Examining the User Experience Implications of Active Variant TLDs

Draft Final Report

January 2013

Table of Contents

Executive Summary

As part of the process of implementing internationalized top-level domains or IDN TLDs, ICANN is undertaking several projects to determine the requirements for activating their *variants*. This report examines potential challenges from a user experience perspective when variants of IDN TLDs are activated.

Definition and activation of variants are determined by linguistic and technical communities who may have differing perspectives. Linguistic communities are primarily concerned with end users, and consequently may advocate for a maximal variant label set to enable diverse linguistic expression and easier accessibility. The technical community, on the other hand, is primarily concerned with the security and stability of the Domain Name System (DNS), and therefore stipulates the minimal number of variant labels (if any) be added to the root zone.

To find the right balance between these two perspectives, within the context of usability, this report proposes seven guiding principles for activating the variants of IDN TLDs. They include: (i) *Minimality* (variants must introduce only the least changes necessary in the DNS), (ii) *Security* (variants must minimize the risks introduced by IDNs), (iii) *Predictability* (variants should behave and function as users expect in their language and script environments), (iv) *Equivalency* (variants must be managed by the same entity and direct users to related content), (v) *Consistency* (variants should behave similarly within and across TLDs and supporting technology), (vi) *Manageability* (variants should be straightforward to visualize and administer with supporting technology), and (vii) *Ease of Use* (variants should be easy to use for new and existing Internet users).

To learn from existing deployments of IDN variants, this report summarizes and compares how nine ccTLD registries are offering variants for Arabic, Chinese, Devanagari and Latin scripts. Analysis shows that these management practices have many similarities. In particular, although the label generation rules for variants differ across scripts, they all treat the variant labels as a set for all aspects of the domain name life cycle, limit the number of activated variants to ensure a positive user experience and withhold other inactive variants for security and stability purposes (except the case of Latin IDN variants). However, although the Chinese script registries share the same tables and have cohesively defined variants for both the top- and second-level domains, the Arabic script community exhibits some differences within the Arabic language and across languages that share the Arabic script.

Active variants of TLDs may introduce challenges to various user communities. To identify these challenges, user communities are first grouped into three categories: (i) *End Users*–those who would use the variants, (ii) *Registrants, Registrars and Registries*–those who would register and manage the variants, and (iii) *Technical Community*–those who develop usability, configuration and diagnostic software that supports the variants. Based on these roles, this report identifies the challenges that may arise as the variants are activated. These challenges are summarized below and explained in more detail in the document.

- Challenges with the Use of Variants

    - User cannot find the complete set of variants for a primary label
    - Variants not intuitive
    - Variants delegated independently
    - Variants defined inconsistently
    - Variants displayed inconsistently
    - User cannot input variants
    - Unable to distinguish specific variants
    - Identifier not bound to all variants
    - Accessibility and privacy impacted
    - Variants not searchable
    - Search rankings unpredictable
    - Search optimization affected by variants
    - Variants not part of URL/URI/IRI
    - Variants cause session re-establishment

- Challenges in the Registration Management of Variants

    - Management across IDN TLDs inconsistent
    - Registration for Second-level Domains across TLDs inconsistent
    - Inconsistent association of ASCII and IDN TLDs
    - Technological support inadequate
    - Registration system not straightforward to localize
    - Registration information inconsistent
    - Trademark protection tracking complex
    - Trademark protection dispute process complex

- Challenges in the Configuration and Diagnostics of Variants

    - Software configuration not supported
    - Cannot associate variants for configuration
    - Compounded certificate management
    - DNSSEC validation inconsistent
    - Log and history searching does not match
    - Network traffic statistics incomplete
    - Caching infrastructure inefficient
    - Diagnostic and troubleshooting tools incompatible
    - Forensics significantly more complicated

To address these challenges, within the context of the seven guiding usability principles, we propose recommendations for ICANN, impacted registries and registrars, and the technical community. The summary of recommendations is given below, with details included in Section 6.

- Recommendations to ICANN

    1. ICANN must implement a well defined and conservative variant TLD allocation process.
    2. ICANN must maintain a repository for Label Generation Ruleset (LGR) for the root zone and IDN TLDs and make it available to users and programmatically processable.
    3. ICANN must develop, to the extent possible, minimal, simple and consistent LGR for the root zone.
    4. ICANN must develop, to the extent possible, a minimal, simple and consistent life cycle for the variant TLD sets (across languages and scripts).
    5. ICANN must define guidelines to evaluate the competence and readiness of the registry to manage variants, to ensure a stable and secure end user experience.
    6. ICANN should require IDN TLD registries with variants to apply the relevant (script) subset of the root zone LGR and state life cycle for variants across second-level domain labels. Deviations should be justified.
    7. ICANN must create educational materials on the use and impact of variants for different user communities.
    8. ICANN must require accredited registrar who supports IDNs with TLD and/or SLD variants to support variants across its registration platform.
    9. ICANN must develop consistent registration data requirements for variants at root and other levels.
    10. ICANN must define technical requirements and engage with standards organizations, such as the IETF, to determine how the IDN variants should be consistently implemented.

- Recommendations to a Registry that Offers IDNs for Scripts that have Variants

    1. Registry must not register any second-level variant labels unless the label registration request has met all approval requirements.
    2. Registry that supports variants must make its updated LGR available to ICANN.
    3. Registry that supports variants should apply the LGR developed for the root across lower-level domains. Deviations from the LGR should be publicly documented and justified.
    4. Registry that supports variants must implement, to the extent possible, state life cycle for the second-level variant recommended by ICANN.
    5. Registry should create educational materials on the use and impacts of variants for different user communities, such as end users, system administrators, etc.
    6. Registry that supports variants must require relevant registrars to support IDN variants across their registration platforms.

- Recommendations to a Registrar that Supports the Registration of Variants

    1. Registrar must update its practice to address requirements specific to the registration of IDN variants.
    2. Registrar should extend linguistic and technical support of IDN variants for registrants.
    3. Registrar must support IDN variants across its registration platforms.
    4. Registrar must support registry policies and associated services for collecting and managing registration data of IDN variants.
    5. Registrar that supports the registration of variants may also update any related services that are impacted by variants.

- Recommendations to the Technical Community

    1. Developers of software tools for the technical community should consider, based on user requirements, enhancing their software to support the administration and management of variants.
    2. Software intended for Internet end users—such as web browsers, email clients, and operating systems—should support variants to the extent necessary to ensure a positive user experience.
    3. To provide end users with a consistent and predictable experience with variants across software applications, developers should, to the extent possible, publicly share best practices and emerging standards in terminology and functionality.

The proliferation of variants across scripts has far-reaching implications on policy, technology, and practice. To help ensure that variants are implemented in a way that is both secure and user friendly, it is important that the stakeholders address these recommendations and continue to work together in the fast-evolving IDN variant ecosystem.

1. Background and Introduction

There has been a long-term need to deploy domain names in local languages, also called Internationalized Domain Names (IDNs), for making the Internet accessible to populations using non-ASCII scripts. The Internet community has worked together for more than a decade to make this possible. The Internet Engineering Task Force (IETF) initially published a set of protocols [9,11,13] to handle Internationalized Domain Names in Applications (IDNA2003). These protocols have been revised based on community feedback [14], resulting in the IDNA 2008 [8,12,15,16,17].

Having some years of experience with IDNs at second and lower levels, recent work focuses on deploying IDNs in the root zone (at the top-level). The ICANN Internationalized Domain Name (IDN) Fast Track process [39], an experimental program for IDN country code Top-Level Domains (ccTLDs), has added 31 IDN ccTLD labels to the root zone. Policy development process (PDP) is underway in the country code Names Supporting Organization (ccNSO) to replace the Fast Track process and allow for more IDN ccTLDs. In addition, IDN TLDs will also be part of ICANN's new Generic Top-Level Domain (new gTLD) program.

One key issue to be resolved in both the ccNSO PDP related to IDNs and the new gTLD program is the inclusion of certain IDN labels called "variants" in the DNS root zone. To address this complex technology and policy issue, ICANN first engaged six script communities that identified requirements of variants for these scripts [1,2,3,5,6,7]. It then produced an integrated report [4] collating and synthesizing these issues associated with the possible inclusion IDN variants in the DNS root zone.

Following the recommendations identified in [4] and subsequent project plan [41], this study was initiated to examine the user experience implications of active variant internationalized Top-Level Domains or variant IDN TLDs. The current report contains the following sections:

- Section 1 provides the reader with a brief description of the objectives and scope of the study as well as terminology used in the report. It also explores the divergent interests and practices around variants to highlight the challenges for finding an acceptable solution for using variants.
- Section 2 summarizes and compares, from a user experience and registry operation perspective, second-level domain (SLD) variant practices across Arabic, Chinese, Devanagari and Latin script IDN ccTLD registries.
- Section 3 proposes a set of principles to be used for determining the usability of variants TLDs, setting the foundation for the rest of the report.
- Sections 4 and Section 5 categorize the user communities and identify how these communities may be impacted by active variant TLDs.
- Finally, based on the usability principles and informed by current IDN variant deployment experiences, Section 6 proposes a set of actions for various stakeholders to ensure a secure, stable and consistent user experience for IDN variants.

1.1. Study Objectives

An IDN TLD label may have many variants. It is possible to not activate any such variant labels and only allow a primary label in such a set to be active. On the other hand, it is also possible to activate all the variant labels. A more practical approach may be to activate some and not activate other variants of such IDN TLDs. The current project considers the user experience implications when one or more variants of an IDN TLD are activated (i.e., have resource records in the DNS). It seeks to answer the following set of questions:

- What are the components of an acceptable user experience for variant TLDs?
- How will various user roles be impacted if variant TLDs are activated?
- What are the necessary rules or guidelines a TLD should operate under in order to provide an acceptable user experience for variants?
- What are the policy/contractual considerations that will make these rules effective?
- How does the impact of variant TLDs on applications have on user experience?
- What other entities have a critical role to play in addressing these issues and what educational or consultative steps could be implemented to generate support and collaboration by these parties?

1.2. Scope of Work

Variants are hard to define and categorize. According to the Integrated Issues Report [4], a variant label of an IDN is a

> "term that has been used in multiple ways, to indicate some sort of relationship between two or more labels or names. It has been used variably to refer to, for example, a particular relationship between specific characters or code points in a particular script, or a set of alternate labels where some linkage relationship is articulated, or a desired procedure whereby names are registered in multiples, or a desired functionality causing shared behavior by some set of identifiers."

For the purpose of this work, a variant refers to a whole label, not just a code point or a character, formulated using a Label Generation Ruleset (LGR) [4] for the particular zone. This is discussed further in Section 1.3.

TLDs can be ASCII labels or IDNs. For those TLDs that are IDNs, many do not have variants. The scope of current work is limited to the usability study of those IDN TLDs that have variants, as illustrated by the shaded region in Figure 1.

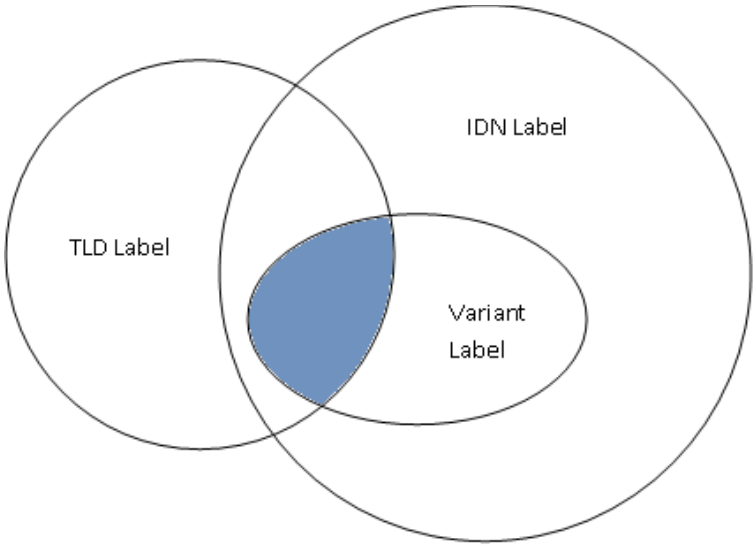

Figure 1. Scope of Study on IDN TLD Variants

Many different kinds of variants occur in different scripts (e.g. see the IDN Variant Issues Reports for Arabic, Chinese, Cyrillic, Devanagari, Greek and Latin and the Integrated Issues Report [1,2,3,5,6,7,4]). The current report considers examples from these scripts. However, the work is more generally applicable to variants across other scripts as well.

1.3. Terminology, Assumptions and Limitations

The terminology used in this document follows the conventions recommended by the community as documented and defined in the Integrated Issues Report [4], particularly in its Appendix 2: Terminology. Readers should understand the definitions for *Variant Label, Label Generation Ruleset (LGR), primary Label, Internationalized Domain Label (IDL), IDL Set, U-Label, A-Label* and other terms used in this document. Readers are also encouraged to consult RFC 6365: *Terminology Used in Internationalization in the IETF* for more general but relevant definitions. The terms *must, should*, and *may* are used in the document as defined by RFC 2119.

For the purposes of this study, we use the term *variants* to strictly mean *alternate labels* or *variant labels*. These variant IDN labels or Internationalized Domain Labels (IDLs) are also cumulatively referred to as a *variant set* or an IDL set (as defined by [4]). The specific characters or code points which cause these variants will be referred to as *variant code points* or *variant characters* but not as *variants*.

