LOS ANGELES – IDN Program Update
Wednesday, October 15, 2014 – 08:30 to 09:45
ICANN – Los Angeles, USA

UNIDENTIFIED SPEAKER: This is the IDN program update. October 15, 2014.

SARMAD HUSSAIN: Good morning, everybody, and welcome to the IDN program update session. We have two sessions back to back today. The first session is a general introduction to the program, and after the session we have another session which goes in more detail, looking into the label generation rule set.

As far as the agenda for the current program is concerned, we will be giving you an update of the IDN program, the scope of the program and the current project it's undertaking. We will then focus on a couple of projects. We will be looking at the IDN TLD program and sharing with you what a maximal starting repertoire is to build the roots on LGR.

We will also take you through the process of generation panels being formed, taking you from a maximal starting repertoire to a label generation rule set. We are currently in the process of defining tools to assist the community and the generation panels to develop the label generation rule set. We will be presenting the requirements for the tool for your feedback, and request feedback so that we can improve the requirements to fit the needs of the community as we go forward with it.

We have a detailed community update in the next session. However, in this session we are still giving an update from two communities – Latin community, which is just forming, and Arabic Generation Panel, Arabic Script Community, which is a community which is ahead of other communities, so you get the flavor of both sides. And then we will end up in a question answer session.

So let's start with the overview of the IDN program. As far as the IDN program is concerned, we are all very familiar with that. Until recently, most of the top level domain names were in ASCII script. That obviously has changed now, and it is possible to have top level domain names in other scripts as well. Basically, the IDN program, among other things, looks at the different projects which looks at allowing or supporting IDN TLDs.

To be more specific, we are actually working on multiple projects which can be categorized at least in three portions. One of the large projects we are currently undertaking is called IDN TLD Program. It basically is focusing on defining a label generation rule set, which will assist in validating and determining the variants for top level – labels for top level.

We are also assisting in the process for the evaluation of IDN ccTLD applications through the IDN fast track program. In addition to these two programs, we also assist in the process of developing IDN implementation guidelines. This is work which has been done by ccNSO and GNSO together, by the community, to develop guidelines which are applicable at second and other levels to address consumer risk and confusion. And of course, we are now actively engaged in

communications and outreach to let the community know what the IDN program is undertaking.

Let's focus on the IDN TLD program. Even when we had ASCII labels, the top level labels have been more special and more conservative than the labels at second and other levels. One of the main constraints at top level has been what is called the letter principle, that top level domain names must be formed through ASCII letters A through Z only and should not contain digits and hyphens which are otherwise allowed at other levels. So, the top level domain root zone has always been more conservative.

The interesting question arises that when we move from ASCII to internationalized domain names, it is not very clear how we define a letter. One of the main, and what should be allowed, therefore, at the top level, for all the different scripts which we are using as far as the [new] levels are concerned.

Basically, the IDN TLD program is currently focusing at answering multiple questions. First of all, given multiple scripts now beyond ASCII and Latin, what are the characters which can form a label? So, what are core points which can form a label? Which of these characters are possibly confusable with each other or variants of each other? And, finally, are there any additional constraints on labels beyond the core point label constraints which are put in by answering the first two questions? For example, are there any well-formed [formedness] constraints for labels in complex scripts?

So, as far as the IDN TLD program is concerned, our idea is to answer these questions for all the scripts which we want to use for the root

zone. To address this challenge, there was a program which was started on the direction of the Board, and the program started back in 2011. Initially, six case studies were undertaken to look at the challenges and define the problem to whatever extent possible. Those issues were integrated into a singular report, which was then used to devise the way forward.

There were then a series or projects undertaken in the third phase of this program which looked at, basically, community giving direction that has to be machine-possible format for organizing the data, and that data should actually come based on scripts from community.

There was a complete procedure which was defined for this program. Just at a very high level the procedure basically was organized in three stages. As a first stage, the core points which were shortlisted by IDN in 2008 protocol were further shortlisted to exclude characters which should not be allowed for the root zone; for example, digits. That was done by an integration panel, which was constituted by ICANN.

Once we had the maximal starting repertoire in place, the procedure asks for going out to the communities, and for each script community forming a generation panel, or panel of volunteers, who would work on that portion of the script which is already shortlisted by MSR, and further shortlist core points to only contain those characters which should be allowed for the root zone, based on the principles which are already existing for the root zone.

Each community develops a proposal for their own script, and so we get many different proposals, one for each script. Those proposals are received by ICANN and are integrated into a single label generation rule

set for the root zone, and that work is, again, done by an integration panel, which is a panel of experts with expertise in IDNA, DNS, Unicode, and linguistics.

At this point, we are now executing this process. Just to give you an overview, we are currently working on a specification. Kim Davies is here, who will give you an overview of specification and share the requirements of this tool, which we are now looking forward to create after forming requirements based on community feedback.

This tool will take an LGR and – well, it will allow us to create an LGR and use the LGR for the various use cases which we have. We'll get into more discussion on that later.

