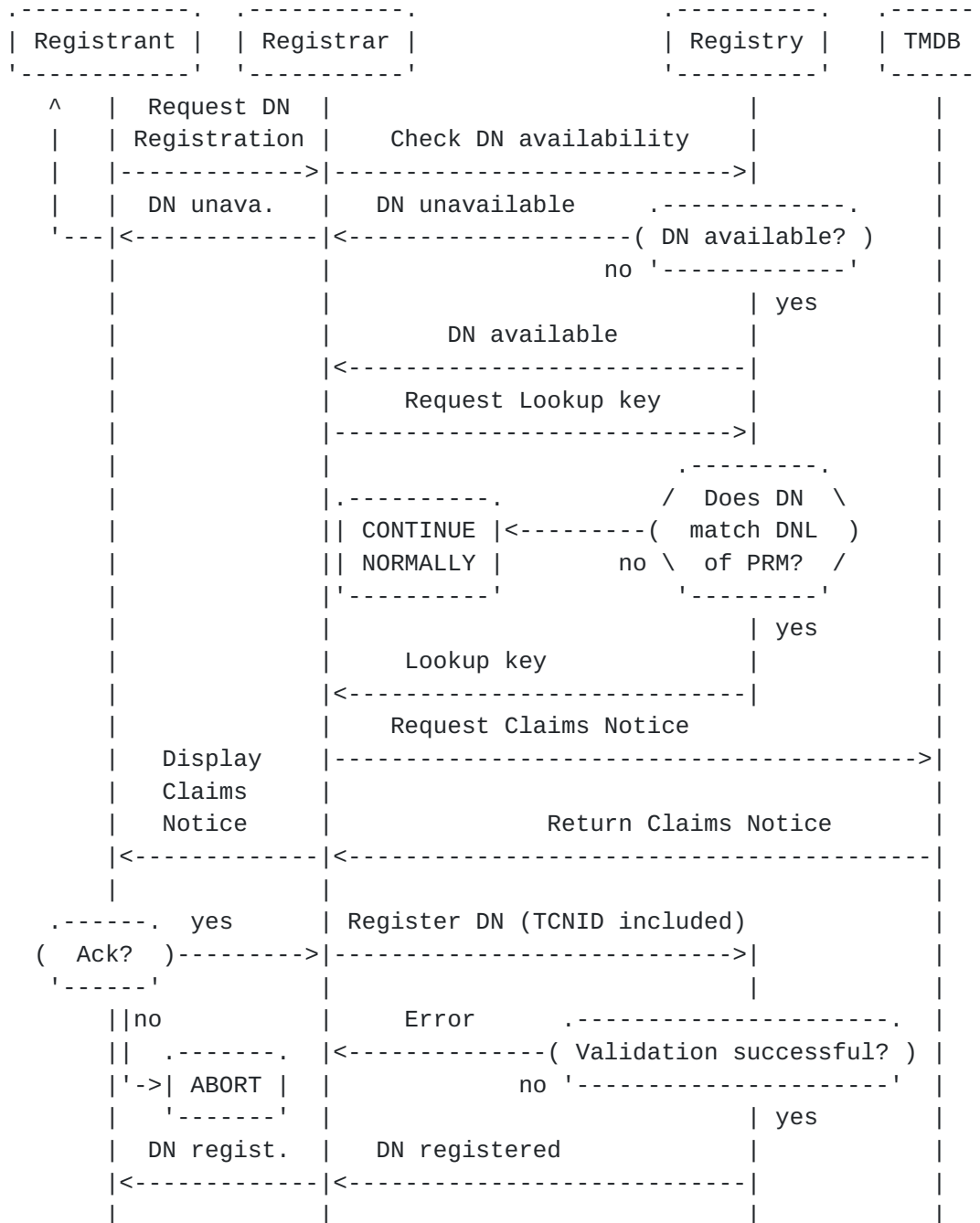


Figure 7



Note: the figure depicted above represents a synchronous DN registration workflow (usually called first come first served).

### **5.3.2. Trademark Claims DN registration by Registries**

During Trademark Claim Periods, Registries perform two main functions:

- o Registries MUST provide Registrars (over the ry interface, [Section 4.3.5](#)) the Lookup Key used to retrieve the TCNs for DNs that match the DNL List.
- o Registries MUST provide the Lookup Key only when queried about a specific DN.
- o For each DN matching a DNL of a PRM, Registries MUST perform a minimum set of checks for verifying DN registrations during Trademark Claims Period upon reception of a registration request over the ry interface ([Section 4.3.5](#)). If any of these checks fails the Registry MUST abort the registration. Each of these checks MUST be performed before the DN is effectively allocated.
- o In case of asynchronous registrations (e.g. auctions), the minimum set of checks MAY be performed when creating the intermediate object (e.g. a DN application) used for DN effective allocation. If the minimum set of checks is performed when creating the intermediate object (e.g. a DN application) a Registry MAY effectively allocate the DN without performing the minimum set of checks again.
- o Performing the minimum set of checks Registries MUST verify that:
  1. The TCNID (<tmNotice:id>), expiration datetime (<tmNotice: notAfter>) and acceptance datetime of the TCN, have been received from the Registrar along with the DN registration.

If the three elements mentioned above are not provided by the Registrar for a DN matching a DNL of a PRM, but the DNL was inserted (or re-inserted) for the first time into DNL List less than 24 hours ago, the registration MAY continue without this data and the tests listed below are not required to be performed.

2. The TCN has not expired (according to the expiration datetime sent by the Registrar).
3. The acceptance datetime is no more than 48 hours in the past.
4. Using the leftmost DNL (A-label in the case of IDNs) of the DN being registered, the expiration datetime provided by the registrar, and the TMDB Notice Identifier extracted from the



TCNID provided by the registrar compute the TCN Checksum.  
Verify that the computed TCN Checksum match the TCN Checksum  
present in the TCNID.

These procedures apply to all DN registrations at the second level as  
well as to all other levels subordinate to the TLD that the Registry  
accepts registrations for.

### **5.3.3. TMDB Claims Services for Registries**

#### **5.3.3.1. Domain Name Label (DNL) List**

A new DNL List MUST be published by the TMDB twice a day, by 00:00:00 and 12:00:00 UTC.

Registries MUST refresh the latest version of the DNL List at least once every 24 hours.

Update DNL List

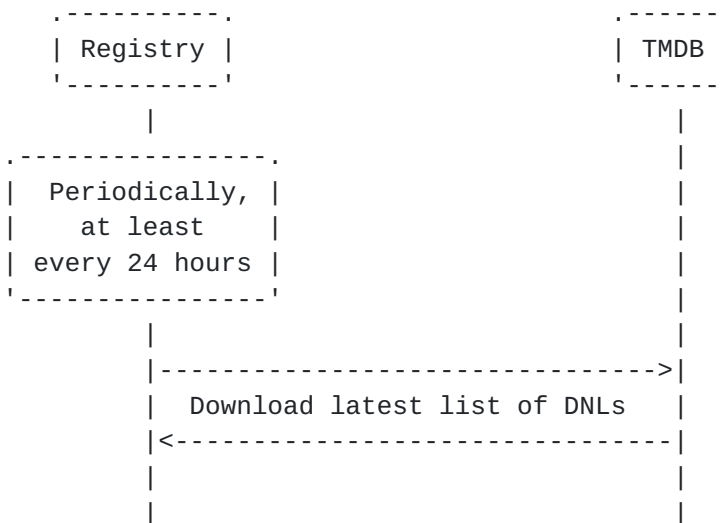


Figure 8

Note: the DNL List will be the same regardless of the TLD. If a Backend Registry Operator manages the infrastructure of several TLDs, the Backend Registry Operator could refresh the DNL List once every 24 hours, the DNL List could be used for all the TLDs managed by the Backend Registry Operator.



#### **5.3.3.2. Notice of Registered Domain Names**

The NORDN process during the Trademark Claims Period is almost the same as during Sunrise Period as defined in [Section 5.2.3.3](#) with the difference that only registrations subject to a Trademark Claim (i.e., at registration time the name appeared in the current DNL List downloaded by the Registry Operator) are included in the LORDN.