The notion of *primary label* is used to refer to one of the labels in an IDL set which is either preferred by a registrant or is pre-determined by a community to represent an IDL set. Labels are used at multiple levels: top-level domain label (TLD), second-level domain label (SLD or 2LD), third-level domain label (3LD), etc. Multiple level labels combine together to form a domain name; for example, in the domain name*www.icann.org* “icann” is SLD and “org” is TLD.

There are ongoing discussions on how variant labels will be activated in the root zone. Though much of this document may remain relevant irrespective of the method used to

activate variants, the final implementation could still have impact on this study, while possibly introducing additional issues not discussed in this document.

1.4. Balancing User Expectations with Consistent and Secure Implementations

The expectations for the use of IDNs and their variants are very different across linguistic and technical communities. Linguistic communities are primarily concerned with serving end users, and require maximally distinct IDN labels from a character repertoire for capturing a diverse linguistic expression and additionally maximal variant label sets for accessibility. Further, they consider these variants in the variant label set as related, i.e. different ways of representing the same label. On the other hand, the technical community is primarily concerned with the security and stability of the DNS, and requires the minimal number of additional labels in the root with minimal (if any) variants. The technical community considers each label as a unique A-label within the DNS and therefore distinct for any other label, where variant labels, as stipulated by a zone LGR, is an arbitrary grouping due to non-technical linguistic reasons.

The current discussions for IDN TLDs and their variants at the root now involve the "script community," which roughly represents the set of linguistic communities using a certain script. The motivation of defining this community comes from the fact that the root is shared by the global community [4], and therefore, though the language delineation cannot be represented at the root, script distinction is (arguably) possible. This implies that the same set of rules should be applicable to all TLDs supporting a script[1]. This creates a third expectation, which has a more diverse "script" expression (containing a common character repertoire of many more characters than would be in an individual language) and more numerous and possibly a much larger set of variants (as a larger script-level character repertoire causes more conflicts).

The challenge is to find the right balance between these competing linguistic, script and technical expectations.

In addition to differing expectations, there can also be very different implementations of IDNs and their variants, if they are not consistently managed. This can be sub-classified into two aspects. First, what are the various states of a variant (see Section 5 of Integrated Issus Report [4]) and how a variant is assigned to a state (e.g. automatically through a LGR or arbitrarily chosen by the end user, etc.)? Second, how will the variants which are active (if any), be implemented? Differences in how variants are assigned different states, and how the activated variants are implemented would cause confusion in both the use and technical implementation. For example, if no common mechanisms are agreed upon, the activation and resolution of variants may result in different user experiences across

[1] Single script loosely refers to TLDs defining their repertoire of characters from primarily one script (e.g. Cyrillic, Greek, Arabic; see http://www.unicode.org/Public/UNIDATA/Blocks.txt), with the understanding that there may be additional characters in it with other Unicode script properties (e.g. Common and Inherited). As per Unicode, "In text processing, the identities of all characters are normally known, but some characters may be shared across scripts or attached to any character, thus requiring special values for Common and Inherited" (Davis and Whistler (2012)).

different gTLDs. The experience across ccTLDs would also remain very distinct, as they serve different linguistic communities. Further, the implementation may also change down the tree, from TLD to SLD and further below. This could add to further challenges in their configuration and use. Consistent guidelines can give users a more predictable experience across and within the TLDs.

Irrespective of their implementation, the way variant sets are defined and managed could still vary greatly down the tree, e.g. between TLDs and SLDs because TLDs are defined at the script-level and SLDs may be defined at the language level. This will be one major source of variation in the way variants are enabled for users. This change in definition of variants across various levels could become a source of confusion and may have a significant impact on the usability of variants. If user confusion is to be minimized, this variation will need to be managed.

How the variants are enabled and expected to be used by a script community may also vary across script. For example, variants in Chinese script may be visually distinct but those in Arabic script may be visually similar. Some variants may be determined by language and some arbitrarily based on the implementation. For a better user experience, such script-level differences should at least be documented, and, whenever possible, minimized.

The user experience with variants of IDN TLDs may also be very different due to the diversity of technology platforms being used. User systems are configured in a variety of ways, including different operating systems, different keyboards, different fonts, different personalized settings, different applications, different locales, etc. Therefore, even if other sources of variations are contained, the range of system configurations – along with varying levels of support for variants – will result in different user experiences.

The final source of variation in user experience is dependent upon users themselves. The competence of users with the script and technology they are using can greatly vary the user experience. Insufficient knowledge of the script or insufficient grasp of technology being used can significantly degrade the user experience. As the IDNs are rolled out, reaching the unreached, most of the new end users will be technologically challenged.

The current work studies the different facets of expectations and variations to determine how and to what extent IDN TLD variants may be introduced to balance between these varied needs and constraints.

2. Existing SLD and TLD Practices

Currently, IDN variants are already in practice at second and lower levels in the Domain Name System (DNS).Therefore, it would be valuable to learn from how the relevant end users are adopting them. Though the TLDs are different, the state of practice at lower levels can still provide some insight in understanding user expectations. Further, in the case of Chinese script, simplified and traditional Chinese IDN ccTLDs that are managed synchronously have been implemented to provide users with an experience similar to IDN TLD variants. Learning from their experience will be helpful in the context of this study. The current section gives an overview of these practices for the ccTLDs.

2.1. IDN Variants at Second-Level

2.1.1. Existing Arabic IDN Variant Management at Second-Level by IDN ccTLDs

Arabic script is used for a variety of languages globally. There has been a lot of work on analyzing variant labels in the context of Arabic script and the languages that use this script. This has included work by the Arabic Script IDN Working Group [21], the Arab League, UN ESCWA, and other national and international efforts. RFC 5564: *Linguistic Guidelines for the Use of the Arabic Language in Internet Domains* [10] discusses in detail the source of user confusion in using Arabic script specifically for the Arabic language. A more comprehensive analysis of Arabic variants at the script level is presented in the *Arabic Script Issues Report* by the Arabic Case Study Team [3].

These guidelines have been largely adopted for the use of Arabic language IDNs by various ccTLD registries in Arabic-speaking countries. For example, dot-Emarat (امارات) IDN ccTLD for United Arab Emirates has an extensive policy to manage variants [29] summarized as follows:

- Registrant applies for a domain name in Arabic language, which is then considered *primary*
- Variants are created due to digits which are semantically same as ASCII digits and letters which are culturally confused due to local conventions
- Variant due to digits are automatically activated
- Variants due to writing conventions are blocked by default, but may be activated by the request of the registrant
- Blocked variants are not available for other registrants
- Variants can be activated at any time through the registrar
- A registrant may activate up to five variants at a time
- All variants are associated as a set to the *primary* and may not be separated

In addition, the variants are also considered a single group for registration data and a single registration data record is maintained for all the variants. When an active variant is looked up, the registration information of the *primary* label is returned. Thus, every time the status of a variant is updated, the registration data entry of the IDL set is also updated. Further, when preparing the reserved list of domain names, the registry also blocks/reserves all the variants of these domain names.

Similar policies are adopted for other Arabic-language IDN ccTLDs, with some variation. For example, Qatar IDN ccTLD considers the characters و and ؤ as variants in addition to other characters considered variants by dot-Emarat. Jordan IDN ccTLD allows up to three SLD variants to be activated.

The IDN ccTLD for Saudi Arabia offers position-level variants for each character [22]. Though the technical solution is more complex, this enables them to offer many more labels to their potential registrants. The IDN policy [25] regulates the labels. The policy also states that it "may establish rules and procedures to resolve the problem of character variants with other Arabic script based languages (e.g., Persian, Urdu)." As explained in supporting

documentation [26], this is to enable resolution of domain name variants generated from keyboards of other languages.

The ASCII ccTLD for Iran (.ir) has been offering IDNs at the third level, under the SLD ایران ("iran") with up to five variant labels for (automatic) registration in addition to the primary domain name requested by the registrant in Persian language. The variants are produced due to Arabic letters and Arabic and ASCII digits [24]. As these variants were offered under IDNA2003 that did not allow the use of Zero Width Non Joiner (ZWNJ), any labels that needed the character were registered without it. The new policy for the IDN ccTLD allows for the use of ZWNJ, as per IDNA2008. It uses ZWNJ to create distinct labels, except after a few letters which do not have distinguishable joined and non-joined forms, as given in the Arabic script report [3] in which case use of ZWNJ will create a variant label. Further, though the IDN domain registration under the ASCII ccTLD and IDN ccTLD will not be tied together, IDN ccTLD for Iran plans to offer a special sunrise period for registrants of 3LD IDN labels within ASCII ccTLD to allow them to register the corresponding labels at SLD level under IDN ccTLD. This is being planned as the IDN domain registrations under the ASCII ccTLD from Iran will eventually be phased out[2].

These TLDs are already offering variant registrations at SLD or below in the IDN ccTLD, except Saudi-NIC, which is currently only offering a single (primary) label at SLD and lower levels, and intends to offer variant resolution if and when the variants are enabled at TLD[3].

2.1.2. Existing Chinese IDN Variants at Second-Level

Chinese variants are defined as "characters with different visual forms but with the same pronunciations and with the same meanings as the corresponding official forms in the given language contexts [5]." Because of the ideographic nature of the Han script, and the evolution of the writing system, Chinese characters have many variants. The Chinese Domain Name (CDN) community has been working together for the past decade to solve this problem.

At the second-level, the variant management system for CDN registries has three key components:

- *A Language Variant Table (LVT)* that defines the variants for each character that can be permitted as Chinese domain names. The CDN ccTLDs and gTLDs all use the language table developed by the Chinese Domain Name Consortium (CDNC).
- *Registration Policy* that is specified by RFC 3743 [18] and RFC 4713 [19]. RFC 3743 defines a set of IDN registration and administration guidelines for applying restrictions to CJK scripts and the zones that use these scripts. RFC 4713 describes how ".CN" and ".TW" apply the principles of RFC 3743 to manage Simplified Chinese and Traditional Chinese domain name equivalence.

[2] Based on communication of authors with IRNIC.
[3] Based on communication of authors with Saudi-NIC.

- *Registry/Registrar Provisioning Systems* that perform functions such as reading the IDN Table, generating variant domain name, transcoding to punycode, and provisioning the zone file if synchronization is used.

In general, the Chinese Domain Name registries exhibit the following practices at the second-level. The primary label and all the variants generated form an atomic package (called an Internationalized Domain Label package – or IDL package – in RFC 3743 [18]). Once an IDL package is created, the name holders can dynamically request the domain registry to activate and deactivate some variant IDLs in the package. However, no IDLs can be inserted into or removed from the IDL package during its lifetime. When the IDL package is destroyed, due to either being unregistered or revoked, all IDLs in the package are available again to all name holders at the same time.

Because the number of variants for the Chinese domain can be extensive, the IDL package can expand and become quite large. For example the IDL package for the label 臺灣大學 (National Taiwan University, U+81FA U+7063 U+5927 U+5B78) contains 30 labels[4]. Thus, if an IDN that contains 臺灣大學 as SLD as well as TLD could have 900 variant combinations. To reduce the complexity, RFC 3743 and 4713 apply the following restrictions:

- RFC 3743 separates labels into activated vs. reserved. Only those labels would be in active usage would be activated to enhance user experience, all other labels are reserved for the applicant for security purposes. In the example above, since the commonly used labels are **臺灣大學**, **台湾大学**, and **台灣大學**, so only those two labels can be activated, all others are reserved for the applicant.
- RFC 4713 applies further restriction by only allows a label with all simplified Chinese characters, a label with all traditional Chinese characters, and one additional label indicated by the user to be activated.

Thus, **臺灣大學** IDL set would only have 2-3 active variants (and for label **臺灣大學.臺灣大學** a total of 4-9 possibilities as compared to 900 variant combinations).

TWNIC has been offering Chinese SLDs (as a solution) under .tw since 2003. However, to manage the mapping between variant labels, the registry requires the registrars to provide DNS hosting and management. As reported, they have managed 200,000 Chinese .tw SLDs so far [27]. Due to its long history of offering domain names in Chinese under the .tw ASCII TLD, TWNIC states in its policy document [28] that it synchronizes the SLDs for not only the two (simplified and traditional) Chinese TLDs but also for the ASCII TLD.

[4] We use the expression **[台, 檯, 簀, 臺, 颱][湾, 灣][大][学, 學, 斈]**to enumerate the set of the variant IDLs in this package. The expression is an OR-AND like expression [IEEE Standard 1003.2-1992], in which the n-th bracketed clause lists the character variants of the n-th character in the IDL and the concatenation of a character in each bracketed clause forms a variant IDL. Thus the total number of variants for the IDL package **臺灣大學** would be 5 × 2 × 1 × 3 = 30. An IDN label **臺灣大學.臺灣大學** would contain 30 × 30 = 900 variants.

2.1.3. Proposed Devanagari IDN Variant Management at Second-level by IDN ccTLD[5]

Devanagari script is used to write many languages including Hindi, Marathi, Konkani, Nepali, Bodo, Dogri, Maithili, Sindhi, Sanskrit and Santhali. Sanskrit and Santhali will be supported by a separate IDN ccTLDs, while all other languages listed will be supported under the same IDN ccTLD label for India. The ccTLD consists of a single character repertoire, which contains all the characters of all the languages supported by the ccTLD label. When a label is formed it first undergoes a structural well-formedness check and then its variant set is generated.

