
HYDERABAD – Internationalized Domain Name Workshop on Root Zone Label Generation Rules
Sunday, November 06, 2016 – 11:00 to 12:15 IST
ICANN57 | Hyderabad, India

UNKNOWN SPEAKER: So, we will start with a short presentation from Mark [inaudible] on LGR toolset, which is now available, both online and in open source. And then we'll hand it over to integration panel, to share their feedback on best practices for IDN LGRs for the root zone, based on the work they've been doing, and interactions they've been having with generation panels.

And then we have a few community updates coming from Lao, which [inaudible] will be presenting. One is joining us remotely, to present on Chinese generation, update on Chinese generation panel. We have [inaudible] here is going to be presenting an update from Japanese generation panel.

And Professor Kim will be presenting an update from Korean generation panel. So, without further delay, let me hand it over to Mark [inaudible], to present on LGR toolset.

MARK: Good morning. Let me see if I can manage this. So, [inaudible] asked me to present a few slides on the LGR toolset's new

Note: The following is the output resulting from transcribing an audio file into a word/text document. Although the transcription is largely accurate, in some cases may be incomplete or inaccurate due to inaudible passages and grammatical corrections. It is posted as an aid to the original audio file, but should not be treated as an authoritative record.

features. So, I will walk through. Those are mainly UI [inaudible], so I was expecting to get a laser, but I don't know if that works or not. We'll see.

So, LGR toolset is, it has multiple, actually, multiple components, one is an editor, a visual web based editor, and then there is a kind of a backend services. And the same, the tools can be used not only on the web interface, but also with command line so you can process stuff, you know, on the command line, or in batch, or in different ways. So, it's all about, you know, managing LGR files.

So, code points, variant rules, [inaudible] roles, and all of that stuff. So you can create a LGR, validate a label, manage LGR. You can do union intersections, gifs, various stuff like that. The good thing is, the good news here, is that the LGR toolset is now open source, so you can get it on Get Hub.

And you can actually, you know, we're happy to receive a poll request for any changes or additions. And as [inaudible] was saying, there is also a virtual machine running this toolset itself, so you can use it. Credits to the developers are shown. [Inaudible] will [inaudible] were the developers.

So, here is the walk through example. I use a French LGR because that's my language. So, simple things such as [inaudible], you know, we'll see if this actually is something that

the Latin generation panel will actually propose, but you know, an example. We will validate with a list of labels, and we do an union and dif of the LGRs.

So, doesn't look good to me, so, these are essentially screen shots of the toolset. So, you can import an already written LGR, or you could create a new. So we're creating one, and you specify the validating repertoire, which is, those are the current two different validating repertoires, but as you know, additional ones, or ones that make sense, you could, one could add new validating repertoires.

So you start with an empty LGR, and then you add code points, and so when you sent it, you see no code points, obviously. So, if you add a code point in this context, we're looking into adding a code point range. So the ASCII lowercase letters from A to Z, and it shows you the list of the actual code points in the range, and then you can unselect some, if you need.

So, it enables you to actually, you know, start with a lot of code points without having to specify each of them. So, when you say yes, then it shows, in this example, I didn't unselect anything. So, it shows as a large range.

And if you, sorry, if you click on the, you know, laser pointer here, on the right side, there is a button that says, expand range. So, what it does, it will actually create one code point row for,

one row for each code point in the range. So it's essentially expanding the range. So, at least from my side, I can't see really this, but so, it's the list of all of the code points that were in the range.

So, the...

At least in front of me, the, it's got a blur, but...

Okay. So, now I'm adding a code point, which is ligature OE, and then I'm adding the actual code point sequence, O then the letter E, the letter O and the letter E, obviously to make them variants to each other after.

So now, I'm adding the variant of OE to O space E, the code point sequence. And then the reverse one. Okay. The reverse one variant. We submit this label list. So, with various labels. And you could see on the list, that there is both ligatures O, E, and then AE. And then what I will do with this list, I will choose tools [update?].

So, in many of those tools, depending on the kind of LGR, you may have thousands of variants for each code point. So, it could take some time to process. So, the way we did it is actually to do it in batch mode, then show the result of the front page when it's done. Then in between, you can receive an email saying, you know, the actual processing is done.

So, that's how we did it, to manage the load. Obviously, most of the time, it's just, you know, seconds, but it all depends on the actual work of the number of variants and cope point repertoire and things like that. So, it's actually appearing on the on page at the bottom of the page, when the processing is done.