#### **5.3.4. Trademark Claims DN Registration by Registrars**

For each DN matching a DNL of a PRM, Registrars MUST perform the following steps:

1. Use the Lookup Key received from the Registry to obtain the TCN from the TMDB using the dr interface ([Section 4.3.6](#)) Registrars MUST only query for the Lookup Key of a DN that is available for registration.
2. Present the TCN to the Registrant as described in [[ICANN-GTLD-AGB-20120604](#)].
3. Ask Registrant for acknowledgement, i.e. the Registrant MUST consent with the TCN, before any further processing. (The transmission of a TCNID to the Registry over the ry interface, [Section 4.3.5](#) implies that the Registrant has expressed his consent with the TCN.)
4. Perform the minimum set of checks for verifying DN registrations. If any of these checks fails the Registrar MUST abort the DN registration. Each of these checks MUST be performed before the registration is sent to the Registry. Performing the minimum set of checks Registrars MUST verify that:
  1. The time when the validation is done is within the TCN validity based on the <tmNotice:notBefore> and <tmNotice:notAfter> elements.
  2. The leftmost DNL (A-label in case of IDNs) of the DN being effectively allocated matches the label (<tmNotice:label>) element in the TCN.
  3. The Registrant has acknowledged (expressed his consent with) the TCN.
5. Record the date and time when the registrant acknowledged the TCN.
6. Send the registration to the Registry (ry interface, [Section 4.3.5](#)) and include the following information:
  - \* TCNID (<tmNotice:id>)
  - \* Expiration date of the TCN (<tmNotice:notAfter>)
  - \* Acceptance datetime of the TCN.



TCN are generated twice a day. The expiration date (<tmNotice: notAfter>) of each TCN MUST be set to 48 hours in the future by the TMDB, allowing the implementation of a cache by Registrars and enough time for acknowledging the TCN. Registrars SHOULD implement a cache of TCNs to minimize the number of queries sent to the TMDB. A cached TCN MUST be removed from the cache after the expiration date of the TCN as defined by <tmNotice: notAfter>. The TMDB MAY implement rate-limiting as one of the protection mechanisms to mitigate the risk of performance degradation.

### **[5.3.5.](#) TMDB Claims Services for Registrars**

#### **[5.3.5.1.](#) Claims Notice Information Service**

The TCNs are provided by the TMDB online and are fetched by the Registrar via the dr interface ([Section 4.3.6](#)).

To get access to the TCNs, the Registrar needs the credentials provided by the TMDB ([Section 5.1.2.1](#)) and the Lookup Key received from the Registry via the ry interface ([Section 4.3.5](#)). The dr interface ([Section 4.3.6](#)) uses HTTPS with Basic access authentication.

The dr interface ([Section 4.3.6](#)) MAY support up to 10 concurrent connections from each Registrar.

The URL of the dr interface ([Section 4.3.6](#)) is:

```
< https://<tmdb-domain-name>/cnis/<lookupkey>.xml >
```

Note that the "lookupkey" may contain SLASH characters ("/"). The SLASH character is part of the URL path and MUST NOT be escaped when requesting the TCN.

The TLS certificate (HTTPS) used on the dr interface ([Section 4.3.6](#)) MUST be signed by a well-know public CA. Registrars MUST perform the Certification Path Validation described in [Section 6 of \[RFC5280\]](#). Registrars will be authenticated in the dr interface using HTTP Basic access authentication. The dr ([Section 4.3.6](#)) interface MUST support HTTPS keep-alive and MUST maintain the connection for up to 30 minutes.



## **5.4. Qualified Launch Program Period**

### **5.4.1. Domain Registration**

During the OPTIONAL (see [[QLP-Addendum](#)]) Qualified Launch Program (QLP) period effective allocations of DNS to third parties could require that Registries and Registrars provide Sunrise and/or Claims services. If required, Registries and Registrars MUST provide Sunrise and/or Claims services as described in: [Section 5.2](#) and [Section 5.3](#).

The effective allocation scenarios are:

- o If the leftmost DNL (A-label in case of IDNs) of the DN being effectively allocated (QLP Name in this section) matches a DNL in the SURL, and an SMD is provided, then Registries MUST provide Sunrise Services (see [Section 5.2](#)) and the DN MUST be reported in a Sunrise LORDN file during the QLP period.
- o If the QLP Name matches a DNL in the SURL but does not match a DNL in the DNL List, and an SMD is NOT provided (see section 2.2 of [[QLP-Addendum](#)]), then the DN MUST be reported in a Sunrise LORDN file using the special SMD-id "99999-99999" during the QLP period.
- o If the QLP Name matches a DNL in the SURL and also matches a DNL in the DNL List, and an SMD is NOT provided (see section 2.2 of [[QLP-Addendum](#)]), then Registries MUST provide Claims services (see [Section 5.3](#)) and the DN MUST be reported in a Claims LORDN file during the QLP period.
- o If the QLP Name matches a DNL in the DNL List but does not match a DNL in the SURL, then Registries MUST provide Claims services (see [Section 5.2](#)) and the DN MUST be reported in a Claims LORDN file during the QLP period.



The following table lists all the effective allocation scenarios during a QLP Period:

| QLP<br>Name<br>match<br>in the<br>SURL<br>List | QLP<br>Name<br>match<br>in the<br>DNL<br>List | SMD was<br>provided by the<br>potential<br>Registrant        | Registry<br>MUST<br>provide<br>Sunrise or<br>Claims<br>Services | Registry MUST<br>report DN<br>registration<br>in <type><br>LORDN file |
|--|---|--|---|---|
| Y  | Y   | Y  | Sunrise   | Sunrise   |
| Y  | N   | Y  | Sunrise   | Sunrise   |
| N  | Y   | --   | Claims  | Claims  |
| N  | N   | --   | --  | --  |
| Y  | Y   | N (see section<br>2.2 of<br><a href="#">[QLP-Addendum]</a> ) | Claims  | Claims  |
| Y  | N   | N (see section<br>2.2 of<br><a href="#">[QLP-Addendum]</a> ) | --  | Sunrise (using<br>special<br>SMD-id)                                  |

#### QLP Effective Allocation Scenarios

The TMDB MUST provide the following services to Registries during a QLP period:

- o SMD Revocation List (see [Section 5.2.3.1](#))
- o Notice of Registered Domain Names (see [Section 5.2.3.3](#))
- o Domain Name Label List (see [Section 5.3.3.1](#))
- o Sunrise List (see [Section 5.4.2.1](#))

The TMDB MUST provide the following services to Registrars during a QLP period:

- o SMD Revocation List (see [Section 5.2.3.1](#))
- o Claims Notice Information Service (see [Section 5.3.5.1](#))



### [5.4.2.](#) TMDB QLP Services for Registries

#### [5.4.2.1.](#) Sunrise List (SURL)

A new Sunrise List MUST be published by the TMDB twice a day, by 00:00:00 and 12:00:00 UTC.

Registries offering the OPTIONAL QLP period MUST refresh the latest version of the Sunrise List at least once every 24 hours.

Update Sunrise List

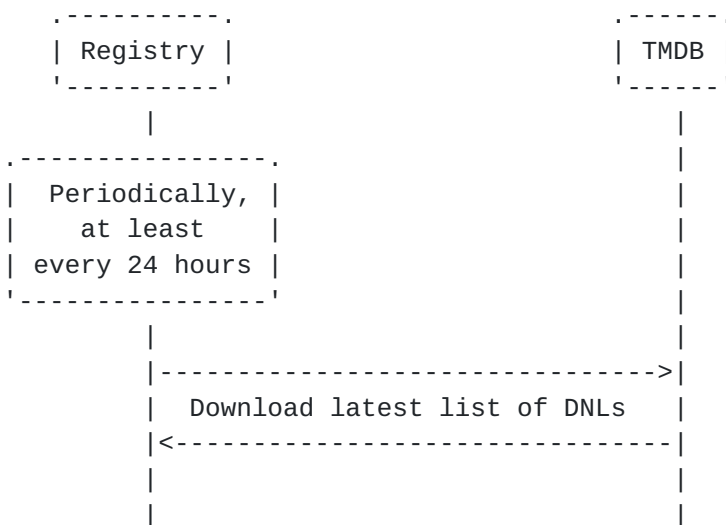


Figure 9

Note: the Sunrise List will be the same regardless of the TLD. If a Backend Registry Operator manages the infrastructure of several TLDs, the Backend Registry Operator could refresh the Sunrise List once every 24 hours, the Sunrise List could be used for all the TLDs managed by the Backend Registry Operator.



## 6. Data Format Descriptions

### 6.1. DNL List file

This section defines the format of the list containing every Domain Name Label (DNL) that matches a Pre-Registered Mark (PRM). The list is maintained by the TMDB and downloaded by Registries in regular intervals (see [Section 5.3.3.1](#)). The Registries use the DNL List during the Trademark Claims period to check whether a requested DN matches a DNL of a PRM.

The DNL List contains all the DNLs covered by a PRM present in the TMDB at the time it is generated.