> "In case of these languages sharing the same ccTLD, having different Language character repertoire table, whole label evaluation rules but same sets of variants, there are some implications. Variants generated for some languages may not contain identifiable characters for that language. E.g. for the given variant set, ञ्य (0931, 094D, 092F) is a variant of -य (002D, 092F). Here 0931 is the character that is only valid for Marathi, Konkani and Nepali, but it would occur in the variant set for other languages sharing the same ccTLD if -य (002D, 092F) comes in the intended SLD" (CDAC, India).

The ccTLD policy further limits the activation to three variant labels.

2.1.4. Existing French IDN Variants at the second level

CIRA, the .ca registry, provides French IDN variants at the second level since January 2013. The policy is summarized as follows[6]:

- The domain name registration process remains the same – search for an available domain name and use a CIRA-certified Registrar to register it.
- Every French character variant of a registered .CA domain name automatically becomes part of an administrative bundle. For example, the bundle for cira.ca would include variants such as cirà.ca, çira.ca, cîra.ca, çïrâ.ca, and all other combination of accented French[7] and standard ASCII characters.
- Once a domain name is registered, all the variants of that domain name in the administrative bundle are reserved, and cannot be registered by anyone except the existing Registrant for the domain name.
- Each variant in an administrative bundle a registrant wish to use must be registered individually, and each registered domain name in the administrative bundle has its own life cycle.
- All domain names within an administrative bundle must be registered to the same Registrant and with the same Registrar. Therefore, a domain transfer involves all variants together.
- Each variant is a separate domain, where different NS servers may be specified and are billed independently.

[5] This section is based on unpublished information communicated by CDAC, India.
[6] See http://cira.ca/why-ca/french-ca/ for further details.
[7] The following characters are supported: é, ë, ê, è, â, à, æ, ô, œ, ù, û, ü. ç, î, ï, ÿ.

- Currently, the CIRA WHOIS service does not provide any information of the other variants of a domain name. Both the U-Label and the A-Label of the queried variant are shown in the output of the WHOIS.
- All the variants and the ASCII version are bundled together.

2.1.5. Comparison of SLD practices across Scripts

Looking at the practices between and across the SLD practices, one finds similarities as well as differences.

- Arabic, Chinese and Devanagari script registries pre-define a fixed set of variants in an IDL set. These share different aspects of operation, including registration data information.
- As this IDL set may be large, the registries set limits on how many variants may be activated. Chinese registries allow for all-simplified, all-traditional and one additional user-defined variant to be activated, limiting the total to three variants. For Arabic registries the limit on the number of active variants varies between three to six labels, arbitrarily chosen by the registrants. There is also a proposed three label limit for Devanagari IDN ccTLD. Interestingly, none of the IDN ccTLDs for the scripts restrict the activation to a single label, for promoting a better user experience. For the Canadian French variants, there is no limit on the number of variants.
- The CDN registries allow certain labels as preferred or primary, as determined by the language variant table. The registries for Arabic, Devanagari and Canadian French domain names do not make such distinction and allow registrants to choose the primary label[8].
- Though the Chinese-script registries share the same table and have cohesively defined variants across all languages using the script, the Arabic script community exhibits many differences within the Arabic language and across the languages using the Arabic script. For example, even though all Arabic-language registries use the same reference language table (RFC 5564), they implement it with slight variations. For example, Jordan has extra variants defined for the letter و. Furthermore, IRNIC and Saudi-NIC intend to implement positional variants (using ZWNJ), whereas others are implementing character-level variants (by not allowing ZWNJ). Registries serving other languages using Arabic script, e.g. Iran IDN ccTLD, use a different language table. Registry for Devanagari defines a single language variant table for multiple languages supported by the TLD, which is the superset of all the code points in these different languages. Thus, there are three different strategies, with Chinese script strategy (for a single table for all languages supported by Chinese script) being most consistent from the end user perspective.
- Generally ccTLDs do not offer IDNs under the ASCII TLD. IRNIC has been offering Arabic domain names at the third level under ایران.ir ASCII ccTLD, but not tying this

[8] Though no restrictions on primary labels are given in the policies of Arabic domain names, still some restrictions may apply in practice.

registration to the IDN ccTLD (except by a special sunrise period). Taiwan has not only been offering IDNs under ASCII TLD for a decade, but also synchronizes an IDN SLD under the IDN and ASCII TLDs. On the contrary, Canadian French variants are offered under the ASCII TLD (.ca).

- The IDN TLDs are using internal custom-built solutions to manage the registration process for IDNs. These systems differ by registry. In some cases, some of the process is still manual.
- The variants are registered to the same registrant (and unallocated variants are withheld). All variants share the same registration data. However, for the Canadian French variants, the NS data is not shared and is independent across variants.
- The registries may also impose additional constraints to manage variants properly. For example TWNIC requires the registrars to manage DNS hosting and operations for the registrant. This solves some challenges faced by Arabic SLD registrants in configuring their system. Some Arabic registries provide configuration support to their registrants in this context.

2.2. Simplified and Traditional Chinese IDN TLDs

In 2010, ICANN approved the proposed delegation of .中国/ .**中國** top-level domains to China Internet Network Information Center (CNNIC) and the proposed delegation of .台灣/ .**台湾** top-level domains to Taiwan Network Information Center (TWNIC). These pairs of TLDs are managed synchronously [37,38], in which the contents of the zones that the TLD operates are synchronized. In this section, we share some statistics and lessons learned.

Both .中国/ .**中國** and .台灣/ .**台湾** follow registration algorithms described in RFC 3743 and RFC 4713 using the master variant table produced by the Chinese Domain Name Consortium (CDNC). As there is currently no standard EPP extension to handle variants, they use proprietary extensions and supply client software to their registrars who wish to generate variants themselves [23]. Table 1 below lists registration statistics for these TLDs.

Table 1: Registration Statistics for Synchronized IDN ccTLDs

	Number of Registrations (provided by registry)	% of Domains with variant forms (provided by registry)
.**中国**/.**中國** (dot China)	320,000	77%
.**台湾**/.**台灣** (dot Taiwan)	43,000	93%

In CNNIC, the delegated variant pairs (SC and TC) resolve to the same nameserver. They employ a practice called “parallel provisioning,” in which the SC and TC label is delegated separately, but the contents of the delegated zones are maintained together and from the

same backend database. Updates to one also update the other. In TWNIC, the delegated variant pairs (SC and TC) also resolve to the same name server. DNAME is used. More information can be found in TWNIC's policy document [27].

To better understand traffic patterns of IDN ccTLDs and IDN synchronized ccTLDs, a measurement exercise was carried out by ICANN. The exercise aimed to measure the relative incidence at L-Root of queries for names under two-character ASCII ccTLDs and for names under their corresponding IDN TLDs. The measurement was conducted from 2012-07-19 1410 UTC to 2012-09-21 1458 UTC and data was collected from all available pcap archives on all active (218) L-Root nodes.

The ratio of DNS queries seen for the two IDN synchronized TLDs are presented in Table 2 below. In particular for dot China, the percentage is calculated as the number of queries to Traditional Chinese TLD .中國/ total number of queries to IDN ccTLD .中国 and .中國. For dot Taiwan, the percentage is calculated as the number of queries to Simplified Chinese TLD .台湾 / total number of queries to IDN ccTLD .台灣 and .台湾.

Table 2: Query Statistics for synchronized IDN ccTLDs

(measured by ICANN on L-root servers from 2012-07-19 to 2012-09-21)

	% of DNS Queries for domains in synchronized TLDs
.中国/.中國 (dot China)	12.2%
.台湾/.台灣 (dot Taiwan)	18.4%

A few lessons can be drawn from synchronized IDN ccTLDs:

- The synchronized TLD query statistics show that Chinese IDN variants are being actively used. Given the fact that Chinese IDN ccTLDs tailor specifically to a linguistic community (e.g. Simplified Chinese community) and that IDN gTLDs tailor to a much more global audience, it is likely that variants for IDN gTLDs would receive more queries.
- RFC 4713 reduces the variants for IDN.ID, thus making this a much simpler problem to manage for the registries.
- Variants for TLDs and SLDs are generated and managed in a cohesive, consistent, and predictable manner from the perspective of the end user.

3. Usability Principles for IDN Variants

As is evident from the practice for the IDN SLDs and TLDs, variants are needed to address the end user expectations. However, even though they are needed, the registries follow a conservative approach in making them available. Based on the practice and other

considerations (like security and stability of the DNS), the following principles should guide how IDN variants and primary labels are allocated, activated, and managed.

3.1. Principle 1: Minimality

Implementing variants must introduce least number of TLD labels in the DNS. Given the inherent complexity of supporting and using variant labels, a conservative approach to adding variants is advisable. Where the case is not convincing, the variant should not be activated. Even where need is evident, a variant may still not be activated if it poses a security challenge (see Security Principle). Appropriate criteria are needed to evaluate variants for activation.

For example, an Arabic IDN TLD may have more than a dozen variant labels. Activating all of these variants may not be in the best interest of the registry, registrants, or users. Active variants may be limited to those which are likely to be used, such as variants that can be typed either by a Farsi keyboard or an Arabic keyboard and do not require toggling between the two keyboards to input a single label.

3.2. Principle 2: Security

Variants must minimize the risks introduced by IDNs. Variant labels are allocated or withheld to minimize security risks due to independent delegation of labels considered similar by a community based on visual and semantic reasons. However, the activation of variant labels should not pose additional security risks.

For example, the Arabic script IDN ccTLD strings for Saudi Arabia السعودية (A-label xn--mgberp4a5d4ar) and السعودية (xn--mgberp4a5d4a87g) are visually identical and could create a significant security risk if delegated independently to different entities. Thus, they should be allocated to the same entity, with primary activated (and other at least withheld, if not allocated).

3.3. Principle 3: Equivalency

Variants must be managed by the same entity and direct users to related content. A variant label should have an equivalent relationship with the primary label. In other words, users expect variant labels to resolve to content that is the same or similar to the content linked with the primary label. This would be possible if the variants are managed by the same entity. As discussed, management by different entities will also pose security risks.

For example, a variant TLD label could take web users to the same web site as the primary label, or take users to content that has been localized based on the specific variant being used (e.g., a Traditional Chinese variant may direct users to a Traditional Chinese site whereas a Simplified Chinese variant directs users to a Simplified Chinese site). Furthermore, users would expect that this similar content be managed by the same entity, with similar expectations regarding security.

3.4. Principle 4: Predictability

Variants should be defined as users expect in their language and script environments. Based on experience, users have expectations for how their languages and scripts are supported by technology. The behavior of variant labels should be consistent with these general linguistic expectations.

For example, Chinese users may expect a domain in Simplified Chinese characters to be equivalent with a domain in Traditional Chinese characters, even though they are visually distinct, while Cyrillic users may not share the same expectations between visually distinct characters. Devanagari script users expect the strings to be structurally well-formed, i.e. two combining vowels cannot occur together.

3.5. Principle 5: Manageability

Variants should be straightforward to visualize and administer with supporting technology. This principle applies to those who are required to register, administer, and manage primary and variant labels. The tools and processes available to these users should support all or any subset of (active or non-active) variant labels with equal ease. The relationship between primary and variant labels should be as transparent as necessary for the intended user group and scenario.

Registrants should have a clear understanding of the variants related to a registered domain. If the registry sets a limit on how many variant labels can resolve to a primary label, registrants should have the ability to understand and manage which labels will be active. Software tools and services should be made available to enable the administration of IDL set for various configuration, management and diagnostic functions.

3.6. Principle 6: Consistency

Variants should be defined similarly within and supported equally across TLDs and supporting technology. Users view domain names holistically (not as individual labels but as a complete unit), and thus expect ASCII TLDs to behave consistently across labels and geographies. Users would expect similar consistency in IDN TLDs and variants. Therefore, they would expect variant delegation rules for an IDN ccTLD to be similar to the IDN gTLD delegation rules for a given script. They would also expect the second-level variants to be consistent with the variants at the top-level. At a more general level, users and application developers will expect a consistent framework for handling IDN variants not just within languages but also within and across scripts.

For example, users will expect that if two strings are variants at TLD level, they are also variants at SLD level, and if these strings are activated at TLD level, they can also be activated at SLD level (and not blocked). Further, if two labels are variants in a script and are allowed for activation in one registry supporting the script, user will also expect them to be variants and allowed for activation by another registry supporting the script.

3.7. Principle 7: Ease-of-Use

Variants should be easy to understand and use for new and existing Internet users. Appropriate interface mechanisms are needed to view and use variants by different users. Active variant labels should not require additional configuration or software to function as well as primary labels. All active variant labels should only use characters that are well supported by technology, across existing software and operating systems.

For example, software interfaces should be able to distinguish visually same variants (as in Arabic script), and should be able to relate visually distinct variants (as in Chinese script), in such a way that users find it intuitive and easy to use.

4. User Roles

Different users interact with systems and use domain names in different contexts. Though there are multiple ways the users can be identified, for this study they are grouped into three categories: (i) *End Users*–those who use the variants, (ii) *Registrants, Registrars and Registries*–those who manage registration of the variants, and (iii) the *Technical Community*–those who deal with usability, configuration and diagnostics of the variants. Collectively, they are termed as *Users.*

4.1. End Users

End users use domain names for a variety of commonly understood functions. These include web browsing, emailing, desktop publishing, file transferring, etc. They access these functions using a variety of devices, user input/output methods software system configurations, software applications, and networks. This variety makes this user role very challenging to assess.

