Examining Top Level IDNs
John C Klensin

I. A Terminology Problem

If ICANN refers to top level IDNs as "IDN.IDN", confusion will be introduced with a number of strategies, some of which are completely unviable in the DNS. In particular, the idea that, if a top-level domain is an IDN, all of the second-level registrations in that domain and all of the subtrees below them are required to be IDNs –possibly IDNs in the same language or script group—may not be viable, enforceable, or good DNS policy. If such a thing is proposed, it should be discussed on its own merits, not assumed as part of "full implementation of IDNs". If the discussion is of IDNs at the top level, or IDN TLDs, it would be far better to use that terminology and not pre-judge the issue.

It is obviously just a personal opinion, but I want to strongly encourage ICANN to understand the position that some proposed uses of IDNs, especially at the top level, could essentially wreck the DNS as the Internet's primary mechanism for unique and unambiguous references to network objects. Worse, because DNS names are key to, and compared in, some security mechanisms, improper use of aliasing techniques or simulation of them could compromise Internet security. ICANN's chief responsibility in the DNS area is to preserve the stable, ambiguity-free, and effective use of the global Internet: that responsibility must take priority over the often short-sighted objective of "letting people do what they want", no matter how powerful, loud, or well-connected those "people" appear to be.

The discussion in this document assumes some minimal understanding of how internationalized names actually work in the DNS and, in particular, how those names are encoded for insertion into, and retrieval from, the DNS. I would hope that ICANN would avoid making policy decisions about any aspect of the DNS without that understanding clearly in mind[1]. In the text below, I have deliberately written possible domain names in punycode[2]. Since that is the form that goes into the DNS, it is the form on which any ICANN policy-making or domain allocations should be based. This is consistent with the recommendations of the initial President's Committee on IDNs.

---

[1] The standard for implementation and use of IDNs is RFC 3490, "Internationalizing Domain Names in Applications (IDNA)". P. Faltstrom, P. Hoffman, A. Costello. March 2003. ftp://ftp.rfc-editor.org/in-notes/rfc3490.txt. An extended tutorial on the technical foundations and operation of the DNS, including IDN mechanisms, appears in the US National Research Council Report, *Signposts in Cyberspace: The Domain Name System and Internet Navigation*, National Academy Press, Washington, DC: 2005. http://books.nap.edu/catalog/11258.html

[2] Punycode is defined and described in RFC 3492, "Punycode: A Bootstring encoding of Unicode for Internationalized Domain Names in Applications (IDNA)". A. Costello. March 2003. ftp://ftp.rfc-editor.org/in-notes/rfc3492.txt

II. Two Separate Types of Domains

Characterizing the issue here as one of IDNs at the top level misses some important points. In the vast majority of cases, the goal is not "an IDN TLD" as an end in itself. Instead, it is better access to the Internet for some population or a domain that matches some population's needs better than the existing ones. We have long had a distinction between "generic" TLDs and country-code ones; that distinction may be important for top-level IDNs as well.

III. Generic Domains

For new generic domains, the original President's IDN Committee recommended that applications for such domains be processed as if they were conventional applications. While there was no announcement during the recent round of sponsored TLD applications, the essence of the Committee recommendation was that IDN proposals could be accepted at any time TLD proposals were submitted and handled identically. From that point of view, an application for *xn--gemtlich-85a* should be treated no differently from an application for either an ASCII representation or transliteration of the same word: if ICANN followed previous procedures, they would need to evaluate the technical and financial integrity of the applicant and, by some criterion, the appropriateness of the domain, if it were sponsored, the appropriateness and reasonableness of the sponsorship claim, and so on. With only a few exceptions, the choice of name for a proposed TLD has not been a prominent issue. As long as the community understands that, worldwide, any string that uses non-ASCII characters may be harder to type (unless the punycode form is chosen) than a purely ASCII domain name, there should be no bar to TLD applications using non-ASCII characters.

There are additional difficulties with domains whose names are associated with some semantic concept in a particular language. ICANN may receive applications for the same, or similar, semantic constructs, expressed in different languages. To some degree, there is already precedent for this. After all, we already have "COM" and "BIZ" with very similar intended semantics and registrations, even though neither is, strictly speaking, a word in English (or in any other language that is broadly represented in the community). However, as in many other areas, the introduction of the range of scripts and languages accessible through IDNs multiplies the question of the appropriateness of semantic redundancy at the top level by a very large factor. As a trivial example, would ICANN accept a proposal for a TLD named *xn--geschft-9wa* if the application were satisfactory except for the choice of name? If not, on what basis would such an application be rejected and would that basis make this an IDN issue or a general TLD allocation one?

