Label Generation Ruleset Process

Working Group Meeting 2012-08-29

Table of Contents

1. Introduction

Kurt Pritz welcomed attendees to the meeting, and thanked everyone for working on the label generation ruleset. ICANN team members are available to help support the team, so please ask if you have anything you need to succeed in the work.

Tour-de-table, introducing meeting attendees: Nicoleta Munteanu, Francisco Aria, Dennis Change, Mirjana Tasic, Michael Everson, Fahd Batayneh, Zhang Zhoucai, Panagiotis Papaspiliopoulos, Daniel Kalchev, Edmon Chong, Yoshiro Yoneya, Dennis Jennings, Andrew Sullivan, Chris Dillon, Sarman Hussein, Naela Sarras, Akshat Joshi, Neha Gupta, James Seng, Steve Sheng, Joseph Yee, Kim Davies, Kurt Pritz and Asmus Freytag. Remote attendees are Alexei Sozonov, Will Shorter, Vaggelis Segredakis [audio difficulties], Alireza Saleh, Raymond Doctor, Vladimir Shadrunov, Linlin Zhou, Rinalia Abdul Rahim, and Iftakhar Shah.

2. Agenda Review

Andrew Sullivan said the basic intent is to walk through the document. Today focus on how the document is designed, and tomorrow discuss the process itself.

3. Overall document structure

Andrew Sullivan: Four parts. Introduction, although the document assumes prior understanding in the document, so doesn't reintroduce the subject matter. Document discusses the proposed methodology, then tries to tie it back to the original principles discussed in the beginning — the IAB principles, and the parameters from the integrated issues report.

(No comment on the document structure.)

Michael Everson said there are errors in the integrated issues report, particularly in Cyrillic. Said that the examples were not clear, and “þ” and “thorn” should be used in examples. Andrew Sullivan explained the rationale behind the examples, although said the examples are light and would appreciate contributions of illustrative examples. The choice of examples was mindful that single characters are not allowed as actual TLDs. Michael Everson suggested “c” and “ç” may be useful. Francisco Arias also noted translations of “example” and “test” are current reserved against use.

4. Discussion of founding premise

Andrew Sullivan: Mnemonics are useful in the root, however that doesn't mean that all possible language words and strings need to be representable in the root zone.

Michael Everson: There are languages spoken by many, and some by few. Is this a concern at this point (if a language group can't represent any words)?

Andrew Sullivan: Yes, it's a problem. There is a tradeoff between accommodating everyone and minimising risk to others, and this is a problem unique to the root zone due to its need to work for everyone.

Asmus Freytag: It would be useful to set as a premise to set the level of utility you would like to have, if you could. For example, you could set a limit on only vowels, that would allow mnemonics, but would be problematic. If the principle of the premise can be made clear, with the stability of the Internet being foreshadowed. Otherwise the principles don't let you evaluate multiple hypothetical proposals. For example, you could create a Cyrillic subset that has no overlap with Latin, it would not be very useful, but it wouldn't violate the current principles.

Daniel Kalchev: There are so many different languages and variations, some used by very small groups of people. Need to make a clear decision whether to let those languages have their own alphabets implemented, or we go for a restricted set, like the choice to limit to ASCII in the past. We have a precedent with “.қаз” (fast track string for Kazakhstan) in the root zone, where the “қ” is not the standard Cyrillic “k”.

Andrew Sullivan: Referring to the ASCII limits in the past, is at best stretched. That was due to the technical limitations of the time, not a conscious choice.

Panagiotis Papaspiliopoulos: Previous work should be taken into account. We need to include as many people as we can, and [facilitate] them as best as possible.

Edmon Chung: Bear in mind when discussing grandfathering, is the new gTLD first round will likely be done before this work is implemented, and all the work needs to be grandfathered. Also noted the potential need to discuss the principles relating to the code point repertoire chosen, and whether that can generate useful strings. Perhaps include the concept that the repertoire produced needs to produce strings of useful mnemonic value.

Asmus Freytag: Unicode has a concept of "usability", but that is only used in the negative sense relating to security. It is not used in the positive sense. Need to recapture that label, give it a different sense, in addition to existing definitions. To do a cost-benefit analysis, there needs to be something on the benefit side to make the assessment.

Andrew Sullivan: This process has been bedeviled by the re-use of terms with slightly different meanings, depending on who says it. Concerned about using existing terms, that is how we got to the "variant" problem in the first place. "Variant" had a very specific meaning originally, but has accreted many new meanings over time.

Asmus Freytag: Create a new principle, call it "utility".

Andrew Sullivan: Restarting the starting premise: It would be useful to add IDN variants to the root, we need generation rules to do that, and those rules will govern permissible U-labels (and therefore A-labels) in the root. This doesn't mean every word needs to be in the root for it to be usable. Need labels that are at the same time safe, and natural [to use]. They needs to be usefully mnemonic to people. As it is not useful to use Latin if your native script is Arabic, it is similarly not useful to limit to only vowels, etc. Also had the suggestion that aim to cover as many people as possible — so rules that tend to exclude large groups of people, or as defined would exclude groups of people in the future. For example, you could create rules for Han and not solve the rules for Arabic or Cyrillic, but that reason that is acceptable is rules for Han don't impact the rules for Arabic or Cyrillic. How do we capture that in a simple one sentence phrase?

5. The IAB Principles

Andrew Sullivan: Internet Drafts can be written by anyone. This draft[1] happens to be adopted by the Internet Architecture Board (IAB). Typically the IAB does not adopt "rank nonsense", so it is safe to say this is the direction the IAB is going with the draft. However it is not adopted as an RFC, so it may change prior to publication. If there are comments on it, they should go the to the IAB to potentially influence its content. For example, Asmus' comment regarding utility may be strained and could be improved would be good feedback.

It is, however, a good starting point. RFC 1123[2] was part of the genesis of the document, which describes rules for domain names. Related the 3com story, that the rule for domain names was extended to allow them to begin with numbers. This is described to be safe, as in a footnote the document says that TLDs will be "alphabetic". There is a debate about whether that is a side comment or a statement of formal policy, particularly as it is written by Jon Postel who ran the IANA at the time. It is important because software often distinguishes between domain names and IP addresses by checking for alphabetic characters in a string.

[1] http://tools.ietf.org/html/draft-iab-dns-zone-codepoint-pples

[2] http://tools.ietf.org/html/rfc1123

In the IETF community, no-one could move things ahead due to differing opinions on RFC 1123, which lead to this IAB meditation on what is safe in labels.

Sarmad Hussein: The IAB document does not list the first two principles — longevity and usability. *[Section 3.1 and 3.2 of the draft] [Dennis Jennings: Noted, will come back to that.]*

5.1. Longevity Principle

Andrew Sullivan: The document is divided into principle applicable to all zones, those applicable to all public zones, and those specific to the root zone.

The first principle is the "Longevity Principle". The idea is there are several different versions of Unicode is use, and we need to work with all the different versions out there. That is a tall order given the number of systems active on the Internet. We know, for example, Windows 98 is active on the Internet today.

Michael Everson: You have to be careful about it — there are new characters that did not exist in Windows 98 time. It would be impossible to support strings based on the older encoding model for Burma, for example. An exception would have to be made, because it should not be required to support the previous version.

Andrew Sullivan: There are other examples too. Needs to develop a procedure that doesn't lead to disastrous rules. If we cleave to hard to some of the principles, we will run into situations like Burma in Windows 98. These compatibility discussions are active in the IETF.

Asmus Freytag: The way I read the language, as it is, not knowing what was in the author's mind — longevity means avoiding relying on characters that are potentially impacted by the Burmese issue. When Korean was new, it went through a gigantic model change, due to political upheaval in the country. I understand it to be, clearly phrased, choose nott o process something that can be yanked out from under you. If it has a track record, it is a safe stepping stone that takes you where you need to go. That is an entirely different argument to ancient software considerations.

Andrew Sullivan: The reason for the principle is because we have all the stuff out there.

Asmus Freytag: As written, though, the IAB document does not lead anybody there. For that to feed into the document, we need to elevate it somehow, as it is not in the IAB's words.

Dennis Jennings: Agree with Asmus, I read it as to stability of the code points. Think there is also an exercise in deciding how "stable" is defined. I read it as stable with respect to multiple versions of Unicode. But who decides what is stable? Those questions are not addressed — they are process questions that need to be addressed.

Asmus Freytag: The likely relevant criteria relate to the Unicode properties, which formally define how the basic fundamental properties relating to the code point is to be used. If they haven't been messed with it is a sign that it is stable. The properties for ASCII characters have not been altered, even though descriptive elements may have been added. Would be happy if there was a side note that alluded to the essential identity of what is being encoded as being stable. [Dennis Jennings: Agree.]

Edmon Chung: Why doesn't the other principles combine to address this principle?

Andrew Sullivan: All the principles are linked — they are not rules. They are not designed to be rules to be plugged into an algorithm, but rather consideration when developing your policy. The reason longevity is called out specifically, is the root zone needs to work for a wide array of deployed systems. If we could be sure everyone is using the same version of Unicode, it would be simpler. But that won't happen. How big is the community that needs to be supported? In the DNS, we can't talk to the consumers of the data, we don't know who they are. That is the reason this is called out, and emphasised separately.

Asmus Freytag: As I interpret it, it is a slightly different reason. My understanding is it reflects a large increase in the repertoire. You don't want identifiers based on any single version of Unicode, and shift them around for the next version. You then have to throw out all the old labels. When you increase the repertoire, you go from ASCII which has a long history and stability. Now you need a principle which is the boundary where you may end up in a morass. The longevity principle tells us to stay away from those groups until the dust has settled. Regarding the multiplicity of principles, each one hones on a specific aspects. It allows you to analyse where they run into each other. I am very fine with that use of principles, the development of the Unicode happened the same way. Unicode was based on a number of sometimes contradictory principles, and was developed trying to find the sweet spot between them.

Andrew Sullivan: Relating to the fact you can't "undo a TLD". [Asmus: That's the stability principle.] In a leaf zone, its very easy to get rid of labels — at a zone I run at home, for things inside my house, I can take a name offline and it is no big deal. If you do that in the root zone, entire trees disappear and that's a big deal. We can't get rid of these labels. It is another reason that longevity is here. I don't think we are in disagreement, but I accept that these are narrow boiled-down bullets of whats going on there. If more background is necessary, then maybe it needs to be captured. I am very close to these documents, so I know the background discussion that went into them. Sometimes I have forgotten not everyone knows the background discussion.