End users may or may not be familiar with the script being used. Even the users who are not familiar with the script may use the IDNs or their variants, e.g. when responding to an email from somebody using a domain name in a different script. Not all end users may be familiar or comfortable with the use of technology or commonly accepted usability practices, particularly those using the IDNs and variants, as they may be new to the Internet and DNS.

4.2. Users involved in the Registration Process

This group includes users involved in registering and managing the domain name life cycle. They include at least the following types of users:

Registrants

Registrants interact through the registrar interface to register, update, renew or delete a domain name. In addition, the registrants also update domain name registration data and may check for confusability of a domain name (e.g. for preventive registration to protect a trademark, etc.). Registrants may also use indirect mechanisms through proxy

or privacy services (identified in the Registrar Accreditation Agreement) to perform these functions. Registrants may or may not be experienced in registration of ASCII or IDN domains, let alone variants.

Registrars

Registrars provide services to a variety of clients, including end users, registrants and registries. Services include registering domain names, billing for registrations, handling domain name disputes, and maintaining registrant information. Registrars also escrow registrant data and interface with ICANN for compliance functions. Within the registrars, there may be different roles, including those involved in policy, marketing and sales and those involved in enabling the back-end systems. As the latter are the same as the technical community, the registrar role primarily focuses on the former set of users, for the purpose of this study. Registrars may have varied degree of expertise with IDNs and variants, and may or may not have personnel and/or tools to manage IDNs and their variants.

Registries

Registries interact with many different users providing a variety of services. They provide an EPP interface to registrars, host the DNS Server and Registration data server, and conduct performance and security analytics. The registries interface with ICANN for payments and performance. The introduction of variants at the TLD may impact one of more of the functions they perform. Registries include ccTLD, gTLD, IDN ccTLD and IDN gTLD. For the purposes of this study, only registries dealing with IDNs are relevant. The gTLDs and ccTLDs are different in multiple ways. For example, gTLDs are likely to offer script-level generic variants, while ccTLDs are likely to be more language specific (which would result in larger SLD to TLD level differences in what labels can be variants). Further, policies on how to handle IDNs and variants are likely to be more similar across gTLDs, if managed through contractual arrangements by ICANN, while ccTLDs develop independent policies. In terms of operations, IDN registries may have staff performing various roles, including those working on policy, sales and marketing, registration review, etc., and those working on system development and testing, operation maintenance, technical support (for registrars/hosting companies), etc. Again, as the latter are the same as the technical community, for the purpose of this study, the registry user primarily focuses on the former set of roles.

4.3. Technical Community

The technical community includes technology professionals engaged in developing, configuring and maintaining systems that handle variants. Their work may involve domain names across multiple scripts. Technical community members may be involved in providing front-end services, which require interfacing with end users or registrants, or back-end services. Those providing front-end services may have to troubleshoot domain names as U-labels and would need to be more aware of relevant scripts, whereas those providing back-end services are more likely to use A-labels.

The technical community comprises of many roles, all of which are critical in the functioning of the DNS and will continue to play a central role in the deployment of IDN variants. These include, at a minimum, system administrators, network managers, security managers, and application developers.

In many organizations, the boundary of system, network and security administrators may not be well defined. The following roles are defined for this study.

System Administrators

For the purpose of this study, a system administrator manages at least the following: configuring and provisioning computers, operating systems, and monitoring services; performance management; management of user accounts and identifiers; and management of certificates.

Network Managers

For the purpose of this study, a network manager accomplishes the following: configuring and provisioning routers, switches and network applications; monitoring network resources; and managing performance.

Security Managers

For the purpose of this study, a security administrator accomplishes the following: configuring and provisioning firewalls, VPNs and security policies on various network resources and servers, and monitoring security and logs. A law enforcement agency performs similar tasks albeit with a broader scope.

Application Developers

For the purpose of this study, we assign a broad meaning to application developers to include developers of operating systems, libraries, desktop applications, mobile applications, web applications, frameworks, etc. in any programming language. Application developers need to make applications aware of variants, where relevant, for a variety of users.

5. Challenges Related to Active Variant TLDs

This section presents challenges that have been identified, and is not intended to be a complete list. The challenges have been grouped into three broad categories, focusing on (i) *use*, (ii) *registration management,* and (iii) *configuration and diagnostics* of variants, in line with the categorization of users in the previous section.

Where possible, the challenges have been kept generic, applicable across various scenarios and implementations of variants. As the list of challenges is presented, each challenge is first summarized. After the summary, a more detailed discussion is given for each challenge, followed by one or two illustrative examples. These examples are presented to help understand the challenge, but should not be interpreted as restricting its scope.

5.1. Challenges with the Use of Variants

The challenges presented in this section are primarily concerned with the end user. Some end users may be able to handle variants more proficiently. However, the current challenges are listed in the context of novice users, though familiar with the script. These challenges are equally relevant for application developers who try to make the interface friendly for end users. Variants of IDNs are a new concept and will surely challenge application developers to find mechanisms to make them easily and securely available to end users. Many of the challenges are generic enough to be applicable across many other users who will be interacting with variants, including registrants, sales and marketing staff with registrars and registries and even more experienced users. For example, the first challenge, related to finding the complete set of variants, universally impacts all users.

The challenges can be logically sub-grouped into those related to the interface (input and output) and those related to their processing (web search, etc.).

5.1.1. Different Users cannot find the complete set of variants for a primary label

There is no single place from which different kinds of users can determine the complete set of variants for a domain name, including TLDs.

While most end users may have little or no interest in understanding the full set of variants for a given primary labels, registration managers and technical community will need easy access to these sets. This is a challenging requirement to meet, as the variants may change depending on registry, level of domain label, and over time. Some users may also be interested in the history of the variants, in addition to their current status.

For example, users may need to view the list of variants to understand how variants may be generated in their script; registrants may need to know if and why two labels are equivalent, and view the list of variants to determine the possible registrations they may undertake. Software developers may need this information to enhance the capacity of the search engines. Trademark protection agents may need to have access to all possible variants for investigating possible dispute cases. Registrars may need to know this information for effective sales. Network configuration and security personnel may need this information for configuration, diagnostics, and forensic analysis.

5.1.2. Variants not intuitive

Though end users generally interpret the use of labels within the context of a language, current discussions suggest that variants for TLDs will be defined in terms of scripts. This may impact user expectations.

Two code points which are considered distinct in a language may be considered variants for TLDs due to restrictions from other languages using the same script. For TLDs, and for many SLDs, to accommodate a more global audience, registries may base variants on common denominators of the script.

For example, in Arabic script U+06A9 and U+06AA are considered distinct in the Sindhi language, but may be considered as variants as these are stylistically different characters in

other languages (e.g. Urdu and Farsi) (Arabic Case Study Team, 2011). This would mean that for Sindhi speakers, two distinct labels may be considered variants, which is not intuitive for users.

5.1.3. Variants delegated independently

User considers two labels as variants of each other due to a certain linguistic context, but the labels are delegated as independent IDN TLDs.

Users expect the independently delegated TLDs to be unique. However, there is a possibility that two independently delegated TLDs are considered variants of each other by users of a linguistic community. This situation may arise if a linguistic community is not considered in the formation of a LGR for their script or the current process independently delegates two IDN labels as ccTLDs and/or gTLDs which are later determined to be variants by the LGR development process.

5.1.4. Variants defined inconsistently

End users may find variants to be inconsistent in how they are defined within a script across both TLDs and SLDs.

Users may assume that variants in a TLD remain the same when viewing an SLD, which may not be the case, leading to potential security issues. These differences could also arise in implementations of gTLDs and ccTLDs.

For example, for a Sindhi-language based SLD, the use of U+06A9 and U+06AA may result in two different labels, but as they are considered variants at the TLD, end users may also consider them as variants at the second-level.

5.1.5. Variants displayed inconsistently

Variant TLD labels consist of different Unicode code points. These code points may not all be displayed properly by the technology at hand.

This inconsistency of support may be caused by many different factors, such as different software configurations (including legacy systems). This includes different operating systems, locale settings, fonts, rendering engines, and applications, to view the domain names. Further, some of the recently encoded Unicode code points may not be supported by technology. Limitations in processing and storage capacity (e.g. on mobile platforms) may further aggravate the challenges.

For example, an Internet café in Iran (Farsi locale) may give a very different view of variants compared with an Internet café in Dubai (Arabic locale) because the computers may be configured with different operating systems, fonts and keyboards. So a user may not be able to properly view a URL embedded within a website. Also, mobile phones may not render a variant as it is rendered on a computer, due to difference in operating systems and/or fonts.

5.1.6. Variants cannot be input by the user

As variant TLD labels may consist of different Unicode code points, it may not be possible to input a variant due to limitations/configurations of the technology at hand.

Due to variation in configuration of the user system, especially in the input method and/or keyboard available, the user may not be able to input all of the variants. Further, it may not be possible to input recently encoded Unicode code points, as they may not be visible and/or available through the input methods. Constraints on mobile technology may also restrict input methods due to processing and space limitations.

For example, a user who has an input method set for Simplified Chinese may not be able to input Traditional Chinese label, and vice versa. So a user may not be able to access relevant content. This may be a more significant issue while using domain names in mobile phones.

5.1.7. Unable to distinguish specific variants

There may be instances where users need to identify a specific variant, such as to log into a system. However, in many cases, the variants may be visually very similar or the same (when displayed as U-label) and A-labels may be mnemonically intractable. Thus, an end user will not be able to determine the specific variant needed in a specific case and will have problems in using such systems.

A user may need to identify a specific (e.g. primary) variant of a variant set. In many cases this may not be feasible because variants may be visually similar or exactly the same when displayed as U-labels. As the A-labels are not good mnemonics, the user will not be able to distinguish the variants from each other.

For example, a user may input an email address for signing in to an e-commerce website. The user would need to remember the specific variant to login.

5.1.8. Identifier not bound to all variants

Many websites ask users to identify themselves through their email addresses or another identifier that includes the domain name. If variants are introduced, users may be able to input different variants of the domain name (knowingly and, in most cases, unknowingly as these variants maybe visually identical). Thus, users may experience systems that do not work as expected.

For example, a person traveling in a different locale (and using a different keyboard) may not be able to sign in to an online email service that uses the complete email address (including the domain name) as the username.

5.1.9. Accessibility and privacy impacted

Many applications manage and log domain names to facilitate privacy and usability. These applications may not perform as expected with variants.

For example, the auto-complete functionality in web browsers uses the history of the user to facilitate typing a domain name. History management interfaces allow the users to delete

certain domain names they have accessed for privacy reasons. These applications may not be able to effectively collate the variants.

If a user deletes certain domains from history, the variants of the domain name accessed by the user (which may have been accessed without explicit user knowledge, e.g. by clicking on a visually same variant) may not be deleted. This will negatively impact the privacy of the user.

5.1.10. Variants not searchable

Search techniques may not consider variant domain labels as related, and thus do not find relevant web pages against a user query.

As variants introduce a new concept, they are not considered equivalent for search at this time. The situation is further complicated as the variant set for a domain name will require using multiple variant definitions at various levels concurrently. Thus it will be very difficult for applications to search variant domain names against a user query.

For example, if a user searches for variant3LD.variant2LD.variantTLD, the user could be searching for a particular variant combination (based on the system configuration of the user). However, the search engine is not able to divert the search towards the website the user is looking for as it does not know all the possible variants or is not configured to point to all of them.

5.1.11. Search rankings unpredictable

Users may not find the content they are looking for because search techniques may not rank variant domain labels equally.

Even if search engines are able to link two different web pages as being indicated by variants of the same TLD (with a variant combination of lower levels as well), they may be ranked very differently instead of being ranked "together" as users may expect.

For example, if a user searches for variant3LD.variant2LD.variantTLD, this user could be searching for a particular variant combination. However, even though the search engine locates the variants, the search results for pages indicated by the variants may be ranked very differently.

5.1.12. Search optimization affected by variants

Pointing multiple variants towards the same website may lower the website's ranking due to search engine algorithms.

As most web frameworks do not support variants or multiple domains, software developers may use the HTTP "Move permanently" response to redirect multiple domains to the same web page. This may significantly increase the number of redirects, lowering the search engine ranking.

5.1.13. Variants not part of URL/URI/IRI

The concept of variants is only being implemented as part of domain name labels, but users may expect similar equivalence in resource identifiers. This will create inconsistent behavior for users.

Labels and domain names are re-used in many places in identifiers, not just in the domain name part. Users who use a label at multiple places, may expect similar behavior. This is a significant requirement creep (beyond variants in domain names) as it has significant implications on not just domains but on filing systems, etc.

For example, users may expect that if www.SLD.TLDvariant1 and www.SLD.TLDvariant2 work, then email addresses TLDvariant1@SLD.TLDvariant1 and TLDvariant2@SLD.TLDvariant2 should also work. However, TLDvariant1 and TLDvariant2 before the '@' sign in the email may not be equivalent. Similarly, the following is not considered equivalent in the URL http://www.SLD.TLDvariant1/prg?t=TLDvariant2. This will confuse the user who is not able to distinguish the two uses of the labels.

5.1.14. Variants cause session re-establishment

Web sessions may need to be re-established if the same website is accessed through a different variant. This may confuse end users, especially when the variants are the same or similar looking.

Mechanisms for web session establishment, either authenticated or unauthenticated, e.g. through cookies, use domain names as the entry key. Since a variant is a different domain name, the session may be considered different by the mechanism and user will need to re-establish the session.