Then here is an example of the elevated result, which actually shows that AE is invalid, because it was not in the repertoire. So, which makes sense. So, we, for the purpose of this demonstration, we create another LGR, and the... For the purpose of showing the union and dif. So we create a new LGR. We do not add OE, but we do add AE. Then make variance of the ligature, and then the code point sequence of each.

And then we annotate the result, and then you see that now OE is invalid, but AE is okay. So, what we are doing now is essentially do a dif of the two LGR. And again, the same thing here, it actually shows a real dump of everything. It compares everything, the meta data, the code point repertoire, the variance.

So, it's often a long listing, so you have to look for it. And you may not see it on the screen, but it's actually shows that there is a difference in the code points. So, you can do a union of the two, and the resulting union actually now is another LGR, which

is shown on the screen, with the union of all of the code points and variants.

And there is a description of the algorithm that is being used for the union. And obviously, if you annotate the result of the union, then all of the code points are okay. So, that's what I had.

UNKNOWN SPEAKER: So, I think we'll continue again with the presentations, and take questions at the end of the session. So, we'll hand it over to the integration panel to take us through the next presentation which is on best practices for developing root zone LGR. So, [inaudible] will start on this.

UNKNOWN SPEAKER: One thing I may want to clarify on the tool is that, the tool on the editing side, you can do almost everything, but it's validate on the processing side. So.

UNKNOWN SPEAKER: Okay, this looks very good. We are going to, lovely, this is a magical interface here. Perfect. Hand waving brings it all to life. Anyway, my name is [inaudible]. I'm going to start off this presentation, then we're going to tag team with my colleague, Michelle [inaudible], who is going to finish it off, and the...

We're going to discuss various different issues having to do with best practices in terms of LGR design, as we have learned from working with some various generation panels that have so far produced drafts, but also from the experiences we have inside of the integration from discussions among ourselves.

We're going to start talking about a particular example of a type of script that has many related instances around the world. And then dive into discussing some things [inaudible] LGR as an example. And ending up in a summary of issues that have to do with LGR specification, how to... Where to start, how to do the code point repertoire, how to ensure that you constantly document everything, and including the variant sets, and how to generate from the complexity of your writing system, and a simplified set of rules.

And we're hoping for those of you that are in the middle of, or starting in the process of LGR specification, some of these pointers can be helpful. So, I'm going to go carry it through up to the beginning of the LGR specification issues today. So, there is a set of scripts called, that are of a class that's not [inaudible] graphic and not alphabetic, and they are called [inaudible].

And these scripts include scripts like, example like the Ethiopic, the Neo [inaudible] scripts from South Asia, [inaudible] from Southeast Asia. And there are various historical relation among

these scripts. And as a result, to some degree, there are certain structural similarities which has led to the classification of these scripts.

Now, I am not going to insist that we have the linguistic precision on identifying what we are trying to discuss. How do we deal with scripts that either have or can be claimed by some to have a relation with each other? And does that show in the LGR? And if it doesn't show, why not?

This is where we're going. So, don't expect us to tell you something that would impress a linguistic professor here. That's not what we're trying to do. So, historically, these scripts have some similarities in their structure, syllables in these scripts have typically a leading consonant, and then some vowels that belong with the consonants, satellite vowels that can occur after, before, above, and below, depending on the script, or attached in some cases.

And in many of these scripts, a consonant that does not have a following vowel, can be derived from a basic consonant by a mechanism like the [inaudible], and unlike, for instance, the Arabic case, in these scripts, all vowels are marked and not assumed and left to the reader.

It is noteworthy, that as far as said, these treatment of these scripts in Unicode is concerned, that the so-called encoding

model of each of these scripts is different. For example, Ethiopic is almost like Korean in the sense that there is separate code for each syllable. The neo [inaudible] scripts can be understood as having vowel code points, that are separate code points.

They always follow constants as combining marks. And there is stacking of consonants involved. The South East Asians have, scripts have vowel code points. Some are independent, like consonants, some occur before the consonants in the syllable, and there is a use of combining marks with signs and tones, and the details of these are all similar, but in each respect, there are some particular differences.

And the interesting part is that the difference is surface very prominently, when you design the LGRs, because the thing that a LGR pays attention to, and Michelle will go onto that detail later, is you try to invalidate those kinds of labels that violate the basic assumption about the structure of the script, to the extent that a rendering engine or something may not produce a reliable result, because it isn't prepared to handle a certain combination.

