ABU DHABI – ICANN GDD: IDN Root Zone Label Generation Rules
Wednesday, November 1, 2017 – 15:15 to 16:45 GST
ICANN60 | Abu Dhabi, United Arab Emirates

UNKNOWN SPEAKER: ICANN GDD IDN Root Zone Label Generation Rules on the 1st November, 2017 in Capital Suite 3 at ICANN60, Abu Dhabi.

SARMAD HUSSAIN: Okay, so thank you all very much for joining the IDN Root Zone Label Generation Rule Workshop at ICANN60. We have the following agenda for our session today. There's going to be an overview of Root Zone LGR version 2 which just came out by Marc Blanchet, who is a member of the Integration Panel. We will also have a more detailed presentation on how we use this Root Zone LGR by Michel Suignard who is also a member of the Integration Panel.

Then, we are actually developing a LGR tool. This is being developed by our team at [inaudible] and we'll have Audric presenting on the latest developments as far as the tool is concerned. He'll be joining us remotely followed by four community presentations. We have a Chinese Generation Panel Update by Kenny Huang and Wang Wei. We have a Japanese GP Update by Hiro Hotta. We have a Korean GP Update from Professor Kim Kyongsok. They're all chairs of the Generation

Panel. We also have a Greek generation panel update from [inaudible], Panagiotis could not be here, so let's move right into the sessions. Over to Marc Blanchet for the overview of the Root Zone LGR version two. [AUDIO BREAK]

MARC BLANCHET: Good afternoon. I'm Marc Blanchet and I'm a member of the Integration Panel, and Michel nearby me. So, this is kind of a summary of LGR version two, a bit of a repeat from what we discussed this morning, but Michel will go deeper into the details and you will also see how to actually execute those kinds of possibilities using the LGR Toolset in the presentation there after.

So, what is Root Zone LGR2? It's a set of normative XML files and informative documents for, currently, six scripts. It's governed the way the root zone is operated for a given set of scripts. It determines which Unicode code points are permitted in U-labels, determines which variants are allocatable or blocked, and obviously the output is used by other procedures determining whether a label is allocated, delegated, and stuff like that.

It's script based, so each label in the root zone belongs to a single script. However, we obviously cover the languages such

as Japanese who have multiple primary scripts. The Root Zone LGR is released in stages. That's why you're seeing LGR one and two and three later to allow some LGRs to be available sooner in the Root Zone or for, for example, applicants. Version one was Arabic and this version two adds more scripts, five more scripts.

These are the scripts that are available in the current Root Zone LGR. Again, the normative files are XML per script which we call the element XML and they are essentially the ones submitted by the generation panels and a common XML file which is mechanically generated from the script files.

Informative documents include an overview of the LGR itself, HTML representation of the XML files, and the repertoire table such as an example on the side here. These are the actual XML files, so they're for your information, and the HTML files are also available in this URL.

Script files are generated by Generation Panels. The files are reviewed and created by the Integration Panel. Common file created by IP is the cumulative set of all integrated LGRs for the repertoire. So, the common file contains the whole repertoire currently supported by that version of the LGR. It also includes all the variants but in a blocked type form. Character classes, the union of character classes, however sometimes between two script LGR, they might be using some classes so they are

renamed to avoid collision. Same way with the WLE rules, so the common file actually has correct syntax to be used.

How you would be using it is covered in the next presentation, which is done by Michel.

MICHEL SUIGNARD: Okay, I think you had the reference. Yeah you had the reference for you before. You jump a bit too fast. That was for you. So yes, there's a list different document links that you can see there. Okay, now we have my presentation, so this is really how to use the Root Zone LGR2, so we'll go a bit into more details. There's some duplication between the different parts, but it's kind of useful to reinforce some of those points.

So, we see now we use this to compare to a large degree what [inaudible] existing TLDs in their values because obviously we're going to apply to an existing [inaudible] of TLDs, some of them already using IDN so we have to make sure that any new TLD in IDN format is going to be compatible with what is already allocated or delegated. That's just a nice pretty picture, but now we're going to get into a bit more details.

So again, this is a bit duplicated. We have the Root Zone LGR split into some of those files, so I'm not going to go over this list again because that was said, but the main things of why we are

using the LGR is, when we apply for a label, first we have to validate that the label you are applying for is valid. Then we have to generate allocatable variants, if any, for that given label. In some LGR you have none, but some of the LGRs, like Arabic, you can create allocatable variants. And then the last step, you have to check that your labels don't collide with existing delegated labels and their own variants.

So, that's three major steps; on the next slide, we'll go through what we do to execute those various tasks. On what we use as well, which file is used for what. I won't see the top here, but -- so applying for a label, we see each label in the Root Zone is in a single script. You have first to determine which script you are going to use on them. And then depending on the script you have for your level, we are going to use a specific Element LGR.

Each Element LGR, as we saw before, is script based, so it determines exactly which of them you should use for the new level. We see the Element LGR will be used to validate checking again code points, so the code points are within the LGR. We also check context constraints. The position of some code points is constrained by what is in front and what is after, especially for a lot of [inaudible] scripts. That's a very key concept.

Then you may have also some constraint not by the position in the level but as a whole level. Some of them apply by what we call context rules and some basically apply as WLE rules that apply per actions at the end.

And then the other point that we have to do, we have to use that again if there is a variant in the LGR, we have to generate those allocatable variants for the label you applied for.