Users may consider variants as the same, especially if the variants look the same and/or they point to the same website. Thus, if users log into a website and save the login information, they would expect to be logged in automatically the next time they access it. If the website is accessed by clicking on a link (which is a variant), re-authentication will be needed and an additional (separate) copy of the information will be saved. Further data entered on the website may not be available or may be lost across these (redundant) sessions.

5.2. Challenges in the Registration and Management of Variants

Registration and management of variants are critical to their deployment, and rely on many user roles, though the registrant holds a central position in the process[9]. Sales and marketing representatives from registrars and registries are involved in serving registrants. Trademark protection professionals are active in protecting the interests of the trademark holders, including registrants. Privacy and proxy service providers address the privacy issues of the registrants, while providing necessary support on their behalf to the registries and registrars.

[9] Privacy and proxy services may be considered to be acting on behalf of the registrant.

Registration management includes determining the right set of variants, presenting the choices to potential registrants, capturing the registrant preferences and information, making registrant information available for registered variants, activating these variants and then maintaining them over time (managing renewals and deletions). In some cases, the policy makers and community need to be involved, to determine the sets of variants, how to manage information for these variants, and how to resolve registration disputes arising from them.

The following are challenges that have been identified in these contexts.

5.2.1. Management across IDN TLDs inconsistent

Lack of consistent guidelines will make the registration process inconsistent across TLDs. The registration process of variants for TLDs and SLDs may require multiple aspects of the registration to be defined. These aspects and how they will be managed are not clearly identified and agreed upon by the community. Issues include at least the following: The relationship between SLDs and TLDs; primary vs. other variant labels; activation states and procedure to set and change these states; limits on activation of variants; and pricing of activating and changing activated variants. Without clear guidelines, the registration will be very different across TLDs confusing the registrants.

For example, it is not clear if all variants could be activated. Further, how would TLDs decide which TLD variant should be primary, in case many can be activated? If a primary TLD variant is determined, can it be changed to non-primary at a later time? What will be the process? How would registrars handle arbitrary registry policies in this context, without confusing the registrants?

5.2.2. Registration for SLDs across TLDs inconsistent

Differences in definition, arrangement, and activation of variants of a SLD across TLDs for a script (gTLDs and ccTLDs) may make the registration process difficult, especially if a registrant is managing similar domains across multiple TLDs.

The variant sets may be different for SLDs across TLDs. Further SLD variants may have different possible activation states for across TLDs. Additional complexity is involved as the SLD and the TLD labels may change their status (e.g. activation states and the choice of primary label at each level). Thus, the registration process may become much more complex and registry/registrar dependent.

For example, it may not be possible to have the same SLD variants to be activated in the same way across the various TLDs of the same script due to differing variant handling policies (activation status over time, primary label definition over time, pricing, etc.). Registrants may find the process for registration, renewal, de-activation and deletion of variant sets so confusing that they may be discouraged from using IDNs.

5.2.3. Inconsistent association of ASCII and IDN TLDs

Some TLDs may associate ASCII and IDN TLDs for registrations, while for others this may not be possible or practical. This inconsistency may confuse users.

ASCII TLDs and IDN TLDs are not variants of one another. However, as a service to registrants and users, registries may package them together in the same way that variants are packaged. This may lead registrants to expect similar levels of support across other TLDs, and may result in additional confusion in the registration process.

For example, TWNIC is already offering registrants the ability to package IDNSLDs across ASCII and IDNTLDs. Similar mechanisms may also be done by gTLDs owned by same entities for more effective user facilitation and sales. However, in other cases, it may not be possible as the ASCII and IDNTLDs may be owned and operated by different entities.

5.2.4. Software support inadequate

Technology to support the registration currently does not handle variants, which may make variant registration challenging and difficult.

Registrants need to view all the variants for the domain name being registered, understand the differences between these variants, decide how to define which of the variants should be primary (if any), and which subset may be activated. Registrars normally offer automated services, through which domain names available through various registries are made available for sales, renewals and deletions. Currently, no well-defined automated services are available to support variants of the domain name, especially in the context of different variants for different registries for different levels. Further, there is no mechanism to update variant information over time, even if it is made available at one time for new and existing domain names. In addition, registrars also have back-end interfaces for communicating with registries, e.g. through EPP based technology.

For example, a user may not be able to access all variant TLDs during registration because the registrar interface is not enabled for variants. Alternatively, if a registrant has already registered an IDN, and variants are introduced afterwards, there is no mechanism defined to contact the user and update the variant activation. This issue is also relevant at other levels of the DNS tree.

5.2.5. Registration system not straightforward to localize

As variants are defined at the script level, it is not possible to predict the language of the registrant for the localization of registration interface.

New registrants and users of IDNs and variants may be monolingual and may only understand a single non-Latin script. Registrars that are providing interfaces may not be able to predict the language of the registrant for a given TLD and thus may not be able to localize the interface. This may make it difficult for registrants to register IDNs and its variants, particularly given the inherent complexity of variants.

For example, a potential Arabic script IDN TLD registrant may speak Urdu, Pashto, Arabic, Farsi, Sindhi, or any other of the many languages using Arabic script. A gTLD may cater to many of these communities. In some cases, even a ccTLD may cater to multiple languages, e.g. the Arabic script IDN ccTLD for Pakistan and India. A registry or registrar may not know which language to offer within the registration interface, or even offer support such a language.

5.2.6. Registration information inconsistent

Variants are not part of the registration data and services infrastructure at this time. Therefore, registrants will not be able to define and access data related to variants consistently across registries, creating usability and security challenges.

The variants assigned to the registrant should point to the information entered at the time of registration. However, it is not clearly defined how variants will point to this information and how this registration information will store the variants (and their properties). Arbitrary solutions maybe developed by registries to collect and distribute this information related to variants. Some registries may provide look up functionality for all variants whereas others may provide look up functionality for only active variants, and still others may provide look up functionality only through the primary label. This would confuse both the registrant in producing the data, and users in accessing and interpreting the data.

For example, users may consider the variants that do not show up in the registration data query to be unrelated, while in other registries all variants may be fully displayed.

5.2.7. Trademark protection tracking complex

Due to potentially large set of variants and their confusingly similar strings, protecting trademarks may become much more complex and costly.

A domain which may be seemingly unrelated may have a variant which can be exactly the same or similar to an existing but different trademark. This may be difficult to anticipate and track, especially due to variety of implementations across TLDs, due to complexity of variant management processes for both SLDs and TLDs, and due to lack of support of registration data and services for variants. With many more TLDs and potentially many more variants and strings similar to these variants, tracking and protecting trademarks will be an uphill task.

For example, if a trademark is similar to a string of an inactive variant of a TLD, it may be hard to identify and monitor (especially if inactive variants are not available through registration data services).

5.2.8. Trademark protection dispute process complex

Variants may introduce new scenarios into the dispute process, which are currently not addressed by the dispute policy. Determining such cases may be more complex.

Current disputes are limited to single strings and confusions arising from them. When more than one label is included to form a variant set and registered to a single registration request, it may interfere with other (seemingly) unrelated strings and lead to more potential disputes. However, as many variants will be automatically generated, and not requested by the registrants, it will be difficult to determine the intent behind the registrations. Further, this will cause more critical cases where variants are introduced at a later stage after the primary labels have already been delegated. A comprehensive policy on how to deal with such cases is still not defined, especially in cases where they have legal implications on the Label Generation Ruleset.

For example, two characters considered distinct in a language may need to be collapsed as they are considered equivalent in another language using the same script (e.g. U+06A9 and U+06AA; see 6.1.2 for more details). However, this may invoke a trademark issue in the context of the community speaking the first language.

5.3. Challenges in the Configuration and Diagnostics of Variants

Configuration is critical for making variants operational and diagnostics are necessary for their continued health. This is primarily undertaken by system support professionals, network managers, security managers and other users, who are expected to be competent in their understanding of how the DNS works.

5.3.1. Software configuration not supported

Tools are currently not available for configuring variants for the DNS, web servers, email servers, etc. System configuration personnel may find it difficult to manage many variant domains.

Variants are expected to support equivalent behavior for users. However, system configuration personnel may find that software may not permit entering more than one domain name or, in cases in which software allows multiple entries, it processes a variant as just another unrelated domain name. Thus, it would be hard to configure the systems to simulate the desired behavior or to make sure each variant entry is appropriately configured.

For example, a webmaster configures the web platform for all activated variants (tens of variants are possible). But because each variant may require a separate, unrelated configuration, the process is both tedious and error-prone.

5.3.2. Cannot associate variants for configuration

System configuration software may display domain names only in ASCII format. This means that IDN labels will be displayed at A-labels (not as U-labels), making it challenging for system configuration personnel to manage them.

Many of the operating and configuration systems may require domain labels to be written in ASCII (A-label), as they may not fully support non-ASCII file formats and filenames. This means that even if variants can be supported, they may still need to be configured as A-labels. However, A-labels, due to the nature of the algorithm that produces them, generate intractable ASCII strings that cannot serve as mnemonics.

For example, the IDN ccTLD label for Pakistan has two variants, پاکستان (with U+0643 as the third character) and پاکستان (with U+06A9 as the third character); the latter string is the primary label approved for delegation. Both U-labels generate exactly the same mnemonic visually. Punycode[10]for the former is “xn--mgbai9a5eva00b” compared to “xn--

[10]Punycode generated from https://www.centralnic.com/portfolio/domains/idn/converter.

mgbai9azgqp6j" for the latter string. So it is difficult to use either representation conveniently to configure systems.

5.3.3. Compounded certificate management

Certificates for authentication are bound with domain names. From a security perspective, a variant is a different domain name and would require a new certificate, if activated. As there may be many domain name variants, certificate management will be challenging and more costly.

For each variant combination of a domain name (which would include variants of TLDs and of labels at lower levels), a separate certificate would need to be purchased and managed. Different certificates may come from different vendors, at different dates and configured on different systems. Thus, for tens or hundreds of variants (which are possible as number of variants at each level have a multiplicative effect on total variants), certificate management may become a difficult proposition. This will have a negative impact on the operational security of the relevant portions of the DNS.

For example, a website administrator would need a different TLS/SSL certificate for each variant. These certificates may have different pricing and renewal dates. This would need to be tracked for all activated variants. See challenge 6.1.13 "Variants cause session re-establishment" which covers the impact of certificates and session management.

5.3.4. DNSSEC validation inconsistent

Depending on the variant being used, users may not be consistently validated through DNSSEC, even though users may be expecting such validation.

Each variant needs to be independently signed for DNSSEC, and may have different expiration dates or different DNS records characteristics. Thus, all variants may not have a consistent DNSSEC validation status, causing challenges in secure access of relevant systems.

For example, a user may access one variant which is DNSSEC validated. Then the user accesses another variant that is not DNSSEC validated, resulting in an inconsistent and potentially less secure user experience.

DNSSEC is required and used in additional and upper-layer protocols such as the DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA. Therefore, the issues listed above are then inherited in those higher protocols, therefore further complicating the issue.

5.3.5. Log and history searching does not match

Logs of domain names accessed by a system are stored by software for multiple purposes. The applications using these logs may need to associate the variants, but may not be able or configured to do so, negatively impacting usability.

Many applications maintain logs of domain names for facilitating use, management, diagnostics and security of the system. System configuration personnel may use the

domain names to filter certain website content. Network managers may use the domain names to monitor traffic and law enforcement agencies and other security personnel may use them to follow up or prevent illegal activities online. In such cases, they would need to have tools and technology that can collate the variants into a single set. However, current systems do not allow these users to collapse the variants into a single set.

For example, law enforcement organization may need to track suspicious activity against a domain name. They would need to manually manage all the (tens or hundreds of) active domain name variants (SLD.TLD) manually and to map the behavior. Furthermore, this issue gets worse when the tactic is used at many levels of the tree at the same time. This may be more time consuming (and may not be easily possible) to accomplish.

5.3.6. Network traffic statistics incomplete

Network administrators look at network flows to manage their network, such as bandwidth allocation. Because variants may not be collated by applications providing these statistics, variants may make it difficult to manage the networks efficiently.

Based on access patterns of different domain names, network administrators may engineer how bandwidth is distributed. However, for effective management, statistics for variants that access the same web content may need to be combined to give realistic numbers. Network managers may not be able to detect patterns if the traffic is distributed across the tens (or hundreds) of variant domains (e.g. variants of 3LD.2LD.TLD). Even when the systems may be developed to collate such traffic, it may still be hard to find out which two domain names are variants as the variant generation rulesets are distributed across multiple entities.

For example, netflows for variant1 will be separate from variant2. However, the network admin may need to see both aggregated into a single netflow.

5.3.7. Caching infrastructure inefficient

Web pages are cached to reduce network traffic and provide better response time to end users against their requests. As caching matches domain names, it may not work effectively with variants.

Variants will have adverse impact on caching. First, as caching management software may consider variants to be unique, it will cache one copy of the same page for each variant. Storing multiple copies of the same web page would mean reducing the number of unique web pages to be cached given the fixed space available. In addition, if a variant which is not cached is accessed, the cached webpage will not be used and a fresh copy will be retrieved, increasing the network traffic and the delay. As domain names may have many variants, this may have adverse impact on the response time.

5.3.8. Diagnostic and troubleshooting tools incompatible

To troubleshoot network issues, low-level command-line tools are used. These tools generally use A-labels for IDNs and will not support variants. In addition, they typically associate a single domain name against an IP address. These constraints may make network management more complex.

