ICANN VIP Project: Overview

John C Klensin[1]

Reviews of the various Variant Issues Project team reports makes it obvious that they raise or identify high-level issues that several (sometimes all) of them have in common. Rather than repeating comments about those issues for each applicable report and doing so without the another script or scripts to be sufficiently out of scope that such comments were avoided even when they understood their relevance. Others of these comments actually summarize a conclusion from the collection of reports that the Variant Information Project, as framed and delegated to the teams, may have been misconceived or structured to produce answers to the wrong questions. I'd like to assume at this point that the project was worthwhile because that conclusion could be reached only one the team reports were in hand. While there were hypotheses about some of these conclusions in advance, they were much less conclusive than is possible today.

Some of the comments here, especially in the first group, are, I believe, important to understanding the role and implications of label variants – especially variants for "alphabetic" or "alpha-phonetic" scripts – in the top level ("root zone") of the DNS. It is worth noting that it is often easy to say "we need…", "use of my script and language would be less appropriate without…", or even "my view of a business case for the TLD I would like to have would be much more profitable if…". It is much more difficult to do the analysis of who or what would be hurt if those requests are satisfied, if only because doing so requires much broader analysis of multiple languages, scripts, the IDNA mechanisms, the DNS and it inherent limitations, and the effects on end users who see the DNS as an identifier and navigation tool and then evaluate and balance all of the tradeoffs involved.

The charge to the VIP teams can be read as asking for exactly those types of easy statements and requirements, rather than requiring deeper analysis and discussion of tradeoffs. In most areas, most of the teams are to be complemented on being very careful to consider the broader issues –at least those within what they perceived as their scope—and avoiding producing a fantasy wishlist.

[1] This review was prepared at the request of the ICANN Variant Information Project Team and partially supported by ICANN. It reflects the author's personal views and may not reflect the views of ICANN staff, the members of the VIP teams, or other personnel associated with ICANN. Sections of it draw heavily on other work by the author that bears on the issues discussed.

Very High Level Issues (but still about the DNS)

As I trust everyone knows by now, the DNS is an exact-match lookup system (the technical term is “known item search”) in which one must know exactly what one is looking for to find it. There are no options for automatic spelling correction, near-matches or matches to the most similar string that is already present in a particular zone (or “domain”). It, especially in combination with a general design style for Internet application protocols that goes back to the 1970s, makes it a poor match for anything involving natural language, even when the relevant natural language was restricted to the subset of English that can be written in the ASCII character set. While it is possible to “fake” some relationships using aliases (i.e., CNAME and DNAME records or different domains names with the same values in their data records), the basic architecture of the DNS -- including the nature of loose synchronization, the need for many applications to know the names that will be used to refer to them, and the difficulty of identifying all of the possible variations on a name that might occur to a user -- is hostile to trying to create equivalences among names by listing all of those variations in the DNS. Other things need to know their own names as well: there are complex relationships between aliases or alternate names in the DNS and efforts to use DNS names as identifiers, e.g., in digital certificates. There are good ways to do those things; the DNS just isn’t good at them and nothing that can be added to the current design will change that fundamental fact. That doesn’t mean that there may not be “good enough” approximations that will work adequately for some cases. But they are essentially tricks. One of many problems with those tricks it that end users will never fully understand them and will, instead, end up with expectations about what should work that will, in turn, create surprising anomalies when they don’t.

As is usual with IDNs, they don’t change that situation or introduce fundamentally new problems. They do, however, magnify them and expand their scope in many different directions. The problems become especially pronounced with languages whose orthographies (as used by people, not just in rules and dictionaries) are not rigidly consistent and with scripts whose Unicode representations do not seem completely natural and predictable to those who have to implement and use those representations.

The result is that every “variant” and variant implementation option -- other than simply blocking registration and entry into the DNS of strings that might somehow be perceived as similar or equivalent to a registered name -- has costs to the perceived predictability and usability, and often to the security and stability of the DNS as seen from the perspective of users and applications as well as advantages in terms of making names easier to guess by treating many similar ones as related or equivalent. Blocking of strings may prevent users from finding what they might be looking for or guessing at, but it prevents errors and conflicts and expedites getting a user who won't find the correct entry by guessing at DNS names to a different and more effective locational or navigational method.