As a broader and more interesting example, suppose a request were submitted for a new TLD to be called *xn--80abmrog4b1d*, with the expectation that people who attributed that category to themselves would register in it. Would *xn--6dbf0as6a* be considered to be in conflict with it? The semantics in the relevant languages are not the same although the

superficial definitions are. A reasonable person might predict that few people would want to register in both domains. After all, *xn--80aa2ba.* *xn--6dbf0as6a* would be a bit unusual, but is certainly is not impossible, and the "bidi"[3] issues it raises for presentation are presumably out of ICANN's scope as long as only single labels are being considered (see "what problem is being solved", below). One would, however, not normally expect a great many overlapping registrations. The commercial efforts of the "protect your name, register everywhere plausible" community might, however, overcome this presumption of reasonableness, resulting in many redundant registrations. Would it be wise to permit top-level registrations for similar concepts using words from every language in the world that had those concepts? If the answer were "yes", would those registrations overwhelm the maximum manageable size and growth rate of the DNS root? If it were "no", how does one decide which languages should be represented, even after deciding which concepts are to be represented?

IV. TLDs Tied to Country Names

A Review of ISO 3166 and Its Relationship to the DNS

Before discussing the relationship between ccTLDs and top-level IDNs, it is important to review what ISO 3166 is about and why the IANA chose to use it. ISO 3166 and, in particular, ISO 3166-1, is an international standard for the identification of countries and similar entities. ISO 3166 has been widely adopted for multiple purposes: the codes are used on international mail bags, in currency exchange (e.g., the notations GBP and CNY consist of the two-letter ISO 3166-1 codes, plus a nationally-designated letter code for the currency), on passports, to identify securities traded internationally, and for a variety of other purposes[4]. The coding system has been endorsed by virtually every ISO (national) member body. While the codes are written in Latin characters, they are nonetheless

[3] "Bidi" is the Unicode Consortium's term for "bidirectional strings", i.e., strings that contain characters from both scripts that normally run from right to left and those that normally run left to right. See Section 4.4 of *The Unicode Standard, Version 4.0*, The Unicode Consortium, Addison-Wesley, 2003 and Unicode Standard Annex #9, "The Bidirectional Algorithm".

[4] See http://www.iso.org/iso/en/prods-services/iso3166ma/04background-on-iso-3166/you-and-iso3166.html for more information.

codes, rather than names of countries[5]. IANA's use of the 3166 table served two very important, but separate, purposes[6]:

- It kept ICANN out of the business of determining which entities were to be treated as countries. Note that the entitlement to a TLD that goes with a ISO 3166-1 is separate from the question of what that TLD is called.
- It kept ICANN out of arguments about what country-entities were to be called and, in the process, created a standard convention about naming: all country-code domains were two characters long and no non-country domains were two characters long.

Part of the current top-level IDN discussion seems to hinge on the principle that countries should be able to be known by whatever name they select, typically an official name in one of the country's official languages. On its surface, this seems reasonable and, to the extent to which names within a country domain are referenced from within the country and the country's official name is not very long, it would even work well. But we need to remember that, if the name is to be used by populations who use very different scripts, it is likely that the TLD name for a country will need to be expressed in the ASCII-compatible form of an IDN, the so-called "punycode"[7]. Punycode has very poor mnemonic value and is not very compact (i.e., it typically requires many characters to represent a smaller number of characters in the original script). It is also worth returning to the ISO 3166-1 theme before moving off this topic: countries do not use their full names, or nationally-selected abbreviations, in postal traffic, in international currency transactions, or in a host of other transactions. They may, and often do, use aliases for the ISO 3166 code in domestic transactions, but not in international ones. It has not been clearly shown that domain names should be any different.

Some Options for IDNs Associated with Countries

For IDNs and country TLDs, several different ideas and proposals have floated around. These ideas are inconsistent in both statement and intent, although the same terminology is often used to describe all of them. This section attempts to summarize those ideas and

---

[5] The choice of Latin script (Roman-based) characters for the ISO 3166 codes was, obviously, not an ICANN or IANA decision. The reasons for that choice should be discussion in ISO contexts, but probably include the observation that Roman-based characters are, by a wide margin, the most widely used and recognized in the world. There are also very few of them and they are normally written and printed separately, rather than joined. Those characteristics, taken together, make them much easier to recognize accurately than most other plausible candidate scripts. It is worth noting that not only ISO, but such UN bodies as the ITU, always use a small subset of Roman characters where character-based protocol elements are required. Except for the amount of usage, several other scripts including Greek, Cyrillic, and Tamil would, in principle, be equally good candidates, but they are not nearly as widely used and recognized as Roman characters.