A note that all Element LGRs are based on the submitted LGR, but they're not exactly identical. We have to modify them slightly to facilitate their integration, but we obviously make sure that we don't introduce regression by running a set of tools. Typically, we like to have those modifications as small as possible. That's why also we tend to be pretty insisting when we get submitted LGRs that they're in a format that avoid for us to make too many changes to avoid further regression risk from our part.

So, the next thing, we check for collisions. That's the next step and that's where we use a different file. This time we use a Merged file that we call sometimes Common. This file contains all the non-reflexive variant mappings, because we don't need the reflexive mapping of the Merged file for what we're doing here.

All the variant mapping types are set to blocked. That's different from the Elemental LGR which typically use -- especially for the one that have allocatable variants, they have to have some of their own variant types which are then resolved within this Elemental LGR. In the common we need to use blocked because the only purpose on the common LGR is to check for collisions, not to generate allocatable variants.

Again, the variant mapping in that Common LGR are symmetric and transitive as usual. Any label or variant label is part of only one variant label set. That's by definition. We see that in each set, all labels are variants of each other.

Then the next thing you want to do, you want to create an index variant. That's basically a way to determine the uniqueness of a label within the variant so each element of the variant set has the same index basically. If you just want to do that you use the smallest point of the variant set because that will define a unique signature that you can use to compare with other labels because then you do the same thing for the existing delegated labels. You compute the index and if you have a match, you know that you have a collision. If you have a collision, obviously that means your variant label cannot be used.

So, that's a small example in Arabic, so there's a string above that I can't read, but someone here maybe can read. So then

you basically determine the base of the LGR that is shown there that you have original label on one allocatable variant. On the other one, they were blocked so they don't really play any role on the process.

So other tasks for the Common LGR that you use this -- yes?

UNKNOWN SPEAKER: What's the difference between blocked variant and invalid variant on your previous slide?

MICHEL SUIGNARD: I don't know, I didn't create that slide, but that's not really very useful. You know that you are creating some invalid variants because they do get on context rule that doesn't make them valid anyway, but there is no difference on the process here basically between blocked and invalid. Invalid will not even be used for the process itself. Invalidity is just for example, they're not really used in the process. You only use the blocked variants. Yes?

MIRJANA TASIC: Mirjana Tasicfor the record. In the previous slide there is a term I am not familiar with.

MICHEL SUIGNARD: Which one?

MIRJANA TASIC: Non-reflexive variant mappings. What exactly does this mean?

MICHEL SUIGNARD: Variant mapping is just a type you define basically; when you do the mapping you define the type.

MIRJANA TASIC: I define what?

MICHEL SUIGNARD: The type.

MIRJANA TASIC: Oh, type.

MICHEL SUIGNARD: For example, typically block is for example an example you can use. But in the Chinese case, you have a lot of different mapping types. You have traditional, simplified, you have both if you want both traditional and simplified. Each Elemental LGR can

define their own mappings types because they're really private to each Elemental LGR, because they basically get resolved by the action at the end. So the action set basically looks into all those mapping types, and then gets to a decision as to whether -- can it be allocated or not. That's the decision you take at the end of the action sequence in the LGR.

UNKNOWN SPEAKER: The question here was about symmetric [inaudible].

MICHEL SUIGNARD: Symmetric, that basically means the idea of mapping between A and B. You have B and A. So symmetric is if you have a mapping between A and B, you also have mapping between B and A. And transitive means that if you have A and B and B and C, you also have A and C.

MIRJANA TASIC: Okay, if I understand, non-reflexive means it's kind of attribute --

MICHEL SUIGNARD: No, reflexive is against yourself. It's basically define the mapping to yourself.

MIRJANA TASIC: Oh, thank you. Now it is clear. Thank you.

MICHEL SUIGNARD: Also, that information is available on various documents that we have on the website, so you can go into more details but in all the system they use for variants, there is also quite a bit of description of variants in the RC about LGR itself, quite a bit of details, including some complicated details on how you do that for the Eastasian writing system, Chinese and such.

Okay, so with a Common LGR you also have other tasks because you have also to make sure that all the Elements LGR comply among themselves, are consistent. We use a Common LGR in fact to do those validations. And again, the merged file is derived from the script LGR by a mechanical process. It's a bit of a complicated process, but in fact it's totally automated, and like we said before, we have to do some renaming to avoid code name space collisions on some minor points, again, edited to make that possible.

Another point that is also useful in this mechanism we use for the root, in fact could be used outside. It could be used for second or third level domain system where they use the script because they are basically the same kind of issues. So, what we

do on the root could in fact be used again in other contexts. In fact, it's a very useful system.

For us, it's a bit new. The first time, the previous version was only one script so we had none of those issues. This time, we have six of them, so that was our first experiment in having multiple scripts on how to do the integration on the context of multiple scripts.

So, the other file that we have in the system, like we said, I'm not going to spend too much time on this because Marc already mentioned that.

How do you make the top to show? I want to unraise the top here. So I just move this thing? How do you get the thing to unraise the top? Am I missing something? Okay. Sorry, the screen here played a trick with us because it can hide things.