As far as the progress of the panels is concerned, the integration panel was seated middle of last year, towards the middle of last year. They worked and developed the first MSR (maximal starting repertoire) which contained 22 scripts. They are now working on the second release of the MSR, MSR-2, which was targeted to come out end of this year, which will contain the six remaining scripts which were not covered in MSR-1. MSR-1 was released in June, and as I just said, MSR-2 is expected towards the end of this year.

The Arabic Generation Panel was the first panel to be seated. They started work in early this year. In February they were formally seated, but their work had started in November last year, and they are now continuing to do their work and it's anticipated that they'll finish their work and submit the LGR proposal for Arabic script towards the beginning of next year, January or February of next year.

The Chinese Generation Panel was also seated in September this year, recently. They have been working for a few months developing the proposal. They are continuing their work, and their proposal is also expected to be received early next year.

Based on inputs, the integration panel will look into these proposals and will aim to get the first release of LGR out by the middle of next year.

This is the status of where we are as far as different communities are concerned for generation panels. We have the Arabic and Chinese seated and doing their work. The Korean Generation Panel is meeting regularly. It's currently working on the proposal to be seated. We have many other scripts which are currently forming. We have community interest, some leadership already identified.

We had two initial meetings here, at ICANN 51, for Latin and Cyrillic scripts. There is some initial interest. But then there are also some scripts, which are showing in red on the screen, for which we still have to start the work. So this is basically our status going forward. If you belong to one of these scripts and you use one of these scripts and you want to volunteer, please e-mail us at idntlds@icann.org, and please contribute to this effort.

We are also, obviously, actively engaged in communicating what the IDN program is doing, and also reaching out to the communities to ask them to volunteer for this work. We have been giving updates to communities at ICANN meetings, but also at other annual meetings going on between, of course, the IDN ICANN meetings themselves. We also are maintaining an active list of web presence with updated documents to guide the community in this regard.

I'll stop here. We'll probably go through the presentations and probably take the questions towards the end. So I'll hand it over to Marc Blanchet, who is an Integration Panel member, to talk about the work that has been going on as far as MSR (maximal starting repertoire) development is concerned. Over to Mark.

MARK BLANCHET: Good morning, everyone. Mark Blanchet here, Integration Panel member. On behalf of the Integration Panel, we'll talk about the maximal starting repertoire. As has been discussed previously, the process of generating the repertoire is by starting with the latest Unicode repertoire, and then limit to the subset of the PVALID code point from the latest registry of IDNA 2008. So this is a registry of code points and additional rules that are generated by IANA based on the RFC that describes the protocol.

Then we apply the letter principle, which is described in procedure. And then we further – so, each step actually decreases the number of valid code points. The next step is to actually limit to the modern scripts that are targeted for the given MSR version. What that means is that the Integration Panel doesn't look at all the scripts possible, but takes a subset of the scripts and then start working on this set of scripts.

Then we exclude code points that are unambiguously limited to technical, phonetic, religious, or [liturgical] use, historical or obsolete writing systems, or writing systems that are not in widespread or common, everyday use, based on the criteria that I will discuss in a few slides.

The result of all these steps is a smaller set of code points. This is really the starting set for which the generation panels pick their repertoire. What that means is that we're actually helping the generation panel by giving them a smaller set to work on, instead of the whole set of code points.

A graphic representation of this process is this, where the MSR is actually the smaller set identified in this picture. This work has been done by the Integration Panel and has been submitted for public comments. We got public comments and then we published a new version based on the public comments.

I said that one of the criteria is to identify if the language and scripts are in use or not. One of the important criteria we use is the Expanded Graded Intergenerational Disruption Scale. This provides an evaluation of language vitality. So you are the URL at the bottom. It provides a use to really understand the effective demand of the writing system itself. It's not based on the population size but by the actual vitality of the writing system.

I'm just getting a bit more – give me a sec. When you're getting old, you need help. It's not a perfect correlation with script use, but a useful criterion. Some writing systems also are not stable or not widely used even if the language is. For the MSR, the Integration Panel uses the cutoff between level four and level five, which I have described below for your information. Level one is widespread use and level 10 is extinct language. Level four is identified as educational.

This is a quote from the criteria. "A language is in vigorous use with standardization and literature being sustained through a widespread

system of institutionally supported education." So if a language is at that level, it's in the MSR. Level five says, "Developing: Language is in vigorous use, with literature in a standardized form being used by some though this is not yet widespread or sustainable." Therefore, scripts and languages in this category are not in the MSR.

I'm asked to move faster. So, MSR-1 is the first release we did. It contains 22 scripts. And then the generation panels create their repertoire based on the MSR. So this is a summary of the content of MSR-1. Essentially, we started with 137 code points in Unicode 6.3 down to 32,000 in MSR-1, with the following scripts listed.

The Integration Panel is actually working on MSR-2. The new version of the maximal starting repertoire is cumulative, so it's actually adding code points to the first version. It has a few scripts that are listed here. The MSR-2 will then contain from 22 scripts to 28 scripts, and add about 700 code points, and is scheduled to be released by the end of this year.