The DNL List is contained in a CSV-like formatted file that has the following structure:

- o first line: <version>,<DNL List creation datetime>

Where:

- + <version>, version of the file, this field MUST be 1.
- + <DNL List creation datetime>, date and time in UTC that the DNL List was created.

- o second line: a header line as specified in [[RFC4180](#)]

With the header names as follows:

DNL,lookup-key,insertion-datetime

- o One or more lines with: <DNL>,<lookup key>,<DNL insertion datetime>

Where:

- + <DNL>, a Domain Name Label covered by a PRM.
- + <lookup key>, lookup key that the Registry MUST provide to the Registrar. The lookup key has the following format: <YYYY><MM><DD><vv>/<X>/<X>/<X>/<Random bits><Sequential number>, where:
  - YYYY: year that the TCN was generated.
  - MM: zero-padded month that the TCN was generated.



- DD: zero-padded day that the TCN was generated.
  - vv: version of the TCN, possible values are 00 and 01.
  - X: one hexadecimal digit [0-9A-F]. This is the first, second and third hexadecimal digit of encoding the <Random bits> in base16 as specified in [[RFC4648](#)].
  - Random bits: 144 random bits encoded in base64url as specified in [[RFC4648](#)].
  - Sequential number: zero-padded natural number in the range 0000000001 to 2147483647.
- + <DNL insertion datetime>, datetime in UTC that the DNL was first inserted into the DNL List. The possible two values of time for inserting a DNL to the DNL List are 00:00:00 and 12:00:00 UTC.

Example for DNL List

```
1,2012-08-16T00:00:00.0Z
DNL,lookup-key,insertion-datetime
example,2013041500/2/6/9/rJ1NrD092vDsAzf7EQzgjX4R0000000001,\
  2010-07-14T00:00:00.0Z
another-example,2013041500/6/A/5/a1JAqG2vI2BmCv5PfUvuDkf40000000002,\
  2012-08-16T00:00:00.0Z
anotherexample,2013041500/A/C/7/rHdC4wnrWRvPY6nneCVtQhFj0000000003,\
  2011-08-16T12:00:00.0Z
```

Figure 10

To provide authentication and integrity protection, the DNL List will be PGP [[RFC4880](#)] signed by the TMDB with the private key of the TMDB (see also [Section 5.1.1.4](#)). The PGP signature of the DNL List can be found in the similar URI but with extension .sig as shown below.

The URL of the dy interface ([Section 4.3.3](#)) is:

- o < https://<tmdb-domain-name>/dnl/dnl-latest.csv >
- o < https://<tmdb-domain-name>/dnl/dnl-latest.sig >



## 6.2. SMD Revocation List

This section defines the format of the list of SMDs that have been revoked. The list is maintained by the SMDM and downloaded by Registries (and optionally by Registrars) in regular intervals (see [Section 5.2.3.1](#)). The SMD Revocation List is used during the Sunrise Period to validate SMDs received. The SMD Revocation List has a similar function as CRLs used in PKI [[RFC5280](#)] / [[RFC6818](#)].

The SMD Revocation List contains all the revoked SMDs present in the TMDB at the time it is generated.

The SMD Revocation List is contained in a CSV-like formatted file that has the following structure:

- o first line: <version>,<SMD Revocation List creation datetime>

Where:

- + <version>, version of the file, this field MUST be 1.
- + <SMD Revocation List creation datetime>, datetime in UTC that the SMD Revocation List was created.

- o second line: a header line as specified in [[RFC4180](#)]

With the header names as follows:

smd-id,insertion-datetime

- o One or more lines with: <smd-id>,<revoked SMD datetime>

Where:

- + <smd-id>, identifier of the SMD that was revoked.
- + <revoked SMD datetime>, revocation datetime in UTC of the SMD. The possible two values of time for inserting an SMD to the SMD Revocation List are 00:00:00 and 12:00:00 UTC.

To provide integrity protection, the SMD Revocation List is PGP [[RFC4880](#)] signed by the TMDB with the private key of the TMDB (see also [Section 5.1.1.4](#)). The SMD Revocation List is provided by the TMDB with extension .csv. The PGP signature of the SMD Revocation List can be found in the similar URI but with extension .sig as shown below.

The URL of the sr interface ([Section 4.3.12](#)) and sy interface



([Section 4.3.11](#)) is:

o < https://<tmdb-domain-name>/smdrl/smdrl-latest.csv >

o < https://<tmdb-domain-name>/smdrl/smdrl-latest.sig >

Example for SMD Revocation list

```
1,2012-08-16T00:00:00.0Z  
smd-id,insertion-datetime  
2-2,2012-08-15T00:00:00.0Z  
3-2,2012-08-15T00:00:00.0Z  
1-2,2012-08-15T00:00:00.0Z
```

Figure 11



### 6.3. LORDN File

This section defines the format of the List of Registered Domain Names (LORDN), which is maintained by each Registry and uploaded at least daily to the TMDB. Every time a DN matching a DNL of a PRM said DN is added to the LORDN along with further information related to its registration.

The URIs of the yd interface ([Section 4.3.7](#)) used to upload the LORDN file is:

- o Sunrise LORDN file:

< https://<tmdb-domain-name>/LORDN/<TLD>/sunrise >

- o Claims LORDN file:

< https://<tmdb-domain-name>/LORDN/<TLD>/claims >

During a QLP period, Registries MAY be required to upload Sunrise or Claims LORDN files. The URIs of the yd interface used to upload LORDN files during a QLP period is:

- o Sunrise LORDN file (during QLP period):

< https://<tmdb-domain-name>/LORDN/<TLD>/sunrise/qlp >

- o Claims LORDN file (during a QLP period):

< https://<tmdb-domain-name>/LORDN/<TLD>/claims/qlp >

The yd interface ([Section 4.3.7](#)) returns the following HTTP status codes after a HTTP POST request method is received:

- o The interface provides a HTTP/202 status code if the interface was able to receive the LORDN file and the syntax of the LORDN file is correct.

The interface provides the LORDN Transaction Identifier in the HTTP Entity-body that would be used by the Registry to download the LORDN Log file. The LORDN Transaction Identifier is a natural number zero-padded in the range 000000000000000001 to 9223372036854775807.

The TMDB uses the <LORDN creation datetime> element of the LORDN file as a unique client-side identifier. If a LORDN file with the same <LORDN creation datetime> of a previously sent LORDN file is received by the TMDB, the LORDN Transaction



Identifier of the previously sent LORDN file MUST be provided to the Registry. The TMDB MUST ignore the DN Lines present in the LORDN file if a LORDN file with the same <LORDN creation datetime> was previously sent.

The HTTP Location header field contains the URI where the LORDN Log file could be retrieved later, for example:

202 Accepted

Location: https://<tmdb-domain-name>/LORDN/example/sunrise/  
00000000000000000001/result

- o The interface provides a HTTP/400 if the request is incorrect or the syntax of the LORDN file is incorrect. The TMDB MUST return a human readable message in the HTTP Entity-body regarding the incorrect syntax of the LORDN file.
- o The interface provides a HTTP/401 status code if the credentials provided does not authorize the Registry Operator to upload a LORDN file.
- o The TMDB MUST return a HTTP/404 status code when trying to upload a LORDN file using the  
https://<tmdb-domain-name>/LORDN/<TLD>/sunrise/qlp or  
https://<tmdb-domain-name>/LORDN/<TLD>/claims/qlp interface  
outside of a QLP period plus 26 hours.
- o The interface provides a HTTP/500 status code if the system is experiencing a general failure.

For example, to upload the Sunrise LORDN file for TLD "example", the URI would be:

< https://<tmdb-domain-name>/LORDN/example/sunrise >

The LORDN is contained in a CSV-like formatted file that has the following structure:

- o For Sunrise Period:
  - \* first line: <version>,<LORDN creation datetime>,<Number of DN Lines>

Where:

- <version>, version of the file, this field MUST be 1.



- <LORDN creation datetime>, date and time in UTC that the LORDN was created.
  - <Number of DN Lines>, number of DN Lines present in the LORDN file.
- \* second line: a header line as specified in [[RFC4180](#)]

With the header names as follows:

roid, domain-name, SMD-id, registrar-id, registration-datetime, application-datetime

- \* One or more lines with: <roid>, <DN registered>, <SMD-id>, <IANA Registrar id>, <datetime of registration>, <datetime of application creation>

Where:

- <roid>, DN Repository Object Identifier (DNROID) in the SRS.
- <DN registered>, DN that was effectively allocated. For IDNs the A-Label is used [[RFC5890](#)]
- <SMD-id>, SMD ID used for registration.
- <IANA Registrar ID>, IANA Registrar ID.
- <datetime of registration>, date and time in UTC that the domain was effectively allocated.
- OPTIONAL <datetime of application creation>, date and time in UTC that the application was created. The <datetime of application creation> MUST be provided in case of a DN effective allocation based on an asynchronous registration (e.g., when using auctions).