While doing investigation or troubleshooting, various low-level and command-line tools such as ping, dig, wireshark, curl, etc. are used. These tools use an ASCII interface and cannot process variants. Therefore, troubleshooting and investigation is more difficult and error-prone and takes more time. Further, many troubleshooting tools are based on the concept that a domain name does not have variants. IP addresses in log records or in traffic interception are translated to names by DNS PTR records.DNS PTR records typically point to a single domain name and do not relate the variants. PTR records are used for troubleshooting, logging and access-control lists.

For example, an investigator wants to test availability of all the websites related to domain name. A tailored ping command for this community would test all variants to make the investigation more comprehensive and complete. As another example, a network administrator sees an IP address in logs. The DNS PTR translation points to variant1; all other variants are unknown and can't be found.

5.3.9. Forensics significantly more complicated

Variant domains should be considered related and equivalent for forensics and should be reported accordingly to aid in investigations.

Forensics software used by security professionals and law enforcement uses various heuristics and databases to detect issues and to provide reports for investigation. These heuristics may consider each variant unique and the analysis would need to be manually configured for variants. The configuration will be further obfuscated by the fact that URLs consist of multiple levels, and each level has different variant rules across different registries, and these rules are distributed across multiple entities located in multiple geographical locations. Further, the rules may change over time and active variant status may also change over time and the history of such changes may not be maintained. The varied, distributed and dynamic nature of variant definition and management makes forensic analysis very complicated.

For example, if an intruder accesses domain name variant1 and domain name variant2, the forensics software may not identify the relationship between the variants. Further, variants may significantly increase the botnet capability. A botnet could use all possible variants to register its nodes. Investigations to correlate all of these nodes would be more complex.

5.4. Traceability Matrix between User Principles and Variant Challenges

			Minimality	Security	Equivalency	Predictability	Manageability	Consistency	Ease-of-Use
5.1.	Challenges with the Use of Variants								
	5.1.1.	Different Users cannot find the complete set of variants for a primary label		x			x		x
	5.1.2.	Variants not intuitive		x		x		x	x
	5.1.3.	Variants are delegated independently		x	x	x			x
	5.1.4.	Variants defined inconsistently		x		x		x	x
	5.1.5.	Variants displayed inconsistently				x	x	x	x
	5.1.6.	Variants cannot be input by the user		x		x	x		x
	5.1.7.	Unable to distinguish specific variants		x		x	x		x
	5.1.8.	Identifier not bound to all variants		x		x	x		x
	5.1.9.	Accessibility and privacy impacted		x		x	x		x
	5.1.10.	Variants not searchable		x	x	x	x		x
	5.1.11.	Search rankings unpredictable			x	x			x
	5.1.12.	Search optimization affected by variants	x		x	x		x	x
	5.1.13.	Variants not part of URL/URI/IRI	x		x		x	x	x
	5.1.14.	Variants cause session re-establishment			x	x			x
5.2.	Challenges in the Registration and Management of Variants								
	5.2.1.	Management across IDN TLDs inconsistent			x	x	x	x	x
	5.2.2.	Registration for SLDs across TLDs inconsistent		x	x	x	x	x	x
	5.2.3.	Inconsistent association of ASCII and IDN TLDs			x	x	x	x	x
	5.2.4.	Software support inadequate		x			x		x
	5.2.5.	Registration system not straightforward to localize	x	x			x	x	x
	5.2.6.	Registration information inconsistent	x				x		
	5.2.7.	Trademark protection tracking complex					x		x
	5.2.8.	Trademark protection dispute process complex	x	x			x		x
5.3.	Challenges in the Configuration and Diagnostics of Variants								
	5.3.1.	Software configuration not supported		x			x		x
	5.3.2.	Cannot associate variants for configuration		x	x		x		
	5.3.3.	Compounded certificate management	x	x	x		x		

	5.3.4.	DNSSEC validation inconsistent	x	x	x		x		
	5.3.5.	Log and history searching does not match		x			x		
	5.3.6.	Network traffic statistics incomplete				x	x		
	5.3.7.	Caching infrastructure inefficient		x		x	x		
	5.3.8.	Diagnostic and troubleshooting tools incompatible		x		x	x		
	5.3.9.	Forensics significantly more complicated		x		x	x		

6. Recommendations

While the challenges in Section 5 are grouped by use cases, this section groups the recommendations by the target stakeholders to facilitate the identification of next steps.

For the second-level, various strategies are currently used to manage variants, such as blocking, automatic allocation, activating an arbitrary subset of variants based on the choice of the registrant, and restricting the activation of variants within a maximum limit (normally between 3-6 labels). These policies are motivated by linguistic/script requirements of the community as well as practical limitations or technical constraints. Registries have not reported any major issues in deploying variants at the second-level. While the variants at the top-level are different and more constrained (e.g., [42]), deployment of variants at the top-level should be informed by those second-level deployments.

While activating variants may better serve specific language communities, this report does not make any specific recommendations on how variants should be technically implemented in the DNS or even whether they should be implemented at all. Understanding that each additional active variant may add complexity and/or security risks for the registrants and end users, the recommendations presented here generally seek to block/withhold a maximum number of variants while activating a minimal number of variants.

From most end users' point of view, a TLD is just one part of an Internet identifier, often embedded in something else, such as a URI or an email address. Thus, apart from technical considerations of the shared root zone across multiple languages and scripts and specific issues regarding code points as discussed in Integrated Issues Report, end users expect consistent support of variants across all levels of the DNS tree.

The DNS is a hierarchical distributed naming system and its namespace has multiple levels of labels, where each sub-tree may be delegated to a different entity. Therefore, there is no possible way for ICANN, TLD registries or others to enforce rules at all levels of the tree. The report makes recommendations to the various stakeholders, recognizing implicitly that it is an imperfect solution that may not be followed down the whole DNS tree. The focus of these recommendations is limited to the variants as defined by the LGR for the relevant zone. Most recommendations are made under the assumption that some IDN variant TLDs are deployed. The recommendations use *must, should* and *may* to distinguish between their priorities, as discussed in Section 1.

6.1. Recommendations to ICANN

6.1.1. ICANN must implement a well defined and conservative variant TLD allocation process

A TLD label application may have many variant labels. The applicant may desire one or more of these variant labels for activation. The minimality principle discussed earlier states that given the inherent complexity of supporting and using variant labels, a conservative approach to adding variants is advisable. Minimal necessary active variants will provide a better user experience, as they are easier to configure, manage, monitor and navigate.

To ensure a minimal number of variants are activated, the following requirements should be considered before any variant label of a gTLD or ccTLD is approved.

1. The approval of an variant TLD must not be automatic, but initiated upon the request of an applicant, explicitly specifying(i) the variant label, (ii) the status for which the variant should be evaluated (activated, allocated but not activated, etc.), and (iii) the need for the variant (e.g., motivated by linguistic, security, usability and/or other considerations). Unless such an application is initiated, all variants generated against a primary TLD application by the root LGR should remain withheld (and un-allocated).
2. A variant TLD application must be accepted only if the applicant clearly demonstrates the necessity for activating the string. Variants that are not necessary, but are desired, must not be allocated and activated.
3. TLD variant(s) must be applied for by and allocated to the same entity or registry that has applied for the corresponding primary TLD label.
4. All requirements for a TLD application approval process also apply to the approval of a variant TLD. These include, for example, requirements for GAC and public comments on the label, string similarity evaluation, DNS stability evaluation of the variant TLD label, etc. ICANN must document this process.
5. The registry delegation and re-delegation processes must be extended to include activated variants of a TLD. The registry contract must be updated accordingly.
6. The registry fail-over plan should be extended to include activated variants of a TLD. The relevant registry contract must be updated accordingly.

6.1.2. ICANN must maintain an LGR repository for the root zone and IDN TLDs and make it available to users and programmatically processable

Variants require specific processing in various parts of the domain ecosystem, such as Internet software on the client and server sides, registrant-registrar-registry interactions, search engines, etc. Therefore, a mechanism is needed to unambiguously determine the existence of a TLD and its complete variant set.

1. ICANN must provide the following, accessible programmatically, e.g., via RESTful API, and through other mechanisms, e.g. web interface.

    a. Root zone LGR
    b. State of each variant (activated, withheld, blocked, etc.)of each allocated TLD
    c. Second-level LGR submitted for each TLD

This repository must conform to a standard specification, developed either within the ICANN community or in collaboration with standards organizations, such as the IETF. The repository must be accessible at least for informational use.

The availability and scalability of the repository for active operational use by software applications may be needed but may put ICANN in active operational role. ICANN should engage with the community in such case to chart the best way forward.

2. Appropriate training materials should be developed for the relevant user communities (as identified in Section 4) to use the repository. Also see Recommendation 6.1.7.

6.1.3. ICANN must develop, to the extent possible, minimal, simple and consistent LGR for the root zone

Minimal, simple and consistent rules will promote adoption of variants by end users and development of variant support by application developers. Thus, acknowledging that languages and scripts have many inherent differences, to the extent possible, variant sets should be generated employing a minimal number of well-documented and consistent rules across languages and scripts.

Such rules should avoid integrating complex script or language dependent deviations to minimize inconsistencies. Any deviation necessary for a particular script or community should be clearly justified and documented in the root LGR.

The following list provides some considerations for the *character repertoire* in the LGR to minimize non-essential variant labels:

1. The code points allowed for LGR must include only those minimally needed by a particular script community. For example, the repertoire should not include dead scripts and code points representing archaic characters that are not currently in use by a script (as per IDN P2.1).
2. If the community cannot agree on the need of a code point, the default decision must be not to include it in the repertoire.
3. Any code point that is optionally written in a script (e.g., combining marks) must not be included.
4. There must be explicit description and justification for inclusion of each code point that causes a variant directly or in combination with other code point(s) by the community developing the LGR.
5. Code points in the LGR for the root should be based on script. Language-specific code points should be minimized, and any such code points should be explicitly justified. This may result in increasing the variant labels but will promote consistency of use across global end users.

The following list provides some considerations for the *variant rules* in the LGR to minimize non-essential variant labels:

1. Code point variants must be minimized, motivated primarily by the secure use of the Internet; each such rule should be justified, stating clearly whether it is motivated by security considerations or community needs or both.

2. Variant rules not motivated by security reasons must only be allowed if there is significant community need. These should be accepted based on documented community need.
3. As TLDs – particularly gTLDs –are used by a global end user community, variants for TLDs must be defined for a script. Language-based variant rules should not be allowed.
4. A simpler variant rule is preferable to a more complex variant rule, in instances in which there are options available; for example, 1:1 code point variant rule is simpler than 1:many code point variant rule; context-free variant rules are simpler than context-sensitive variant rules, etc.
5. Variant rules should not produce a contradictory situation that allows code points to form variant labels for one community, but create unique labels for another community. This is possible both by presence vs. absence of a single code point (e.g., a combining mark) or by mapping between two different code points. Such contradictions should be avoided both within and across scripts.

6.1.4. To help ensure that users have a more predictable and consistent experience registering and using primary and variant labels, ICANN must develop, to the extent possible, a minimal, simple and consistent life cycle for the variant TLD sets (across languages and scripts)

The Integrated Issues Report [4] identifies the following states for variant labels: *blocked, withheld, allocated, activated, delegated,* and *mirrored.* The life cycle for how variant labels for the TLDs can change between these states should be clearly defined and kept as simple and consistent as possible for end users. These include how primary labels may be defined, which subsets of variants are activated, how they may change their activation states, etc.

While developing the life cycle, the following consideration must be taken into account.

1. Complete sets of (any) reserved labels and their variants must be developed and publicly announced for the root zone.
2. For each TLD application, all possible variants generated through LGR should be withheld by default, and not available to any other applicant.
3. A variant must be in a withheld state for it to be activated after an appropriate evaluation process (see Recommendation 6.1.1).
4. A variant must be in an activated state for it to be delegated/mirrored.
5. A separate technical analysis should be conducted to determine whether a delegated variant of a TLD can be changed to a mirrored state or vice versa.
6. It should not be possible to make a delegated/mirrored variant inactive (withheld, blocked or reserved), as it will have an adverse impact on existing registrations.
7. A separate analysis should be conducted to determine whether an allocated variant that is not activated may be unallocated (withheld, blocked or reserved).
8. Appropriately detailed procedure must be developed to apply for change in state of any TLD variant. Such procedure must include a DNS stability evaluation if the target or source state of the variant is "activated."
9. The updated states of all TLDs and their variants must be available publicly, as discussed in Recommendation 6.1.2.

6.1.5. ICANN must define guidelines to evaluate the competence and readiness of the registry to manage variants, to ensure a stable and secure end user experience

Activating variants requires technical and linguistic competence and appropriate internal policy augmentation of the relevant registry. ICANN should clearly define these requirements for activation of a variant TLD. This should be in addition to the evaluation ICANN conducts for the applicant of any other IDN TLD label during the delegation process. Activation of a variant TLD from the registry should be deferred until such pre-requisites are met.

In developing such procedures, the following additional recommendations on linguistic capacity should be addressed:

1. The registry must have a demonstrated linguistic capacity to develop and maintain the second-level LGR.
2. The variant TLD must use the same second-level LGR as the second-level LGR being used by the corresponding primary TLD label. Registry contract should be updated to address this requirement.
3. The applicant must have a clear and consistent linguistic policy for the second-level (including LGR and variant state life cycle for second-level) that is not in conflict with the root-level LGR and state life cycle.

