Country-Code Top-Level Domain Best Current Practices Info

Last Revised November 1, 2006

Zita Wenzel, Randy Bush, Steven Huter

Country-Code Top-Level Domain Best Current Practices

Copyright Notice

Abstract

This document describes the issues and best current practices for the technical organization, operation, and management of country-code top-level domains (ccTLDs).

1. Introduction

A top-level domain (TLD) is one of the domains directly under root in the Domain Name System (DNS) organization of the Internet. The most common generic top-level domains are .com, .net, and .org.

A country-code top-level domain (ccTLD) is one of the top-level domains (TLDs). There are approximately 23 9 two-letter codes
(primarily defined by ISO 3166-1) that designate countries and entities of the world. See ISO 3166-1 [2].

This document summarizes best current practices of ccTLD technical design, operation, and management. Its main objective is to provide helpful information and guidelines for the administrators and technical staff who operate ccTLD Registries to serve their local Internet communities.

For more background information on the Internet's administrative procedures, see "Guide to "Administrative Procedures of the Internet Infrastructure" [28].

1.1 Definitions and conventions

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 RFC 2119 [3].

Note that "cctld" in lower-case is used to represent a single country-code top-level domain (ccTLD), for example .kh for Cambodia or .sn for Senegal. ccTLD refers to all country-code top-level domains.

1.2 Description of the Domain Name System

The Domain Name System is defined by RFC 1034 [4] and RFC 1035 [5], with clarifications, extensions, and modifications given in RFC 1123 [6], RFC 1996 [7], RFC 2181 [8], among others. Also see RFC 434 [9] which updates RFCs 1034, 1035, and 2181.

Over the years, many different words have been used to describe the components of resource naming on the Internet (e.g., URL, URN). To make certain that the set of terms used in this document are well defined and non-ambiguous, the definitions are given here.

1.2.1 Zone file

A zone file contains the domains names and related data for a specific portion of the name space. The zone file is the internal representation the software uses.

1.2.2 DNS records

Domain Name System (DNS) records are specific records that hold data about a specific domain name. They are called Resource Records or RRs. Each RRs has one label (or name), a class, a type, and rdata (data on the right-hand side). The RRs for DNS are A (address), CNAME (canonical name), PTR (pointer), SOA (start of authority), and NS (name server). A zone MUST have SOA and NS RRs. CNAME (canonical name) is rarely used, and only with great caution. A and MX (mail exchanger) are used by the mail sending application.

See RFC 1035 [5] for more information. For more background information, see the "DNS and BIND" book [27].

1.2.3 Primary server

A primary server for a zone holds the original authoritative copy of the DNS records for that zone.

This copy is stored in a zone file. This is the location of the zone file where changes are made.

1.2.4 Secondary server

A secondary server for a zone also holds a complete copy of the records for that zone, which it obtains by transferring them from the primary server whenever a change is made there.

Primary and secondary servers are listed in the NS (name server) records for the zone, and are termed authoritative servers.

It is to be noted that clients looking up names in the DNS do not differentiate between the primary server and secondaries.

1.2.5 Caching server

A caching server holds temporary copies of DNS records; it uses resource records to answer queries about domain names. Since it does not have a copy of the zone file, but merely caches individual resource records it has fetched, it is not authoritative for data in the zone(s).

Further explanation of these terms can be found in RFC 1034 [4].

1.2.6 DNS names

DNS names can be represented in multiple forms, with different properties for internationalization. See RFC 4343 [9]. The most important ones are:

Domain name:

A domain name is a binary representation of a name used internally in the DNS protocol. This consists of a series of labels of 1-63
octets, with an overall length limited to 255 octets (including the length fields).

Email:

In most implementations of applications today, domain names in the Internet have been limited to the much more restricted forms used,
e.g., in email (Simple Mail Transfer Protocol or SMTP), which defines its own rules. Names can be entered in a case-sensitive fashion, but they are interpreted in a case-independent fashion. They are limited to the upper- and lower-case letters a-z, the digits, and the hyphen-minus, all in ASCII. In addition, the specification in RFC 2821 [10] does not allow the components of a domain name in SMTP to start or end with a hyphen.

Internationalized Domain Name:

In the DNS protocols, a name is referred to as a sequence of octets.
However, when discussing requirements for internationalized domain names, the question is to find ways to represent characters that are meaningful for humans.