And because that is the goal for specifying LGRs, you end up being very sensitive to these kind of basic differences in the way the encoding of the script is handled. The concrete example that we are choosing to present today would be derived from

recent drafts of [inaudible] LGR. We have strictly a case where tones would be marked only in consonants at the head of a syllable, but the LGR does not enforce that restriction because the rules are simplified to only take into the account the immediate context of characters with each other, not model the entire segmentation of the label into its syllabic components.

So, this restriction on point one is only approximately modeled in the LGR. And that is okay, because the rules focus on the important aspect to make sure that rendering of a label is predictable. In [inaudible], you further have certain linguistics constraint on syllable structures.

These constraints are relaxed for labels, even though they exist in the writing system, for labels we need to relax them because, for instance, in regular words, you don't have arbitrary strings or consonants, but when you use consonants to write abbreviations [inaudible], then arbitrary strings of consonants are suddenly possible.

And we know that IDNs, or domain names in general, often have abbreviations as their source, not just words. And so, we have to make sure the LGR does not try to be like an over fussy teacher, you know, marking up things as wrong, that in flexible, every day practice actually exist and are used for identifiers.

Another similar issue is a particular code point, a letter in [Lao] has the code point zero, E, B, C, and this code point... So who is running that mouse? Thank you.

That would be the semi- consonant [inaudible], which is used to mark a final R sound in consonant clusters. Two consonant clusters are given, and it turns out that in earlier writing rules, this had happened after several different consonants. And the semi-consonant, as you can see from its shape, is always applied underneath the proceeding consonant.

With the spelling forms in [Lao] in the 1960s, only one combination was left over, which meant that this character, the semi-consonant would only apply after a host zone. And since then, of course, there has been a rapidly accelerating issue of [inaudible] with some other languages, and whenever they have a R sound in them, that's where the semi-consonant [inaudible] is used.

So, now it can again appear under many more different consonants, then just the [inaudible]. And we call this an evolving treatment of the letter, because it has changed its nature and its usage over time. And we need to make sure the LGR rule that defines the context for the semi-consonant [inaudible] follows actual usage, and is not trying to enforce some kind of idealized usage, but is limited to what...

Allows the actual usage of it. Now, it turns out, I mentioned the reason we have these restrictions is to account expectations of rendering engines. Now rendering engines will usually be field tested so that they cover basically the whole spectrum of actual writing that people are using.

So, in this particular case, we can confidently expect rendering engines to handle things like [inaudible] words, and we also expect that [inaudible] words will show up in identifiers in particular IDNs, so we need to make sure that such rules do not try to restrict certain letters to the ideal case, or the classical case, or the formal case, or the case you would taught it in school.

That's not the purpose of it. So, with that, we are going to switch over to some pointers on how you can create a good and solid LGR specification. And I'm passing the baton over to my colleague Michelle [inaudible].

MICHELLE:

Okay, so the main point that LGR specification is made of probably three consonants. There is LGR specification, there is a XML file, and there are also what we call test files, you know, which basically contain [inaudible] that are for the writing system, that allows to validate the LGR. So, this is a LGR, the

XML specification is probably the main component. That is kind of what you would call normative part of the specification.

But at the same time, it's very important to document the LGR on that's what we use the proposal, or the specification. So, we do expect that document to be consistent. We use a template on the, one of the writers of the LGR to follow that template, because it does help us in the structure. We have a repertoire definition, we an overview of the writing system, we have, you know, basically a description of the values type of characters in the repertoire. [Inaudible] place for the rules.

We have a place for the... That also allows us to compare, for example, the rules as specified in the specification with the XML equivalent. So, it's very important for us to basically do a review of the document that every specification for the same amount... For that, we have a template. The link is on the screen.

We have, as well, some examples, so people can use, you know, all of those elements to create their own LGR specification.

So, code points. We see that, as seen as probably the main point for many LGR of the specification. You have to be conservative. MSR would provide you the maximum envelope, but there is no... We don't expect necessary all of the MSR content for a given script to be included. It's basically a maximum envelope. So every code points that you had on the repertoire needs to be

justified by references to, you know, an existing character set or documentation.