James Seng: On the topic of longevity, we need to bear in mind this document does not exist in isolation, it is based on IDNA2008, which is based on Unicode 5.2. The longevity principle, in a way, violates this. You can't say it needs to be compatible with Unicode 3.2, when the underlying protocol needs 5.2 or higher. It is an unfortunate consequence that we have to deal with, but it may not be addressed by the manner described in the document.

Daniel Kalchev: My question to Andrew is "What will happen if we decide on the certain components without present knowledge of the Unicode and IDNA at the current state, and in a few years a new version of Unicode or IDNA emerges that makes those code points obsolete?" We can not remove them from the root zone. We have a problem I think.

Andrew Sullivan: The short answer to the question — If that happens, we have bigger problems than this. If the Unicode Consortium tomorrow changed the properties of "n", we have big problems. As Asmus noted, we are building upon a presumption about how the Unicode Consortium will act.

Dennis Jennings: These questions are not discussed in complete isolation, and the work here will also feed back into the Unicode Consortium etc. Or does this happen in complete isolation?

Andrew Sullivan: One triggering event was a code point moving from PVALID to disallowed [from 5.2 to 6.0?]. [Michael Everson: Need to know what it is. Andrew: Will dig it up and

provide it. It was an extremely obscure codepoint.] One IETF member was very exercised about this as they were promised it would never happen, and it did.

— Morning break —

Vaggelis Segredakis: I feel the Conservatism Principle is very questionable. The paragraph: "Public zones are, by definition, zones that are shared by different groups of people. Therefore, any decision to permit a code point in a public zone (including the root) should be as conservative as practicable. Doubts should always be resolved in favor of rejecting a code point for inclusion rather than in favor of including it, in order to minimize risk." presents the question what happens with characters like omikron, latin o and cyrillic o. Will they be excluded on principle because they might get confused?

Panagiotis Papaspiliopoulos: On the simplicity principle, there is the final sigma, a character we only have in Greek.

Dennis Jennings: We haven't gotten to those principles yet — we are only dealing with the longevity principle so far. Note those comments for later discussion.

Andrew Sullivan: I note people's uncomfortability with the term longevity, and I am worried about instability of the terms. We should possibly provide feedback to the IAB on these issues.

Asmus Freytag: In direct response to this, the majority of the principles are probably fine with a bit of wordsmithing, and are reasonable stated in the IAB draft. We just have to be sure to use them the way they are stated. Am happy to support the fact that they are arbitrary labels, and we will stick to them. We have to be careful when we use "stable" in the English sense, not in the "stability principle" sense. Another comment is that the IAB statement doesn't provide guidance on how to evaluate against the longevity principle. As an appendix, it would be useful to provide guidelines on how to practically evaluate against the principles, using copious examples. We can never get to rules for rule making if it doesn't work. Incidentally, that is precisely the methodology for creating the character encoding standard. An appendix like that should be on our list of work items.

Andrew Sullivan: I wonder if that should go in an Appendix, or in a section ...

Asmus Freytag: It is very detailed, and the details should be sequestered, but it should be referenced.

Edmon Chung: Dennis — I keep hearing Andrew punting to the IAB document. It is a draft and this group could influence the draft, so I don't want to stop all the time and direct it to the IAB. We should have discussion about it. Would that be more fruitful than waiting for the IAB before coming back.

Andrew Sullivan: The worry that I have, is to not use the same term as the IAB document, and use it differently. If we are unhappy with one of these things, or the principle as such, it is no problem discussing it. I just want to use different names. The difficulty is we get to a term, you ask 10 people what it means, and you get 15 answer back. It is impossible to converge on that. We either need to use the IAB's term in the exact same sense, or we use a different term. We shouldn't nuance the IAB discussion, that is the reason I am pushing back.

Asmus Freytag: One thing to remember is these are principles. Principles tend to be abstract, and hard to pin down, and often try as you may, you often can't satisfy all of them. In searching for a solution to measure how closely to apply all of them. There is always, in these principals, an overriding one. In this sense, it is conservatism. In this case, if you look through a solution that doesn't meet the requirements, you throw it out. In Unicode it is the opposite, the overriding principal is [inclusion] when there is doubt.

5.2. Usability Principle

Andrew Sullivan: The next principal is the Usability Principle. This is intended to be a minimal meaning of usable. It seems to me from the discussion so far there could be a principle of "Utility", but it is not the same as this.

Asmus Freytag: I wonder in this case, as Usability is such an odd principle, try to find another term and then explicitly acknowledge "we use this term as it is defined by the IAB". [Andrew Sullivan: A suggestion for a word?] Michael suggested "Practical", also "Non-unusable" or "Non-confusable"? Just throwing words out to point out the direction. A simple sentence like "This name of this principle is a misnomer, what is meant is anti-abusable…" [sic?]

James Seng: Suggest "reliable" as a possible term.

Akshat Joshi: [inaudible]

Sarmad Hussein: All the proposals seems to go a single direction, to constrain the set of allowable code points. To have a healthy balance there should be principles that pull in the direction of more expression for each language, so that figures into the discussion.

Asmus Freytag: In principle, there is no problem with principles going in one direction i.e. restricting. You need to have a counterbalance that brings you in the direction that when you bring something to the table, it is evaluated against the restrictions. There is text in the document that talks of the goal of the document, to cover communities etc. If you stick to the goal, with a few parameters like utility, reviewed against the conservative principles, can work.

Sarmad Hussein: I think what I am suggesting, is the goal is OK, but perhaps sink it down to the principle level as well. Some way to balance it. I am perfectly fine with conservatism as the "final" principle. Looking for an outcome that tries to get as many strings on the table as possible. The principles are not representing the demand side.

Andrew Sullivan: The overall section is called "Principles to constrain the label generation rules". That is not an accident. I don't believe we are wanting for people that are not going to ask for things to be in the zone. We've already seen a number of ideas that are emphatically bad, so I am not concerned that we will lack those inputs. What we need is a mechanism that explains why we are going to limit the decision. I know the IAB is very concerned about that. We know there are people that are very anxious to add things to the root zone that are done without an appreciation for the consequences for people outside their linguistic context. Im perfectly OK with the suggestion that we add in an earlier section an expanded version of the starting premise, that we're trying to include as many language communities as we can. But I think these principles really are an attempt to narrow down.

Asmus Freytag: I have a brief suggestion on what it might look like. A couple of characteristics, independent of security: 1. mnemonics should have a certain utility to them; 2. it should be comprehensive, so has not to exclude a certain script community; 3. it should not be arbitrary

in the sense that security concerns are evaluated more tightly for one community over another; 4. and it should not be biased in the sense that you only care about communities more than 100,000 or whatever. Those four characteristics can help define the goal for the project, and it provides a framework for the project that allows you to distinguish between various different bad ideas. Those four points should be somewhere near the premise section.

Sarmad Hussein: Talking about longevity principle, another issue came to mind. In documenting some of the languages in Pakistan, [inaudible]. We did some work encoding languages, and there is an urgency to get people to start using those characters. The work we did for dot-pakistan for example, the label generation ruleset, includes those characters precisely to go against the longevity principle. Some of the characters are from Unicode version 6.0. I want to put on the record there are cases where longevity [inaudible].

Andrew Sullivan: I take the point that there are often other socially-desirable goals that may go against the principle. Do you really think those other socially desirable goals belong in the root zone, or should be further down the tree?

Sarmad Husseum: I'm talking generally about all zones.

Dennis Jennings: Some comments in the Adobe Connect to catch up on from Raymond Doctor.

Chris Dillon: We should be looking for principles that are overlapping. Specifically the two principles that are ringing an alarm-bell, are the longevity and stability principles. There may be a case where there is a degree over overlap where we may want to merge the principles.

Andrew Sullivan: The stability argument is about labels, not code points. Once you have a label delegated it is very hard to remove it, take a look at the Soviet Union label. That is different from the other principles about how you go about this. This is a reminder about the consequences of what you have done.

Asmus Freytag: I think it applies to the repertoires as well as labels — once there is something in the repertoire it is hard to remove. ... There is a weaker form of the conservatism principle, [about how to resolve issues]. I have no problem with principles having fuzzy edges, because they are guidelines to inform thinking on issues, they are not rules. Principles that are guideposts can work, even if they are not super-distinct from each other. I think the IAB draft is pretty well thought out, they are not the product of "monkeys at a typewriter".

Michael Everson: [Question about the distinctipn between the two panels]

Asmus Freytag: The primary panels will be closer to a specific community, the secondary panel is charged with maintaining the integrity of the whole system as its primary function, and it will maintain the conservatism principle with more "oomph".

Dennis Jennings: I see the primary panels as more on the demand side [...]

Edmon Chung: We were talking about the utility principle, how do we come to this?

Andrew Sullivan: A.3.1 needs to be expanded quite a bit, that is text that needs to written, not around the table but the next draft can contain a first attempt at that.

Edmon Chung: So we are not debating it as principle, but in the premise.

Andrew Sullivan: Yes, I want to keep the principles as constraints, and the utility principle is an expansion.

Daniel Kalchev: Should take care to incorporate the [4 characteristics Asmus gave], and where they fit, as the comments are very appropriate. As mentioned earlier, it is better to not overload existing terms when we can use more exact terms to refer to what we are trying to avoid or implement.

Dennis Jennings: What I am hearing is more expansive definitions of the goals of what we're trying to allow.

Andrew Sullivan: Discussion around the usability principle is where that lives. The point of the principle is that when you permit something it shouldn't cause the users of it grief. That is really what it comes down to. One kind of grief is you are lead astray because you see something and you had another idea in mind. On one of the discussion lists there were suggestions for alternative words. Confusability is not a separate thing, and for this group, my understanding is they are separate from this draft.

Francisco Arias: We have a string similarity process already in use, and there is an intention to modify those, and my belief is we will eventually use some of the tools we use here, like the LGR, but the idea is to separate visual similarity aspect from exchangeable codepoint variants. The best discussion of this we have now is what is in the integrated issues report.

Daniel Kalchev: We could include in the usability principle that we include non-confusability etc. but aren't we overloading it too much?