There are current attempts to define the requirements for an
"Internationalized Domain Name" (IDN) to allow for other scripts and characters. IDN is defined as a sequence of characters that can be used in the context of functions where a name is used today, but contains one or more characters that are outside the set of characters specified as legal characters for names RFC 1123 [6].

Therefore, formally the DNS protocol can transport any octets as a name. However, there are many applications that place restrictions on what they accept. So we suggest that you limit registered names to the narrow acceptability of SMTP. You can actually register whatever you wish, but you SHOULD warn registrants of names that 1)
MAY NOT work with many applications, and 2) MAY be interpreted differently in different (national, language, etc.) contexts.

1.2.7 Registry and registrar

A registry serves as the authoritative repository for all information REQUIRED to resolve domain names registered in the registry's top-level domain (TLD), or second-level domains (SLDs) if the reserved SLD mode is used (e.g., co.uk, ac.nz). The registry also maintains additional information such as the administration and technical contacts for the domain name, the billing contact, and the registrar who registered the domain name.

A registrar provides services to the registrant (the person who registered a domain name) and provides the information to the registry. The registrar provides domain information (servers and contact and billing information) to the registry. The registrar MAY also provide additional value-added services to the registrant such as email, web hosting, etc.

The registrant is the individual end-user who is requesting the domain name.

Normally, the registry and registrar organizations are separate.
There is one registry which SHOULD be administered as a national trust because it is a natural monopoly by definition, and multiple registrars provide competition in registering names with the registry.

A country MAY begin registry services by also acting as the sole,
initial registrar. These functions

MAY be kept separate and the registrar MAY eventually be transitioned away from the registry as one of many registrars.
However, combining the functions MAY also provide a simple, more efficient, organization with less overhead. Note, however, that this would now be a monopoly on two levels and separation later may be problematic.

2. Human resources RECOMMENDED for ccTLD administration

There must be a designated manager for supervising each domain's name space. In the case of a ccTLD, this manager must supervise the domain names and operate the Domain Name System (DNS) in that country.

Two points of contact (POC), with different responsibilities, are REQUIRED.

2.1 Administrative Point of Contact (Admin POC)

The Registry's Administrative POC's role is to make simple,
publishable rules that the applicants and registrars can follow unambiguously. It is a good idea to think of each situation as if it had to be automated. For example, given an application for example.com.ng, you want to be able to write a script which sends a query to some whois.registry-of-companies.gov.ng and see if the street address is the same as the registered company. The Administrative POC SHOULD be representing the local Internet community and be ensuring that the ccTLD is being run for the benefit of the country and its citizens.

2.2 Technical Point of Contact (Tech POC)

The Technical POC's role is to maintain the contents of the zone and to make the system work. This person SHOULD be a programmer or someone familiar with UNIX or Linux, UNIX or Linux tools, and a DNS expert.

2.3 Programmers and technical staff

The human resources necessary to run a ccTLD registry SHOULD typically start with an Administrative POC to handle policy issues,
and a Technical POC to run the domain. A DNS expert, a UNIX systems administrator, and a UNIX tools and web programmer SHOULD be added.
These SHOULD be unique individuals in well-defined roles.

Note that the technical staff SHOULD be carrying out policy decisions and not making policy.

If you are charging any fees, you SHOULD also have a financial person or billing manager. This is especially true if your registry and registrar functions are combined.

A lawyer is also RECOMMENDED.

3. How to structure a ccTLD and why

Each country is free to develop their own system of domain naming within their ccTLD.

3.1 Flat versus hierarchical designs

A flat design allows any name directly under the top-level country-code domain (i.e., the second-

level domain or SLD). For example, mycompany.cctld.

A hierarchical design provides categorized or affinity groups at the second-level. For example, mycollege.edu.cctld, where "edu"
specifies educational institutions.

3.1.1 Two- versus three-letter SLDs

Another choice is whether to use two or three-letter second-level names. For example, some ccTLDs use "ac" for academic while others use "edu" for education. Some ccTLDs use "co" for corporate or commercial while others use "com" for commercial. This is largely a matter of preference, but note that you SHOULD NOT change it later.

Also consider whether using "com" will be easier to match or harder to discriminate from the .com top-level domain (TLD).

3.2 Pros and cons of design choices

Flat designs are easier to start because no definition of groups is necessary (and therefore no decisions need to be made about what names go into which groups) and everyone has equal access.