Obviously, for example, historical points. You see, we tend not to have added the code points in the MSR, so that should not be an issue, but even some code points may be programmatic, so you have to make your own judgment. We do use tags. It's basically a way to subset code points, and to be able to group them for further processing for context or rules.

You don't have to necessarily, but very often, it makes documentation much clearer, you know, you can, for example, define a tag for consonants. You can define a tag for vowels, or even subset of them. Or you can have multiple tag for [inaudible] code points. So, you can be pretty precise on how to do the subsets.

Code points, you know, it's very different from the current IDN tables you saw in the past. The LGR do allow you to define not just code points, but also sequence of code points. That's a very powerful mechanism in many writing system, where some character used with basically a preceding code points like, you could even use that in Latin, for example.

Some combining characters only use the base characters, so it makes sense to only include those, not to be overly productive on letting a combining character to be used, you know,

indefinitely. So, on, [inaudible] in some of the [inaudible] system, where you could have, in fact, writing system, you could have some sequences exact. That does restrict the usage, and also make the specification a bit simpler.

In this case, on Thai, the first character can be used as a singleton, but in fact, the second character is only used in this combination, and so it makes sense to only define those two. Then you can specify rules on those sequences as well.

So, like I said before, the documentation for code points, we need references. To be clear, references, one of them is going to be the Unicode version, where the character was first defined. That's pretty easy to find. You can just look at the Unicode standard.

The second one, you know, the next one, would be references that are more based on international standards, or you know, any documentation that is specific to a writing system that would come finding references. We do need accessible reference. They can't just be, it's very important for us, especially for, you know, rare characters are characters that were documentation is a bit difficult to find.

It's very important for us that we have, you know, some evidence of use. That's true, not just for repertoire, it's also true for variance on rules. So, every part of the specification need to

have some references so we can refer to, you know, for the very [inaudible] of those elements.

On the notation for Thai context and rules should also be, don't take [inaudible], so we expect to be, consistency on this. For the rules, there is also a mechanism that you need to put the sequence in order for... So, the attack, the subset need to be defined before they use in the section of the, where the rules are defined in XML.

So, it's pretty important that you don't use a rule before you define the content of the rule. In fact, most of the tool will detect that enormity on which fix will basically just detect that mistake. Yeah, maybe.

UNKNOWN SPEAKER: [Inaudible], we have just discovered, if you use the ICANN LGR tool, you need to make sure you run the validation phase to enforce this restriction, not just edit. If you don't run the validation phase, you get things that are out of order.

MICHELLE: Yeah. [Inaudible] it's very important that you define the rules in the same way, between the specification on the XML. We do [inaudible] for that, and we will make sure that when you

describe specification on the rules in the specification, that you do define them in the same way in the XML.

So, we can do a match between those two sides. Variant sets. Again, this is, you know, we are seeing that most in the [inaudible] SDK side of things, we have a few variance in the South Asian, but very few. We didn't really get... So, this is going to come mostly for the SKD sets. For that, we really expect variant set to be well-defined in existing documentation.

For example, the existing IDN tables has a very important source for us. We will compare any new variant set with those sources. Obviously, we expect them to be consistent, so they have to be [inaudible], so that means, you know, you have to have the set is... If you have multiple variants, they have to be... For each entry, you will have the same variance in the tables.

That would be, obviously, enforced by us when we receive a XML. We always verify that this is true. [Inaudible] ICANN Tool, okay, so use the tool so that will be enforced. But for some, you know, if you do that, then you're on your own. You will have to make sure that is also respected.

Typically, reflexive mapping is something when you're using some specific case, like for [inaudible] case, mostly for traditional, simplified. It's not something that is [inaudible] typically for the variant set. But you know, if you get confused,

in those things, the AP will be happy to help you on these finding what you need to do for variant sets.

Yes, then we have some constrain on the [inaudible]... Yeah. Then, yeah, you can't use context on variance. That's something we have seen in the past on, this is not, it's really not something we encourage. Okay.

Yeah, variant set. We have a lot of discussion about allocate able variants on typically, we don't like allocate variance. It's just because they do create a lot of issues on the... When you kind of go against conservative principle.

So, we... So, a number of allocate able variants, especially if you use a completely free form allocate able, when the label become longer and longer, and you add more and more of code points that are allocate able variant, you get basically, you know, a massive explosion of [inaudible] labels that would be all allocate able to the same entity, or that become quickly an issue on...