Anyway, the script LGR proposals are archived, so it's very important that we present a good description of the Element LGRs. They are really useful for references. We keep them. They're part of the references in the XML. XML do point to this document. This document stay on the ICANN website, so it's a very good documentation on why you got to where you are, so a description of the repertoire, on the rules, because we see that XML itself is very dry. There's not really a lot of detail on how we

have those rules, on how and when and so it's really important that the GPs do a good job on describing their own Element LGR in the proposal and like also having the references for the decision or repertoire decisions they have when they came to those decisions.

Then there's a script LGR, we see an input to Element LGR. The purpose is the same thing. The Element LGR, if you want, is a result when we do an integration pattern as edited them and the script LGR is kind of the input to it. Again, the last we have to do when we do the integration the better it is because that avoids any risk of making mistakes on adjusting various elements.

Then there is the issue of out-of- repertoire variants which always is a bit tricky. It's basically when you need coordination with different Element LGRs, it's very important that you think a bit about the repertoire because what is possible is that one Element LGR can in fact affect another Element LGR through the other script variants. We see that through the CJK context. So we also see that in the Cyrillic, Latin, Greek context where you could have a level in one script blocking in fact another element in another script, or another LGR I would say.

Okay, I think I'm done here. So there's more documents here that have more details on what I just said. Let me see there. I see there's a lot of documents in the IDN TLD portal for the Root

Zone LGR2. That's all I have. Obviously, I'll be open for questions later or now.

SARMAD HUSSAIN: We can do some questions now, so Akshat.

AKSHAT JOSHI: This is Akshat for the records. It's just an implementation detail I just wanted to know from IP, when it comes to variants, when you generate the variants are the generated variants also checked for compliance with the whole level evaluation rules or do they just get generated like that?

MICHEL SUIGNARD: No, that's [inaudible]. There's no dependency on the [inaudible] rules, on the variants, I think. I'm trying to remember. They go through the same process, so there's no differences. I must not understand the question you're asking.

AKSHAT JOSHI: My question was, when we interact with the tool, we give a label. It gives the validity status.

MICHEL SUIGNARD: Oh, that's Marc [inaudible] toolset.

AKSHAT JOSHI: So it gives a validity status and then it generates a set of variants for it. Does it generate the variants regardless of whether they are valid or not, or it takes them through the whole level evaluation and then only generates those which are valid because those which are not valid, they may not even be registered as variants. [Inaudible] be not able to be registered. Why I am asking this is because in our case, I kind of see there could be a possibility that if an ill-informed label is there, it can generate an invalid variant.

MARC BLANCHET: Marc here. The next presentation is about the toolset. I suggest -- and the guys who actually wrote the code are going to present. Let's talk to them after their presentation and I'm sure they'll give you the answer.

SARMAD HUSSAIN: Okay, so we'll park that question until after the next presentation. Any other questions for the Integration Panel?

Okay, so then let's move to the next presentation on the LGR toolset update where we are as far as toolset implementation is

concerned, and Audric will be joining us remotely to present this. Can you hear us, Audric?

AUDRIC SCHILTKNECHT: Yes, hello? Can you hear me?

SARMAD HUSSAIN: Yes, we need to increase the volume a bit. So, please go ahead and you also have the control, so would you like us to move the slides from here, or would you want to do it directly?

AUDRIC SCHILTKNECHT: Either way is fine with me. I will move the slides myself.

SARMAD HUSSAIN: Okay. Please go ahead, then. We can hear you.

AUDRIC SCHILTKNECHT: Okay. Hello everybody. This is Audric Schiltknecht and Julien Bernard from Viagénie. We are going to give you an update of the new LGR Toolset software. As for the agenda of the presentation, first, we will try to briefly summarize what the LGR Toolset is. Then, we will walk you through the new features that we have implemented in the toolset, such as the management of

the sets of LGR files and the impact on label validation. That has already been covered by Marc and Michel, but we will go over that once again. Then the new HTML export feature and finally some of the interface improvements.

So, [inaudible] the toolsets have been made to allow human being to create, update and use the Label Generation Rules. Indeed, these files are the final XML files and might be a bit daunting for non-technical people to edit and manage. That's the reason why this tool has been created. Using the toolset you can also validate labels, you can generate and check variants and ensure that there are no collisions. Every operation that you want to do on LGR files, you can do it through the tool's nice web interface.

So the tool is available. It's an Opensource and also there is a webserver available for you if you don't want to install it and if you want to just use it. The tool is actually divided into two parts. You have cmdline and libraries in python, so you can also use [inaudible] of the toolset meaning [inaudible] processing intelligence by yourself if you want to implement your own interface or your own process, but you can also use directly the web interface.

So now, Julien will talk a bit about the improvements and the new feature about LGR Sets.

JULIEN BERNARD: Hi, this is Julien. So far the new feature of the LGR Toolset, we implemented the sets of LGRs in order to implement the Root Zone LGR structure. As explained in the previous presentation, the sets of the LGR is [inaudible] of Element LGRs. We manage the repertoire which will be a cumulative repertoire of all the Element LGRs. The variants will be the unions of the variants mapping from the Element LGRs with the blocked type because the type is specific to the Element LGR so we cannot guess [inaudible] to get another type for variants in the Merged LGR.

The classes and the Whole Label Evaluations rules and actions unions of those from Element LGRs. Their name are prefixed by the script of the Element LGR in order to find where those classes, or rules or actions came from.