One point to make – and that's my final slide – is that we're working on the Unicode 7, which has been released. However, given that the IDNA 2008 registry is still based on Unicode 6.3, we're actually working on the subset related to Unicode 6.3. What that means is, depending on if the IANA table is updated to 7, related to Unicode 7, by the time we're ready for MSR-2, we will be able to jump in. If not, then we will release based on Unicode 6.3 as the IANA registry is.

I think that, Patrick, you had some presentations on Monday about this [inaudible], right? Or – no? Okay. Whatever. So, for the purpose of our work, we're still using this 7.0 and you will see, depending on the

timeline of the [inaudible], if we're all going to go back to 6.3 or 7.0 depending on the IANA registry.

SARMAD HUSSAIN: We'll move right ahead. We'll come back for questions and comments. Once we have the MSR, then we need to take the generation panels from MSR to proposal for LGR. And we have Asmus Freytag here. Again, he's a member of Integration Panel. And he'll take us through the process which generation panels follow, taking MSR to LGR.

ASMUS FREYTAG: Thank you so much. What I'm going to present here is a very quick and high-level summary of the process that the generation panel would go through in creating an LGR. It starts with the most recent edition of the MSR. It creates a repertoire. It looks at question, whether variants are needed and how to define them, and whether some labels need additional restrictions, which would be expressed by these so-called Whole Label Evaluation rules (WLEs).

After the decisions are done, they need to be documented and their rationale provided. They need to be put in a special format, XML format for the repertoire variants and WLEs, and then finally they're submitted for public comment and eventually review by the Integration Panel.

I'm going to look at each of these in a little bit more detail now. After starting out with the maximal starting repertoire, the generation panel looks at the collection of code points that are used with the script that are defined in its scope. From the intersection of the MSR and the code

points required for the script, it further selects the code points needed for top level IDN domain names.

This selection process has to be based on the principles that are laid out in the procedure document. A link of that is provided at the end of the presentation, also. That describes and defines and governs the whole root zone LGR development process.

These principles are fairly detailed and basically describe the parameters that should govern the development process, and in particular, the process of selecting code points for the repertoire. One of the principles calls for an avoidance of any significant systemic risk by including certain code points, and they also, among others, call for an avoidance of any geographical or language bias, that is to the extent possible within the generation panels are urged to cover all the languages and all the user communities that use that script.

It is important that for each code point that is added to the repertoire, there is some rationale provided either individually or summarily if the code point is part of a set, like an alphabet. The rationale is intended to be expressed that it tracks the decision down to how they satisfy the various principles.

In essence, you just saw the graphic earlier of how the Integration Panel created the MSR, and that is the light blue box in the middle of the diagram. The next process of repertoire selects further. First, each generation panel identifies a script. I'm giving an example of script A and script B. Inside the total sum of characters that fit the MSR and are required for the script, the LGR selects a subset.

After the repertoire has been done, an additional question needs to be answered, and that is, “Does the script require variants?” Not all repertoires contain code points that could be considered variants. Some or maybe many generation panels will just simply answer no to this question, and then don’t have to be concerned with this part of the process. Code points that the users would consider the same as some other code points or some other code point sequence are potentially candidates for variants and need to be researched by the generation panel.

If variants are identified, there is two possible types. Some variants lead to labels that are mutually blocked. So a label containing one variant and a label containing the other variant, either one of them could be registered, but not both. And some other variants lead to labels that could be simultaneously allocated to the same applicants.

There is a document called Variant Rules that contains considerably more detail on this question. In particular, it describes in more details the constraints that exist on defining variants, how to make sure that mappings are symmetric and transitive, and how to assign the types on the variants themselves. There is an overall constraint on variants which is defined in the procedure. In addition to the principles, it contains the directive to maximize the number of blocked variants and minimize the number of allocatable variants.

The reason for this is that there is a steep complexity cost associated with creating allocatable variants in the DNS, and so it is really important to limit those as strictly as possible to what is absolutely required for the writing system.

However, blocked variants have the opposite effect. They don't create complexity cost, but can make the LGR more robust by removing confusing and similar – not removing, but not letting confusing and similar labels exist simultaneously.

The next question to be answered is whether WLEs (whole label evaluation) rules are needed. Are there sequences of code points, for instance, in the repertoire that should not exist because perhaps they provide rendering problems, or are there certain sequences that a code point must be in because it's only used in that sequence?

The next question need to be answered, if one contemplates adding a WLE, does the increase in complexity offset by a reduction in risk to the DNS? Does the whole WLE satisfy the principles? The Integration Panel expects that most generation panels would answer no, and not actually proceed with WLEs. But if a WLE is contemplated, it would be really useful to have an advanced discussion with the Integration Panel, so all sides understand the issues involved up front and they are not discovered when it's too late during the review process.