## Example for LORDN during Sunrise

```

1,2012-08-16T00:00:00.0Z,3
roid, domain-name, SMD-id, registrar-id, registration-datetime, \
  application-datetime
SH8013-REP, example1.gtld, 1-2, 9999, 2012-08-15T13:20:00.0Z, \
  2012-07-15T00:50:00.0Z
EK77-REP, example2.gtld, 2-2, 9999, 2012-08-15T14:00:03.0Z
HB800-REP, example3.gtld, 3-2, 9999, 2012-08-15T15:40:00.0Z

```

Figure 12

## o For Trademark Claims Period:

- \* first line: <version>, <LORDN creation datetime>, <Number of DN Lines>

Where:

- <version>, version of the file, this field MUST be 1.
- <LORDN creation datetime>, date and time in UTC that the LORDN was created.
- <Number of DN Lines>, number of DN Lines present in the LORDN file.

- \* second line: a header line as specified in [[RFC4180](#)]

With the header names as follows:

```

roid, domain-name, notice-id, registrar-id, registration-
datetime, ack-datetime, application-datetime

```

- \* One or more lines with: <roid>, <DN registered>, <TCNID>, <IANA Registrar id>, <datetime of registration>, <datetime of acceptance of the TCN>, <datetime of application creation>

Where:

- <roid>, DN Repository Object Identifier (DNROID) in the SRS.
- <DN registered>, DN that was effectively allocated. For IDNs the A-Label is used [[RFC5890](#)].



- <TCNID>, Trademark Claims Notice Identifier as specified in <tmNotice:id>.
- <IANA Registrar ID>, IANA Registrar ID.
- <datetime of registration>, date and time in UTC that the domain was effectively allocated.
- <datetime of acceptance of the TCN>, date and time in UTC that the TCN was acknowledged.
- OPTIONAL <datetime of application creation>, date and time in UTC that the application was created. The <datetime of application creation> MUST be provided in case of a DN effective allocation based on an asynchronous registration (e.g., when using auctions).

For a DN matching a DNL of a PRM at the moment of registration, created without the TCNID, expiration datetime and acceptance datetime, because DNL was inserted (or re-inserted) for the first time into DNL List less than 24 hours ago, the string "recent-dnl-insertion" MAY be specified in <TCNID> and <datetime of acceptance of the TCN>.

Example for LORDN during Claims

```
1,2012-08-16T00:00:00.0Z,3
roid, domain-name, notice-id, registrar-id, registration-datetime, \
    ack-datetime, application-datetime
SH8013-REP, example1.gtld, a76716ed9223352036854775808, \
    9999, 2012-08-15T14:20:00.0Z, 2012-08-15T13:20:00.0Z
EK77-REP, example2.gtld, a7b786ed9223372036856775808, \
    9999, 2012-08-15T11:20:00.0Z, 2012-08-15T11:19:00.0Z
HB800-REP, example3.gtld, recent-dnl-insertion, \
    9999, 2012-08-15T13:20:00.0Z, recent-dnl-insertion
```

Figure 13



### **6.3.1. LORDN Log File**

After reception of the LORDN file, the TMDB verifies its content for syntactical and semantical correctness. The output of the LORDN file verification is retrieved using the yd interface ([Section 4.3.7](#)).

The URI of the yd interface ([Section 4.3.7](#)) used to retrieve the LORDN Log File is:

- o Sunrise LORDN Log file:

```
< https://<tmdb-domain-name>/LORDN/<TLD>/sunrise/  
<lordn-transaction-identifier>/result >
```

- o Claims LORDN Log file:

```
< https://<tmdb-domain-name>/LORDN/<TLD>/claims/  
<lordn-transaction-identifier>/result >
```

A Registry Operator MUST NOT send more than one request per minute per TLD to download a LORDN Log file.

The yd interface ([Section 4.3.7](#)) returns the following HTTP status codes after a HTTP GET request method is received:

- o The interface provides a HTTP/200 status code if the interface was able to provide the LORDN Log file. The LORDN Log file is contained in the HTTP Entity-body.
- o The interface provides a HTTP/204 status code if the LORDN Transaction Identifier is correct, but the server has not finalized processing the LORDN file.
- o The interface provides a HTTP/400 status code if the request is incorrect.
- o The interface provides a HTTP/401 status code if the credentials provided does not authorize the Registry Operator to download the LORDN Log file.
- o The interface provides a HTTP/404 status code if the LORDN Transaction Identifier is incorrect.
- o The interface provides a HTTP/500 status code if the system is experiencing a general failure.

For example, to obtain the LORDN Log File in case of a Sunrise LORDN file with LORDN Transaction Identifier 00000000000000000001 and TLD



"example" the URI would be:

```
< https://<tmdb-domain-name>/LORDN/example/sunrise/  
00000000000000000001/result >
```

The LORDN Log file is contained in a CSV-like formatted file that has the following structure:

- o first line: <version>,<LORDN Log creation datetime>,<LORDN file creation datetime>,<LORDN Log Identifier>,<Status flag>,<Warning flag>,<Number of DN Lines>

Where:

- + <version>, version of the file, this field MUST be 1.
- + <LORDN Log creation datetime>, date and time in UTC that the LORDN Log was created.
- + <LORDN file creation datetime>, date and time in UTC of creation for the LORDN file that this log file is referring to.
- + <LORDN Log Identifier>, unique identifier of the LORDN Log provided by the TMDB. This identifier could be used by the Registry Operator to unequivocally identify the LORDN Log. The identifier will be a string of a maximum LENGTH of 60 characters from the Base 64 alphabet.
- + <Status flag>, whether the LORDN file has been accepted for processing by the TMDB. Possible values are "accepted" or "rejected".
- + <Warning flag>, whether the LORDN Log has any warning result codes. Possible values are "no-warnings" or "warnings-present".
- + <Number of DN Lines>, number of DNs effective allocations processed in the LORDN file.

A Registry Operator is NOT REQUIRED to process a LORDN Log with a <Status flag>="accepted" and <Warning flag>="no-warnings".

- o second line: a header line as specified in [[RFC4180](#)]

With the header names as follows:



roid,result-code

- o One or more lines with: <roid>,<result code>

Where:

- + <roid>, DN Repository Object IDentifier (DNROID) in the SRS.
- + <result code>, result code as described in [Section 6.3.1.1](#).

Example for LORDN Log file

```
1,2012-08-16T02:15:00.0Z,2012-08-16T00:00:00.0Z,\
00000000000000478Nzs+3VMkR8ckuUynOLmyeqTmZQSbzDuf/R50n2n5QX4=,\
accepted,no-warnings,1
roid,result-code
SH8013-REP,2000
```

Figure 14



### **6.3.1.1. LORDN Log Result Codes**

In Figure 15 the classes of result codes (rc) are listed. Those classes in square brackets are not used at this time, but may come into use at some later stage. The first two digits of a result code denote the result code class, which defines the outcome at the TMDB:

- o ok: Success, DN Line accepted by the TMDB.
- o warn: a warning is issued, DN Line accepted by the TMDB.
- o err: an error is issued, LORDN file rejected by the TMDB.

In case that after processing a DN Line, the error result code is 45xx or 46xx for that DN Line, the LORDN file MUST be rejected by the TMDB. If the LORDN file is rejected, DN Lines that are syntactically valid will be reported with a 2001 result code. A 2001 result code means that the DN Line is syntactically valid, however the DN Line was not processed because the LORDN file was rejected. All DNS reported in a rejected LORDN file MUST be reported again by the Registry because none of the DN Lines present in the LORDN file have been processed by the TMDB.