For the interface, we made some better changes to support the sets. First, when you try to import an LGR, you can now import more than one file and then when you import more than one file, this will create a set and you will have in the import interface a new box that you will have for a set name so that you can name your sets as you want. Then, when you click on Import, you go to the main LGR interface with a new tab, the Embedded LGRs tab. You should click on this tab.

Go to the next slide. Okay. If you click on this tab, you have the list of Element LGRs in the set. Then you can click on an Element LGR and see this Element LGR as if you imported it apart from the set. The way to see if your LGR is in a set or it was imported outside the set is, in the top of the webpage you have the path of the LGR and you see here that LGR is in the Root Zone, which is the name of the set here.

Okay. For the tools in many of the label validation part, there is two new elements. You can provide the list of labels as a file, which will open the delegated labels that will be used to check for collisions with the label you want to validate and its variants. And you have also to choose the scripts from the Element LGR that will be used to check if the label is valid.

The process is described in the next slide. So you first retrieve the Element LGR from the script you have selected. Then you validate the label in this Element LGR for classic label variations. The difference is that if you have a list of delegated labels, you will check for collisions and the collisions are checked using the Common-Merged LGR, and finally, if you don't have any collision, you will generate the variants using the Element LGR.

So the results will be displayed like that. If everything is fine with no collision, you will get this type of screen and you can't see the lines, but they are generated and are [inaudible]. If you

have collisions, you will get this type of display telling you that you have collisions and you won't have variants generated.

Okay, so you didn't move her into the slides. Okay, I'll go back to the previous one. This is when everything is fine and this one when you have collision. And we also updated other tools. When this is applicable because not all tools are fir for sets, but we basically did the same two new parameters as for the label validations so you can in some tools provide the allocated set labels file that will contains labels to check for collisions and in some tools you can select a script in order to evaluate the labels in a specific way.

Finally, there is a new tool to check for cross-script variants in a label. Audric will do the next part. Thank you.

AUDRIC SCHILTKNECHT: So a new feature that was added is the HTML export. Basically, we have the toolset which is already a visual way of representing an LGR, but the requirement here was to create also a human-readable version, kind of a one page, HTML version of an LGR in order to ease the exchange and the comparison of LGR without using the tool.

This functionality obviously supports LGR sets and is accessible as part of the LGR Toolset web interface for any of the tools, but

also as a standalone cmdline, however that needs some configuration because it relies on [inaudible] and so on.

So, for an example of the tool, I'm not sure if everyone can see correctly the display. In the top left corner we have all the metadata of the LGR; for example, the version, the date, language tags and unicode version. Then we have the table of contents which allows you to navigate in the page, and we have the description. So, if your description in your input LGR file is in HTML and has the correct type text /HTML, then it will be rendered as a valid HTML on the output of this tool.

Next we have the repertoire. So, a brief scenario with the number of elements, ranges and sequences, and then you have the exhaustive list of the repertoire contained in your LGR. So, for code points, we have basically the code points; some information like the script, the name, associated tags, and we also have two columns that display the context rules, if any, and the variants. You can click on the link in these columns and it will navigate automatically to the correct section later on the page. I will talk a bit about this section next in the final [inaudible].

Then you have all your variant sets that [inaudible]. So like previously, we have a scenario with a number of sets, the largest variant sets and the number of variants according to type. For

this example, it was Merged LGRs so that explained why all the variant types are blocked. Then you have the list of your sets with the numbers, types and some comments.

Now you have the class here, so what part of the LGR is properly evaluated. You have the name of the class, and you have the number of quick points that belong to the class, and you have the numbers. So, the numbers can either be the literal quick points if you explicitly define a class as a list of quick points. Or, it can also be like in the bottom example, if you define your class as a combination of other classes, then you get kind of a visual representation of that with links to the classes you used to define your class. Sorry, I think I missed one.

Finally, we have the rules part of the LGR. So a rule is, you have the name of the rule. Then you have the regular expression which is used to define the rule and later on is used in the tool to evaluate labels and quick points. So you cannot get a visual representation of your rule which can be really useful if you are trying to develop some of the Regex used in WLE. Then you get some flags to indicate whether or not the rule is used as a trigger so in an action, or as a context, for example for code points or for variants, if a rule includes an anchor and references and comments.

Finally, we have written some improvements into the general interface. Basically, we just reorganized things to group things together. So for example, the results of tools have been grouped together on the right of the page. Then on the top left of the page you have a list of loaded LGRs and LGR sets. After that you get also all the links to create and to import existing LGRs. And I think we are done. Any questions? [AUDIO BREAK]

PITINAN KOOARMORNPATANA: We have questions from online participants. From Dennis Tan, questions, what are the practical applications of a class within a script LGR?

AUDRIC SCHILTKNECHT: I'm not sure I understand the question actually.

MICHEL SUIGNARD: Well, I can try to answer that. I mean, a class is basically something you use on the rules later, so it's basically an element of Regex that you have to apply later, so the role of classes is in fact to be used as a part of existing rules that you define either in the context or in the WLE. That's the point, I think.

MARC BLANCHET: Marc to complement. So, my understanding of your question is actually generic to LGRs, not specifically for the toolset. The toolset just expose and make it you to write them and use them, but the use of a class is more an LGR concept itself, so it depends on your need of your script if you need to group or classify stuff for the design of the LGR.

MICHEL SUIGNARD: You will see that, for example, if will have a class for consonant, you have a class for vowels, or you would even subclass some elements if you have more than one type of consonant. Then you use those to apply your context for when you create your context rules.