In developing such procedures, the following additional recommendations on technical capacity should be addressed:

1. The IDN TLD applicant requires maintenance and implementation of LGR at the second-level within its registration and resolution process. The registry must have technical capacity for this purpose. This includes at least the following:

    a. A registration system supporting the registration and activation of variants
    b. A DNS zone generation supporting variants
    c. A registration data query system using the standard protocol supporting variants

6.1.6. ICANN should require IDN TLD registries with variants to apply the relevant (script) subset of the root zone LGR and state life cycle for variants across second-level domain labels. Deviations should be justified

Minimal, simple, and consistent solutions across the DNS tree enable software developers to more consistently and easily integrate support for variants across applications that interact with the DNS. End users, in turn, benefit from a consistent level of support for variants across applications. Conversely, any deviations by registries risk generating variant sets that are not compatible with general software applications. Therefore, registries must be required to adopt the relevant portion of the root-level LGR at other levels in the DNS, and carefully consider and justify any deviations to help ensure a more consistent and predictable experience for registrars, registrants and, ultimately, end users. Any additional code points not in the root LGR (e.g., digits, hyphen, etc.) and corresponding rules should be clearly documented.

Registries should also be encouraged to adopt at the second-level the same state life cycle defined for variants at the root so that users have a consistent experience across the DNS tree. Any deviations should be carefully considered and justified.

Specific recommendations include the following:

1. ICANN must evaluate the second-level LGR submitted by the IDN TLD applicant to ensure that it conforms to the LGR for the root. ICANN must explicitly require justification of any deviation to ensure consistency of experience at the second-level and to avoid contradiction or conflict (e.g., code points that are variants at the root level should not be delegated independently at the second-level).
2. ICANN must develop a variant state life cycle for second and other levels, in line with the life cycle at the root zone. ICANN must evaluate the second-level variant state life cycle submitted by the IDN TLD applicant to ensure that it conforms to the guidelines by ICANN. ICANN should explicitly require justification of any deviation to ensure consistency of experience at the second-level.
3. ICANN must update IDN TLD registry contract to acquire, evaluate, and publicly publish second-level LGR and variant state life cycle, where applicable.
4. ICANN must require a registry to allocate all the second-level variants to the same registrant in line with the policy at the root level. ICANN should also encourage the registry to advise the registrants to point SLD.TLD variants to the same or similar content.

6.1.7. ICANN must create educational materials on the use and impact of variants for different user communities

Educational materials are needed to raise awareness about the variant in the different user communities (identified in Section 5). These materials should explain IDN variants, how they are determined and used, and their potential impacts on the community. These materials should be available in multiple languages with localized examples.

1. ICANN must create training material on the use and impact of variants for end users in all the languages for which there are variants for TLDs (and potential variants for SLDs), in collaboration with the relevant registries.
2. ICANN must create training materials on best practices for registration of variants and maintaining relevant and accurate registration data. The materials should include use of EPP with registries, and guidelines for appropriate display of variants for potential registrants.
3. ICANN should create training materials on how to configure and manage variants for various clients and servers for system administrators and network managers.
4. ICANN must create information materials on the impacts variants may have on network security management.
5. ICANN may create training materials to make developers aware of the possible impacts of variants on end user applications (e.g., web browsers, email clients, search engines) to help developers understand the challenges and encourage support for variants where needed to improve the user experience.

6.1.8. ICANN must require any accredited registrar that supports IDNs with TLD and/or SLD variants to support variants across its registration platform

Current levels of support for IDNs are limited in the services provided by some registrars. Variant sets add an additional level of complexity. ICANN should develop consistent guidelines and registration practices for the support of IDNs in general and variant sets where relevant. Any changes to the Registrar Accreditation Agreement should be discussed and included.

1. ICANN must require registrar offering a TLD which has variant(s) at TLD and/or SLD levels to provide adequate (updated) interfaces for registrants to view variant sets of domain names being offered and to allow registrants to register any relevant sub-set.
2. ICANN must require registrar offering a TLD which has variant(s) at TLD and/or SLD levels to provide adequate interfaces for registrants to request change in state of a variant.

6.1.9. ICANN must develop consistent registration data requirements for variants at root and other levels

A consistent set of requirements for registration data should be developed for variants in consultation with the user community.

1. ICANN must clearly define the application data requirements for a variant TLD for each state.
2. ICANN must clearly define how this data will be updated as variant TLD changes state.
3. ICANN must define registration data requirements and accuracy measures for second-level variants, and update the Registrar Accreditation Agreement to collect and maintain the relevant registration data.
4. ICANN should incorporate measures to gauge and implement the accuracy of such data for variants into its contractual compliance process.
5. ICANN should ensure that the relevant data is included in the Registration Data Escrow (RDE) program, to protect registrants.

6.1.10. ICANN must convene relevant experts involved in domain name disputes to determine any new issues introduced by variants and update existing dispute resolution processes accordingly

Variants may have an impact on disputes and their resolution process. Appropriate experts from the relevant communities should be consulted to understand the extent of issues and solutions. The consultation should include all the stakeholders, e.g., the registry/registrar community, legal experts who understand trademarks and the associated dispute resolution process (e.g., UDRP), and script communities involved in developing variant set guidelines.

6.1.11. ICANN must define technical requirements and engage with standards organizations, such as the IETF, to determine how the IDN variants should be consistently implemented

Many registries have already implemented variants at the second-level. ICANN should clearly document the requirements for implementing variants for top-level and second-level, in consultation with the community that is already implementing them and the community that will need to implement the variants. ICANN should engage with IETF and other relevant organizations to at least document the best practices to implement variants at these levels.

1. ICANN must develop requirements and seek guidance from the protocol standard organization IETF on effective and consistent mechanism to enable variants both at the TLD and lower levels.
2. Standards are a key for widespread deployment and interoperability. ICANN must develop technical requirements and then work with the protocol standard organizations, such as IETF, to develop or update relevant protocols to ensure interoperability of IDN variants. Examples include, but are not limited to, domain provisioning protocol (EPP), domain registration data lookup protocol, etc.

6.2. Recommendations to a Registry that Offers IDNs for Scripts that have Variants

6.2.1. Registry must not register any second-level variant labels unless the label registration request has met all approval requirements

Many variants may result from the combination of one or more variants of a TLD and variants of at second-level (i.e., SLD.TLD.). Some of these labels are desired for activation by the registrant. To ensure a minimal number of variants are registered, the following requirements should be considered before any variant label is allocated.

1. The registration of a variant label must not be automatic, but initiated on the request of a registrant, explicitly specifying (i) the variant label and (ii) the state for which the variant is being requested (delegated, allocated but not delegated, etc.). By default, all variants should be withheld (and un-allocated).
2. A variant must be registered to the same registrant who has applied for the corresponding primary label.
3. All requirements for a label registration process must also apply to the registration of a variant label.
4. The registration of all variants should be tied to the registration of the primary label. For example, if the latter expires, the former should also expire.

6.2.2. Registry that supports variants must make its updated LGR available to ICANN and the community

Variants may require specific processing in various parts of the domain ecosystem, such as Internet software on the client and server side, registrant-registrar-registry interactions, search engines, etc. To address this need to unambiguously determine a complete variant set, the second-level LGR of the registry should be uploaded to the ICANN LGR repository

(see Recommendation 6.1.2.). The registry should ensure that it timely contributes the second-level LGR and any revisions to this repository.

6.2.3. Registry that supports variants should apply the LGR developed for the root across lower-level domains. Deviations from the LGR should be publicly documented and justified

To provide end users with a consistent experience across the DNS tree, a registry should adopt the LGR available for the root for the relevant script at other levels as well. However, if deviation from the root LGR is necessary, the registry must consider the minimal deviations necessary from the root LGR, and document such cases. Registries should note that all changes are not equal, and the following cases have varied degree of impact on user experience (ranked from best to worst cases):

a. Relevant (script) portion of the root LGR and the second-level LGR is exactly the same (best end user experience).
b. The second-level LGR is a subset of the relevant portion of the root LGR, with all relevant rules for relevant character repertoire included.
c. The second-level LGR includes a subset of the root LGR, with all relevant rules for relevant character repertoire included, but also some additional code points and rules are included (e.g., digits, hyphen, and other characters not allowed in the root).
d. The second-level LGR includes a subset of the root LGR (with or without any additional code points) but does not include all relevant rules from root LGR, though does not create a conflict in variant sets across the two levels.
e. The second-level LGR has additional rules that conflict with the root LGR; for example, two code points that are variants in the root LGR are not variants in the second-level LGR or vice versa (not recommended).

In addition to the Recommendation 6.1.3 for developing the LGR for the root, the following should also be considered:

1. There must be explicit justification for exclusion of each code point or variant rule from those already included in the relevant (script) portion of the root LGR.
2. There must be explicit justification for inclusion of each code point or variant rule, in addition to those already included in the relevant (script) portion of the root LGR.

6.2.4. Registry that supports variants must implement, to the extent possible, state life cycle for the second-level variant recommended by ICANN

The life cycle of how variant labels can change between states should be clearly defined and kept as simple and consistent with ICANN guidelines (see Recommendation 6.1.6 (2) for further details).

6.2.5. Registry should create educational materials on the use and impacts of variants for different user communities, such as end users, system administrators, etc.

Educational materials are needed to raise awareness for the relevant user communities (identified in Section 5). These materials should be available in multiple languages with localized examples.

1. Registry should translate training material on the use and impact of variants for end users in the languages it is serving.
2. Registry should disseminate the training material developed by ICANN on how to configure and manage variants for various clients and servers to system administrators and network managers.
3. Registry should use the training material on best practices for registration of variants and maintaining relevant and accurate registration data developed by ICANN.

6.2.6. Registry offering variants must require relevant registrars to support IDN variants across their registration platforms

Current levels of support for IDNs are limited in the services provided by some registrars. Variant sets add an additional level of complexity.

1. Registry should require registrars to provide adequate (updated) interfaces for registrants to view and select from variant sets of domain names being offered, and to allow registrants to register any sub-set.
2. Registry should require registrars to provide adequate interfaces for registrants to request changes of state of variants corresponding to the primary domain names.
3. Registry must require that second-level variants belong to the same registrant.

6.3. Recommendations to a Registrar that Supports the Registration of Variants

6.3.1. Registrar must update its practice to address requirements specific to the registration of IDN variants

1. Registrar must allocate variants of a domain name to the same registrant.
2. Registrar should disclose any fees related to the processing and activation of variants.
3. Registrar should work with ICANN to develop materials (in appropriate languages) to help registrants understand variants and the associated registration process (e.g., how to select which variants to register, which states variants may have, how the variant subset registration and their states may be updated, etc.).
4. Any service by the registrar which packages multiple non-variant domains together, such as ASCII domain with IDN and IDN variant domains, must make it clear to the registrants that such an offering is a value-added service not related to variants. Providing such clarity helps provide consistent user experience with variants.

6.3.2. Registrar should extend linguistic and technical support of IDN variants for registrants

Clear and consistent guidelines and practices should be developed for the support of IDNs in general and variant sets in particular.

1. Registrar should update its technical and linguistic expertise to handle the additional complexity introduced by variants at all levels across the DNS. This includes expertise to understand the LGR for the script and languages being supported.
2. Registrar should have technical and linguistic staff to address any relevant queries from the registrants for interpreting LGR and its implications.

3. Registrar should update its registration interfaces to display variants and allow registrants to understand, prioritize, and select/update variant subsets and their states for registration. Registrants should also be made aware of pricing and service level implications related to variants.
4. Registrar may point registrants to the guidelines developed by ICANN on how to configure their servers to handle variants.

6.3.3. Registrar must support IDN variants across their registration platforms

A registrar who supports IDN variants should update its back-end systems to interact with relevant registries to determine the complete set of variants available for registration (processing the different LGRs), and update the registry with registration information for new and updated registrations, including the subset of variants and their individual states (and registration data, where relevant). This should be managed through the adoption of standard provisioning systems (e.g., using EPP) instead of customized solutions.

6.3.4. Registrar must support registry policies and associated services for collecting and managing registration data of IDN variants

Registrars must conform to registry policies regarding the collection, management, and sharing of registration data for variants

1. Appropriate internationalized interfaces should be developed for the collection and maintenance of registration data for variants. These interfaces should be localized for relevant languages being served by the TLD being supported.
2. Additional measures should be incorporated to ascertain and maintain the accuracy of internationalized registration data for variants; for example, the owner of a variant label should be the same as the owner of the corresponding primary label.
3. Registrar should update its systems to escrow registration data associated with variants under any updated guidelines for the Registration Data Escrow (RDE) program by ICANN.

6.3.5. Registrar that supports the registration of variants may also update any related services that are impacted by variants

Registrars often offer additional services to registrants, including hosting services, certificates, privacy services, and proxy services. These services maybe enhanced and updated to support IDN variants.

6.4. Recommendations to the Technical Community

6.4.1. Developers of software tools for the technical community should consider, based on user requirements, to enhance their software to support the administration and management of variants

It is assumed that the technical community will require more clarity into variant sets and their status as well as more granular control of how such variants are managed across software and services. Feature requirements may include, but are not limited to the following:

1. Displaying the current status of IDN variant labels (delegated, blocked, active, etc.).
2. Displaying both A-labels and U-labels as required for management.
3. Pattern-matching and searching tools to assist users in identifying and managing variants.
4. Client/server software that is "variant aware" to enable enhanced monitoring and management of data traffic. For example, web traffic from primary and related variant domains may be aggregated for reporting and/or management purposes. Caching services may also be enhanced to identify variant labels as related to primary labels.