I'm going to give an example of a possible candidate for WLEs. The way this presentation works, it was not supposed to advance the complete slide but do that in pieces, so it's a little bit much here. So let's focus on the top left of the slide. If you look at the way the Indic scripts are encoded into code points, you'll notice that they're coded as individual constituents, but the people writing the Indic script think of their writing system as being written in whole syllables. And you can create a set of rules that define what a whole syllable would constitute – which we don't need to go into detail, so I put an example on the top left.

It turns out, if you create a string that is an incomplete syllable, it might cause difficulties for systems trying to render it. So the question is, is there a reasonable way that we could define the requirement that prevents badly-formed, badly-displaying syllables to be entered as labels in the DNS?

On the middle, there is an example of a syllable according to the first rule, either a vowel or a vowel followed by a modifier letter. In the right column, there is an example of syllables created according to the second rule: a consonant, a nuktated consonant, a nuktated consonant with a vowel killer, and so on.

And if you look at these examples and the rules, you can distill four possible requirements that certain types of code points must follow or not follow certain other kinds of code points. And that particular kind of requirement might be a possible candidate for a Whole Label Evaluation rule.

Once you have in the LGR defined your repertoire, defined your variants, and defined your WLEs, it is time to create that into a formal LGR proposal. The requirements for the proposal format are in the document called Requirements for LGR. There is an XML format for the repertoire and the variants and the WLEs, but in addition, the Integration Panel needs examples of labels, variant labels, and locked variant labels for its review, and the documentation for everything that contains the rationale.

The completed LGR proposal would then be submitted for ICANN so it can be released for public comment, and then eventually released for the Integration Panel for review. After the public comments are

resolved, the Integration Panel will attempt to integrate the LGR into the whole root zone LGR. And if it that process is successful, the Integration Panel will submit this LGR as part of a larger integrated LGR for public comment consolidating several scripts.

If it's not successful, the Integration Panel is not unanimous about accepting the proposal, it will reject the proposal and communicate with the generation panel which features made it impossible to integrate the LGR proposal.

Now, during the entire process, it really helps if a generation panel can keep the Integration Panel in the loop, because by the procedure the Integration Panel is restricted to look at an LGR proposal entirely as an up-down proposal. That could lead to the situation that a single issue would prevent an otherwise useful and competent LGR proposal from being accepted. Early discussions are a useful way to exclude that possibility.

The generation panel throughout the process is expected to follow the procedure, both to adhere to the principles that are laid out in the procedure, but also there are several detailed prescriptions in the procedure that govern the process. The procedure is the ultimate authoritative prescription for the entire process.

There is a document called Considerations for Designing an LGR for the Root, and this document is intended as a helpful list of issues that generation panels can consider during the process of developing an LGR.

In certain cases, the generation panels will work in a script that is related to other scripts, like the scripts using the Han repertoire and Latin-Greek Cyrillic, for instance. In those cases where scripts are related, the Integration Panel will look for consistent and compatible treatment of similar issues in repertoires, and if the generation panels coordinate among themselves, there is less of a chance that the Integration Panel will run into issues during integration. The overall goal of this is to have a consistent user experience.

The issues needed to be coordinated might be, for repertoire, that one has consistent treatments of similar repertoires, for example Indic, that definitions of variants are compatible, for example in the Han script, or that character set might be variants of cross scripts are handled correctly. And similar things for the WLEs.

That concludes the overview, a very quick and very summary and very cursory overview over the process. ICANN in general, and the Integration Panel in particular, have developed a number of resources. This slide summarizes some of them, and they are intended to provide further input to generation panels as to how to best approach the task. Thank you.

SARMAD HUSSAIN: Thank you, Asmus. And we'll now go into the next presentation, by Kim Davies. He's going to talk about a tool which is being built to assist the generation panels and the community to create, first of all, an LGR, and then once LGRs are created, to use the LGRs for label generation. So, over to Kim Davies.

KIM DAVIES: Thank you, Sarmad. This isn't working at this distance. I'm going to move. Okay, so. What we're going to talk about today in this section is what we're calling the LGR Builder. In essence, what we've talked about until now is using LGRs as an important piece of the variant project. We've developed a specification for a file format to be used for expressing LGRs. That file format is based in XML. XML is not necessarily the easiest thing to write out manually. So there's a recognition that tools would be beneficial to the process to take the output of the generation panels and to support the work of other people using LGRs to have some kind of tool that will help them author these files.

An authoring tool would have additional benefits, as well. The benefit of the LGR format is it's XML. It has a schema. This means you can validate the format, check for errors. An authoring tool would therefore allow you to do additional things during the compilation that you wouldn't otherwise be able to do: check for formatting errors, check for illegal values, and so on.

Just a recap, for those that haven't been following too closely, what is an LGR? A label generation rule set, apart from obviously being a set of rules, when we express it in one of these files according to the format, it contains a number of elements. It contains the allowable code points per the LGR, it contains the variant rules for specific code points, and it contains rules, whole label rules, that apply across the stream – kind of like regular expressions, if you're familiar with that. It also includes processing rules. So once you've determined that a string matches an

LGR, what do you actually do with that string? Do you allocate it? Do you block it? And so forth.