SARMAD HUSSAIN: So, we are running slightly behind schedule. We will take one quick comment or question from [inaudible] and then we should move on.

UNKNOWN SPEAKER: Yes, I have a question. We have done an application based on the code as it was in the beginning of the summer. Do you have information on changes that this update introduces?

MARC BLANCHET: I can answer that question. So, this is all on GitHub, it represents all open source, so you have release notes in GitHub and you can even --was that your question?

UNKNOWN SPEAKER: I couldn't see in the release notes when I looked there. Do you have that?

MARC BLANCHET: I'm pretty sure, but I can verify offline.

UNKNOWN SPEAKER: Oh, there? Okay.

SARMAD HUSSAIN: Yes. So I guess we can take that offline. Let's also come back to the question which [inaudible] had raised, so Audric or Marc, would you be able to answer that?

MARC BLANCHET: I think all of us were not sure about the question, so could you repeat please?

AKSHAT JOSHI: My question was when variants get generated, does the tool send them back through the label validation process before showing all the possible variants or just generates out the variants without regard to the Whole Label Evaluation aspect of it?

JULIEN BERNARD: Julien here. Do you mean in the Merged area? If you mean that, the variants are generated in the Merged area and we just keep the type blocked and there's no more process in that.

SARMAD HUSSAIN: This is Sarmad. I think in the interest of time, we should move forward and what we will do is, I do understand the question and I will reach out to Viagénie and we'll get you a concrete answer to that. Let's take the other questions after. Also, could you please pass the clicker?

So let's move forward, so next we have Wang Wei, who is chair of the Chinese Generation Panel, along with Kenny Huang who will provide an update from the Chinese Generation Panel.

WANG WEI: Hi, everyone. This is Wang Wei from the CGP, and Kenny and I are co-chairs of the Chinese Generation Panel. Also, we have Dr.

[inaudible] who is sitting back there from CNIC [inaudible] many supportive work to the whole CGP work.

Let's take a review of the CGP work. The panel was formally set up in September 2014. [AUDIO BREAK]

I'm sorry, the panel was formed up in September 2014 and in the next year we had a couple of coordination meetings with Japan and with Korea. We held a meeting in Dallas, ITF meeting, and in Seoul to discuss about the coordination between the three panels because the Chinese characters are used not only in China, the Chinese language, in Japan and Korea.

That year, we reviewed the CGP repertoire and some variant extension because basically, there's an organization that we call CDNC which is Chinese Domain Name Consortium. That is founded by the .CN .TW .educate Hong Kong and [inaudible], who have been working on the Chinese Domain Name for over 10 years, so we have a pretty good basic document based on the CDNC work, but what we need to do is to think to what extent we should inherit the work from CDNC and to what extent we should make changes to the CDNC's work.

So we tried to submit a first draft in 2016 and we got feedback from the IP for the first version. According to the feedback from IP, we had another couple of meetings with Japan and Korea,

specifically with Korea, because there were some variant groups which were hard for the Korean experts to accept. So we discussed a lot, we spent a whole year on the coordination, and by the end of 2014 we almost made the consensus on that.

During the process, we submitted two versions to IP and we got the feedback from IP. According to the statistics of IP, there's almost nine versions of proposals, and I'm sorry that we made up so many versions and that we are still working on it. I hope we can finish the work when we have the 10th version.

So, let's have a quick overview of the CGP Proposal of the current version. Chinese characters are used in the Chinese language area, including China mainland, Taiwan, Hong Kong, and also in Malaysia and Singapore some people now use it. It is also used in Japan and in Korea. The relationship between Chinese-Chinese character we call it Hanzi. The Chinese character in the Japan writing system and the Chinese and Korean writing system is like the graph on the slide. There's kind of an overlapping set between the three of us.

We have 23 experts from 10 countries including China mainland, Taiwan, Hong Kong, Macau, Singapore, Malaysia, as well as members from Europe and North America. Luckily, we have the advisor Edmon Chung appointed by ICANN to help us on that.

We also have the CJK coordination framework to help us on the work.

Currently, we have a repertoire that includes 19,746 characters. I know it is a big number but just imagine that every Chinese character is a word. It has its own meaning, so imagine how many words are in your dictionary, then how many characters we have in Chinese. Currently, there are two characters in the CGP repertoire not included in MSR yet. That's two characters from the Hong Kong local community. We will apply to add them into the MSR follow the standard process.

Most of them inherit from the CDNC table and some, there are 124 from the dotAsia table which reflected a requirement of the Hong Kong local community. We have an extra 18 characters from the Table of General Standard Chinese Characters, which was published in 2012 by the Chinese government.

Finally, we have 43 characters imported from the Japanese Generation panel and the Korean panel. We don't intend to import all of the Chinese characters from Japan and from Korea, but if it happens to be in the iCore set, our expert will review the characters and decide to import them into the Chinese proposal. So, you will see there about a couple of thousand characters overmapping between C and J and K. So, I think all three parties have agreed on that.

Let's go to the variant section. The definition for a Chinese variant is characters with different visual forms but with the same pronunciations and with the same meanings as the corresponding official form in the given language context. To understand that, just imagine that in English we have two colors: color and colour. When you write it in the form of Chinese, you can have a rough understanding of what the Chinese variant is.