Obviously, you can look at the case of Chinese [inaudible] allocate able variant, but they created a system, in fact, to limit so a number of variance, depending on any of the sizes, are able to in fact, free variance, you know, traditional, simplified or original. So, it's not as bad as it looks, or in fact, you should just do a pure allocate able system, you will have much worse without.

So, again, allocate for variance, we something that you should really be very careful in advocating for them, or if you have too many of them, the IP would come back to you, we will be very, very, I would say, straight on this aspect.

Obviously, no, with the RFC, that's for the IGR, that's where you [inaudible] in fact, documentation that specification about the variance. In fact, a lot of examples, even some fairly complex examples of variance. So, do read that document carefully. And also, look at the existing LGR out there.

In fact, we, there is, in fact, now a fairly complicated variant, [inaudible] variants, for example, someone was published for Chinese is out there, who is based on dot Asia. Losing it. So, you can use it [inaudible] but also the Arabic LGR is also available out there, or you can, again, look at how they did variance on various definition of how to process them.

Yes, appendix B is where we talk a lot about the [inaudible] side of variance.

Okay, I go fast.

Yeah. So, it says that LGR is not a spelling specification. That was already wrote before, so I'm not going to go in details on this. We see [inaudible] improving security, so we're trying to

avoid ambiguous rendering. Also, sequence where the rendering would not be consistent.

Like, for example, you may have between platforms, different renderings that would be a good sign, as far as this sequence is probably not safe. We tend also to prefer simple rules, so sometimes, you know, LGR would be sent to us with complex mechanism. We always aim at simplifying them. We will help you sometimes to do that.

We, you know, if you can use contacts, it's always a good thing. We have, in fact, seen a lot of contacts, basically where you, a code point can only be used in the context with defining what is before, after, or both.

That is, in fact, a very powerful mechanism to restrict usage. You can also, we see this context, continue to put code points, so you're not just restrained to a single code point rule, you can also define sequence can also have context.

On the last point, I think test layers is very important that we get [inaudible] of content for your writing system, so we can validate the LGR, so we can make sure that it does pass what people would consider being valid in your running system. And also, flags really considered being invalid. So, we use that, that's basically where the data that we use to make sure that the NGR

do pass correctly for, you know, common words, or you know, things that you would expect to see in the label.

So, we do use those test labels files to determine that the LGR is, in fact, correctly done. And also, when we do simplification, you know, or modification, LGR, we make sure that that simplification does not change on what is valid and what is not valid.

Obviously, we always make sure, you know, [inaudible] system, we do look at the, the fact on dedicated TLDs that have already used those scripts on making sure they do, we don't really create a situation where an existing TLD could be invalid. That would be kind of bad.

Okay, I think that's the end of it. Yeah.

UNKNOWN SPEAKER: So, we'll continue and we'll actually take questions at the end. So, I will then pass on the, to [inaudible], who is going to be presenting on Lao generation panel.

UNKNOWN SPEAKER: Good morning everyone. My name is [inaudible], I'm from Ministry of Post [inaudible] Telecommunication of Laos, which

whole managing the ccTLD of dot LA in Laos. I'm here for, as a Lao generation panel representative.

This is what I am going to present today. I will start with some introduction about Lao script and Lao language. And then I will give some overview of Lao generation panel members, and challenges in developing a Lao LGR proposal, current focus, and timeline.

Sorry.

This is the, some introduction about Lao script. Lao script is used to write the Lao language, and the language is spoken by approximately 13 million people, mainly in Lao, [inaudible], and neighboring of Cambodia, China, Myanmar, and Vietnam.

The complexity of Lao language is the... There are Lao [inaudible] around the syllable, as you can see from here. There is no space between words. How to...?

UNKNOWN SPEAKER: You may, the point will not be visible to the rest of the people, so you just need to guide us through words.

UNKNOWN SPEAKER: And there is no space between words and syllables. There is only through the human process, by reading multiple words and

extract them. But this can be really difficult for the computer system. That's why, as you can see from the picture. So, we have defined the rules for the syllable breaking.

As you can see, the X4 and X5 here, X4 is the position of the top [inaudible], X5 is for [inaudible] mark. And X2 is only specified for the [inaudible] consonant and should come with [inaudible]... said earlier. And X3 is for the [inaudible] vowel.

So, the Lao [inaudible] has defined in detail, its rules for each corrector in the proposal. So, please feel free to come and give your comment and [inaudible] on it.