Andrew Sullivan: The text in the IAB document is on the screen, do not think the way we're using usability is different from the way the IAB is using it. However, it is very different from the ordinary meaning of the word usability. This is partly an artefact of the IETF/IAB being bad at user interface issues. That feedback to the IAB may be useful. We should be as consistent [in use of terms] as possible.

On the point of not ignoring confusability, it is foreshadowing tomorrow's discussion, given the way the panels are constructed, I don't think the outcome is that terms deemed confusable by both panels are going to be allowed. The procedure should automatically lead us to the correct conclusion. As Francisco mentioned, the string similarity component is separate to this. While Cyrillic and Latin may be very similar, there may be context that makes them not-confusable. There may be two strings that look identical, there may be a string similarity panel that rules them the same, or there may be a variant issued that awards them to the same party, I don't know what the rules will be on that.

Asmus Freytag: Explicitly, the usability principle, as stated in the IAB document, the text divided into two halves — one of them, the first two sentences, that is rather hand-waving and confuses two principles. We should ignore that section, as it is not helpful. The third sentence, "Zone administrators should consider whether a candidate code point could be used maliciously…" [is useful]. If you feel Andrew that that needs to be made explicit, then we can have an appendix that tells them, that we have explicitly trimmed that part of it off as it wasn't well drafted. That is a specific suggestion on how to deal with the usability principle.

On confusability, not sure how it is going to play out. We have the bad case between Latin and Cyrillic, and needs to be addressed in the repertoire creation, rather than relying on the similarity panel. I don't think that work [on string similarity] is going to immediately will solve the extreme collision between those two scripts.

Panagiotis Papaspiliopoulos: I beleive the principles are good, but I believe [there needs to be more focus on] more [code points] should be included. Maybe we could start with [inaudible]. We are using the IAB draft as a basis for the document, maybe we could do an appendix as Asmus suggested, or we could do something else.

Dennis Jennings: You made a good point, as stated before, these are principles to constrain but there needs to be goals of inclusiveness, and the document needs to be clearer on that.

Panagiotis Papaspiliopoulos: In the stability principle, in our draft, would cause me trouble. I don't want the document to refer to specific Greek letters. Want to male some basic principles for our work, but how we interpret these principles in our own languages.

Zhang Zhoucai: In our experience in CJK unification, our major lesson was to minimise the exceptions. Too many exceptions will destroy the standard. Some of the principles reflect such lessons — the conservatism principle is good, but it is not enough. How do we minimise the exceptions? The second example, the badge/mask/group inclusion is very difficult. Originally we decided to include all national standard characters, but that made a lot of problems. Each inclusion must be checked individually, one-by-one, specifically. The inclusion principle seems to state that, but it is unclear.

Andrew Sullivan: This idea of minimising exceptions is exactly what the simplicity and predictability principles are intended to capture. Remember that once these rules are in place, there will be software that tries to evaluate these rules to identify if a domain is valid. Just like with software rules that has caused problems for .INFO, .MUSEUM etc. These kind of naive implementations area problem, often implemented in Javascript that doesn't involve a call-back. It needs to be simple enough that a junior programmer can implement it in an afternoon. If it is not that simple, it is a long list of exceptions, and it will be implemented wrong. The consequence of that, and I want to emphasis this, that stuff that is really important for some languages is not going to work in the root zone. We can't do it in a way that a junior programmer can implement it.

Michael Everson: What stuff? We need to know.

Asmus Freytag: We will not really know how hard this bites that when we evaluate it against certain kinds of use cases. But right now we're discussing principles. The unicode standard, I am familiar with, is a complex beast. Implementing it can be simple if standard gives you tables to implement it. Simplicity ought to be understood that if it can be represented as a table with simple driver code, that it would satisfy my definition of simplicity. Rules that are encoded in regular expressions can be more constraining than simple tables and implementation code.

Andrew Sullivan: What we've learned from the deployment of DNS software: We know that, even today, 10 years after a number of TLDs were added to the root zone that have more than 3 characters in them, we have email validation forms that reject them as not being valid domain names. Those kinds of examples are out there, because people found them in a Google search and put them in their code. So if the rules can not be implemented in a trivial amount of Javascript, then the rule is too complex. That is my [personal take] on the test.

Asmus Freytag: When you internationalise anything, and domain names is not an exception, you find that 90% of code that deals with ASCII is no longer valid. Even simple things like determining if a character is a letter, such as in a regular expression. We say it needs to be simple in the context of an internationalised solution. It is going to cause pain, in that you can't do your old ASCII tricks. There is no longer a single range for letters, for example. You need to increase the power of your solution at that level, but we can say there shouldn't be other complex rules. It may be some things are not as complex as they appear initially. I sense some people representing certain communities, that an overly simplistic — not simple by simplistic — approach may rule out certain things. In my characteristics was a caution about not being arbitrary. A simple solution shouldn't be arbitrary.

Fahd Batayneh: Regarding the simplicity principle, in Arabic we have diacritics not permitted in IDNs, but we need those diacritics as they can change meanings. Another issue that crosses my mind is we don't [have acronyms] i.e. you can't write ICANN in Arabic. A third issue is we don't have spaces, so we need to use hyphens when there are more than one word in a label, so that would be an obstacle. I raise these issues as they connect to the simplicity principle. When I write I never use diacritics, because I know based on context what it means, but in some contexts it is needed.

Sarmad Hussein: Mathematically speaking, there are four levels of complexity from the "Chomsky Hierarchy". Inherently, each level is more complex than the other – more rules, context sensitivity etc. Writing systems across the world are not all at the same level, so you may have writing systems that are context-insensitive or context-free, and those more complex. If you had a rule that favoured context-insensitivity it would automatically bias against certain writing systems.

Akshat Joshi: Are we only talking about permissible sets of code points, or the formation of permissible labels?

Andrew Sullivan: We are talking about guiding panels in their deliberations, which will guide what is permitted in the root zone. There are two parts — the codepoint repertoire themselves, and then what consequences that flow from that, the codepoint variant rules. Those are strictly speaking separable components of the label generation rules. WHen you implement, for example, email software it needs to understand the variant generation ruleset to understand what domains an email could come from. All of these principles have different weight depending on what point you're looking at the question.

Akshat Joshi: There could be an API developed that an implementor could use, that could be one of the ways of dealing with this problem. Code samples could be provided for people to put in their web pages.

Dennis Jennings: One reason we don't use IDN tables, but use LGR ruleset, is to maintain a level of abstraction. The IDN table is an implementation.

Joseph Yee: Even though not everyone agrees to the Unicode rules, some implement the rules differently. If it is not simple enough, the tools used to implement IDNA/Unicode are bad enough already. If people ask me for advice on implementing a string in Unicode, I tell them to focus on ASCII. If the rule is not simple enough to write out the program, you'll have problems with the implementations and the adoption of them. Right now, as this is only a principle, further down is a question of code points that are easy for programmers to write. JavaScript already has problems with Unicode strings [regarding normalisation rules]. In open source,

people fork it, which brings more complications. Even HTML 5, they implement an email recognition pattern, and they don't support IDNA — only ASCII.

Raymond Doctor: *[Reference to Chomsky hierarchy and aspects of Indian languages, that the language is comprised of building blocks that are not specific characters, difficult to follow due to poor audio quality]* Request that the principle of simplicity needs to consider scripts where characters are not the basic building block of the language.

Michael Everson: It may be the case that all that read Brahmi-derived scripts, notice the orthographic symbol first, but that's a matter for a reading rule. The process is according to the underlying characters. So this isn't a problem, unless there are certain [strings] that are not representable in some why. I don't see the impact of this, as we are dealing with characters, not with how they are displayed — which is a different level of abstraction.

Raymond Doctor: In the Devanagari report, you should be able to better understand the issue. Indian languages, by definition, fall on the third level of Chomsky's hierarchy, they are not context free, they are context bound. [inaudible]. We need to see a way a given script is visualised by the native user, and for us, we view the syllable as our processing unit and not the character. [...] The way the user perceives the script should be considered.

Akshat Joshi: The reason why the building block came up. The question si whether you are only making a code point repertoire the issues doesn't arise, but when you are thinking of validating a label — you can't do it based solely on code points. In the case of Brahmi-based Indian languages, the validation needs to be done on the basis of syllables. There are instances, in top-level domains, of domains being registered that do not meet syllabic rules but are being registered. These should not be permitted for registered.

Michael Everson: I know what you mean but I need examples.

Asmus Freytag: I would like to take this entire input and basically, lets go through out approach to the principles in the panel procedures, and see if we can come up with a test case that would fail for a particular language or script family, as a concrete litmus test to see whether we are setting up a procedure that is bound to fail something that we were not expecting to fail. We are not at a level yet to solve that issue, we are hearing the concerns, and we should make a mental note that we need to work out how to handle this procedurally. If we start to speculate [on this topic], we are going to way off the discussion of principles.

Sarmad Hussein: Going back to the simplicity principle, inherently some scripts will perhaps have more complex rules than others, and that is not because the rule itself wants to do something more complex, but because the script itself is more complex in its presentation. Simplicity principle should be sensitive to the script it is being applied to. Second thing I see missing, is the constraints are focused on how the character is encoded/represented, but not how it is rendered. Sometimes what is clear in a Unicode table may not be visible to the user, and it is not clear how it is handled in this process.

Andrew Sullivan: On the question on rendering, that is too bad for us — we have no access to the rendering, and no context in the DNS about how it is going to be rendered. If we could control that, we could cut off a large number of problems that we have. We already have this problem with ASCII, so it is not unique to IDNs. We are just out of luck there, there is nothing we can do about it due to the layered nature of the network protocols. What troubles me about the direction of the discussion, is we haven't addressed how people can register things that don't work for a particular language, such that the resulting label is garbage. For instance,

in the case of Brahmi-derived cases, we have examples of people that have registered things that don't adhere to the syllabic rules, and similarly we have examples of things that would cause difficulties in Arabic script; and even before we have examples of diacritics that are, or are not, there depending on the rules. My question is — depending on the rules that could result in labels that are incoherent for the user of the system, does that do any harm? Presumably the result is no-one will register them because they are not useful. I guess the concern is the potential for abuse, for phishing etc. Given the current procedures for registering TLDs, do you think there is any example of a phishing TLD that is worth paying $250,000 for? One thread could be the cost is going down. Another thread is that such an example is possible. A third answer is that I don't think it is likely to happen.