However, hierarchical designs allow more unique domain names and provide fewer disputes over who has the "rights" to a name. For example, does My Single Company get msc.cctld or does My Small Cooperative get msc.cctld.

SLDs can be used to solve this problem. Assuming the company is a commercial concern and the cooperative is not, My Small Company could register msc.com.cctld and My Small Cooperative could register msc.org.cctld, where "com" is for commercial entities and "org" is for non-profit organizations.

Another possibility is to not allow abbreviations at any level.
my-small-company.cctld and my-small-coop.cctld are uniquely identified and easily found. So are mysmallcompany.cctld and mysmallcooperative.cctld.

It is in your, and your users, best interest to choose a design and be consistent.

See section 3.4 on Changing the Structure.

3.3 Survey of other ccTLD structures

Some examples are:

Country ccTLD Example

====== ===== =======

Flat (non-hierarchical) organization:

Senegal SN ucad.sn (Universite' Cheikh Anta Diop)

Hierarchical, affinity-based second-level domains:

Japan JP co.jp (co stands for corporations), ac.jp (ac stands for academic), etc.

Hierarchical, affinity-based second-level domains:

Uganda UG sc.ug (sc stands for non-baccalaureate schools), etc.

Geo-political organization by state (second-level domains) and city
(third-level domains):

United States US example.los-angeles.ca.us (ca is the state code for California)

3.4 Changing the structure

You SHOULD NOT change the structure. Changing the structure later will cause a number of problems and disputes.

If Mary registered mary.co.cctld and Maryanne registered mary.or.cctld, who would get mary.cctld after a conversion to a flat structure with no SLDs?

Because people in general prefer short names, another possible problem is if a person who has registered earlier has harry.cctld and a person who is currently registering must register under a SLD
(e.g., mary.co.cctld), there MAY be complaints. This is also true for the reverse situation, i.e., if the first person had to register harry.co.cctld and a new person registering can be registered directly under the ccTLD (e.g., mary.cctld).

3.5 Distributed administration

There are pros and cons of the hierarchical distribution (as the United States ccTLD, .us, was originally organized) because the administration is delegated along with the zone (see RFC 1480 [11]).

For example, the name los-angeles.ca.us is registered and then delegated to an entity chosen by the city. Therefore, requests for names under los-angeles.ca.us go to that entity and not to the .us ccTLD

Registry. An advantage is less work for the Registry. However, it is important to have a

contractual relationship with the delegated registrars so that they are following all the rules and regulations of the ccTLD Registry (and RFC 1591 [12]). If you don?t, you MAY have problems with inconsistent policies and implementation. It is important to communicate regularly with all delegated registrars.

4. Technical requirements for ccTLD administration

For hardware, you will need a primary server, one or more secondary servers, and preferably, a test server. These servers SHOULD be robust, geographically and physically separate, and on diverse networks. Note that you SHOULD own and operate the primary server,
but the secondary servers are usually owned and operated elsewhere and you SHOULD have an agreement with those organizations to run your secondary service. In fact, if you can get competently managed and well-connected servers (like NS.RIPE.NET from RIPE) as your secondary servers, you will be in a more secure and reliable position.

Technical stability is the key concern.

It is the goal for the primary server to be in the country, but it is not mandatory. During the startup phase, the primary server MAY be out of the country.

4.1 Secondary servers

See RFC 2182 [13] for rationale and RECOMMENDATIONS about the selection and operation of secondary servers. Secondary servers are REQUIRED for the continued operation of the Internet and it is up to the ccTLD Registry to be firm in enforcing this requirement for the ccTLD and all sub-delegations.

Geographical diversity of DNS servers of the ccTLD is REQUIRED. See section 5 of RFC 2182 [13]. Having secondary servers on different continents is RECOMMENDED.

The NS RRset in the zone MUST match that in the root zone. A similar requirement SHOULD be imposed on delegations below the ccTLD.

4.1.1 Secondary servers for ccTLD sub-delegations

More than one secondary server SHOULD be allowed. More are better.

Of the server set, two MUST satisfy RFC 2182 [13]. It makes no difference which two, they could be secondaries.

4.2 Physically separate networks

It is VERY STRONGLY RECOMMENDED to maintain name servers on physically separate international backbones and physically separate Points of Presence (POPs).