One of the design bases for creating the LGR format was the idea that a registry implementing an LGR can simply slot in the LGR into a piece of software, and the software would have all the information it needs to process against that LGR. We're really trying to the remove manual processing aspect from applying an LGR to a string.

I can't even read that. What this diagram is designed to illustrate is kind of the ecosystem of the LGR software. I think once this technology is fully applied, we'll see all these elements available to the community. The underlying foundation for LGR is software libraries. So I'll take the bottom box first.

Our vision is there'll be one or more software libraries that understand the LGR file format that can perform processes on it such as loading an LGR, saving an LGR, validating an LGR, modifying an LGR, or using an LGR. This could be in multiple languages. We've talked in previous meetings that our expectation is that will be at least one or two different open source implementations of the LGR specification. This will provide a good foundation for the four elements at the top.

Walking through the four elements that would depend on these software libraries, the first is a Graphical User Interface. This is the LGR builder that I'm going to talk about today. This is a web-based application for using LGRs in an intuitive fashion. So you can bring up a web interface and it will allow you to interact with an LGR, compile an LGR, and so on.

The second box, command line tools. This is essentially what we have in differing forms today. Folks like Asmus are using command line tools that have been developed that understand the LGR format, process against it, and so on. So in compilation of the MSR that you heard earlier, command line tools are being used to interact with the LGR file format.

The third piece is registry systems. I mentioned earlier that the goal of the LGR format was to allow registries to automatically utilize LGRs. So our vision is in the long term that if you're a TLD registry, for example, if you wish to support IDNs, IDN variants, and so on, all you need to do is implement one of these software libraries into your platform. You don't need to develop a specific context for your registration. You simply implement a generic library, and you insert an LGR file in, and the library understands the file, and all the instructions in the file advise the registry on the processing requirements.

The last box is IANA systems. For those that aren't aware, IANA maintains today what we call the IDN Table Repository. Our vision there is to replace that with an LGR repository. So we would migrate all the existing IDN tables from different TLD registries in existence today into the LGR format, and we'd use the similar benefits that these other tools would have from using that common library to support that system.

I think a slide is missing. There is a phase one. Phase one is, in essence, the minimum functionality. So we're looking at, to do an LGR builder, we want to have a minimum viable product that we can release into the community early, followed by phase two. Phase two would be the desirable functionality required of an LGR builder. So we want to

separate our requirements based on community input as to what is the minimum amount of things such a tool needs to do, and in the fullness of time, what would be all the great things such a tool would be able to accomplish.

Some of the desirable functionality is having a comprehensive UI that understands all the different aspects of LGRs, the ability to do the testing of tables, and some of the more complex features of the LGR that we don't expect most authors to need. I think in the beginning, most of the generation panels are likely to not use a lot of the complexity that the LGR format allows for, but obviously, over time, we want to support all of that.

So this is just a rough sketch of how a UI might look. Ultimately, have some sort of building interface where it shows you all the code points. You can go through, select, deselect, and so on, save them, and so on. So just to give you a slight idea of the benefits of using a web application is we can utilize all the benefits of the web. We can use fonts, we can use layout, and we can use structure to make it a much more intuitive process to author these files.

What are we doing now? Right now we're developing the draft requirements of what this builder tool would need to do, dividing it into a phase one, as I mentioned before, and then a phase two. Today is the first step in doing consultation with the community, to firstly ensure the idea is sound. Is this something useful for you to work on LGRs with? Make sure it's useful. And also identify the requirements. So if listening to this has given you some ideas about exactly what you'd want such a

tool to do, then please relay it back to us, and we'll definitely take that into consideration as we develop the idea.

Then we're going to identify developers and execute on building this tool. We've received a lot of expressions of interest from the community in the LGR development, and certainly some of ICANN's partners want to work on LGRs in general, so we want to identify who can develop this tool so we can get it done.

And then finally, as with all the code we're developing associated with LGRs, our goal here is to make it open source. So whatever libraries or software we develop in the course of this, we will share that so that it can be reused by others that want to make use of the technology as well. Thank you.

SARMAD HUSSAIN: Thank you very much, Kim. We will be releasing the requirements for the LGR tool which we are developing for public comment and public feedback. So please do look out for that in coming weeks, and please do give us feedback on it.

Now we have our five minute quick updates from two different communities, Latin Script and Arabic Script. As I said, we go into much more detail from community feedback in the next session. But over to Cary for a five-minute update the need and the status of where Latin Script generation panel is right now.

CARY KARP: I'm going to be moving from the higher-level discussions of all of this to about as close to the grassroots as it can get. This is material that is probably familiar to everyone, that's why I've only got five minutes to breeze through it.

This is going nuts, here. All right. The Latin script differs from all the others we're considering in two very significant regards. The one is it does not have to justify its way into the root zone. It is already there, which will reduce the degree of motivation for some communities to conduct the next fuss, and raise a red flag for others. Again, familiar stuff. But it should be noted, it has to be noted, that the basic twenty-six letters of the Latin alphabet, A to Z, A to Zed, however one sees it, aren't sufficient for very many of the orthographies that actually use the script. That even includes English.