So, we define some subtypes for these variants. Basically we have two types: blocked or allocatable. Of course, there's a third type, which is invalid or out-of-repertoire; for those characters we import from Japan and Korea. But we define them as subtypes. It is based on the definition of simplified and traditional.

Based on these types, we generate the rules to allocate the variant labels. The principles are pretty simple. We just allocate the originally applied one and all simplified labels and all traditional labels and block all the others. Of course, we did some coordination with C,J and K especially on the variant mappings; that was unacceptable to the Korean community. Luckily, we made it after almost a two years' coordination.

This slide explains why we need so many characters. There's a concern about if it’s really needed, that we have about 20,000 characters. I have to say that for Chinese, we already have about

4,000 characters since the 16th century BC, so the number of the repertoire increase over time and according to the variant dictionary published in Taiwan 2004 we already have over 100,000 Chinese characters. But luckily, the linguistic experts have done some job on that, so they picked about 20,000 characters which is suitable for domain names and could be used in domain names.

And then s paper shows that there's a Survey on Chinese Weblog Wording, shows that the characters used on weblog exceeds 20,000, which has included all the repertoire we proposed. The China Ministry of Civil Affairs issued a notification last year requiring government information systems related to naming function have to cover the characters in national standard GB13000 and GB18023, which has a huge number of characters.

So the key issue left now is the overproduction issue. So basically, we only generate original label or simplified or traditional, but the problem is that in some cases we have multiple allocatable simplified variants and multiple allocatable traditional. So here's an example. If you input AD, the original label, the allocatable label besides AD, it will be BE and CF. All the others will be blocked.

But when we input an HL, because H has two allocatable simplified variants and L has two allocatable simplified variants.

So the number of combination will be four and the traditional will be two, so totally there will be seven allocatable labels. So when the number of characters with multiple traditional variants increased so the number of allocatable labels were increased.

If you input 10 characters like H, if you input 10 H, you might have 10 power of two, the number of allocatable labels. To fix this issue, we proposed two solutions. The first solution is to see if we could just limit the number of allocatable labels, but ask if it's possible for the applicant to reactivate some desired labels. But anyway, this solution has been denied to cope with the whole framework of LGR.

So we tried to propose another solution which is to create some new subtypes like we mark the J and N in red, which means when the generation rules meet these red characters, it will try to block it. But of course, we don't intend to block them directly, but create some new subtypes to keep it consistent with the current practice at the second level domain to keep the repertoire and the variant groups almost the same as a second level domain, but we create some specific new types at the root level and let the root LGR.

So, another issue for the next step is that we imported about 43 characters from the J and K. Here are some examples, and the

Chinese linguistic expert reviewed them and set up the variant mapping relationship between the characters we imported and the original Chinese characters. So we might list them as out-of-repertoire and set them as invalid and blocked in LGR.

Thank you all, especially for the IP. I'm sorry for the nine versions and probably for the 10th version, and thanks to Edmon who coordinates a lot between C and the IP [inaudible] of course, and our Singapore office. Thank you.

SARMAD HUSSAIN: Thank you, Wang Wei. We'll move right into the next presentation, which is on update by the Japanese Generation Panel by Hiro Hotta, who is the chair of the panel.

HIRO HOTTA: Yes, it should be very brief because there has been no big advance in these seven months. So, update. I can skip this. Of course, you know what the LGR steps are.

What JGP should care about; of course, as Wang Wei said, we coordinate among CJK, and of course, coordinate with IP and coordinate with Global communities and the Japanese community. We have experienced the 19 years of IDN.jp, the

second-level registration, so as far as possible we want to align with the rules for second-level domain names.

Step 1: Manning JGP. I'm a chair and one other member is there, who is Murakami, who is a trademark and gTLD market expert. First version of Japanese LGR. The scopes of the character codes it's a kind of unique language. It means that the Kanji, Hiragana, and Katakana, three kinds of scripts are used in a mixed way. So, we can make a word with three scripts in a mixed way.

` The variants are in Japanese; usually all the characters are considered to be different. It means that there are no variants as defined, at least originally. But, as Kanji is shared with C and K, we have to import the definition of variants from C and K. So, the final Japanese LGR will import variants of Chinese LGR and Korean LGR. Eventually, we do have variants in Japanese. And WLE, maybe there's a very small number of WLE rules. May I have the question here?

UNKNOWN SPEAKER: Would you not say that the Hiragana and Katakana characters would be variants one from the other?

HIRO HOTTA: No. They are not considered to be variants because they do have the separate usage, so we don't define that.

CJK Coordination; this was explained by Wang Wei, so I will skip this.

Now, the consultation with IP and Japanese community. We do have mainly two issues to be solved. One is the reduction of the number of allocatable labels. This issue also annoyed C and us. CGP turned to JGP to resolve that, but we are still thinking about how to solve that.

So, JGP is trying to solve it by limiting allowed strings by employing the notion that allocatable labels basically consist of day-use Japanese characters. So we do have a kind of definition of day-use Japanese characters. Our repertoire has 6,000, and among them, 2,000 is defined as day-use, so we do use this kind of notion to reduce the number of allocatable levels.

However, it seems the Japanese community is not comfortable with this solution because most gTLDs in Japanese scripts may not be general nouns but trade names or geo names that often encompass personal names or geo names, so we do have to consult with the Japanese community and of course with IP, how to reduce the number of allocatable labels. So, we are still seeking a way to reduce all those.