Asmus Freytag: What was just enumerated is the hard core of the usability principle. The potential for confusion and misuse as opposed to confusability. I think this is what the IAB intended to cover with the usability principle. If we can reign that in as far as possible without conflicting with the simplicity principle. The simplicity principle has two different sides — as simple as possible is one side, as simple as necessary is the other, and they are not quite the same. Joseph and Andrew have raised the issue of simplicity from the implementation point of view, others have raised the issue of complex issues you have to being with. It needs to be noted these two faces of simplicity, so when we write the proposed guidelines to the panel, and also point them out in guidelines to the secondary panel. What is a permissable level of complexity of a solution, given the requirements of both implementability and solving the problem?

— Lunch break —

5.3. Conservatism Principle

Andrew Sullivan: We've talked about the two first principles, then branches a little on some of the other principles. Would like to now talk about the conservatism principle.

Vaggelis Segredakis: [restate comment from after morning break]

Asmus Freytag: The answer is this is a misunderstanding of taking the principle, and assuming it means exclusion. Because we are all aware of these particular characters as being so confusible as to be indistinguishable, we know its something the panels will review. In some of the sections at the end that deal with "concretisation", it should consider these specific issues. You can not look a principle and say a specific character will be excluded, that is for the panels.

Francisco Arias: Vaggelis' issue also seems to relate more to visual similarity than exchangeable code points.

Asmus Freytag: Don't understand the distinction, from a naive external perspective.

Andrew Sullivan: String similarity and other variant things have turned out to collapse on one another, but we have an existing ICANN procedure for string similarity overall. We have made a conscious effort to punt that part of the problem off to another committee. When these panels finish their work, the chances that really hard examples of indistinguishable codepoints don't get handled are almost zero. Functioning panels should do something about those cases. I want to remind us that the origins of this term "variant" did not come from code

points that were indistinguishable from one antoher, it came from the problem of traditional and simplified Chinese. In many cases those code points are clearly distinct from one another, but to speakers of Chinese they are the same thing, so it is almost the opposite problem. The problem has accreted these other meanings, because other writing systems have other problems that may be considered similar.

Asmus Freytag: Part of the problem arises from the encoding. Linguistically we know that Cyrillic and Latin come from the same thing, even though Unicode encodes them differently, but in fonts they are rendered the same. In the history of encoding, one observation is that it used to segregated by region — so there are multiple distinct sets coded differently; and other aspects benefit from a separation of these things: sorting, capitalisation rules, searching, etc. The decision in Unicode to not unify them like Han characters were have resulted in a lot of benefits, it just turns out that is creates the one problem we are dealing with here that has to bear the cost. The fact is these entities are not accidental similarities, but have real world shared origins, I view the issue as different from other string confusables. Maybe you can make the same rules work for accidental similarities versus fundamental similarities. It seems to me with accidental similarities you have a gray zone about how far you want to track them down. And the further you go down the less risk people are going to be confused as their fonts distinguish them. But with fundamental similarities, no matter what font you use, you will not be able to distinguish them without analysing the underlying code points.

Francisco Arias: Visual similarity is probably better analysed at a string level rather than code point level, even though certain similarities be found at a code point level.

Asmus Freytag: Especially between Latin and Cyrillic, the number of related characters means you can create a large universe of strings that are “identical”. As a user you don’t even know if you alternated Latin and Cyrillic (based on first come first served) because you can’t tell. Its a particularly thorny issue, that basically means that you must have that one solved. As Andrew says, you posit that people will be reasonable and solve it.

Francisco Arias: Domain Names in the DNS are comprised of several labels, to be visually similar they need to be visually similar in all the labels. If there is a string that is completely used in Cyrillic characters, all the labels, and someone only knows Latin, and I see that string, if I were to type it using a Latin keyboard I would not get anything. So there is no real harm here.

Asmus Freytag: Take a domain name spelled out in a printed source, and you can not get to it because each alternating sub-string is a different script. It may all look Latin, for example, and each string meets the one script per label rule. That seems like a violation of the predictability rule, and utility, so I see harm there.

Edmon Chung: Support Francisco’s comments, and worried about the path that Asmus is describing. It doesn’t seem to me to be an issue this group should solve. It is important to leave it to the string similarity review. We’d have to stop the whole new gTLD programme as it implicates ASCII TLDs too, so I’d caution us to stay away from that issue.

James Seng: I want to confirm Andrew’s version of events based on being the author of RFC 3753. The term variants was developed there was no concept of visual similarity, it came from the Unicode term of “Z-variants”.

Francisco Arias: This project is only about exchangeable code points, not visually similar variants. If there is something wrong with this approach, that is fine — lets define that. But the conclusion of the previous phase was to treat these two problems separately.

Dennis Jennings: I am not sure that it does. My sense was visual similarity was for strings, but my understanding is there are characters in Latin, Cyrillic etc. that are more than one code point and therefore variants.

Asmus Freytag: It is interesting to note that the business of characters that were once the same thing, but have acquired multiple code points, is an ongoing process. "Q" and "W" were re-encoded recently in Kurdish. In scripts other than Han, we see people borrowing characters from one script and using them in another. You have a situation where they kind of are the same thing, even though they are different code points. If you limit it based on script, you are forced to use two different code points. But if you look at a typesetter, he only has one metal piece in his case. It is useful to distinguish the historical identity of the underlying entity, from mere visual similarity issues. If you don't address some remote visual similarity you have may increased infinitesimal exclusion possibility. Personally underwhelmed by the work to date in the confusable realm, because they do not distinguish between those things that can not be distinguished form those that can not easily be distinguished.

Mirjana Tasic: At the moment I see two problems: If we use both scripts (Latin/Cyrillic) there are two characters that can make one label, that is one problem; secondly you can not leave it like this. You need something like a "variant label", the same label, looks the same, composed of the same labels, those labels should be treated as variant labels even though they are written in different scripts. It would be best to give the same variant label to the one who has first registered the TLD, and to point to the same domain, otherwise the end user will be confused.

Edmon Chung: Back to Asmus' point, there are things we have discussed in the past. The reason I said to move away from it is not to avoid the issue. There are two areas to address the issues — if the Cyrllic or Greek community that come up with proposals that include them as variants, we could review that as part of reviewing the actual rulesets and repertoires. And the reason we come back to the string similarity review, the exact visually similar string, it is reserved in a way. Once you put one in, the others are conceptually reserved, not in a list but in practice. The string similarity review and string contention process already do this and is not precluded by this current discussion.

Michael Everson: It is frustrating that we do not have examples of these things. There are not real world examples of these issues.

Andrew Sullivan: That is not true. There is no requirement that labels be meaningful words. Look at the root zone, most are not words.

Michael Everson: Don't refer to entities with a same presentation form as the same "character". This is why they have been split. One of the implications of the work here, if we decide that scripts can not be mixed — and I believe we should — then this has implications for the Unicode consortium, if for instance the Tatar language community goes from Cyrillic to Latin. That would imply the Unicode consortium would need to encode a new character. I think that would be a good idea. That is why the Cyrillic "Q" and "W" is encoded separately from Latin, so it could be sorted correctly for Kurds. Please lets not misuse terminology, but we need to recognise that those are different from a reason.

Asmus Freytag: The term “character”, even in Unicode, many different shades of meaning and it frustrates people on the outside sometimes. It is entirely the case that an encoded character, somehow relates to an “entity in the real world”. The identity of those entities depends on the problem. If in Unicode you accept largely the idea that script is a primary separator. Not all things that function separately are not separately encoded. In Scandanavian, there are some variants of the letter “a” sort after the letter “z”, which is different from other Latin usages. However they are not encoded differently. The particular choices that Unicode made in the encoding are to a large degree “conventional”, and now they are convention they feed back into our sense of reality. But due to the history of how characters encodings were made, choices may have been made differently [today]. TeX for example, does not have separate entities for Latin and Greek characters that look the same, as it is font-based. As the outlines are precisely the same, the internal representation is the same. Michael is correct in one particular thing, if you have strong rules that underscore the latent Unicode rule that different encodings per script, it forces Unicode to keep doing that. Maybe in 400-500 years, things drift and they look conventionally different, but Latin and Cyrillic have co-existed for a long period and the appearance has not drifted substantially. Glyphs are often reused in fonts, for example. Maybe the context in this realm is not to define a solution, but define requirements for a solution, with the awareness we have of this particular problem, so that the second level panel does not issue any final recommendations [that are problematic]. Not comfortable trusting to the magic of the process here.

(Topic held in abeyance for later discussion by Dennis Jennings.)

Andrew Sullivan: I’m very concerned that this discussion has avoided the mechanism that we’ve suggested, and if the next principle isn’t going to investigate this rat-hole again, we need to be clear what is proposed so far. The overall proposal in the document makes it possible for the secondary panel, at least, and probably the primary panel; to create a set of rules that if something uses the Latin “o”, that it would block non-Latin “o”s in the same location. Similarly, therefore, it is possible for the the label generation rules to result in a system such that if you have all Latin characters such that Latin “sex” prohibits the Cyrillic “sex”. This has two consequences — one is Asmus’ concern, that fundamentally indistinguishable characters can’t be coped with, can be coped with under this proposal; and that the string similarity issues are not as different as we thought.

5.4. Inclusion Principle

Andrew Sullivan: This should be relatively non-controversial.

(No comment).

5.5. Simplicity Principle

Edmon Chung: On this principle, generally I agree with it. I would caution what Asmus mentioned earlier. Another thing important to think about is, how are these rulesets going to be implemented, who is going to use it, and how it is going to be used. Fundamentally, the only people to use it are the IANA folks. Unless you want others to use these rules — and we said earlier that we don’t know if these rules will be used in other zones. In the Universal Acceptance work of ICANN, we’d encourage implementors to use those principals rather than these to consider what is a valid domain or TLD. If you expect an implementor to implement these rules, to figure out if a label is “correct”, that should be sent to the Universal Acceptance part not by rule-setting here.

Andrew Sullivan: One of the criticisms that the Integrated Issues Report received, is the discussion of issues bound up are sent to five different committees and those committees don't talk to one another. In this particular case, that criticism is extremely acute. We can not say "We have this set of rules, we use it to govern what goes into the root, but we encourage others to not use them and do it a different way". These things for the root zone need to be sufficiently comprehensible so they can be used in software. If we think variants, especially TLD variants, are sufficiently important, then software is going to have to be able to figure out what they are. If there are six variants and one shows up on day, I can't tell everyone down the tree to inform them there is another variant, it needs to be software implemented. My mail server needs to be able to compute what its variants are. That is something that has to be automatable, and be implemented by other parties we don't know about.