And next, would be the overview of the Lao GP member. The Lao GP is a group of members having experience in Lao language development. The competition linguistic, and Lao linguistic, and also the IDN. The draft of LGR proposal was actually developed at [inaudible] by the member of Lao GP, together with consultation from additional experts as needed.

And there are about 13 members in Lao GP. So, the contact information and yeah, the name list, are already in the proposal, so feel free to make the contact with them. And next, I will come to some challenges in developing the LGR proposal. The first challenge would be very small, no standup, for writing the Lao context. For example, the corrector, zero EBC, which is as [inaudible] said before.

So, this corrector in the grammar is only used with [inaudible], to like, to make the combination consonant, but in the reality, people can use it really often with many, many corrector, like program, electronic, and so on. So, this, we have spent more time to discuss about this before finalizing the rule.

And also for the site, zero ECC, this one is already some [inaudible] cut out, but still induce for some specific term like for the person names, or surname. So, the last GP has decide to keep this site for the root zone. The second is the complexity of the syllable of the writing structure, especially in case of the three corrector together can form the main consonants.

This has happened really [inaudible], but still induced for some word, like a Vietnamese name, and so on. So, we decided to keep it because it's still in use. And the next is the tool after Lao come together to form the [inaudible]. This [inaudible] only one case happen, but this word, this [inaudible] is really famous, in use. That's why we keep it in our proposal.

There is some lost challenge, like the case of representation mark and combine like [inaudible], same as in Thailand, but I have prepare for this.

So, next will be our current [focus?]. As [inaudible] said, we are finishing the submission and currently open for public comment. So please come and give your comment on the LGR proposal.

We will appreciate and happy to see your comment and association.

And, for the time [inaudible] that LGR, the Lao LGR proposal, will be integrated into the subsequent version of [inaudible] by the end of December. However, again, we are looking forward to see some public comment and suggestions to fulfill our proposal.

That's all for my presentation. Thank you.

UNKNOWN SPEAKER: Thank you. And now we'll move on to our next presentation.

So, we have an update from [inaudible], who is going to be joining us remotely, to present on Chinese generation panel. [Inaudible], can you hear us?

UNKNOWN SPEAKER: Yes, I can hear you. Can you hear me?

UNKNOWN SPEAKER: A bit faintly. Can we get the volume up?

Can you speak closer to the mic and a bit louder so that we can hear you? We can hear you, but you're just a bit faint.

UNKNOWN SPEAKER: Okay, is this better? Hello? Is this better?

UNKNOWN SPEAKER: Yeah, this is much better. Thank you. So, please, go ahead. Can you control the slides?

UNKNOWN SPEAKER: Yeah, thank you. [Inaudible] I would like to make the presentation about the updates from the CDP for the past three, no, for the past four months. Yeah, thank you. Next, please.

Yup, the [inaudible] actually, in June, the CDP submit a proposal, a proposal draft to the [inaudible], in which the size of the repertoire is [inaudible]... The basic [inaudible] is from the [inaudible]... character sets, and plus the characters from the numeral [inaudible] come use from the [inaudible] and from the [dot Asia?]. Next, please.

And in October, PDP [inaudible] proposal draft version two, in which the number of repertoire can [inaudible], the basic [inaudible]... Chinese [table?]. And we [inaudible] the Chinese table, we noticed that there were 108 24 characters [inaudible] were not included. So we add those 124. And for the, [inaudible] in dot Asia, there were 18 [inaudible] for common use.

And also, we have 43 characters, which are [inaudible] and also, as well as [inaudible] or KGP characters, and have, you know, various relationships with the two points [inaudible] in dot Asia characters. So, we took all of these characters from [inaudible] and 19,746.

And there are two [inaudible] characters that are not included in the [inaudible]. Yet, we suppose that these two characters will be included in the next version of the MSR.

Next, please.

And for the [dot Asia?] characters, for the 124 characters [inaudible] not included in the [inaudible]... two characters are listed in the slide. Some of them are also requested by Hong Kong, by [inaudible] on behalf of Hong Kong community, to add into the PGP next version [inaudible].

Next, please. And also in dot Asia, tables. There are 62 character points, which offer the supplementary ideographic plan, which is plan two in the Unicode. They're hard to [inaudible] in some system, and some application systems. So, I just include them into the [inaudible] version of CGP request, but I'm not sure if they, you know, considering about this problem, I'm not sure if we need to keep them in the final repertoire.