To the extent to which trying to provide more aliases or equivalent domain names doesn't fully do the job of satisfying user expectations based on natural language and other experience, trying may reinforce and accelerate some existing trends. Many users of the Internet today don't try to remember or guess at DNS names or complex URI-type identifiers. Instead, they use search engines or other tools that are good at precisely the things the DNS is poor at – spelling variations, guesses from context about what the user wanted, finding URIs and content rather than host names, and so on. That trend parallels another one that sometimes uses similar techniques – identification of Internet resources by speech to text methods, i.e., talking to computers or other interface devices rather than typing exactly-spelled names. While the questions of who benefits from those approaches and whether the approaches are desirable or not depends on one's perspective, there is no question about the trends themselves and little question that they make trying to fine-tune variations on DNS names less relevant to the user because what those techniques need from the DNS is a collection of stable and unambiguous references. A large number of alternate names might actually make their work more difficult by increasing the perceived size of the name-vocabulary with which the need to deal.

It is necessary to ask "what pain or risks will be caused by adding alternate names" and not just "could alternate names help match some special properties of this language or script". That distinction is fairly well developed in some of the reports but not in others. A similar distinction (again made more clearly in some of the reports than in others) is that something that might be a good practice within a enterprise domain may be less appropriate in a second-level one and inappropriate in the root. Among other things, the latter inherently has less context for actual names of hosts and other systems available than the second and third level names (and beyond) do.[2]

Even after these reports, there is no agreement about what the term "variant" means. The usage in the original JET document that defined the term (RFC 3743) is reflected only in the report on the Chinese script. The hope that this project will clarify the usage of that term within ICANN remains in the category of future work and consensus. If that consensus cannot be achived, I believe that ICANN should prohibit the use of the term as hopelessly confused and confusing unless it is carefully qualified in each instance of us. I do have one specific suggestion, which is that all notions of treating label strings in some special way because of visual confusion issues be treated separately from variant issues. However "variant " ends up being defined, it would be wise to try to confine its use to relationships that result from inherent properties of writing systems, scripts, orthography, or encoding design decisions. By

[2] This issue is discussed in more detail from a specific perspective in my Latin review.

contrast, “visual confusion” is always largely a matter of perception and often depends on rendering and typeface decisions as well.

In summary, I would encourage ICANN to use great care in permitting entitlement to multiple names in the root, independent of whether those labels are delegated as TLDs or handled through some aliasing mechanism. It may be safe to do so when a given string can have, at most, exactly one alternative to be considered as a related name and sufficient administrative precautions can be guaranteed to prevent causing, rather than reducing, confusion. For the other cases, ICANN should consider whether the best interests of the Internet call for non-DNS mechanisms or helping users adapt to one (and only one) way of doing things with no “equivalences”, just as users have adapted to lack of DNS-based equivalences in the ASCII/LDH world[3]

What Characters are in a Script?

Several of the reports point out that adequate use of a script requires that one or more characters from what Unicode characterizes as the “Common” and/or “Inherited” scripts be used along with characters identified with the relevant scripts. Permitting these characters violates the existing ICANN principle prohibiting labels that contain characters from more than one script. Unlimited use of “Common” or “Inherited” characters together with a particular script could easily lead to other problems that none of the teams have addressed. I believe that the only rational approach to the problem will require ICANN tables of valid characters associated with each script. Those tables would exclude characters that Unicode associates with the script that are DISALLOWED by IDNA but include Common or Inherited characters that were necessary for use with the script. ICANN would have to study whether simply including those characters as part of the script would be adequate or whether they would need special treatment similar for CONTEXTO or CONTEXTJ characters within IDNA2008. They would presumably be supersets of any language-based tables for the relevant script. Unfortunately, such tables would require examination and possible updating for each version of Unicode: while changes would be unlikely, they might occur.

[3] The oft-cited case of ASCII lower and upper case equivalency are discussed in my Latin review. It may be worth noting that many DNS experts, having examined the implications of case distinctions for non-ASCII strings but case insensitivity (not equivalence) for ASCII, have come to the conclusion that the later was a mistake. Not a mistake that can be corrected at this late date, but one that we should attempt to propagate.

Specific Issues and Further Study

Some of the reports represent an impressive level of scholarship and tutorial quality about issues with the relevant languages and scripts. Those same reports may be less good about how those languages and scripts interact with the inherent properties of the DNS and/or with subtle Unicode characteristics that we have to accept and work with. Drawing all of those threads together into a complete understanding may still be ahead of us. Some of the reports have responded to those, and other, issues by identifying topics, even including topics that bear on issue identification, as needing future study. That has historically been acceptable in ICANN (although a cynic might describe it as "if you cannot agree, form another committee"). But, in this case, these teams may be the best and moist expert groups ICANN can find to address the issues. If they have not been able to find definitive answers or even definitive agreement on what the real issues are, it is necessary to wonder whether other groups will be able to do better and why or whether more study and committees are likely to merely create further delays or further obscure the issues. I believe it would be helpful to focus the issues by having the ICANN community respond to every suggestion about more work by asking if it is appropriate to block any IDN TLD applications for that script until the issue can be resolved. I believe that might provide the necessary focus and encouragement for getting consensus on answers where they are available, reaching agreement that some issues are not important enough to affect variant (or other) choices, and evolving appropriate strategies for the other cases.