Dennis Jennings: For any system to process a set of strings from the root, it needs to be implementable by software.

Andrew Sullivan: If thorn and thorn are the examples, then "some.string.down.the.tree.thorn" and "some.string.down.the.tree.thorn" are variants, I have no way of informing the operator of the zone that those are variants. The only reliable way for a mail server there to start up is to know what those variants are.

Edmon Chung: The only way for those rules to work is for those rules to be the same. Even if the rules are right for the TLD, they may be a different set down the tree.

Andrew Sullivan: You are assuming the rules are hardcoded, but they need to be dynamic.

Daniel Kalchev: This is currently being dealt with, with XML schemas (IDN tables), so when you go to the next level

Dennis Jennings: The point emerging from this discussion, is how we perceive this will be used need to be made more explicit

Akshat Joshi: When you said that suppose there is a variant for a certain TLD, there should be a way for a software develop itself to know in advance by way of a rule or a list maybe. Talking about browsers, it is not the case that every browser knows every possible domain or not, so it needs to do a DNS lookup. Can such a service be used in such a case where the rules are not easily implemented.

Andrew Sullivan: In order to do this, remember you have to create a way to look it up, whether in the DNS or other, you need some mechanism. One way is to parse the rules, and then lookup whether those are in the DNS or not. If we do this with a new RR Type then it is going to take a long time (10+ years). If this is really going to work, makers of sendmail, Apache etc. will look up using whatever encoding they have, look up all those domains and see if they work. As the administrator of servers on the Internet, you need to configure servers so they know their own names. There are a lot of examples of this with web, mail, HTTP; SIP has NAPTR records that makes it more complex.

Dennis Jennings: As the configurer of my mail software, I install software and it does this for me?

Andrew Sullivan: You set up your server as you would today, and you set it up for one hostname that you know and for mail it rejects mail for other things that it doesn't know about. As you only speak English and only write in Latin, you don't know that you may be

known by these other variant domain names. Your mail server should look that up automatically and figure it out. Alternatively, I'd configure the mail server to explicitly reject anything I don't set it up for.

Asmus Freytag: The automated process, would that crank through all the string similarity rules?

Andrew Sullivan: The idea is that it will go through the label generation rules at each level, having looked them up somehow, with something like SRV records that tells you where they are. These have to be perfectly automatable rules.

Francisco Arias: What Andrew is referring to is active variants, or mirroring variants, which is only one kind of variants. In the visual similarity case, the only possible treatment is blocking, so there is no way to have active variants that are visually similar. So you don't need to handle that here.

Asmus Freytag: In the case of Latin and Cyrillic use case I have in mind, as you can request nonsense letter combinations like "ICANN" that aren't real words in the language, in the Latin and Cyrillic case the keyboard constrains the user to only their script. You find people looking at these perfectly identical domains on a business card, and in these cases, you must have variants to enable usability. If you can not assume which environment you're in, you can not know at the root zone, if you don't have mirroring variants. That is different from "L" and "1" standing in for one another. This is one where you have no chance of even guessing which one it is.

Andrew Sullivan: I want to be very clear, in this supposed variant generation in our example, in the root zone goes the string "example" and "example" (xn--eample-bsf). … Very concerned about promising of mirrored delegation, as they are not that reliable.

Edmon Chung: As you go down the path, you get to the conclusion from before, that the different between "I" and "L", you don't want to want to mirror "AS1A" or "corn". Just because we know English, the same kind of rule should apply in that sense. It should be flagged in string similarity, but shouldn't appear in the root.

5.6. Stability Principle

Andrew Sullivan: My impression for the earlier discussion that text is needed here to clarify, but that it is generally well understood.

5.7. Letter Principle

Andrew Sullivan: This letter principle lands us squarely back to RFC 1123. Part of the reason that it is important is that RFC 1123 has the top-level is "alphabetic", and the letter principle is an attempt to re-engage that somehow. In addition, it discourages things like apostophoids or other confusable letters with punctuation. The IAB draft calls this out as only applicable to the root zone.

Asmus Freytag: You could be more conservative if you want to. One of the reasons you don't want random symbols is it can be difficult to enter them. You could say only allow things in that are in major, well defined, keyboard standards. Of course, keyboard layouts can be created, but the general idea of restricting things to the most usable subset, might be a useful thing to think about.

Edmon Chung: How does this reconcile with ideographs in Han?

Andrew Sullivan: Numbers are not allowed, but they are not allowed in the root zone today.

Asmus Freytag: Ideographs in Unicode are considered letters, and ideographs that potentially function as numbers are considered letters.

Andrew Sullivan: These specifically avoids Unicode categories, because in Brahmi there are cases where characters that are not "L"-type are needed. I expect the panel evaluating it need to produce arguments about this, and those used commonly in words may need to be kept, and those not commonly used may need to be left out; and overriding is the conservatism principle.

Michael Everson: Remembering that there are 320,000 Icelanders, for the languages of the Pacific there is the U+02BB, upside down apostrophe (ʻ), that would not be allowed. Is that letter going to be in, or out, or in the exceptional list?

Andrew Sullivan: On the text, you're reading, is the maximal set. Common is excluded because its used by multiple scripts. There are still examples of this kind of thing, is clearly, in some circumstances a thing is commonly used a letter in a context, and in another, not considered a letter. In those cases they'll be excluded because they are not letters all the time.

Michael Everson: That will be a problem with the use of apostrophes etc. that are not this character. The only people with this character on there keyboard need it. Are we banning Polynesian characters from the DNS, because that is what we're saying.

Andrew Sullivan: This principle does lead to the effect that some code points that are important to some people are left out. The question is does it prevent all useful mnemonics for that language or not? If it means there is no useful mnemonics that is a problem. I'm thinking, in particular, of joiners (ZWJ, ZWNJ). They are not allowed under this principle.

Michael Everson: That would exclude some words, sure, but in Polynesian the exclusion would remove over 50% of words.

Andrew Sullivan: The consequences cut two ways — it does constrain flexibility drastically in some cases, but it does help the rest of the Internet community.

Asmus Freytag: What prevents one random chosen punctuation look-alike character, being used in any script in places where you need some separation, and you can make it as a fall-back in these other places where you need another character. I won't say "hyphen", but a high utility character like that, that would enable certain types of structuring of mnemonics, deliberately chosen not to be ideal but as a fall-back workable choice in a large number of cases.

Andrew Sullivan: There are people in the Internet community who regard almost all of the IDN stuff, particularly close tot he root, as an incredibly bad idea. Historically, the root was extremely special and extremely constrained. The current expansion of the root is considered a bad idea, and the future massive expansion is the worst idea yet. That is a group of people that would present the argument, that every addition of the repertoire of the root, represents more risk, because there are too many people whose use patterns we don't understand, and once we add things to the root we can't remove them. This is the hard-boiled conservative point of view. One response could be you're trying to legislate policy for the root in the

repertoire, and that's not the right place, and you should engage the right ICANN communities. We could try to lay this out conservatively, but in order to get a considered response, we need to write up how this would work.

Asmus Freytag: A strawman proposal on this may be useful, because there is a number of places where the alternative would end up disfavouring a specific community rather harshly. You'd have a good case in saying, that if you want to do this, you put this one thing in, and can be used in place of a ZWNJ in Person, and instead of an apostrophe, etc. It would explore reasons to violate the letter principle in a controlled fashion.

James Seng: In RFC 1123, the alphabetic principle is in a discussion section only. It is not a standard.

Andrew Sullivan: I know, but we can point to software all over the Internet that has considered it a normative rule.

James Seng: IDNA contravenes that principle, so we should not have IDNA at all.

Andrew Sullivan: The reason we have RFC 1123, hyphens are allowed in other levels, but not at the root. Therefore, hyphens are not allowed a the top-level. People who would like to create a crisis about this, say that the addition of test IDNs to the root consider it a violation of RFC 1123.

James Seng: The letter principle, as written in the IAB document, prohibits IDNs.

Andrew Sullivan: The IAB statement, elsewhere in the draft, explicitly allows IDNs. The idea of the letter principle is once you get to a U-label, you refer to letters.

Michael Everson: At some point the gods will die, and sometime the nettle needs to be grasp. In 1989, they weren't thinking about Devanagari.

6. Parameters for decision making

Andrew Sullivan: These come from the discussion in the integrated report from last year. In that report we saw possibilities on how the rules could be made up, "axes" or "knobs" that can be tuned. It should be noted there was some conflation of these in the report, so there are now four dimensions rather than three.

The first is comprehensiveness. The hardest position is to expect to create a rule for every piece of Unicode. The least restrictive position is to have a rule for one code point, proceed with that, and build it up over time.

The second is expertise. The idea here is, "How many experts do you require for this?" Does this require academics, and people who have worked on Unicode, and protocol geeks all need to be involved?

The third dimension is qualification, or the extent to which it can be derived from some formal quality of a codepoint. That would be the reason why Common and Inherited wouldn't be included. The least restrictive version of this is putting arbitrary groups of codepoints together.

The final dimension is centralisation — who gets the make the rules? You could have one panel make all the rules, or you could have different groups that specialise in different parts of Unicode.

Michael Everson: Who is going to decide against these parameters?

Andrew Sullivan: The procedure will be put somewhere along these four axes, then we can ask ourselves to what extent we set out to solve what we needed to in this document, and how does the proposal fit on the axes.

Dennis Jennings: The ultimate product is a procedure to be vetted by the ICANN community and voted on by the ICANN Board.

James Seng: We know there are about 100 new gTLD applications for IDNs, from 11 languages; so there should be certain priorities in terms of urgency.

Asmus Freytag: Want to reiterate that these are tools to allow us to grade our work when we're done. We can explain we made the right choice because of the compromises made on these axes. It sets you up for a backward look at what you've done, but acts as a subtle guidance.

Dennis Jennings: And sets a framework for what the decision will be.

Asmus Freytag: And a way of describing how satisfied we are with the outcome, even though the decision will be made on other factors.