LORDN Log Result Code Classes

| code | Class                      | outcome |
|------|----------------------------|---------|
| ---- | -----                      | -----   |
| 20xx | Success                    | ok      |
| 35xx | [ DN Line syntax warning ] | warn    |
| 36xx | DN Line semantic warning   | warn    |
| 45xx | DN Line syntax error       | err     |
| 46xx | DN Line semantic error     | err     |

Figure 15

In the following, the LORDN Log result codes used by the TMDB are described:

LORDN Log result Codes

| rc | Short Description |
|----|-------------------|
|    | Long Description  |



- 
- 2000** OK  
DN Line successfully processed.
- 2001** OK but not processed  
DN Line is syntactically correct but was not processed because the LORDN file was rejected.
- 3601** TCN Acceptance Date after Registration Date  
TCN Acceptance Date in DN Line is newer than the Registration Date.
- 3602** Duplicate DN Line  
This DN Line is an exact duplicate of another DN Line in same file, DN Line ignored.
- 3603** DNROID Notified Earlier  
Same DNROID has been notified earlier, DN Line ignored.
- 3604** TCN Checksum invalid  
Based on the DN effectively allocated, the TCNID and the expiration date of the linked TCN, the TCN Checksum is invalid.
- 3605** TCN Expired  
The TCN was already expired (based on the <tmNotice:notAfter> field of the TCN) at the time of acknowledgement.
- 3606** Wrong TCNID used  
The TCNID used for the registration does not match the related DN.
- 3609** Invalid SMD used  
The SMD used for registration was not valid at the moment of registration based on the <smd:notBefore> and <smd:notAfter> elements.  
In case of an asynchronous registration, this refer to the <datetime of application creation>.
- 3610** DN reported outside of the time window  
The DN was reported outside of the required 26 hours reporting window.
- 3611** DN does not match the labels in SMD  
The DN does not match the labels included in the SMD.
- 3612** SMDID does not exist



The SMDID has never existed in the central repository.

**[3613](#) SMD was revoked when used**

The SMD used for registration was revoked more than 24 hours ago of the <datetime of registration>. In case of an asynchronous registration, the <datetime of application creation> is used when validating the DN Line.

**[3614](#) TCNID does not exist**

The TCNID has never existed in the central repository.

**[3615](#) Recent-dnl-insertion outside of the time window**

The DN registration is reported as a recent-dnl-insertion, but the (re) insertion into the DNL occurred more than 24 hours ago.

**[3616](#) Registration Date of DN in claims before the end of Sunrise**

The registration date of the DN is before the end of Sunrise and the DN was reported in a claims LORDN file.

**[3617](#) Registrar has not been approved by the TMDB**

Registrar ID in DN Line has not completed Claims integration testing with the TMDB.

**[3618](#) Registration Date of DN in a QLP LORDN file outside of the QLP period**

The registration date of the DN in a QLP LORDN file is outside of the QLP period.

**[3619](#) TCN was not valid**

The TCN was not valid (based on the <tmNotice:notBefore> field of the TCN) at the time of acknowledgement.

**[4501](#) Syntax Error in DN Line**

Syntax Error in DN Line.

**[4601](#) Invalid TLD used**

The TLD in the DN Line does not match what is expected for this LORDN.

**[4602](#) Registrar ID Invalid**

Registrar ID in DN Line is not a valid ICANN-Accredited Registrar.

**[4603](#) Registration Date in the future**

The <datetime of registration> in the DN Line is in the future.



**4606 TLD not in Sunrise or Claims**

The <datetime of registration> was reported when the TLD was not in Sunrise or Claims.

In case of an asynchronous registration, the <datetime of application creation> is used when validating the DN Line.

**4607 Application Date in the future**

The <datetime of application creation> in the DN Line is in the future.

**4608 Application Date is later than Registration Date**

The <datetime of application creation> in the DN Line is later than the <datetime of registration>.

**4609 TCNID wrong syntax**

The syntax of the TCNID is invalid.

**4610 TCN Acceptance Date is in the future**

The <datetime of acceptance of the TCN> is in the future.

**4611 Label has never existed in the TMDB**

The label in the registered DN has never existed in the TMDB.

Figure 16



#### **6.4. SMD File**

This section defines the format of the SMD File. After a successful registration of a mark, the TMV returns an SMD File to the TMH. The SMD File can then be used for registration of one or more DNS covered by the PRM during the Sunrise Period of a TLD.

Two encapsulation boundaries are defined for delimiting the encapsulated base64 encoded SMD: i.e. "-----BEGIN ENCODED SMD-----" and "-----END ENCODED SMD-----". Only data inside the encapsulation boundaries MUST be used by Registries and Registrars for validation purposes, i.e. any data outside these boundaries as well as the boundaries themselves MUST be ignored for validation purposes.

The structure of the SMD File is as follows:

- o Marks: <marks>
- o smdID: <SMD-ID>
- o U-labels: <comma separated list of labels in presentation form (i.e., U-labels or LDH as the case may be)>
- o notBefore: <begin validity>
- o notAfter: <end validity>
- o -----BEGIN ENCODED SMD-----
- o <encoded SMD (see [[I-D.lozano-tmch-smd](#)])>
- o -----END ENCODED SMD-----



Example for SMD File (shortened at [...]):

Marks: Example One

smdID: 1-2

U-labels: example-one, exampleone

notBefore: 2011-08-16 09:00

notAfter: 2012-08-16 09:00

-----BEGIN ENCODED SMD-----

PD94bWwgdmVyc2lvbj0iMS4wIiB1bmNvZGluZz0iVVRGLTgiPz4KPHNtZDpzaWdu  
ZWRNYXJrIHhtbG5zOnNtZD0idXJuOm1ldGY6cGFyYW1zOnhtbDpuczpzaWduZWRN  
YXJrLTEuMCIgawQ9InNpZ25lZE1hcmsiPgogIDxzbWQ6aWQ+MS0yPC9zbWQ6aWQ+  
CiAgPHNtZDppc3N1ZXJJbmZvIGlzc3VlcklEPSIyIj4KICAgIDxzbWQ6b3JnPkV4  
YW1wbGUgSW5jLjwvc21kOm9yZz4KICAgIDxzbWQ6ZW1haWw+c3VwcG9ydEBleGFt  
cGxlLnRsZDwvc21kOmVtYWlsPgogICAgPHNtZDp1cmw+aHR0cDovL3d3dy5leGFt  
cGxlLnRsZDwvc21kOnVybd4KICAgIDxzbWQ6dm9pY2UgeD0iMTIzNCI+KzEuNzAz  
NTU1NTU1NTwvc21kOnZvaWwNlPgogIDwvc21kOm1zc3VlckluZm8+CiAgPHNtZDpu  
b3RCZWZvcuU+MjAwOS0wOC0xNlQwOTowMDowMC4wWjwvc21kOm5vdEJlZm9yZT4K  
ICA8c21kOm5vdEFmdGVyPjIwMTAtMDgtMTZUMDk6MDA6MDAuMFo8L3NtZDpub3RB  
ZnRlcj4KICA8bWVfYazptYXJrIHhtbG5zOm1hcms9InVybjppZXRmOnBhcmFtczpq4  
[...]

UFMxOWw3REJLcmJ3YnpBZWVpMGpLV1Z6cnZtVjdUQmZqeEQzQVFvMVIKYlU1ZEJy  
NklqYmRMRmxuTzV4MEcwXJHN3g1T1VQdXVyaWh5aVVSceZEchdIOEtBSDF3TWND  
cFhHWEZSdEdLawp3eWRneVZZQXR5N290a2wvejNiWmtDVlQzNGdQdkY3MHNSNitR  
eFV5OHUwTHpGnUEvYmVZYVpweFNZRzMxYW1McKfKwGl0VFdGaXBhSudlYtLsRUdG  
TTBM0StCZzdYek5uNG5WTFhva3lFQjNiZ1M0c2NHNlF6blgyM0ZHazwvWduOUNl  
cnRpZmljYXRlPgo8L1g1MDlEYXRhPgogICAgPC9LZXlJbmZvPgogIDwvU2lnbmF0  
dXJlPgo8L3NtZDpzaWduZWRNYXJrPgo=

-----END ENCODED SMD-----

Figure 17



### **6.5. Trademark Claims Notice**

The TMDB MUST provide the TCN to Registrars in XML format as specified below.

An enclosing element `<tmNotice:notice>` that describes the Trademark Notice to a given label.

The child elements of the `<tmNotice:notice>` element include:

- o A `<tmNotice:id>` element that contains the unique identifier of the Trademark Notice. This element contains the the TCNID.

The TCNID is a string concatenation of a TCN Checksum and the TMDB Notice Identifier. The first 8 characters of the TCNID is a TCN Checksum. The rest is the TMDB Notice Identifier, which is a zero-padded natural number in the range of 00000000000000000001 to 9223372036854775807.

Example of a TCNID:

370d0b7c9223372036854775807.

Where:

+ TCN Checksum=370d0b7c

+ TMDB Notice Identifier=9223372036854775807

The TCN Checksum is a 8 characters long Base16 encoded output of computing the CRC32 of the string concatenation of: label + unix\_timestamp(`<tmNotice:notAfter>`) + TMDB Notice Identifier

TMDB MUST use the Unix time conversion of the `<tmNotice:notAfter>` in UTC to calculate the TCN Checksum. Unix time is defined as the number of seconds that have elapsed since 1970-01-01T00:00:00Z not counting leap seconds. For example, the conversion to Unix time of 2010-08-16T09:00:00.0Z is shown:

unix\_time(2010-08-16T09:00:00.0Z)=1281949200

The TMDB uses the `<tmNotice:label>` and `<tmNotice:notAfter>` elements from the TCN along with the TMDB Notice Identifier to compute the TCN Checksum.