Another issue we have is that some characters in our repertoire look very similar, and how to handle such similar looking characters. Do they have to be variants or is there a way to solve that. So we're thinking about that. Thanks.

SARMAD HUSSAIN: Thank you, Hotta san. We'll move directly into our next presentation. We have Professor Kim to give an update on the work being done by the Korean Generation Panel.

KIM KYONGSOK: Thank you. I'm Kim Kyongsok and I'm the Korean GP Chair. My presentation will be very short. I'll give an introduction and will explain the K-LGR version 0.7. It was announced in March this year. I will explain the K-LGR proposal and XML which are sent to IP and we got feedback. The rest of the issues are just for your reference.

First, we have both Hangul syllables and Hanja characters in K-LGR. As I said, in March this year we announced version 0.7 and it contains 11k Hangeul syllables and 4,758 Hanja characters with 152 variant groups. The number of Hanja characters, 4,758, came from the union of two sets. One is KS X 1001; it was used for a long time. The other is IICORE set in ISO 10646, and we made a union and the number of characters is 4,758. Now,

there's no conflict in variant groups between K-LGR version 0.7 and C-LGR. We confirmed it in March of this year. If there's no serious change in C-LGR, it will be stable.

And now I'll explain IP's comments and feedback. I'll mention just the last part. This document was prepared before I got the response from IP, and actually about three weeks ago we sent a revised proposal and XML to IP, and a few days ago, KGP got a response from IP and the comments were mostly editorial and KGP assumes that the Korean proposal and XML are quite stable and hope to finish by the end of this year, and if possible [inaudible] before the meeting in November. I will now explain histories. You can see. Okay, thank you.

SARMAD HUSSAIN: Great. Thank you, Professor Kim. As you heard, the work with the Korean LGR is coming to a conclusion, and as soon as it is finalized it will be released for public comment for everybody to review, and that should happen soon.

So let's move forward to the last presentation. This is going to be done by Vaggelis on the status of the work with the Greek Generation Panel.

VAGGELIS SEGREDAKIS: Hello, all. My name is Vaggelis Segredakis. I'm filling in for Panagiotis Papaspiliopoulos, who is the regular chair of the working group. Okay, it started. Of course, the script we are discussing is Greek. You see that in the multiple ISO entries. The language is hellenica, the Greek language.

We have the people that are actively involved in this working group have been selected by the Greek government. They have been involved in creating this working group and we come from different aspects of the domain name world and outside. We have linguists amongst that. We have people from the registries of Greece and Cyprus. We have people from the regulators both from Greece and Cyprus, and in general, I have to say to you that Greek is used mainly in Greece, Cyprus and different parts of the world that Greeks reside because in past years we had a lot of people migrating to other countries.

We started in December 2015, as you can see there. October 2016 was the official formation. The work is kind of slow, I must admit. We had the initial meeting face to face, and then, for the rest of the meetings, everything was done through email, telephones and stuff like that.

We have tried to pinpoint the languages that use the Greek script for their writing. We already knew Arvanitika the last time we came here. We had some comments from linguists on the

panel that there are other languages like Karamanlidika, which is a language that spells Turkish in Greek characters, because we had people from Turkey many years ago that lived in Greece; they were speaking Turkish, but they were writing them in Greek characters.

We tried to study this issue. We tried to contact linguists. We tried to ask people who had resources to help us identify what we should do in these cases, and as you will see later on, most of these languages are dead languages. So, although they might have used these characters, they are not actually using them now. The population is either happy using their local characters or not at all.

I won't bother you with the whole procedure we had, but we had some useful key points here. Sorry, that's the next slide. I have to change both. The first one is the Pomak language, which is today used by almost 30,000 people. We talked to somebody who actually has created a dictionary for this specific language. We asked him what were the characters that we should use to better present it if they felt that they needed it, and they were happy to use regular Greek characters that they were already in the repertoire we had presented. So, everything okay from that side.

We had a Greek language question which is a question that goes back at least a century in Greece. It has been resolved for the Greek language. We used to have a language that was formal and it was called Katharevousa. It was used only in writing things, and the spoken language was Dimotiki, which was a simplified version. Back in the 80s, Katharevousa was withdrawn from the -- it's not used as a written language anymore in public records and stuff like that that Katharevousa was used before. So, we only have Dimotiki.

But we had to take into account that there were still some texts, some documents that were of a different era maybe, but was there any value for ICANN to have a TLD in Katharevousa? The answer was no.

Anyway, so in the Greek orthography we have within script variants, characters that look alike. We have cross script variants with Latin, with Cyrillic languages. We found Georgian, we found similarities with some other languages; we discussed about it, we had a proposal about it, and the final issue that we had to take into account was that in the Greek language, you have accent points, and as you go back in time, you have more complicated accent points.

The accented characters in Katharevousa or in ancient Greek languages were heavily accented with different pronunciations

depending on the accents that they had. In the contemporary language, Dimotiki, the accent just shows you how to pronounce it, but it's just one sign, it's called tonos, and it is on a specific position of its word.

So, there was a question: What should we do? Should we take the upper half one? Is it the pointer as well? There you see two versions of the same thing. The upper one is with all the pronunciation points as it used to be in Katharevousa or in the Ancient Greek Language and it is quite complicated. In the time of the Ancient Greeks actually these points were changing how you were pronouncing it, so today we don't have that one. We don't know how to pronounce it with the ancient type of accents, and we use the lower one which is just one point over a specific letter in each word.