— Afternoon break —

Andrew Sullivan: Tried to discuss what the trade-offs are in each case. On comrpehensiveness, for example, if you try to cover all over Unicode — including Linear-B and other things no-one will use — it will take forever. Similarly, in the other direction, the danger of going too lax is you get rules that don't take into account other parts of Unicode, but even in their own case are not too stable. If you tend to go piece-by-piece, people will pick up what they are interested in and ignore the bits they are not interested in, and later an nasty interaction will be found that will force the rules to be reconsidered. If a subsequent iteration made a big rule change, for example, tried to remove something from the repertoire or if it changed the rule such that it became incompatible with an established rule, it would be an indication that the entire procedure is broken. So the procedure should be designed to be safe enough to allow for iteration. You should be reasonably sure that once you've covered a chunk of Unicode, that you don't need to go back over it.

One of the things that remains controversial about the way we have proceeded so far, is that to date there have been submissions of IDN tables to IANA, and that has been used as an indication that there is a rule, but those tables are not evaluated by IANA. THe problem is there is a relatively small number of people in the known universe that understand Unicode, so this is a constraint on how it can be developed.

The qualification parameter we've circled around a little, the is the code point qualification — whether a code point is allowed in or not. The idea is that the strongest example would derive all of the rules from some formal property, or set of formal properties, of the code point. For example, the Script property, the General Property, etc. We could build another set of rules on top of that using formal properties to make this a much more automatic procedure. The

difficulty with that is that the properties for various other kinds of uses — none are exactly built so that stable identifiers can be usefully determined from them. The new script extension appears to be an attempt to add something along those lines, but not sure how people are happy with it.

Asmus Freytag: The original idea of script identification in Unicode, is to give exactly one identifier per code point. There are characters, like combining marks, that can be used with multiple scripts; punctuation and digits too. Some of these were given the "Common" property, some were given "Inherited" because they attach to a base character that has a script. The problem is that, even though Unicode does not restrict use, in practice certain characters do not appear together (Indic on Han characters, for example). The idea was therefore born to capture known information about where things are used as part of everyday writing, and to give these generically named characters multiple script properties as a set for each character. That is a work in progress. There are some characters that have a fixed script identity, but have been born into other scripts without being duplicately encoded. Lets say, hypothetically, before the "q" and "w" were duplicated for Kurdic, they possibly would have been given a script extension. I say it is a work in progress, which is interesting given we've been harping on about stability. However it is largely an additive process, such that if it is a determined the Question Mark is used in another script, that could be added to the set. As it is the best available information, it would be suitable to be fed into the human process involved in the panels. You can't take the tables and put them into an algorithm, as the algorithm wouldn't be stable, but it could go into an evaluation process that results in a static result. The same goes for a number of other Unicode properties, where they contain the best available information of a character, and our aim is to improve the information as we go without altering the underlying identity of the character. There is also the risk that a character will be disunified under pressure, and there is probably no way to guard against that, and there is probably nothing that can be done about it.

Daniel Kalchev: It is my understanding that this is what is being referred to as "tags". We may want to use these tags to mark the characters that belong to the same language. The Cyrillic script is huge, most are not used any more. Further, when you talk about Cyrillic with someone from Bulgaria, they will think the characters in the Bulgarian alphabet, and the traditional characters from the Russian alphabet. Most won't imagine others used by other cultural groups. It would be very beneficial for our work to use these tags to tag the set of characters that belong to a particular language, and if we can put a versioning system to represent the evolution of those language sets.

Andrew Sullivan: The first issue is the tags I mentioned on the list, is one possible way of pulling out subsets of Unicode, without relying of formal properties of Unicode. Once idea that has been presented in the past, which we've heard more than once. The Applicant Guidebook requires all code points in a label to be in the same Script, but even LDH doesn't meet that as numbers are not part of Latin. This points out another interaction between these parameters — qualification and comprehensiveness are complimentary of one another. You need a way to identify which chunk of the repertoire, and which set of rules, are you trying to invoke in adding a string. I am foreshadowing tomorrow's discussion, but one thing that will be necessary is a mechanism of saying "this is the intent of this label, I am trying to address this community, and I am not worried about people outside of that". The idea of the secondary panel is the rules of registration in a zone do need to worry about everyone else, but the person advocating the label doesn't need to worry about them. The Label Generation Rules are those that are designed to make the root zone safe for everyone, without requiring everyone considering everyone else.

Zhang Zhoucai: The comprehensiveness and qualification are two sides of the same thing — can they be merged into the one thing?

Andrew Sullivan: I think they are different, because the comprehensiveness is a question of how much of Unicode you cover in one go. I could imagine a system where there is a requirement to be high on one and low on the other.

Dennis Jennings: i think comprehensiveness is answering the question of "How many?" and qualification is "Of what type?" Is that correct?

Andrew Sullivan: That conforms with what I thought, that doesn't mean its correct.

Edmon Chung: My take was, comprehensiveness is "how many have you considered?", and qualification is "how many end up in the repertoire?"

Andrew Sullivan: The extreme of comprehensiveness is requiring Linear-B to be addressed before there can be any labels in the root. People have objected to this idea. However, Linear-B is allowed according to IDNA. The qualification thing — and these are named awkwardly — means "can I automatically qualify the code point", might be better considered whether it is algorithmic. The extreme there is purely using Unicode's script property to determine eligibility.

Sarmad Hussein: These two are two different things. I agree they should be separate. Secondly, are we starting to discuss how to calibrate these parameters?

Andrew Sullivan: If we try to discuss how to calibrate them, I am not bothered. It should be a discussion today to lead-in to tomorrow's discussion of the actual proposal.

Sarmad Hussein: As far as comprehensiveness is concerned, if conservatism is the main underlying principle, the place to start from is we say that we only include those codepoints for which we have a TLD string applied-for or already existing in the root zone. Could that be a place to start?

Andrew Sullivan: My view is if we only consider the code points that have been registered, we are in grave danger of future stability issues. The chances are good in those circumstances that we'll fail to consider the circumstances for some character for someone else. So if we look at a code point — and this is part of the reason for the secondary panel — the main thing they are supposed to do is look at the consequence of input from primary panels and say "you haven't considered how that impacts Polynesians", for example. That doesn't meat the rule of equity. It is important to include as much as is practical from ranges we think we understand. That doesn't entail waiting until we are sure we understand everything that could possibly impinge. Then there will be political pressure to get things done and to shift the tasks to some other organisation. We want to be as comprehensive as we know how to be, without being maximally comprehensive.

Sarmad Hussein: It is really hard to define, going by that definition, because we could do the complete script section, but still that is not the whole set. Doing a complete script section may require too much expertise that may not be available. The solution would be a subset of that, but then which subset? There is no easy way to define that other than the set that has been applied-for, as that is the only deterministic set.

Andrew Sullivan: I don't think there will be an automatic rule for this, it will rely on the judgment of humans, on the panels. For example, there may be a letter that is not used in the root zone today, and if we disallowed it from the root zone on the basis that it is not in use. People would be completely justified in saying that is not valid, and they would be fair in that criticism.

Sarmad Hussein: It is exactly this kind of argument that will make it very hard to decide to include only applied-for TLDs, but other uncommon code points. THen you end up needing to process all of Unicode.

Asmus Freytag: None of that will survive the conservatism principle, so I don't know why this becomes an issue. As you get to the fringes, for example, in the Han script we heard earlier about only allowing vetted characters. The longer I listen to this, I think the situation is the same that in order to do something that works for the characters that you do include, you have to have very good knowledge about how they are used. While it would be nice for the panels to do risk analysis on what might happen if you add other characters later, whether that is possible is open, it would definitely be preferable to adding them to the repertoire on spec. Unicode itself is conservative in encoding characters. Michael is often frustrated when he comes forward with proposals and it doesn't survive in the room, and if Unicode can stick to its guns on that level, you might rethink how you build repertoires and limit it to only stuff you truly understand. That would force you to have a mechanism in place to deal with the inenvitable growth in knowledge. It is similar to character set development in the past. We need to consider the concept of dealing with the boundary of knowledge. If we have a good scenario of the set of risks that might need to be entailed — give us that example, we suspect this particular Arabic character may have issues, for example. Having a few concrete examples of potential damage may help us come up with the right answer to increasing the stability of the solution. It is inescapable that the process needs to allow for an additive solution, the enviroment it works in is additive, it is based on Unicode which is growing, and the user community is growing. It is additive in many different ways. Andrew is entirely right — you can never been wholly comprehensive. You can do it on a rough level, like IDNA2008 did. It may be wrong in one-or-two characters, but out of thousands that's not a big deal. Further than that you can not be comprehensive as you don't understand the intricacies. We've heard today about quirks in Latin, Arabic, Han etc. You have, each panel, an outer boundary of what it understands, and no desire to have the panel to know more than it does. Once you select a concrete set of experts, there is a boundary of their knowledge, and you want them to stick to things they know about. Then you don't have mistakes. That is a kind of conservatism. We need to deal with comprehensiveness can not be approached in an additive process.

Andrew Sullivan: There are different orders of ignorance in these cases. In the case of Latin, for example, assume all of the labels in my zone to date used the first half of the English alphabet. I have zeroeth-order of ignorance, but I know enough about the rest of the alphabet to know how to deal with it, even though I've never received requests about it. But I have first-order ignorance about things with diacritics like å, about what to do about that. I think the cases Asmus is talking about is second-order ignorance and above. I think it is OK for the panels to review everything we understand, and in that process we may learn about new areas that we don't understand, but we are not going to investigate things we don't understand at all. I think that might be OK. The panel would be taking a risk, and you say to the rest of the Internet community, "we evaluated it, we understood it". Those orders of ignorance we could potentially work with.

Asmus Freytag: There is a universe of 60,000 characters, that structurally behave very similarly, but you still want to investigate them individually. There is one sense of what is

manageable in terms of workload? If you can not do the workload, the conservatism principle suggests you make a cut-off on what you can manage.

Michael Everson: I really think we need to not talk about Han, as it is not comparable to any other script. Half of those characters are not in use, and are from ancient books. You can't compare that to Devanagari, Latin, etc.