[Inaudible]

The third part is 18 characters from the [inaudible] for common use listed here. Next one, please. The final part is from the, you know, the [inaudible] from the [inaudible] or KGP, for these [three?] characters. Next, please.

Okay. Now, we have the current version of CGP repertoire, 19,746. The overlapping register between KGP and [inaudible] are listed in the slide. You can see there are about 4,000 overlapping components with KAP repertoire and CGP repertoire, and about 4800 code points [inaudible] between K and C.

Thank you. Next one please. In the first version of CGP proposal draft. CGP forget to add the subtype of R [inaudible], so that caused a [inaudible], especially for [inaudible] to use the whole proposal. So next slide, please.

So, in the second version, we just add the [inaudible] back into the XML. Next one, please.

This is an example to show how we add [inaudible] into the XML. Next one, please.

So, when we have the repertoire, the next work is to define the variance of every single component. I will notice that, besides the original [inaudible] mapping, and 172 [inaudible] characters, which are from [inaudible] and that would be the [inaudible].

And then the [seven?] dot Asia characters, which just got viewed in September. There are 59 planned [inaudible] characters, and 62 [plan two?] characters, whose [inaudible] but different from the dot Asia via mappings. [Inaudible] and dot Asia delegates, or representatives, which was [Edmund Chung?].

We discussed this issue in [inaudible] meeting, and [inaudible] meeting, the dot Asia, they agree to accept the count variant settings in [inaudible], and in their past work from the [inaudible]. Thank you. Next please.

The next issue is about the [acceptable?] variants, proposed by K. So the [KGP?] raised up a question that there are about 200, 259 variant groups, [inaudible]. So, we just had a coordination meeting in late September. We increased the number of unacceptable variants from the 259 to [60?].

So in the next IETF meeting, held in next week in Seoul, the K and C will [inaudible] coordinating on this [inaudible] on this less than 40 variant groups. We hope that we could reach the final conferences in the final IETF meeting. Next please.

Another issue is that whether we should, or how we could limit the number of allocate able labels, I mean [inaudible] the motivation why we should limit the number of labels, and will read the feedback from the IP to the first [inaudible] and propose the drafts. Next one, please.

But we found that it is very hard to define any rules to reach the applicant goals. To have what they desire. What kind of variant labels do they desire. We give up a current example to [inaudible] suggestion to eliminating variant mappings. We think we cannot, you know, simple eliminating variant mappings will cause problems for applicants, when they desire some specific labels.

So, we provide another suggestion in proposal draft of version two. Next, please.

The next slide, please.

Which is, if it's possible for us to [inaudible] times, if we could do that... Actually, the [inaudible] proposed a similar process in their [inaudible], in their letter too. If we can do that, you know, as a compensation, we could figure out a new kind of subtype. We call it market pole, simplified, or [inaudible].

So, they could be blocked. When you're on LGR, but the... For the applicants could get multiple [inaudible] labels by running the LGR for multiple times. [Inaudible] they prepared a letter to explain the motivation, and then the rationale of this idea. And [inaudible] same concern about this, about this issue. So we agree with [inaudible] to give the [inaudible] to IP, to hope the IP could give us, provide [inaudible] back to this letter, to this idea

of the multiple execution or complimentary process for the [inaudible]. Next one, please.

This is an example of [inaudible] rules, if we import this new subtypes. We call them multiple simplified, or multiple traditional. Next one, please. For next step. For the issue for [CDP?] is reduce the letter of [inaudible] and mappings to K.

We have, I think we have a [central?] committee in September, we have reduced the number from 259 to 50. And we hoped that we could have this done, this work done, in the next week, in the IETF meeting. The second work is [inaudible]... variant mappings.

I suppose that for K and for C, there might be dozens [inaudible] maps, we cannot, you know, agree with each other. So, we should find out conflict on the rules, to handle them, to include them from the repertoire, or accept them as independent characters. We will discuss this issue in the next ICANN meeting.

The third work is to limit the number of [inaudible] labels. We still need to [inaudible] with the J and with IP, of course, to on how to [inaudible] of limited number of [inaudible] labels. I mean, the IP proposed to eliminating the multiple variant mappings, but we [inaudible] that so we proposed another option. We do not know the, this option works for IP, so we need to communicate with IP.