We discussed if there was any value to have the old one; in our registry for .gr, we use it. We don't have any registration with this kind of tonos, but you can actually use it if you wish. So, there were some interesting discussions on that.

I go to the next slide and we decided that monotonic characters of the contemporary character set are to be allowed because we don't really think that somebody is going to come and ask for something from the Ancient Greeks. We think that they offer no significant advantage for the average user. It's quite

complicated for them to use it, so monotonic is an okay choice. Other than that, if the registers wish to give polytonic characters below that level, on the second level, they are more than welcome and I think there might be somebody there.

As we said, the Pomak language does not affect the formation of domain names with the Greek characters' set.

We have a variant case, which is sigma and final sigma. Sigma is the s in Latin languages, but if it is within the other letters of the word, then we use the normal sigma which is that one there, the small one. If it is at the beginning of the word, the beginning of a sentence, it's the upper one, which is this one, but the strangest one is this one which is used only at the end of a word.

So, at IDNA 2003, everything was mapped to everything and we had a bad representation of the final sigma of course because every time the browser tried to represent it from xn-- to Greek, you were always getting the small sigma instead of anything else.

In 2008, the final sigma is allowed as a different character. The browser can of course display it as it should be, but at the same time, it's a variant because you cannot distinguish these two words. If it is in capital letters, you get to have small sigma. If it is in small letters at the end of the word, you get to have the final

sigma, so you cannot give these two words to different people or you have to consider if you are going to give it at all. This is within script variant for us.

You see some of the characters with the vowels with tonos there. Unfortunately, it's not very big to be very visible, but we have dialitica, which is called diaeresis here, which are two dots over specific letters if there are three words that can be combined. If you use these letters, they are read separately; if you don't, they are combined. So you have to put them in specific words to give the word the full meaning. So that is within script variant as well.

We have everything under process. Sorry, we haven't finished yet. We are examining the Greek and Latin, Greek and Cyrillic, Greek and Armenian, Greek and other scripts with the help of [inaudible] in our mind, some things became much more clear, and we have to manage to convince ourselves that we shouldn't put every case in there and we should limit ourselves.

We are creating tables per script to see these variants. We're considering various fonts and various sizes because sometimes we have such similarity issues and other things like homoglyphs and some other things.

We know some things about homoglyphs and homographs -- three minutes? Okay, I'll be fast. Because since 2005, we're giving them for .gr and we already have some idea of what to put in there. These are Greek and Latin, you see many letters depending of course on the fonts, are very similar or quite similar. This is Greek and Cyrillic letters that you wouldn't expect to look very familiar from one language to the other. Greek and Armenian, same case. And in Georgian, it pretty much depends on the fonts and the sizes, but sometimes you get to see something that looks a lot like the other one.

So, as we said, we have some issues to take into consideration. We have questions about the context of the domain should be somehow counted in our decision. We look at the existing rules that we have since 2005 to see if they are fit for purpose, and of course we have the experience of the Greek users who are not unfamiliar with Latin characters and many times they use them instead of Greek characters for domain names.

There was some heavy workload, so it was difficult for us to proceed very fast. Heavy workload from other things; unfortunately, not the Greek Generation Panel. However, we feel that we have discussed some issues and we are trying to finalize if possible within this year. I think panels have but

December, 2017 as the finish line. Let's see. Thank you very much.

SARMAD HUSSAIN: Thank you. We're almost to the end of our session time. However, I do want to open up the floor for any final comments or questions. If anybody has one, and I'll put myself in the queue. Please, I'll go after you, so go ahead.

UNKNOWN SPEAKER: Just a small -- maybe it has been taken care already. When I saw the LGR Toolset presentation, who gave it -- I think remotely he gave it. On the first page he showed a link which seemed to be a 2015 link for the toolset. So, I was just wondering whether LGR2 has been taken care of in the toolset and whether it has been updated or not updated. What is the status?

SARMAD HUSSAIN: I think the website link may be a bit misleading because I think those pages, the way ICANN web team works is, those links are frozen on the date they are created. The links don't change as the webpage is updated. So the contents of the webpage may actually be updated, but the link date is still the same date as when it was created.

UNKNOWN SPEAKER: So LGR2 is taken care of, the toolset.

SARMAD HUSSAIN: Yes. Actually, a toolset is generic; you could load LGR2 files to manipulate the tools on LGR2. One question which comes to mind looking at the analysis; you are creating variants of Greek letters with and without marks. I'm just raising this because we have Latin and Cyrillic members here as well, and I think we should probably take some opportunity after the session to get together and discuss this, but just through transitivity, if you have a code point in Latin which is a variant of a code point in Greek and then you have a code point in Greek which is a variant of another letter with, for example, two dots above in Greek, that may actually then have a transitive implication on making the Latin with and without or Cyrillic with and without that mark as a variant in Cyrillic and Latin as well.

So, I think this is something which probably needs to be discussed between the three panels, and I think we really need to have that discussion and have some coordination on that issue, so I would request the relevant members of the GP who are here to take that opportunity and discuss that among each other.

With that, we are two minutes over time, so thank you all for your participation. We'd like to thank the panelists for their presentation and the audience for their excellent questions. So we close the session. Thank you.

[END OF TRANSCRIPTION]