There are some scripts we don't need a panel for. Thaana, in the Maldives, you can just accept them all. Arabic — due to its shaping behavior — is difficult. The dotless beh is a dangerous character that has to be out. Does anyone know why every character in the Devanagari set are there? [No.] I want to talk about what Sarmad said, as the Arabic set is particularly problematic. I would start with stuff you can easily look up in Wikipedia, like what are the top ten scripts (sic) using the Arabic script. Then then next ten scripts. And work like that. You are going to end up with little piles of characters, some safe, some unsafe, that is how the process should work. Some scripts — Armenian, Georgian — they are done. You probably need a group to deal with scripts that don't need a group. The problematic ones will be Latin, Arabic, Devanagari, perhaps some other Indic ones, Cyrillic, Greek; and then the relation between Cyrillic, Greek and Latin. Will the next version of this document, prior to Toronto, outline at that level of detail — is that what we discuss tomorrow?

Andrew Sullivan: Yes, we'll discuss tomorrow. For the scripts that we know how to do them, there is no need for a panel, and no-one has ever asked for them. My sense is we don't worry about them because no-one has asked for them.

Michael Everson: I can see what you're saying, don't put them in until they are needed, but ICANN would be wise to perform contingency on dealing with them as it would be an efficient use of resources.

Sarmad Hussein: The initial sets for characters will be very dependent on the choice of experts. To me, it is going against the conservative principle, that if you take such a panel that has a script, that tries to deal with everything [inaudible].

Andrew Sullivan: Part of the reason the secondary panel is there, is to guard against that. Potential TLD operators could develop a repertoire that allows lots of nifty tricks that would be bad for the rest of the Internet. The secondary panel would be in a position to say "No". If the primary Arabic panel says "we're including all these African characters we know nothing about", the secondary panels job would be to filibuster them until they resolve the issue.

Sarmad Hussein: I am not talking about characters that are not clear, those are clearly out. I am referring to characters that [inaudible]. Even though the TLD application process requires [inaudible].

Asmus Freytag: Noting that the conservative principle should take into account the level of doubt. If there is absolutely no doubt then the principle should not apply. There is a section in the document where there is a breakdown of considerations by script or script group. Scripts are not at all equal. I find it intriguing that Michael is confident that CJK experts have everything in hand, but we haven't heard from the experts if they can handle a batch process. There is pollution in the CJK tables. The issues are subtly different for different groups. What worries me about Michael's idea of a short-circuit procedure, is you lose the benefit of two panels, and this interplay between the panels is useful. Perhaps develop a "Panel for the easy cases", so someone can make the choice and defend it, so the secondary panel can question whether they are really open-and-shut cases. Michael is correct, you can construct a panel

with multi-script experience to identify if things can be assessed quickly. There should be a discussion in the document about what would constitute evidence for that.

Edmon Chung: How will the parameters be used? Will there be one set of parameters to determine the boundaries of each primary panel?

Andrew Sullivan: Asmus had it right before the break, this is a way to structure our procedure. Our proposed procedure is an instantiation at some point on all four axes. These parameters are a measure for us to use after we're done to say "are we comfortable with that?" If we came up with a procedure that said "the entire thing is centralised, controlled by 3 guys in a room, and no-one can send them comments, and white smoke comes out". If our procedure looked like that, and we asked ourselves if we're comfortable with it, and went to the ICANN community, [it would be problematic].

Dennis Jennings: Are the parameters different from the primary and secondary panels?

Andrew Sullivan: No, these are parameters for the entire process. The obvious example is a given primary panel is much less comprehensive than the entire process, as they deal with only a specific part of the process.

Edmon Chung: When the panels are formed, what parameters should they work from?

Andrew Sullivan: The principles, which are there to constrain decision making.

Edmon Chung: Then how would the panels work?

Andrew Sullivan: That is tomorrow morning's discussion.

Asmus Freytag: I mentioned the need of an appendix that provides more detail, with example cases, to the primary and secondary panels on how to conduct their work.

Dennis Jennings: Do we need more parameters, for example, separability? Is that a parameter that we need?

Andrew Sullivan: For the overall process, no. One thing that has caused a great deal of consternation — there is exactly one set of label generation rules for a zone at a given time. Now you might be able to useful describe that set in terms of discrete subsets, but the rules must be internally consistent. So in one zone you can't have two sets of rules for the same code points.

Asmus Freytag: The idea of separability, I don't know where it fits into the principles, I think it is a useful concept. Some of Michael's example are good examples, Kannada is one example. You could maybe qualify certain repertoires as more or less separable in their behaviour, then you have two properties: you can add it later with low risk, and you can add it now with low risk. Seperability is a distinguishing characteristic of some sub-repertoires, and it doesn't help the Han case if you also try to investigate the sub-case of [inaudible].

Michael Everson: The valid set of combining characters, in the general combining block — they are normatively used with Latin and Cyrillic. The diaresis and the circumflex are used in Latin, and in the Georgian script for minority scripts in Georgia and Abkhazia.

Andrew Sullivan: The string must be in NFC for IDNA, is this an issue?

Michael Everson: For Georgian there is no composition or decomposition. There is a bunch of combining marks, not all used in natural languages. It may be prudent to limit the available ones. Some script panels should review this. Would it be, for instance, possible a rule that says these can't be used with CJK characters? Or is that just irrelevant?

Andrew Sullivan: The text about no context rules was with a specific issue in mind. In IDNA we have CONTEXTO and CONTEXTJ rules. The joiner context rule, CONTEXTJ, requires the character before to need a joiner, essentially. The problem with those cases is they turn out to be harder in practice than they are in theory. The question is how often are we willing to make the rules more complicated, to make it more flexible for a given linguistic community?

Edmon Chung: Responding to Dennis' question about other parameters, perhaps then "Efficiency", i.e. lead time to implementation, is an important parameter when we make a decision on a number of things.

Andrew Sullivan: In some sense, its a consequence of how we've set other parameters, i.e. a "time to delivery". We dont want Han users to wait for Polynesian langauges to be solved. To date we have had no requests for U+02BB, therefore we don't want to make others wait for that to be solved.

Edmon Chung: How do we capture that in our decision making parameters? Shouldn't it be included in this section?

Andrew Sullivan: I think it is implicit. One place arguing for a lower comprehensiveness parameter, says explicitly that there will be tremendous pressure to "do something" if you wait for more comprehensiveness. The problem I have with adding it, is I don't want to say one of our main criteria is delivering something quickly, as that is the first thing conservativeness will throw out. In a choice between "Fast, cheap and good", "fast" is the least important.

Asmus Freytag: Difficulty with Edmon's comment is, the section as it exists today, is not well suited to accommodate such comments. What is missing is the section that looks back on the finally chosen set of procedures and guidelines, and has the evaluation. The proposal that we end up with as the consensus proposal has certain characteristics, and these are the reasons we think that process is inevitable, good or whatever we come up with. In terms of front-loading the process by coming up with many parameters, it is difficult, as Andrew has identified the low hanging fruit on the speculative level, as we don't have the solution yet so it is hypothetical. It makes for very hard reading, as it is very vague, not very concrete. We need to be patient and hold off until we have the end of the paper in hand, and then we need to look back to see if we have enough foreshadowing. Let's revisit the idea to see if we have enough ways of grading the result.

Dennis Jennings: This is going to be an iterative process, these two days are just the first pass.

Sarmad Hussein: I'm still uncomfortable with a panel coming in and saying "these 100 characters that are allowed, these are the variants, and this is how it works", and then switching it on without any experience. Then when we have TLDs, we see a problem and have to take something back. That becomes more of a stability problem. Versus, you do the bare minimum that you need to do, and get some experience.

Edmon Chung: I brought my point up as I thought it was important. It is important in the sense that I know we shouldn't fashion it as an early parameter, but we also need to consider who will be reading the document. There is a number of readers of the document that

consider urgency important, and we need to speak to that issue in the document. The message needs to come out clearly, so people in the community see that.

Andrew Sullivan: It sounds to me like Sarmad's concern — I agree with it — it is a constraint on what the primary panel ought to do. As part of its deliberation, it should consider if they are going too fast. Similarly, the secondary panel, probably needs to say "Are you sure 100 variant rules is really what you want to do?" It could be the time the conservatism principle is invoked, by the panels. If your concern is the panels won't do that — I don't think we can write the rules in a way that can't be peturbed by people who don't believe in the principles. I am not sure how to write a procedure that would prevent that. I will point out that there is a procedure, that has been proposed by Patrik Fältström, which is you allow anyone to apply for anything, allow anyone to object, and if someone objects then the answer is "no". That is a way to ensure that no-one could abuse the procedure in favour of things that are dangerous, at the expense of perfectly safe stuff that people just don't like.

Dennis Jennings: Building a procedure that invites objections wouldn't progress very well.

Sarmad Hussein: What is the criteria for when the panels should stop?

Andrew Sullivan: There is nothing in the procedure that guarantees a result. It relies on people working it out amongst themselves. I am predisposed to a procedure that guarantees a result, but my feeling in this case we are stuck with human judgment.

Sarmad Hussein: If we start just with [applied-for] TLDs, what is a problem with that process?

Andrew Sullivan: I know of at least one application for a string that appears to be in German. If we were to believe variant rules for Latin were possible, there would be severe consequences for that application. If we only worked on strings that were applied for, it would have severe impact on other non-applied for strings. For example, there are no applied for strings for Swedish, but we know there are conflicts between the orthographic rules of Swedish and German.

Asmus Freytag: It seems the comprehensiveness parameter needs to be considered. Andrew has tried to valiantly make the point. There are, in all languages and all scripts, there are natural collections. In some cases we call them alphabets. We have the standard repertoires for certain languages. There is a comprehensiveness argument, that you investigate the entire alphabet — not the entire script, but the alphabet. For example, you'd investigate "a" through "z" in Latin, even if they only used 5 letters in the actual string applied for. I imagine there are similar natural breakpoints in the Arabic scripts, and it would behoove the panel to make sure that the investigation is carried out to at least to investigate all the natural members of the set that is being opened up by a particular repertoire. The primary panel needs to define what they think that natural boundary is, then perform their investigation. The implicit goodwill in that process may need to be made more explicit, and the secondary panel can just say no until the primary panel makes sense in their findings. If you have a process where everyone can say no, and no-one can say yes, nothing happens. In this proposal, while the secondary panel is empowered to say no as long as it wants to, it has the power to say yes. Some of the discussion we are having is in circles, as we haven't written certain sections and we are debating details that should be discussed in terms of text modifications. We need to postpone this discussion, get concrete draft language, and discuss that.

Dennis Jennings: To paraphrase, it should be sufficiently comprehensive to cover ...