A Registry MUST use the leftmost DNL (A-label in case of IDNs) of the DN being registered, the expiration datetime of the TCN (provided by the Registrar) and the TMDB Notice Identifier



extracted from the TCNID (provided by the Registrar) to compute the TCN Checksum. For example the DN "foo.bar.example" being effectively allocated, the left most label would be "foo".

Example of computation of the TCN Checksum:

CRC32(example-one12819492009223372036854775807)=370d0b7c

- o A <tmNotice:notBefore> element that contains the start of the validity date and time of the TCN.
- o A <tmNotice:notAfter> element that contains the expiration date and time of the TCN.
- o A <tmNotice:label> elements that contain the DNL (A-label in case of IDNs) form of the label that correspond to the DN covered by a PRM.
- o One or more <tmNotice:claim> elements that contain the Trademark Claim. The <tmNotice:claim> element contains the following child elements:
  - \* A <tmNotice:markName> element that contains the mark text string.
  - \* One or more <tmNotice:holder> elements that contains the information of the holder of the mark. An "entitlement" attribute is used to identify the entitlement of the holder, possible values are: owner, assignee or licensee. The child elements of <tmNotice:holder> include:
    - + An OPTIONAL <tmNotice:name> element that contains the name of the holder. A <tmNotice:name> MUST be specified if <tmNotice:org> is not specified.
    - + An OPTIONAL <tmNotice:org> element that contains the name of the organization holder of the mark. A <tmNotice:org> MUST be specified if <tmNotice:name> is not specified.
    - + A <tmNotice:addr> element that contains the address information of the holder of a mark. A <tmNotice:addr> contains the following child elements:
      - One, two or three OPTIONAL <tmNotice:street> elements that contains the organization's street address.
      - A <tmNotice:city> element that contains the organization's city.



- An OPTIONAL <tmNotice:sp> element that contains the organization's state or province.
- An OPTIONAL <tmNotice:pc> element that contains the organization's postal code.
- A <tmNotice:cc> element that contains the organization's country code. This a two-character code from [\[ISO3166-2\]](#).
- + An OPTIONAL <tmNotice:voice> element that contains the organization's voice telephone number.
- + An OPTIONAL <tmNotice:fax> element that contains the organization's facsimile telephone number.
- + An OPTIONAL <tmNotice:email> element that contains the email address of the holder.
- \* Zero or more OPTIONAL <tmNotice:contact> elements that contains the information of the representative of the mark registration. A "type" attribute is used to identify the type of contact, possible values are: owner, agent or thirdparty. The child elements of <tmNotice:contact> include:
  - + A <tmNotice:name> element that contains name of the responsible person.
  - + An OPTIONAL <tmNotice:org> element that contains the name of the organization of the contact.
  - + A <tmNotice:addr> element that contains the address information of the contact. A <tmNotice:addr> contains the following child elements:
    - One, two or three OPTIONAL <tmNotice:street> elements that contains the contact's street address.
    - A <tmNotice:city> element that contains the contact's city.
    - An OPTIONAL <tmNotice:sp> element that contains the contact's state or province.
    - An OPTIONAL <tmNotice:pc> element that contains the contact's postal code.



- A <tmNotice:cc> element that contains the contact's country code. This a two-character code from [\[ISO3166-2\]](#).
- + A <tmNotice:voice> element that contains the contact's voice telephone number.
- + An OPTIONAL <tmNotice:fax> element that contains the contact's facsimile telephone number.
- + A <tmNotice:email> element that contains the contact's email address.
- \* A <tmNotice:jurDesc> element that contains the name (in English) of the jurisdiction where the mark is protected. A jurCC attribute contains the two-character code of the jurisdiction where the mark was registered. This is a two-character code from [\[WIPO.ST3\]](#).
- \* Zero or more OPTIONAL <tmNotice:classDesc> element that contains the description (in English) of the Nice Classification as defined in [\[WIPO-NICE-CLASSES\]](#). A classNum attribute contains the class number.
- \* A <tmNotice:goodsAndServices> element that contains the full description of the goods and services mentioned in the mark registration document.
- \* An OPTIONAL <tmNotice:notExactMatch> element signals that the claim notice was added to the TCN based on other rule than exact match as defined in [\[ICANN-GTLD-AGB-20120604\]](#). The <tmNotice:notExactMatch> contains one or more:
  - + An OPTIONAL <tmNotice:udrp> element that signals that the claim notice was added because of a previously abused name included in an UDRP case. The <tmNotice:udrp> contains:
    - A <tmNotice:caseNo> element that contains the UDRP case number used to validate the previously abused name.
    - A <tmNotice:udrpProvider> element that contains the name of the UDRP provider.
  - + An OPTIONAL <tmNotice:court> element that signals that the claim notice was added because of a previously abused name included in a court's resolution. The <tmNotice:court> contains:



- A <tmNotice:refNum> element that contains the reference number of the court's resolution used to validate the previously abused name.
- A <tmNotice:cc> element that contains the two-character code from [[IS03166-2](#)] of the jurisdiction of the court.
- A <tmNotice:courtName> element that contains the name of the court.

Example of <tmNotice:notice> object:

```
<?xml version="1.0" encoding="UTF-8"?>
<tmNotice:notice xmlns:tmNotice="urn:ietf:params:xml:ns:tmNotice-1.0">
  <tmNotice:id>370d0b7c9223372036854775807</tmNotice:id>
  <tmNotice:notBefore>2010-08-14T09:00:00.0Z</tmNotice:notBefore>
  <tmNotice:notAfter>2010-08-16T09:00:00.0Z</tmNotice:notAfter>
  <tmNotice:label>example-one</tmNotice:label>
  <tmNotice:claim>
    <tmNotice:markName>Example One</tmNotice:markName>
    <tmNotice:holder entitlement="owner">
      <tmNotice:org>Example Inc.</tmNotice:org>
      <tmNotice:addr>
        <tmNotice:street>123 Example Dr.</tmNotice:street>
        <tmNotice:street>Suite 100</tmNotice:street>
        <tmNotice:city>Reston</tmNotice:city>
        <tmNotice:sp>VA</tmNotice:sp>
        <tmNotice:pc>20190</tmNotice:pc>
        <tmNotice:cc>US</tmNotice:cc>
      </tmNotice:addr>
    </tmNotice:holder>
    <tmNotice:contact type="owner">
      <tmNotice:name>Joe Doe</tmNotice:name>
      <tmNotice:org>Example Inc.</tmNotice:org>
      <tmNotice:addr>
        <tmNotice:street>123 Example Dr.</tmNotice:street>
        <tmNotice:street>Suite 100</tmNotice:street>
        <tmNotice:city>Reston</tmNotice:city>
        <tmNotice:sp>VA</tmNotice:sp>
        <tmNotice:pc>20190</tmNotice:pc>
        <tmNotice:cc>US</tmNotice:cc>
      </tmNotice:addr>
      <tmNotice:voice x="4321">+1.7035555555</tmNotice:voice>
      <tmNotice:email>jdoe@example.com</tmNotice:email>
    </tmNotice:contact>
    <tmNotice:jurDesc jurCC="US">UNITED STATES OF AMERICA</tmNotice:jurDesc>
    <tmNotice:classDesc classNum="35">
      Advertising; business management; business administration.
    </tmNotice:classDesc>
    <tmNotice:classDesc classNum="36">
      Insurance; financial affairs; monetary affairs; real estate.
    </tmNotice:classDesc>
    <tmNotice:goodsAndServices>
      Bardus populorum circumdabit se cum captiosus populum.
      Smert populorum circumdabit se cum captiosus populum qui eis differimus.
    </tmNotice:goodsAndServices>
  </tmNotice:claim>
</tmNotice:notice>
```



```
<tmNotice:claim>
  <tmNotice:markName>Example-One</tmNotice:markName>
  <tmNotice:holder entitlement="owner">
    <tmNotice:org>Example S.A. de C.V.</tmNotice:org>
    <tmNotice:addr>
      <tmNotice:street>Calle conocida #343</tmNotice:street>
      <tmNotice:city>Conocida</tmNotice:city>
      <tmNotice:sp>SP</tmNotice:sp>
      <tmNotice:pc>82140</tmNotice:pc>
      <tmNotice:cc>BR</tmNotice:cc>
    </tmNotice:addr>
  </tmNotice:holder>
  <tmNotice:jurDesc jurCC="BR">BRAZIL</tmNotice:jurDesc>
  <tmNotice:goodsAndServices>
    Bardus populorum circumdabit se cum captiosus populum.
    Smert populorum circumdabit se cum captiosus populum qui eis differimus.
  </tmNotice:goodsAndServices>
</tmNotice:claim>
<tmNotice:claim>
  <tmNotice:markName>One</tmNotice:markName>
  <tmNotice:holder entitlement="owner">
    <tmNotice:org>One Corporation</tmNotice:org>
    <tmNotice:addr>
      <tmNotice:street>Otra calle</tmNotice:street>
      <tmNotice:city>Otra ciudad</tmNotice:city>
      <tmNotice:sp>OT</tmNotice:sp>
      <tmNotice:pc>383742</tmNotice:pc>
      <tmNotice:cc>CR</tmNotice:cc>
    </tmNotice:addr>
  </tmNotice:holder>
  <tmNotice:jurDesc jurCC="CR">COSTA RICA</tmNotice:jurDesc>
  <tmNotice:goodsAndServices>
    Bardus populorum circumdabit se cum captiosus populum.
    Smert populorum circumdabit se cum captiosus populum qui eis differimus.
  </tmNotice:goodsAndServices>
  <tmNotice:notExactMatch>
    <tmNotice:court>
      <tmNotice:refNum>234235</tmNotice:refNum>
      <tmNotice:cc>CR</tmNotice:cc>
      <tmNotice:courtName>Supreme Court of Justice of Costa Rica</
tmNotice:courtName>
    </tmNotice:court>
  </tmNotice:notExactMatch>
</tmNotice:claim>
<tmNotice:claim>
  <tmNotice:markName>One Inc</tmNotice:markName>
  <tmNotice:holder entitlement="owner">
    <tmNotice:org>One SA de CV</tmNotice:org>
```