6.4.2. Software intended for Internet end users—such as web browsers, email clients, and operating systems—should support variants to the extent necessary to ensure a positive user experience

Software developers should be aware of IDN variants and how they may positively and negatively impact user experience. The degree of support required for variants will depend on the audience and specific use cases. Just because an IDN variant URL is successfully resolved within a web browser does not mean the user will have a positive experience overall. A variant URL may not be active or may resolve back to the primary URL. The first step towards ensuring an acceptable end user experience is to ensure that developers are aware of the usability issues posed by variants, such as:

1. Search engines not treating variants as equivalent to primary domains, potentially frustrating or confusing end users.
2. Users who rely on variant labels for user IDs, email addresses, etc. expecting the labels to be treated as equivalent with the primary labels across various software and services.
3. Native keyboards and IMEs not supporting variants as required by users.
4. Deletion of web page history logs for privacy settings not deleting variant domains.
5. Auto-complete functionality for domains not variant sensitive.
6. Usage sessions not consistent across primary domains and variants.

6.4.3. To provide end users with a consistent and predictable experience with variants across software applications, developers should, to the extent possible, publicly share best practices and emerging standards in terminology and functionality

Inconsistent support for IDNs and variants is expected to hamper their use among end users. And while these inconsistencies may be driven by companies pursuing the same goal via different approaches, a baseline of harmonization on how variants are managed across, for example, web browsers will not only benefit end users but developers and content creators who must support these different platforms. This recommendation may be expanded to apply to IDNs in general.

7. Concluding Remarks

<TBD>

8. Authors

This report is written by the following (in alphabetical order by last name):

Marc Blanchet
Dennis Chang
Sarmad Hussain
Steve Sheng (ICANN staff)
John Yunker

9. Acknowledgements

We would like to acknowledge the following individuals (in alphabetical order by last name) for providing feedback and/or supporting the process:

Greg Aaron	Baher Esmat	Christopher Mulligan
Abdulrahman Alzarooni	Patrik Fältström	Rod Rasmussen
Abdulaziz Al-Zoman	Robert Flaim	Peter Roman
Francisco Arias	Asmus Freytag	Alireza Saleh
Fahd Baynteh	Frédérick Gaudreau	Siavash Shahshahani
Marc Boivin	Nai-Wen Hsu	Jonathan Shea
Sébastien Bourdon-Richard	Manal Ismail	Terri Stumme
Cary Carp	Gary Kibbey	Andrew Sullivan
Edmon Chung	John Klensin	Wang Wei
Kim Davies	Sushil Manwar	Joseph Yee
Chris Dillon	Margie Milam	Hongbin Zhu
Mohammed El Bashir	Marc Moreau	

We would also like to thank the following organizations for providing their feedback and/or for responding to our surveys:

At-Large Advisory Committee, ICANN (ALAC)
Center for Development of Advanced Computing (CDAC), India
China Internet Network Information Center (CNNIC)
Hong Kong Internet Registration Corporation Limited (HKIRC)
National Information Technology Center Jordan (NITC)
Royal Canadian Mounted Police
Serious Organised Crime Agency (SOCA), UK
South Asian Network Operators Group (SANOG)
Sureté du Québec, Canada
Taiwan Network Information Center (TWNIC)
UAE Telecommunications Regulatory Authority (Operator of .ae IDN ccTLD)
US Department of Justice (US DoJ)
US Federal Bureau of Investigation (FBI)

References

Relevant ICANN Documents on Variants TLDs

1. Frakes, J, *et al.,* "Considerations in the use of the Latin script in variant internationalized top-level domains: Final report of the ICANN VIP Study Group for the Latin script",(Marina del Rey, California: ICANN, October 2011). http://archive.icann.org/en/topics/new-gtlds/latin-vip-issues-report-07oct11-en.pdf

2. Govind, *et al.,* "Devanāgarī VIP Team Issues Report", (Marina del Rey, California: ICANN, October 2011). http://archive.icann.org/en/topics/new-gtlds/devanagari-vip-issues-report-03oct11-en.pdf

3. Hussain, S, *et al.,* "Internationalized Domain Names Variant Issues Project Arabic Case Study Team Issues Report", (Marina del Rey, California: ICANN, October 2011). http://archive.icann.org/en/topics/new-gtlds/arabic-vip-issues-report-07oct11-en.pdf.

4. Internet Corporation for Assigned Names and Numbers, "The IDN Variant Issues Project: A Study of Issues Related to Management of IDN Variant TLDs (Integrated Issues Report)", (Marina del Rey, California: ICANN, February 2012). http://www.icann.org/en/topics/idn/idn-vip-integrated-issues-final-clean-20feb12-en.pdf.

5. Lee, X., *et al.,* "Report on Chinese Variants in Internationalized Top-Level Domains", (Marina del Rey, California: ICANN, October 2011). http://archive.icann.org/en/topics/new-gtlds/chinese-vip-issues-report-03oct11-en.pdf.

6. Segredakis, V., *et al.,* "Study of the Issues Present in the Registration of IDN TLDs in Greek Characters",(Marina del Rey, California: ICANN, October 2011). http://archive.icann.org/en/topics/new-gtlds/greek-vip-issues-report-07oct11-en.pdf

7. Sozonov, A., *et al.,* "IDN Variant TLDs – Cyrillic Script Issues", (Marina del Rey, California: ICANN, October 2011).http://archive.icann.org/en/topics/new-gtlds/cyrillic-vip-issues-report-06oct11-en.pdf

Relevant IETF RFCs

8. Alvestrand, H., Ed., and C. Karp, "Right-to-Left Scripts for Internationalized Domain Names for Applications (IDNA)", RFC 5893, August 2010.

9. Costello, A., "Punycode: A Bootstring Encoding of Unicode for Internationalized Domain Names in Applications (IDNA)", RFC 3492, March 2003.

10. El-Sherbiny, A., Farah, M., Oueichek, I., and A. Al-Zoman, "Linguistic Guidelines for the Use of the Arabic Language in Internet Domains", RFC 5564, February 2010.

11. Faltstrom, P., Hoffman, P., and A. Costello, "Internationalizing Domain Names in Applications (IDNA)", RFC 3490, March 2003.

12. Faltstrom, P., Ed., "The Unicode Code Points and Internationalized Domain Names for Applications (IDNA)", RFC 5892, August 2010.

13. Hoffman, P. and M. Blanchet, "Nameprep: A Stringprep Profile for Internationalized Domain Names (IDN)",RFC 3491, March 2003.

14. Klensin, J., Faltstrom, P., Karp, C., and IAB, "Review and Recommendations for Internationalized Domain Names (IDNs)", RFC 4690, September 2006.

15. Klensin, J., "Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework", RFC 5890, August 2010.

16. Klensin, J., "Internationalized Domain Names in Applications (IDNA): Protocol", RFC 5891, August 2010.

17. Klensin, J., "Internationalized Domain Names for Applications (IDNA): Background, Explanation, and Rationale", RFC 5894, August 2010.

18. Konishi, K., Huang, K., Qian, H., and Ko, Y., "Joint Engineering Team (JET) Guidelines for Internationalized Domain Names (IDN) Registration and Administration for Chinese, Japanese, and Korean", RFC 3743, April 2004.

19. Lee, X., Mao, W., Chen, E., Hsu, N., and J. Klensin, "Registration and Administration Recommendations for Chinese Domain Names", RFC 4713, October 2006.

Relevant Unicode Documents

20. The Unicode Consortium (2012a). "Unicode Technical Standard 46: Unicode IDNA Compatibility Processing", M. Davis, M. Suignard (Eds.). Retrieved fromhttp://www.unicode.org/reports/tr46/.

IDN ccTLD Variant Practices

21. Arabic Script IDN Working Group (2012). Available at: http://asiwg.org/wiki/Main_Page

22. Al-Zoman, A. and Al-Fayze, R. (2008) "Proposed Solution for Writing Domain Names in Different Arabic Script Based Languages", ASIWG 3rd meeting, Cairo 2008. Retrieved from http://xn--ghb5rwd.com/docs/Resources/ASIWG-proposal-Nov08.pdf

23. China Internet Network Information Center (CNNIC) (2010) "Implementation Plan of .中国 (xn--fiqs8S) and 中國 (xn--fiqz9S)". Retrieved from http://www1.cnnic.cn/html/Dir/2010/06/12/5852.htm

24. Iran Network Information Center (IRNIC) (2010) “Internationalized Domain Names”. Retrieved from http://www.nic.ir/Internationalized_Domain_Names

25. Saudi Network Information Center (Saudi-NIC) (2010) “Guideline Rules for writing Arabic IDNs under the IDN ccTLD”. Retrieved from http://nic.sa/en/view/writing_arabic_idn_guideline

26. Saudi Network Information Center. (2010) “Arabic Domain Name Registration Regulation for Saudi IDN ccTLD”, Retrieved from http://www.nic.net.sa/docs/ADN/Arabic_Domain_Name_Registration_Regulation_For_Saudi_IDN_ccTLD_V1.pdf

27. Taiwan Network Information Center (TWNIC) (2009) “TWNIC’s Registration Policy and Technical Solution for the .台灣 and .台湾 IDN ccTLD”. Retrieved fromhttp://www.twnic.net/english/dn/pdf7a.pdf

28. Taiwan Network Information Center (TWNIC) (2010) “Registration Policy and Rules for .台灣 Top-Level Domain Names in the Chinese Characters”. Retrieved from http://www.twnic.net/english/dn/policya.pdf

29. UAE Telecommunications Regulatory Authority. (2012) “Arabic (dot-Emarat) امارات. Domain Name Policy”. Retrieved from http://cp.nic.ae/NIC/uaenic/english/pdf/emaratdomain/Arabic(dot-Emarat)DomainNamePolicy.pdf.

Usability Resources

30. Grudin, J. (1992). “Utility and Usability: Research Issues and Development Contexts”, *Interacting with Computers*, 4(2): 209–217.

31. International Organization for Standardization (ISO) (1998). “Ergonomic requirements for office work with visual display terminals (VDTs) - Part 11 : Guidance on usability”, ISO 9241-1998(E), International Standards Organization (ISO), Geneva, Switzerland.

32. Nielsen, J. (1993). *Usability Engineering*, Morgan Kaufmann, San Francisco, CA, USA.

33. Tulllia, T. and Albert, W. (2008). “Measuring the User Experience: Collecting, Analyzing, and Presenting Usability Metrics”, Morgan Kaufmann, San Francisco, USA.

Other References

34. ICANN Generic Name Supporting Organization (GNSO) (2007a). “Final Report: Introduction of New Generic Top-Level Domains”. Retrieved fromhttp://gnso.icann.org/en/issues/new-gtlds/pdp-dec05-fr-parta-08aug07.htm.

35. ICANN Generic Name Supporting Organization (gNSO) (2007b). "Outcomes Report of The gNSO Internationalized Domain Names Working Group (IDNWG)". Retrieved from http://gnso.icann.org/en/drafts/idn-wg-fr-22mar07.htm.

36. Internet Corporation for Assigned Names and Numbers (ICANN). (2009d). "Registrar Accreditation Agreement", Marina Del Rey, CA: ICANN. Retrieved from http://www.icann.org/en/registrars/ra-agreement-21may09-en.htm#2

37. Internet Corporation for Assigned Names and Numbers (ICANN). (2010a) "Board resolution (2010.04.22.10) CNNIC Request for (Simplified and Traditional Chinese) IDN ccTLDs", Marina Del Rey, CA: ICANN. Retrieved from http://www.icann.org/en/groups/board/documents/resolutions-22apr10-en.htm#synchronized

38. Internet Corporation for Assigned Names and Numbers (ICANN). (2010b) "Board resolution (2010.04.22.11) TWNIC Request for (Simplified and Traditional Chinese) IDN ccTLDs", Marina Del Rey, CA: ICANN. Retrieved from http://www.icann.org/en/groups/board/documents/resolutions-22apr10-en.htm#synchronized

39. Internet Corporation for Assigned Names and Numbers (2012a). "IDN ccTLD Fast Track Process", retrieved from: http://www.icann.org/en/resources/idn/fast-track

40. Internet Corporation for Assigned Names and Numbers (2012b). "Updated Final Implementation Plan for IDN ccTLD Fast Track Process". Retrieved fromhttp://www.icann.org/en/resources/idn/fast-track

41. Internet Corporation for Assigned Names and Numbers (2012c). "IDN Variant TLD Program – Revised Program Plan". Retrieved fromhttp://www.icann.org/en/resources/idn/idn-variant-tld-updated-plan-23aug12-en.pdf

42. The Internet Architecture Board. (2012). Response to ICANN questions concerning "The interpretation of rules in the ICANN gTLD Application Guidebook". Retrieved from http://www.iab.org/documents/correspondence-reports-documents/2012-2/response-to-icann-questions-concerning-the-interpretation-of-rules-in-the-icann-gtld-applicant-guidebook/

43. ICANN Security and Stability Advisory Committee (SSAC). (2012) "SSAC Advisory of the Delegation of Single-Character IDN Top-Level Domains", (SSAC publication No. 052).

44. ICANN Generic Names Supporting Organization (GNSO). (2012) "Draft Final Report of Internationalized Registration Data Working Group", Marina Del Rey, CA: ICANN. Retrieved May 17, 2012, from http://gnso.icann.org/issues/ird/ird-draft-final-report-03oct11-en.pdf