4.3 Physical and electronic security

All of the servers, especially the primary server SHOULD be physically and electronically secure. The primary server SHOULD be in a burglar-proof building with sensor monitors or security guards.
There SHOULD be an Uninterruptible Power Supply (UPS) and a power generator for electricity failures.

o See RFC 2870 [14] and consider which of its recommendations are appropriate for the scale of your particular zone.

o RECOMMEND the use of TSIG for zone transfer to your secondaries. See RFC 2845 [15].

o See RFC 2182 [13] for RECOMMENDATIONS.

o The CERT Coordination Center is a center of Internet security expertise. See CERT [16].

o The SANS (SysAdmin, Audit, Network Security) Institute is a cooperative research and education organization. See SANS [17].

See 10. Security considerations for more information on DNS security.

4.3.1 Zone transfer access

For each zone for which a server is primary, it SHOULD limit "axfr"
zone transfer access to agreed secondary servers for that zone.
Secondaries SHOULD NOT allow "axfr" zone transfer access to the zones for which they are secondary.

4.4 Quality of service (QoS)

You SHOULD aim for 24 hours/7 days/365 days a year connectivity,
availability, and ability and responsiveness to handle the query load.

4.5 DNS software and tools

4.5.1 Operating system and BIND software

You SHOULD NOT use Microsoft servers for DNS service; they behave oddly, scale poorly, and have very bad security problems. UNIX or Linux is RECOMMENDED as the operating system of choice.

To effectively run Internet services, and manage, upgrade, and care for your systems, there is much to learn. While it might initially seem to be easier with Windows, there is a lot to learn no

matter which operating system one chooses. So we have found that it is best to move to UNIX or Linux as soon as possible, and learn as you go.

The Internet Systems Consortium, Inc. (ISC) is a not-for-profit corporation dedicated to developing and maintaining production quality Open Source reference implementations of core Internet protocols. ISC produces and maintains BIND (Berkeley Internet Name Domain), which is an implementation of the Domain Name System (DNS)
protocols and provides an openly redistributable reference implementation of the major components of the Domain Name System,
including:

o a Domain Name System server (named)

o a Domain Name System resolver library

o tools for verifying the proper operation of the DNS server

The BIND DNS Server is used on the vast majority of name serving machines on the Internet, providing a robust and stable architecture on top of which an organization's naming architecture can be built.
The resolver library included in the BIND distribution provides the standard APIs for translation between domain names and Internet addresses and is intended to be linked with applications requiring name service.

BIND software is complex, therefore it has a many year track record of bugs and security issues. It is important to stay very current.

See [18] for more information on ISC and BIND.

Note that there are other implementations of the DNS other than BIND.

4.5.2 Web-based registration software

Web-based registration software is also RECOMMENDED. This software SHOULD be easy-to-use and well documented. Use of pre-written text responses (and FAQs) for common problems is also RECOMMENDED.

Scripts are also RECOMMENDED.

4.5.3 Tools

Tools available and RECOMENDED for setting up on-line registration,
running, and management of a ccTLD include:

The use of the "dig" command (tool) is RECOMMENDED:

o "dig" sends domain name query packets to name servers.

"dig" (Domain Information Groper) is a flexible command line tool which can be used to gather information from the Domain Name System
(DNS) servers. Dig has two modes: simple interactive mode which makes a single query, and batch which executes a query for each in a list of several query lines. All query options are accessible from the command line.

o "ping"

The use of the "ping" command (tool) is RECOMMENDED to measure times to sites to determine if a server is up and running. Note that
?ping? will not pass firewalls.

o "traceroute"

The use of the "traceroute" command (tool) is RECOMMENDED to follow routes and to evaluate routing efficiency.

"traceroute", from a diversely and well-connected host, is the best way to look at RFC 2182 [13] compliance.

4.6 Whois data

The maintenance and availability of registration information via a
"whois" server is RECOMMENDED, but it is not an RFC 1591 requirement.

"whois" data is information relating to the registration of a domain name. These data MAY include name, address, and contact information. Use the RIPE software:

ftp://ftp.ripe.net/ripe/dbase/software/whoisserver-latest.tar.gz

"rwhois" (remote whois service) is NOT RECOMMENDED.

Off-site backup for "whois" data is RECOMMENDED.

4.7 Serial numbers (yyyymmddnn)

See RFCs 1982 [19] and 2182 [13].

Before 1999, serial numbers were often in the form of 99mmddnn
(where mm is the numerical value for the month, dd is the numerical value for the day, and nn is the consecutive number of modifications done in that day).