Where am I supposed to be pointing this thing? It's extraordinarily unresponsive. Okay. Just noting here, for example, taking a vowel and a consonant, noting the amount of decoration that is available for each of them, and for somebody whose language uses one or another of those, the availability of those characters is extraordinarily important. For someone who doesn't use them, the semantics of the other guy probably are unknown and may or may not be of any particular interest.

The distinctions between the forms are, however, enormously important for the communities that use them, and two different communities that use the same decorated form may regard them in significantly different manners.

Next slide, please. If one is dealing with a zone which is essentially any zone below the root zone that serves a community with a clear sense of

linguistic nexus, linguistic identity, all sorts of nuance can easily be accommodated and probably has to be accommodated.

Noting, nonetheless, that you don't go through every detail that you might find in a style guide for the full orthography of that language, so compromises are necessary anywhere.

Next slide, please. In English, for example, the difference between naivete and naïveté is of no significance. Any English speaker will recognize that these are the exact same two words, although almost every publishing house will at least use the double dots over the I when setting naïveté. These, however [on the slide: resume and résumé], are entirely different words. There is a contrastive significance to those markings, and again, it may or may not be regarded as worth the fuss. Some communities certainly will [militate] for it, others will not.

Here is a cute little example. This is Swedish. The top word is nörden, the [nord]. The bottom word is norden, the north. Those are not the same things. And how one might decompose those dots is, it's an interesting question. Swedish does not decompose an umlauted O to an o-e digraph. However, next slide, some other people might. But you don't do it with proper names. The Swedish name Göthe, which is the second one, is written that way. Goethe's name is written as above. And these are not equivalents. Okay? Next, please.

The root zone, as we've noted in any number of these presentations, has no linguistic attribute. Every language that appears in the domain name, however orthographically sufficient that might be, ultimately uses the root zone, recognizing only the script attribute of the label that's there. IDN policies do, however, need to accommodate as many

of the communities that share the root as they possibly can. Again, that's already been noted. Next slide, please.

The VIP phase of all of this, the Latin Script study group, noted that there's a whole lot of characters here. The pass of deciding which of them are actually necessary to a given speech community is a massive task. The problem of noting which characters might be considered variants of each other is normally context-dependent, as we noted with the umlauted O. Some people will say you decompose a decorated character to a digraph, others will say you decompose it to the base character minus its decoration. We can't know this stuff.

The LGR Integration Panel has reduced this to a maximum starting repertoire that takes out a whole bunch of stuff that's just simply not appropriate to the root, and we need now to address the more pointed thing that the IDNA protocol says, the operator of a zone needs to collate a reasoned subset of the universe of available characters that are suitable for the use of that zone. That's one way of defining the purpose of the present exercise. The facet of that exercised intended to address that particular task is the generation panel, the script generation panel, that is in gestation. That's it. I'm done.

SARMAD HUSSAIN: Thank you, Cary. And now on to our last presentation before the question answer session. We have Meikal Mumin on behalf of the Task Force on Arabic Script IDNs, also acting as the Arabic Generation Panel.

MEIKAL MUMIN : Thanks. Hello, good morning. My name is Meikal Mumin and I was asked to speak on behalf of the Task Force for Arabic Script IDNs, the Arabic Generation Panel.

What is the Task Force on Arabic Script IDNs? Well, it's a creation in oversight by the community-based Middle East Strategy Working Group. It has various objectives, and one of them is creating the Arabic script label generation rule sets, as well as further ones. You can see a whole list of it here, and you can see some information on the Middle East Strategy Working Group on the URL provided on the slides.

The TF-AIDN currently consists of 29 members and those members come from 17 countries. Among those members are members of nine language communities who use Arabic script, as well as members who have further expertise in the use of Arabic script from east Asia, south Asia, Middle East, and Africa, that is, language communities and script-using communities from that areas. The members come from very diverse disciplines.

How do you reach out to TF-AIDN? Well, the membership is open. It's community-based. Details and interest of members are posted on the website, and the discussions are being publicly archived. You can find details on the URL provided there, as well as some background on the formation of TF-AIDN in the introduction. There is also an online work space, news and document archive, and an e-mail archive.

Here you can see that, by now, 21 Arabic script TLDs have been assigned or delegated, and they're in various processes. And this is kind of the scope of our work, I guess. TF-AIDN has tried to outreach to the community, which we consider also part of our job. So more recent

presentations have included those at the APTLD meeting, at the ICANN London, the IGF meeting in Istanbul, and the Arabic IGF Lebanon where some members submitted a proposal for a workshop, in fact.

What is our progress and what work have we accomplished? We have formed the Arabic Script Generation Panel, which was accepted by ICANN, as I understand. We have established principles for inclusions, exclusion, and deferral of Unicode code points, and we have published these. We have conducted an analysis of the maximum starting repertoire as published by the Integration Panel, and provided feedback to them during the public comment phase. We have conducted, just recently finished, an analysis of the code points for the label generation rules.