An Even Higher Level Issue

Returning to the basic lack of suitability of the DNS for some of these problems, there is an old saying that, to someone whose only tool is a hammer, everything looks like a nail. In the ICANN community, it is extremely tempting to assume that the solution to all problems with names, naming, and matching issues should be solved in the DNS. In addition to familiarity, there are some advantages to the "just use the DNS" approach: it is there, the administrative models (e.g., registries and registrar relationships) are in place, and efforts like this Project show that we have mechanisms for exploring and making policies. Opinions differ about the typical effectiveness of those mechanisms, but this Project and these reports are almost certainly among the better examples. Blocking and reservation approaches aside and to a greater or lesser extent, alias and multiple delegation approaches stress the DNS design at the same time that they are unlikely to provide a truly satisfactory user experience. For example, if we could redesign the DNS's lookup and matching algorithms, the Greek Tonos problem and other cases in which accents or character decoration are semi-optional but, if present, must be correct, could perhaps be solved by entering (registering) only the Tonos-containing form into the database but permitting an undecorated lookup to match it. More extremely (but also more practical than trying to design and deploy a major change to the DNS), one could think about whether language-sensitive processing in the user interface prior to lookup might provide a better user experience than trying to enter all of the possible user-perceived forms of a label into the DNS.

It is probably worth keeping in mind that the issues outlined in these various reports can be responded to in a variety of different ways in addition to "get used to it" and "try to simulate a solution by using the DNS" (neither of which is really just a single option). With that in mind, it would be useful to examine each issue or requirement by asking questions like "can a DNS-based mechanism actually provide an adequately effective solution for this issue?", "if DNS-based mechanisms are used but require five or ten alias names or delegations per base label, could that result in enough of a root name explosion to stress the safe limits suggested by the DNS Scaling report?", and "if other requirements – requirements not specific to domain names – imply specialized processing in the user interface anyway, could more be accomplished for a good user experience by doing some small additional work in the user interface processing and relying less on the DNS?".

The Arabic Script report may provide an example of, and interact with, a place where that last question becomes particularly important. From the standpoint of a rational and predictable user experience, the many important and complex issues raised by that report are likely to be completely dominated by one that lies far outside the team's scope: entering and rendering IRIs (the internationalized forms of URIs and URLs) that contain a mix of ASCII protocol identifiers and possibly keywords, delimiters with either left-to-right or direction-neutral properties, and parameter values (including domain names) that are likely to be in Arabic characters in whole or in part. The Unicode algorithms for handling bidirectional text ("Bidi") handles right-to-left and bidirectional running text fairly well but is not optimized (or really satisfactory) for complex identifiers, like IRIs, that are very different from natural language. Perhaps we need processing and rendering rules that are specialized to IRIs (and their domain name subsets). Such rules would require the ability to identify IRIs and domain names embedded in running text. Doing that identification accurately and consistently almost certainly requires reviewing the decision (made long before non-Latin Script strings were considered) to not require delimiters around URLs (and later URIs).

However, if one can identify IRIs -- no matter what context they appear in – and process and render them in a way that makes good and predictable sense to the user – could the required processing mechanisms be used to provide conversion to a canonical form during parsing and lookup? That conversion that might provide a much more satisfactory solution to many of the orthographic variations, coding and rendering differences between [Western] Arabic and

[Eastern] Perso-Arabic, and so on, rather than trying to cover over those differences by creating "variant" aliases in the DNS. That approach might also be effective for some of the issues that the Greek report tries to identify and some of the uncertainties noted in the Cyrillic report. It would almost certainly be unsatisfactory for many of the Chinese script issues because those issues appear to actually require separate DNS delegations.

In the interest of satisfactory user experiences and the stability and predictability of the DNS, I hope that, as ICANN and the broader community consider how to respond to the issues and requirements identified in these reports, the consideration process does not start and end with the assumption that the only possible solutions lie in either forcing something onto the DNS (including tricking the DNS into doing something that isn't a natural part of its design) or on the user. There are roles for both of those approaches, but there are also other options.