[6] These principles are implied, although not precisely spelled out, in the discussion of country-code domains in RFC 1591, "Domain Name System Structure and Delegation". J. Postel. March 1994. ftp://ftp.rfc-editor.org/in-notes/rfc1591.txt.

proposals and make suggestions about considerations in attempting to meet the perceived needs.

Simple Aliases

Description: The Country wishes to keep its 3166 domain but create one alias, in a national language, primarily for internal use and to prevent people who use IDNs from within the country from having to deal with a two-character ASCII TLD.

Possible option: Install these aliases in the root using DNAME records.

Discussion: While there are many circumstances in which DNAME is not appropriate, or behaves in ways different from what users predict, it appears to be quite safe when used to define a reference within a single zone file and quite predictable when there are no records other than the DNAME associated with the DNAME's label. DNAMEs would, however, require the same types of authorization management as ordinary root zone records: some process would be needed to determine who had the authority to ask for one and where it should point. More important, many countries have more than one official name, and sometimes several additional unofficial ones, reflecting different languages and scripts in use in those countries. It may simply not be practical to restrict a country to a single local-script alias, especially when it is constitutionally prohibited from favoring one official language over another. If it is not, then the number of these DNAME records could become a management problem, possibly creating pressure against establishing more new "real" TLDs. Finally, it may be worth mentioning that one vendor of popular operating systems, email clients, and a web browser does not yet support DNAME in its software (to date, it does not support IDNA either, so issues with DNAME may not make things worse).

Country Name in Lieu of the ISO 3166 Code

Description: It is possible to make a case that, while the role of ISO 3166 in determining what is and is not a country should remain unchanged, the use of its two-letter codes is no longer appropriate culturally and internationally. Countries which wish to opt-out of the 3166 code system should have the option to do so.

Possible option: If a country that now has a 3166-based country code domain wishes to shift to a domain named in the language and script of the country, ICANN should accept the request to make that change. Of course, such a request would require a plausible plan to transition out of the current domain, with firm dates, etc.

Discussion: The choice of whether or not to use the 3166 code should be left up to the relevant country, with the issue hinging on the balance between local needs and easy access internationally. Of course, if a country has multiple languages

that must be treated equally, any "one IDN name" or even "one extra IDN name" (see below) is not particularly helpful.

Additional Domains Reflecting National Languages and Scripts

Description: Several groups have suggested that countries should be entitled to one additional domain reflecting the country's name in a national language. Some approaches to these additional domains suggest that they should be completely separate from the existing country-code domains, others that they be somehow linked in terms of names and structures. The "completely separate domain" approach is unquestionably technically feasible, at least as long as the number of domains added per year is kept relatively small, but raises a number of difficult policy issues (see "discussion", below). After considerable discussion, the idea was rejected by the original President's IDN Committee, precisely because of those policy issues. By contrast, most of the ideas of partial or complete linkage between an internationalized TLD name and a ccTLD one are not technically feasible: if complete linkage is desired, than an alias (see above) should be considered. Linkages that contemplate, e.g., translation or transliteration of names throughout the domain tree are not feasible given the design and implementation of the DNS and the important principle of distributed administration of subdomains.

Possible option: Proposals in this area take several forms. Perhaps the most common is the suggestion that each ccTLD administrator be given one additional domain to use for a non-3166 domain name as it considered suitable.

Discussion: The approach of entitlement to exactly one additional name, presumably to represent the name of the country in national characters, raises a whole series of issues. ICANN should not embark on this path without examining its implications, where one stops, and how decisions should be made. In particular, unless other policies and attitudes are changed, there is no such thing as an experiment in the top-level domain space: unless clear and sustainable policies are in place, protection of users as well as registrants implies that any new TLD is forever and must be maintained even if those to whom it is originally allocated lose interest. If TLDs are granted on a fixed policy basis, such as an entitlement for existing ccTLD administrators, even the policy is probably forever: it would be unreasonable to give additional domains to countries who ask for them early on but deny them to those who are later to adopt the needed technology. Some of the key issues with this "one domain entitlement" approach are:

- Some countries have multiple official languages and are constitutionally prohibited from favoring any one over the others. A "one extra domain for the national language" approach effectively blocks those countries from participating and is hence massively discriminatory. Conversely, given each country with multiple national languages as many domains as it has languages could easily overload the DNS root and is likely to be considered discriminatory by those countries with fewer languages.

- Most serious studies of the stable and secure operation of the DNS root have concluded that, if new TLDs are to be added, they should be added in relatively small increments. While many people have predicted that few countries would, in practice, decide to operate additional domain names, matters of national pride as well as economics may predict otherwise. Without some limiting mechanism (at least with regard to rate), we could see 200+ domains (or many times that if the number were controlled by the number of national languages) added within a small number of years, creating a stability threat.