We also had issues along that process, and those issues included that scripts are often much represented as a proxy of languages when we look at materials, and we are supposed to define rules for the [inaudible] script community, where all sources kind of look at language. We have a general lack of data on orthographies. And there is a lack of representation of script language communities. So there is a wide number of Arabic script-using communities, but we only have a limited number of people who have joined the group.

Here is a summary of the code points. I'm not sure it will be readable. So, in total, we considered 343 code points. Of these, 289 have been PVALID according to IDNA 2008. Within the MSR, this number was further reduced to 239, and TF-AIDN has found evidence for the use of 167 of these code points.

One of these code points – in fact, my colleague, Sarmad Hussain, informed me that this example of the code point short list that excluded in the IDNA, it's not excluded by IDNA but by MSR. Anyhow, it's one of the code points which excluded, which is a number, it's a number 2 – it's a numeral, but it's used as a letter in Jawi language, which touches, again, on the earlier issue we had discussed, about the definition of letters across languages.

There are five specific code points which we had listed because of evidence of use but which are excluded in MSR-1. There is an example from Sindhi language. There are 32 further code points where TF-AIDN has found evidence of use, but which are out of scope based on the rules around MSR. So mainly, rules excluding threatened or declining languages, as well as the exclusion of optional characters. This leaves us with the total of 129 characters, which we will suggest for inclusion.

To finish this up, here are some of our next steps. We will have to: produce an XML representation of those rules; finalize discussion on the variants, which we have started recently; publish whole label rules and release them for public comments; and finalize the whole job. So thank you for your kind attention.

SARMAD HUSSAIN: Thank you, Meikal. And now let's open the floor for any questions you may have, or comments. Yes, please.

HIRO HOTTA: Thank you for good presentations. Good morning. My name is Hiro Hotta from .jp ccTLD. I'm named a chair of Japan's JGP. JFP currently has

seven members with various domain expertise. It is not yet formally proposed to ICANN, but has been working inside Japanese language community from the end of August, and have met three times so far. Of course, we are preparing to ask ICANN formally to be perceived as Japanese GP in [inaudible].

The following is the question from me regarding the future evolution of root LGR itself. It means that, is the LGR itself expected to evolve in several years? Can I assume the LGR will be changed in the future?

For example, let's think about the case where a character A and B are defined as variants at this moment, and A is registered and B is blocked. In this case, if A and B becomes nonvariants in several years, I believe B will be released from the blocked set of characters and become able to be registered by other registrants.

If so, I may dare to say that having more variants is safer for the time being. Am I correct? That's the question.

ASMUS FREYTAG: There's two types of variants: allocatable and blocked variants. Even though both of them are variants, they're really different animals. In general, by having blocked variants, you ultimately reduce the namespace of labels that can be registered. If the namespace is a little less dense because variants are blocking each other, that can make the LGR more robust. That's why the procedure had the call for "maximizing the number of blocked variants." That's not to be taken literally; you don't want an infinite number. But when reasonable, yeah, go for more of them rather than fewer.

Have I answered your question or have I told you something completely different?

HIRO HOTTA: Yeah. Half of my question. Is the LGR expected to evolve in several years?

ASMUS FREYTAG: The idea is it would be ideal to have a relatively stable LGR. So if you can solve a problem once and for all by including the relevant communities' needs completely in the initial version of the LGR, that would be preferable.

However, there's various reasons why that ideal of perfection may not be reachable and it is recognized that the LGR may be incrementally added to over time. It is very difficult to ever remove anything, so let's just think of it as additive.

One reason might be a generation panel lacks the expertise for some detailed use of its writing system by a particularly remote community that is difficult to get information from. Rather than holding the process indefinitely, it might make sense to defer that to an update, preferably after studying as much information as is available to get a good idea of whether adding these code points later is risky by their nature or easy to do.

The next thing is that even the Unicode standard is still adding code points. There is going to be a new version of the Unicode standard from now on every year, and while most editions are not the kind of

characters that go into a root zone label, occasionally there might be. Finally, there are some languages that, at certain points, undergo larger or smaller orthography reforms and might require a new character or a new code point. That can happen.

So at all of these times, it would be appropriate to consider those developments and reflect them in an updated LGR if appropriate. Does that answer your question?

HIRO HOTTA: Thank you. So I perceive your comments that the generation panel should come up with a rule as perfect as possible, and don't think it's an interim one. Right?

ASMUS FREYTAG: It's an optimization problem. And there is a sweet spot that you should aim for, which is as complete as you can reasonably get it in a reasonable time. But not to get hung up over perfectionism, but also not say, "Okay, fine. I have one large language covered so I don't care anymore." Neither of those two extremes are acceptable. But in the middle, there is a good gray zone that you can find a comfortable spot.

HIRO HOTTA: All right. Thank you.

SARMAD HUSSAIN: Edmon?