<tmNotice:addr>

Lozano

Expires January 24, 2016

[Page 61]

```
<tmNotice:street>La calle</tmNotice:street>
<tmNotice:city>La ciudad</tmNotice:city>
<tmNotice:sp>CD</tmNotice:sp>
<tmNotice:pc>34323</tmNotice:pc>
<tmNotice:cc>AR</tmNotice:cc>
</tmNotice:addr>
</tmNotice:holder>
<tmNotice:jurDesc jurCC="AR">ARGENTINA</tmNotice:jurDesc>
<tmNotice:goodsAndServices>
  Bardus populorum circumdabit se cum captiosus populum.
  Smert populorum circumdabit se cum captiosus populum qui eis differimus.
</tmNotice:goodsAndServices>
<tmNotice:notExactMatch>
  <tmNotice:udrp>
    <tmNotice:caseNo>D2003-0499</tmNotice:caseNo>
    <tmNotice:udrpProvider>WIPO</tmNotice:udrpProvider>
  </tmNotice:udrp>
</tmNotice:notExactMatch>
</tmNotice:claim>
</tmNotice:notice>
```

For formal syntax of the TCN please refer to [Section 7.1](#).



## 6.6. Sunrise List File

This section defines the format of the list containing every Domain Name Label (DNL) that matches a Pre-Registered Mark (PRM) eligible for Sunrise. The list is maintained by the TMDB and downloaded by Registries in regular intervals (see [Section 5.4.2.1](#)). The Registries use the Sunrise List during the Qualified Launch Program period to check whether a requested DN matches a DNL of a PRM eligible for Sunrise.

The Sunrise List contains all the DNLs covered by a PRM eligible for Sunrise present in the TMDB at the time it is generated.

The Sunrise List is contained in a CSV-like formatted file that has the following structure:

- o first line: <version>,<Sunrise List creation datetime>

Where:

- + <version>, version of the file, this field MUST be 1.
- + <Sunrise List creation datetime>, date and time in UTC that the Sunrise List was created.

- o second line: a header line as specified in [[RFC4180](#)]

With the header names as follows:

DNL,insertion-datetime

- o One or more lines with: <DNL>,<DNL insertion datetime>

Where:

- + <DNL>, a Domain Name Label covered by a PRM eligible for Sunrise.
- + <DNL insertion datetime>, datetime in UTC that the DNL was first inserted into the Sunrise List. The possible two values of time for inserting a DNL to the Sunrise List are 00:00:00 and 12:00:00 UTC.



## Example for Sunrise List

```
1,2012-08-16T00:00:00.0Z
DNL,insertion-datetime
example,2010-07-14T00:00:00.0Z
another-example,2012-08-16T00:00:00.0Z
anotherexample,2011-08-16T12:00:00.0Z
```

Figure 18

To provide authentication and integrity protection, the Sunrise List will be PGP [[RFC4880](#)] signed by the TMDB with the private key of the TMDB (see also [Section 5.1.1.4](#)). The PGP signature of the Sunrise List can be found in the similar URI but with extension .sig as shown below.

The URL of the dy interface ([Section 4.3.3](#)) is:

- o < https://<tmdb-domain-name>/dnl/surl-latest.csv >
- o < https://<tmdb-domain-name>/dnl/surl-latest.sig >



## **7. Formal Syntax**

### **7.1. Trademark Claims Notice**

The schema presented here is for a Trademark Claims Notice.

Copyright (c) 2012 IETF Trust and the persons identified as authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- o Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- o Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- o Neither the name of Internet Society, IETF or IETF Trust, nor the names of specific contributors, may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

BEGIN

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="urn:ietf:params:xml:ns:tmNotice-1.0"
  xmlns:tmNotice="urn:ietf:params:xml:ns:tmNotice-1.0"
  xmlns:mark="urn:ietf:params:xml:ns:mark-1.0"
  xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

  <annotation>
    <documentation>
```



Schema for representing a Trademark Notice.

```
</documentation>
</annotation>
<import namespace="urn:ietf:params:xml:ns:mark-1.0"
  schemaLocation="mark-1.0.xsd"/>
<element name="notice" type="tmNotice:noticeType"/>
<complexType name="holderType">
  <sequence>
    <element name="name" type="token" minOccurs="0"/>
    <element name="org" type="token" minOccurs="0"/>
    <element name="addr" type="tmNotice:addrType"/>
    <element name="voice" type="mark:e164Type" minOccurs="0"/>
    <element name="fax" type="mark:e164Type" minOccurs="0"/>
    <element name="email" type="mark:minTokenType" minOccurs="0"/>
  </sequence>
  <attribute name="entitlement" type="mark:entitlementType"/>
</complexType>
<complexType name="noticeType">
  <sequence>
    <element name="id" type="tmNotice:idType"/>
    <element name="notBefore" type="dateTime"/>
    <element name="notAfter" type="dateTime"/>
    <element name="label" type="mark:labelType"/>
    <element name="claim" type="tmNotice:claimType" minOccurs="0"
      maxOccurs="unbounded"/>
  </sequence>
</complexType>
<complexType name="claimType">
  <sequence>
    <element name="markName" type="token"/>
    <element name="holder" type="tmNotice:holderType"
      maxOccurs="unbounded"/>
    <element name="contact" type="tmNotice:contactType" minOccurs="0"
      maxOccurs="unbounded"/>
    <element name="jurDesc" type="tmNotice:jurDescType"/>
    <element name="classDesc" type="tmNotice:classDescType"
      minOccurs="0" maxOccurs="unbounded"/>
    <element name="goodsAndServices" type="token"/>
    <element name="notExactMatch" type="tmNotice:noExactMatchType"
      minOccurs="0"/>
  </sequence>
</complexType>
<complexType name="jurDescType">
  <simpleContent>
    <extension base="token">
      <attribute name="jurCC" type="mark:ccType" use="required"/>
    </extension>
  </simpleContent>
```



```
</complexType>
<complexType name="classDescType">
  <simpleContent>
    <extension base="token">
      <attribute name="classNum" type="integer" use="required"/>
    </extension>
  </simpleContent>
</complexType>
<complexType name="noExactMatchType">
  <choice maxOccurs="unbounded">
    <element name="udrp" type="tmNotice:udrpType"/>
    <element name="court" type="tmNotice:courtType"/>
  </choice>
</complexType>
<complexType name="udrpType">
  <sequence>
    <element name="caseNo" type="token"/>
    <element name="udrpProvider" type="token"/>
  </sequence>
</complexType>
<complexType name="courtType">
  <sequence>
    <element name="refNum" type="token"/>
    <element name="cc" type="mark:ccType"/>
    <element name="region" type="token" minOccurs="0"
      maxOccurs="unbounded"/>
    <element name="courtName" type="token"/>
  </sequence>
</complexType>
<complexType name="addrType">
  <sequence>
    <element name="street" type="token" minOccurs="1" maxOccurs="3"/>
    <element name="city" type="token"/>
    <element name="sp" type="token" minOccurs="0"/>
    <element name="pc" type="mark:pcType" minOccurs="0"/>
    <element name="cc" type="mark:ccType"/>
  </sequence>
</complexType>
<complexType name="contactType">
  <sequence>
    <element name="name" type="token"/>
    <element name="org" type="token" minOccurs="0"/>
    <element name="addr" type="tmNotice:addrType"/>
    <element name="voice" type="mark:e164Type"/>
    <element name="fax" type="mark:e164Type" minOccurs="0"/>
    <element name="email" type="mark:minTokenType"/>
  </sequence>
  <attribute name="type" type="mark:contactTypeType"/>
</complexType>
```