- Since ISO 3166 country codes are codes, rather than strings in any particular language, if TLDs are to be created for the national names of countries, any country could reasonably request one. In principle, a TLD named ".UnitedKingdomOfGreatBritianAndNorthernIreland" (in English) is no more or less reasonable as ".PeoplesRepublicOfChina" (in Chinese).

- ICANN and many national bodies have taken the position that competition should be encouraged at the top levels of the DNS. Automatically giving new domains to current TLD administrators would seem to contradict this principle; it would be consistent with it to bar existing ccTLD administrators from acquiring control of any new country-based domain(s). If ICANN were to adopt a plan about additional country-code domains, that plan should include working out appropriate competition policies on a country-by-country basis, rather than automatically giving new domains to existing operators.

As suggested above, the conclusion of the original President's IDN Committee, after examining the issues above and some others, was that IDNs should be approved as TLDs only if they went through the normal TLD application process, without any inherent linkage to existing domains. I believe that conclusion is still the correct one.

What Problem is Being Solved

Any examination of top-level IDNs should be clear about the problems that are being solved. If new gTLDs are to be created, there should be no intrinsic reason why a name based on English or Roman-character acronym should be preferred to one based in any other language or script. However, if there are proposals to create domains, including non-ASCII ones, to make things easier for users, ICANN needs to remember that "ease for users" is a human interface issue and only secondarily a protocol one. Users rarely use domain names, but instead use email addresses, URIs[8], and perhaps now the more general IRIs[9]. Making a fully-qualified domain name homogeneous with regard to

[8] The syntax for URLs and the more general URIs is defined in RFC 3986, "Uniform Resource Identifier (URI): Generic Syntax". T. Berners-Lee, R. Fielding, L. Masinter. January 2005. ftp://ftp.rfc-editor.org/in-notes/rfc3986.txt

[9] RFC 3987, "Internationalized Resource Identifiers (IRIs)". M. Duerst, M. Suignard. January 2005. ftp://ftp.rfc-editor.org/in-notes/rfc3986.txt

language and script does not impact the protocol elements and syntax delimiters of a URI. For example, no amount of effort in IDNs will eliminate the http:// and other syntax components of a URL. Writing an entire domain name in Arabic will not inherently resolve the question of whether the labels in that domain name run from right to left (with the root at the left), although the characters in each label certainly will, nor will it eliminate the need for ASCII characters and left-to-right order in a URL in which the domain name is embedded.

Each of these problems can be, and probably should be, resolved in user interfaces that are properly localized and designed to be culturally sensitive to the local environment. The requirement that the protocol identifier and syntax delimiters of an IRI be in ASCII characters when the IRI is used "on the wire" should not influence the user interface, especially when that is culturally uncomfortable. Even the resolution of a domain name which has one or more internationalized elements in the DNS in punycode into local characters is a user interface issue – no end user, anywhere in the world, is going to be happy to look at punycode although some will be less happy about the Roman characters and others will be unhappy about incomprehensibility even given those characters. Instead, all of those elements should be written, and presented to the user, in a form that is locally appropriate and convenient, with software translating to the canonical URI or IRI or email address formats only at the interface to the protocols themselves. But, if that sort of user interface translation is applied to deal with the protocol elements of URIs, IRIs, or email addresses, then translation of the top-level domain names themselves may be more appropriate than attempts at simple substitution of an IDN TLD. If nothing else, such translation makes it possible to express the name of any country or other domain of interest in the local language and script, not just the language of some host country[10]. If, instead, the goal is to accomplish localization to the local language and culture by changing domain names and protocol identifiers alone, the result, if successful, would ultimately fragment the Internet: so far, we have no Polish-specific version of the electronic mail transport protocols, no Chinese-specific hypertext transfer protocol, and no Arabic-specific version of the DNS protocols themselves. Should such protocols develop, the almost inevitable net effect would be to limit those protocols to those language communities and to those people who are equipped with the corresponding software, dividing the network along those lines and making network use nearly impossible for travelers to or from the relevant countries.

---

[10] Part of this approach is discussed in more detail in ISOC Member Briefing #18: "Internationalizing Top-Level Domain Names: Another Look", http://www.isoc.org/briefings/018/briefing18.pdf and, from a somewhat more recent and technical point of view, in RFC 4185, "National and Local Characters for DNS Top Level Domain (TLD) Names". J. Klensin. October 2005, ftp://ftp.rfc-editor.org/in-notes/rfc4185.txt