Asmus Freytag: There is the minimal alphabet required for a language, and also, if the existing registrations exist for the script and there is experience, and they have gone beyond a certain minimal thing, then you need to go to the next level. The next logical subset needs to be investigated. If you had Persian use of Arabic, Arabic use of Arabic, the natural subset would be those code points used by both, and then you could go further.

Edmon Chung: Is there anywhere where accountability and transparency is considered?

Dennis Jennings: The whole process is designed to be community driven.

Andrew Sullivan: This is actually the expertise setting, and partly the centralisation parameter. If you have a very high expertise setting, it doesn't matter what the community says. I've heard lots of comments asserting the ways that Unicode and DNS say, and they are simply wrong. It is perfectly OK where the expertise is set significantly high, and the experts have a say over what the community asserts as fact. Need to ability to say "we know there is a tremendous demand from your community for it, but our expertise says it is too dangerous". Of course, the output of this needs to be subject to discussion, but we need to be able to say sometimes that the experts are just right, and we need to accept their judgment. That is an important feature of the process.

Edmon Chung: Somewhere in the document should explicitly explain that, to note that we did consider the matter.

Sarmad Hussein: Based on these discussions, is there a need for the secondary panel to have competence in the script?

Andrew Sullivan: The secondary panel will be very hard to fill. It needs to have a reasonable level of expertise in the script, to understand the script well enough to know the rules aren't good enough, or to know the rules are too complicated for them to understand — and given their expertise in Unicode to know that is a problem. I am very nervous about this aspect of the proposal, it may bias it toward the Latin script as there are more Latin script experts out there.

Asmus Freytag: The first requirement of the secondary panel, is they must have a general script expertise, and passing familiarity with specific scripts, so they can evaluate the argumentation in discussion with the primary panel. The need the be educated enough such that the logic makes sense to them, while having a broad understanding of Unicode issues. Other experts need to be well-versed in risks to the DNS. That is one way of putting it. Another observation is that I see the primary panel as the ones that take the most community input. They will be lobbied by people involved in the script, and the can consume, consolidate and coalesce that input. The accountability comes from the fact that someone has to ratify the secondary panel finding.

Steve Sheng: It would be helpful to document why this particular approach is taken, because there will be members of the community that think we should use a different approach. These parameters I see as knobs to tune. Is there a way to know after the fact that, perhaps, instead of tuning this knob this way, we should aim it higher or lower. How will we know that we tune to the right parameter?

Andrew Sullivan: The glib answer is that no-one complains too loudly. The true answer is, without running the experiment in an alternate universe, is we can't. We only have one root zone so we can only do it once. That is the reason for the conservatism principle. I know there

are people that are uncomfortable with that, but we don't have a spare test Internet to test things out on.

Steve Sheng: What if we have a review process, after the first "batch" of evaluations, have a review process to see if the parameters need to be tuned?

Andrew Sullivan: I have no objective, but on a practical matter, there will be pressure on the review process will be to allow more than the current process. The same pressures are at work on endlessly expanding the root zone. If it meant that "it didn't do any harm, lets leave it alone", I'd be comfortable with it, but if it is to liberalise the process, I wouldn't want to do that.

Panagiotis Papaspiliopoulos: Based on experience with the GAC, having paragraphs explaining this would be useful.

Asmus Freytag: Like where this discussion is going, but we are piling things up rapidly that we need to draft, as we don't have good text on it now.

Andrew Sullivan: There are several suggestions today about text being written, is it useful to have draft text written tonight. If I skipped dinner, would people read the text before tomorrow?

Asmus Freytag: In general, at least the way I work, even if I have to multitask and read when I am in a meeting, it is still easier to do this if there is some text. Our schedule is unfortunately very compressed, and we don't have the luxury of blue-skying tomorrow.

Dennis Jennings: If we do that, we risk editing text in committee tomorrow. I'd be more comfortable to proceed as planned, not putting a burden on Andrew.

Asmus Freytag: In that case, we need an outline, so we have a place to hang suggestions on. Whether it is just headings, or sentences that say "This section will cover X and Y", so we don't drift around in our discussion tomorrow. It will help us avoid common issues of fixing spelling mistakes etc. Need more backbone than just the agenda for tomorrow.

Joseph Yee: We have .test in many languages, so maybe it is possible to try something.

Andrew Sullivan: The blunt fact is no-one used the .test domains. It makes the damage undetectable. All it showed that you can put A-labels in the root and nothing caught on fire. That is the limit of the test that we did there, that is so uninteresting in day-to-day use to be meaningless. For the root it is important to consider the implications. To draw a parallel, for signing the root zone, we had 48 hour captures to all root servers and all those things. But no-one knew until the last server served that signed zone as to whether it would take significant chunks of the Internet offline. We only knew when no-one complained after the last root server changed over. We are going to do the same experiment when we add variants to the root, we have no way of knowing in advance.

7. Review of Feedback or Comments

Dennis Jennings: Would like to take a few minutes to go around the table to hear feedback.

Zhang Zhoucai: When is the concern regarding irrelative [sic] scripts mixed in a label?

Andrew Sullivan: Is this referring to CJK, as opposed to just Han?

Zhang Zhoucai: Han characters mixed with Pinyin characters, possibly full-width. Also Korean. Even for a relative script language, if a simplified and unsimplified script is used in a label, how do we deal with that? In Taiwan, maybe sometimes they use Han characters with Bopomofo together.

Andrew Sullivan: I think there is a gap in the document about this. It is hinted at a little bit. As we set the comprehensiveness parameter down a bit, so there is no automatic way that codepoints are linked together. So a primary panel is free to create a repertoire that includes these different types of codepoints, all in one chunk. At that point we have a registration protocol, that says "I intend this label to be used with this kind of thing", i.e. a "language tag" although it might not be a language. Then you check at registration time, "Is this section of the repertoire all-encompassed by that tag, or not?" This makes the task of the secondary panel a little more difficult.

Edmon Chung: Echoing Asmus, a very short, form of what we discuss and how it fits back into the document would be useful. It would help us avoid repeating.

Yoshiro Yoneya: The association of code point with script or language is [inaudible] script mixing.

Joseph Yee: Pinyin and half-width issues are already resolved due to IDNA restrictions. Japanese is an outstanding issue, a special case. Hangul can be resolved through NFC, and are not mixed with Unified Han during normal use.

Michael Everson: I oppose script mixing in a label. Tibetan and Han shouldn't be mixed. Just like mixing Arabic and Han shouldn't be mixed.

Daniel Kalchev: I can see your point as someone from Unicode, but were facing here is that the Internet is used by normal people who don't know anything about Unicode. They don't have any idea about these scripts and anything like that. It involves a lot of decision makers. The Cyrillic block in Unicode include characters that look the same as characters in the Latin block. I understand your point about being string in separating scripts in the Unicode terminology but we'll have to find a way to map this to the real world.

James Seng: I think Michael is saying that you are opposed to mixing languages, not scripts.

Michael Everson: I am saying scripts [Japanese needs three scripts.] Japanese is a unique case. [Chinese requires Latin and Han] In a TLD label? [Don't know.]

8. Update on Project 6: Usability of active TLD variants

Steve Sheng: Goal is to inform you of the ongoing work in this area, invite you as a reviewer for the document, and to try and identify questions where the LGR process will have an impact on end user experience.

The project tries to answer four questions:

1. How will various user roles be impact if variant TLDs are activated? Note there are various roles like system administrators, beyond end-users. We are also constrained by active variants, not considering variants that are blocked etc.

2. What are the components of an acceptable user experience?

3. What are the necessary rules or guidelines a TLD should operate under to provide an acceptable user experience?

4. Are there policy/contractual considerations to make these guidelines effective?

The scope considers a large problem space, and we want to constrain ourselves. This is a focus on the intersection of IDN labels, TLD labels and variant labels.

In defining what is an acceptable user experience. A starting thought is to aim for as close to the existing experience with ASCII TLDs and current IDNs. It needs to be predictable to the user, and meets the user demand. The user must also not be surprised.

User roles that have been identified include end users, registrants, registrars, registries (both ccTLDs and gTLDs), system administrators, network manager, security administrator/law enforcement, and application developers. These roles were identified by the integrated issues report.

For end users, what is acceptable? Three points are: the variant labels are no more challenging to understand and use than the primary label (this has implications for WHOIS); all variants are allocated to the primary label registrant; some variants are active and redirect, others are blocked.

Current progress is we've identified user roles, we have started to asses the impact to these user roles. We sent surveys to the IDN ccTLDs that have either variants at the second level, or the synchronised TLDs at the top-level. Another ongoing parallel work is trying to distill some high level usability principles, using general usability as a background for literature.

The project team plans to publish an interim report for community consideration after the ICANN Toronto meeting [in October 2012], and a draft final report sometime in February 2013.

Many of you are experts in the area, or direct operational experience with the end user community. One call from me, is the project will really use some good review on this. I will send an email to P2.1 to review this.

Edmon Chung: Is there any part where we talk about who should do what to make things usable. There may be some things that ICANN should do, hosting providers should do, and so on.

Steve Sheng: That is the plan.

Joseph Yee: As user roles. Is Yahoo considered a System Administrator, in terms of hosting, email, etc.? [Yes] Is the team making the distinction to improve the current user experience [rather than maintain it].

Edmon Chung: I hope not acceptable means we are thinking about not implementing variants, as that is a different kind of non-acceptable.

Sarmad Hussein: There are going to be significant overlaps between the work in P2.1, and the impact on users. Some of the questions here would have significant impact on users. Is an IDL set flat or has some hierarchy? For example, is there going to be a primary versus non-primary label? Should there be more distinctions: Primary, Secondary and Blocked? How do we decide that? Based on what we decide there are associated questions. How are those labels going to be predetermined?

Another questions is how many statuses for a label (activated, blocked, delegated, etc.), who determines the status of a variant, and how can a label change a status?

Dennis Chang: The team is going to work and produce and interim report, and likely there will be a draft before Toronto. Please take a quick look or pass it on to your colleagues, so we get some quality review.

Dennis Jennings: It is helpful to identify other experts in usability, and refer them to Steve and others to help review the document.

Regarding Sarmad's slide — from the user experience perspective, what demands are you placing on the LGR, rather than the other way around?

Steve Sheng: In the upcoming interactions with P2.1, one of the principles under discussion is consistency in P6. That will have some implications on how labels are generated. We will be in communication on those.

— End of meeting —