EDMON CHUNG: I think that's a very good point. The balance needs to be struck. When we discussed the whole LGR process, I think we have to admit that there would be possible changes in the future. Language is a living thing, so there are potential changes in the future. That being said, it's important to be as thorough as we can to start with, but time is of the essence as well, because in the new gTLD program, as well as in the IDN ccTLD program, has already started. I think the need for having it sooner is important.

Building on that, I do have a question. I apologize, I came in late. I was listening, but I was a little bit distracted here and there, so I might have missed something. I want to make a comment on the presentation on the tools. I want to make an observation that it appears that the tools being suggested cover a larger set of potential policies than what is requested from the Integration Panel to the Generation Panel.

The tool actually can reflect policies a little bit more defined, for languages or scripts. I think that's a very important point that I want to make. It's that the Integration Panel looks at a subset of a series of policies possible. I know we're transitioning into the LGR workshop, but I think this is a kind of question maybe to be answered in the next session. But for generation panels, I think it may be a good idea to encourage the language communities to make specifications or provide recommendations beyond only what the Integration Panel will look at.

The reason for that is it helps feed into what is going to be done in the latter parts of the project, for implementation into IDN gTLD and IDN ccTLD. So I think it should be encouraged for the generation panel to

explicitly provide that input and recommendation for the larger community to take into consideration as well.

I make it more specific to say that for example, let's take an example of Chinese domains and Chinese IDN variants. In the process for the generation panel to the Integration Panel, the concept of preferred variant is not explicitly included. However, that is a concept definitely that is useful for later implementation in the larger gTLD and ccTLD process. So am I correct to identify that the tools allow for that specification? The Integration Panel is not asking for that information, but the generation panel should be encouraged to provide such input.

SARMAD HUSSAIN: As far as the tool is concerned, we are currently looking at three kinds of users. First, the current focus of the tool is to give an interface for generation panels to actually create an LGR – and that is really the work at hand at this time – because the specifications are complex and communities don't necessarily have access to expertise which can create the specification, even though they have expertise to create the data, or content for that specification.

The current focus of the tool is going to be on providing that interface for assisting communities to generate, create, or build an LGR proposal for submission to IP. There are going to be potentially subsequent phases of this development, which will focus on user LGR and then also use of LGR by the community, and then also perhaps internal use of LGR by different organizations, including IP for example, to test the proposals which are being received.

As I said, we are developing the requirements, and we should actually be sharing the requirements for public comment before we actually embark on creation of this tool, to make sure that we have the rights that are requirements going forward, and we have the input of the community from all these perspectives so that we can actually have something which is usable in the long run.

We will reach out to you, and please provide us comments on the requirements which are initially defined, and we will refine the requirements based on your feedback.

ASMUS FREYTAG: From the perspective of the Integration Panel in particular, while the volunteers that form a generation panel are free to think through problems and make public statements and recommending this, that, and the other thing, I want to be very clear about one thing: when it comes to formally submitting an LGR proposal to the integration panel, that is expected to stick to the framework and constraints and limitations of the LGR root zone development process, and not burden proposals with additional information that is not part of that particular procedure.

We have very carefully published instructions on the format for submission, and in there, we outline very carefully which subset of the XML specifications for an LGR that we expect to be used by generation panels. So we need to separate the efforts of the community to address a larger problem, which it's entirely entitled to do, from the specific actions inside the root zone LGR development procedure.

SARMAD HUSSAIN: There was a question at the back? And I guess we can come back to you.

BEN PHISTER: I'm with the OP3FT in Paris. I have a question concerning the rollout of the LGR program. From what I understand, the first LGR-1 should be published sometime around the middle of next year. However, you showed a slide with a number of languages and writing systems which will not well have a generation panel seated or rules available. When someone submits a label for a TLD, it isn't one of those writing systems for which LGR rules have not yet been defined. What would be the policy of ICANN to accepting that label? What rules will be used to determine whether or not the label is acceptable?

SARMAD HUSSAIN: I think that policy is probably going to be defined when the next round is finalized, and that will be part of the process. At this time, that policy is not defined. Does that answer your question?

We are running over time. We'll just take one quick comment from Edmon, and then we'll close.

EDMON CHUNG: Thank you. So, thank you Asmus for clarifying that. I think that's very useful. And that actually brings the comment to Sarmad and the team to think about the generation panel. Because I can totally imagine there are more recommendations that – it's pretty hard to convene these generation panels already. I think we all come to appreciate that. Let's

try to provide them, let's try to squeeze every mile of recommendation we can get from them. So perhaps what we need to think about is, in the generation panel report, there would need to be two parts; one part that is submitted to the Integration Panel, and that's within the integration panel definition. And an additional part that can inform further policies and further implementation for gTLDs and ccTLDs. I think that's probably a good approach to think about it.

SARMAD HUSSAIN: Thank you, Edmon. And I can certainly see that happening when communities go into discussions and start looking at MSR and LGR. So we will certainly take that input and put into consideration.

With this, thank you all very much for attending. We apologize for going slightly over time, and we'll close the session. Thank you.

[END OF TRANSCRIPTION]