```
</complexType>
<simpleType name="idType">
  <restriction base="token">
    <pattern value="[a-fA-F0-9]{8}\d{1,19}"/>
  </restriction>
</simpleType>
</schema>
END
```

## **8. Acknowledgements**

This specification is a collaborative effort from several participants in the ICANN community. Bernie Hoeneisen participated as co-author until version 02 providing invaluable support for this document. This specification is based on a model spearheaded by: Chris Wright, Jeff Neuman, Jeff Eckhaus and Will Shorter. The author would also like to thank the thoughtful feedback provided by many in the tmch-tech mailing list, but particularly the extensive help provided by James Gould, James Mitchell and Francisco Arias.

## **9. Change History**

[[RFC Editor: Please remove this section.]]

### **9.1. Changes from [draft-lozano-tmch-func-spec-06](#) to [draft-lozano-tmch-func-spec-07](#)**

1. Added result codes: 3611, 3612, 3613, 3614, 4607, 4608, 4609 and 4610.
2. Added mechanism to retrieve the LORDN Transaction Identifier based on a previously sent <LORDN creation datetime> element.

### **9.2. Changes from [draft-lozano-tmch-func-spec-07](#) to [draft-lozano-tmch-func-spec-08](#)**

1. Updated result codes: 3613.
2. Added result codes: 3615, 3616, 3617 and 4611.
3. Removed result codes: 4605 and 4606 to support Limited Registration Periods as defined in the latest RPM Requirements document.
4. Minor editorial fixes.

### **9.3. Changes from [draft-lozano-tmch-func-spec-08](#) to [draft-lozano-tmch-func-spec-09](#)**

1. Added support for QLP.
2. Updated result codes: 3605.
3. Added result codes: 3618 (QLP) and 3619.



4. XML Schema fix.
5. Minor editorial fixes.

#### **9.4. Changes from [draft-lozano-tmch-func-spec-09](#) to [draft-lozano-tmch-func-spec-10](#)**

Draft expired.

### **10. IANA Considerations**

This document uses URNs to describe XML namespaces and XML schemas conforming to a registry mechanism described in [[RFC3688](#)]. One URI assignment have been registered by the IANA.

Registration request for the Trademark Claims Notice:

URI: urn:ietf:params:xml:ns:tmNotice-1.0

Registrant Contact: See the "Author's Address" section of this document.

XML: None. Namespace URIs do not represent an XML specification.

### **11. Security Considerations**

TBD

### **12. References**

#### **12.1. Normative References**

- [I-D.lozano-tmch-smd]  
Lozano, G., "Mark and Signed Mark Objects Mapping",  
[draft-lozano-tmch-smd-03](#) (work in progress),  
September 2013.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), DOI 10.17487/[RFC2119](#), March 1997,  
<<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", [BCP 81](#), [RFC 3688](#), DOI 10.17487/RFC3688, January 2004,  
<<http://www.rfc-editor.org/info/rfc3688>>.



## **12.2. Informative References**

- [ICANN-GTLD-AGB-20120604]  
ICANN, "gTLD Applicant Guidebook Version 2012-06-04",  
June 2012, <<http://newgtlds.icann.org/en/applicants/agb/guidebook-full-04jun12-en.pdf>>.
- [ISO3166-2]  
ISO, "International Standard for country codes and codes  
for their subdivisions", 2006.
- [QLP-Addendum]  
ICANN, "Qualified Launch Program Addendum", April 2014, <<http://newgtlds.icann.org/en/about/trademark-clearinghouse/rpm-requirements-qlp-addendum-10apr14-en.pdf>>.
- [RFC1034] Mockapetris, P., "Domain names - concepts and facilities",  
STD 13, [RFC 1034](#), DOI 10.17487/RFC1034, November 1987,  
<<http://www.rfc-editor.org/info/rfc1034>>.
- [RFC1035] Mockapetris, P., "Domain names - implementation and  
specification", STD 13, [RFC 1035](#), DOI 10.17487/RFC1035,  
November 1987, <<http://www.rfc-editor.org/info/rfc1035>>.
- [RFC1591] Postel, J., "Domain Name System Structure and Delegation",  
[RFC 1591](#), DOI 10.17487/RFC1591, March 1994,  
<<http://www.rfc-editor.org/info/rfc1591>>.
- [RFC1918] Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G.,  
and E. Lear, "Address Allocation for Private Internets",  
[BCP 5](#), [RFC 1918](#), DOI 10.17487/RFC1918, February 1996,  
<<http://www.rfc-editor.org/info/rfc1918>>.
- [RFC1952] Deutsch, P., "GZIP file format specification version 4.3",  
[RFC 1952](#), DOI 10.17487/RFC1952, May 1996,  
<<http://www.rfc-editor.org/info/rfc1952>>.
- [RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H.,  
Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext  
Transfer Protocol -- HTTP/1.1", [RFC 2616](#), DOI 10.17487/  
[RFC2616](#), June 1999,  
<<http://www.rfc-editor.org/info/rfc2616>>.
- [RFC2818] Rescorla, E., "HTTP Over TLS", [RFC 2818](#), DOI 10.17487/  
[RFC2818](#), May 2000,  
<<http://www.rfc-editor.org/info/rfc2818>>.
- [RFC3339] Klyne, G. and C. Newman, "Date and Time on the Internet:



Timestamps", [RFC 3339](#), DOI 10.17487/RFC3339, July 2002,  
<<http://www.rfc-editor.org/info/rfc3339>>.

- [RFC4180] Shafranovich, Y., "Common Format and MIME Type for Comma-Separated Values (CSV) Files", [RFC 4180](#), DOI 10.17487/[RFC4180](#), October 2005,  
<<http://www.rfc-editor.org/info/rfc4180>>.
- [RFC4346] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.1", [RFC 4346](#), DOI 10.17487/[RFC4346](#), April 2006,  
<<http://www.rfc-editor.org/info/rfc4346>>.
- [RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", [RFC 4648](#), DOI 10.17487/RFC4648, October 2006,  
<<http://www.rfc-editor.org/info/rfc4648>>.
- [RFC4880] Callas, J., Donnerhacke, L., Finney, H., Shaw, D., and R. Thayer, "OpenPGP Message Format", [RFC 4880](#), DOI 10.17487/[RFC4880](#), November 2007,  
<<http://www.rfc-editor.org/info/rfc4880>>.
- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", [RFC 5246](#), DOI 10.17487/[RFC5246](#), August 2008,  
<<http://www.rfc-editor.org/info/rfc5246>>.
- [RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", [RFC 5280](#), DOI 10.17487/RFC5280, May 2008,  
<<http://www.rfc-editor.org/info/rfc5280>>.
- [RFC5730] Hollenbeck, S., "Extensible Provisioning Protocol (EPP)", STD 69, [RFC 5730](#), DOI 10.17487/RFC5730, August 2009,  
<<http://www.rfc-editor.org/info/rfc5730>>.
- [RFC5890] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework", [RFC 5890](#), DOI 10.17487/RFC5890, August 2010,  
<<http://www.rfc-editor.org/info/rfc5890>>.
- [RFC6818] Yee, P., "Updates to the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", [RFC 6818](#), DOI 10.17487/RFC6818, January 2013, <<http://www.rfc-editor.org/info/rfc6818>>.

[WIPO-NICE-CLASSES]



WIPO, "Nice List of Classes", <<http://www.wipo.int/classifications/nivilo/nice/index.htm>>.

[WIPO.ST3]

WIPO, "Recommended standard on two-letter codes for the representation of states, other entities and intergovernmental organizations", March 2007.

## **Appendix A. Document Changelog**

[RFC Editor: This section is to be removed before publication]

### Author's Address

Gustavo Lozano  
ICANN  
12025 Waterfront Drive, Suite 300  
Los Angeles 90292  
US

Phone: +1.3103015800  
Email: [gustavo.lozano@icann.org](mailto:gustavo.lozano@icann.org)