And next one, please. So, actually I just received a letter, a feedback letter, from IP in which the [inaudible] the PDP is currently open to provide the rationale or the, why we need so many Chinese characters. I mean, [inaudible] repertoire, the number of repertoire is 19,446. It's almost the same as the size of MSR2.

So, I know it's a big number, and we are, we make the size of the repertoire so big because we just accept all the original [inaudible] from the [inaudible] cable, and also from the dot Asia and some other steps.

I will discuss the [inaudible] and the dot Asia [inaudible] if you should, you know, reduce the number to about 10,000, based on the historical [inaudible] registration [inaudible] and the table from the memorized [inaudible] for common use.

Still, the members of the [inaudible] registrars, they hope we could make a CDP repertoire, which is [inaudible] as a PDP repertoire, which we will make the second level domain registration easier in the future. So, it...

UNKNOWN SPEAKER:

Sorry [inaudible], we have to interrupt. But actually, we're running a bit late, so could I request you to please conclude the presentation? Thank you.

UNKNOWN SPEAKER: Yeah, thank you. Yeah, that's... I mean [inaudible] the new feedback from the IPs. [Inaudible] think about it, and keep talking and working with the [inaudible] guys to see if we could, you know, reduce the number. Thank you. That's all.

UNKNOWN SPEAKER: Thank you. We are running a bit late. We started a bit late. I realize that, so we may actually go a bit over time, if that is okay for the attendees. And we have the room. But without further delay, let me pass it on to [inaudible] to present on behalf of the Japanese generation panel.

UNKNOWN SPEAKER: So this is [inaudible] from JPR. JJP on behalf of [Hiro] of the JJPR.

Okay. So this is a brief update from JPR, JJP. So, the most recent meetings we had, the [inaudible] coordination meeting in Marrakech, they... Beijing, and Helsinki, and Taipei.

And JJP, Japanese, and Chinese, and Koreans, have the same script [inaudible], so we have to coordinate how to unify our variant. So we are [inaudible] having a coordination meeting.

Okay, so skip. And JJP already decided [inaudible] characters. And we decided, we originally had no variant.

[Inaudible] is very simple. And but, we are going to import variant from China and Korea. So, JJP, we provide, what happened? JJP, we provide variant which is imported from C and K. So, we have to decide the subtype for each variant.

So, we are going to minimize the [inaudible] variant, but it is difficult to predict which characters are to be allocated, and which is to be [inaudible]. So as, when we said JJP also want to have a multiple execution for the LGR. For if one applicant wants to have two or more, maybe two or three, the variant labels as they are delegated.

So, such process should be defined, conjunction to original process, but it is out of bound of LGR itself. So, this could be more difficult thing, but we are now talking with ICANN and CGK. Okay, that's all.

UNKNOWN SPEAKER: Thank you. And let's move on to the next presentation. Professor Kim, who is the chair of the Korean GP.

KIM: Okay, thank you. I'm Kim [inaudible], and I'm Korean GP chair. Let me see.

Okay. It is composed of four part introduction, and list of [inaudible] syllables and [inaudible] characters. And review of variants and timeline. Characters include for Korean GP, Korean label, is both [inaudible] and [inaudible] characters.

And we published for the [inaudible] point five in September 28th of this year. And it has 11 [inaudible] syllables, and about 48, 1900 characters. We have 50 variant groups in the [inaudible] character set. The number of variant groups will probably change, increase according to the discussion and conclusion between KGP and CGP.

And for the list of 100 syllables, it is from 8800 to 783, and the list of [inaudible] characters, is a union of five sources, as shown here. And the total number of [inaudible] characters is 4819.

And in Chinese LGR, published in July this year, there are about 3500 variant groups. And we analyze this, and found that there are about 304 Chinese variant groups, where there are two or more characters. In this case, Korea and China need to review the variant groups carefully, for the remaining 3200 variant groups, since there are no character in the variant group, or just one character.

C in Korea didn't actually build those variant groups. Among 304 variant groups, where there are two or more Korean characters in Chinese variant groups, Korea first classified 46 as acceptable, and 258 as unacceptable. KGP and CGP are discussing to decrease the number to 58 to 50 or so. There are a few [inaudible] scenarios, and I'm showing two.

Scenario one is, KGP accept CGP position. That is, CGP's variant group is maintained, and [inaudible] Korea says [inaudible] as independent, whereas China sees these two as variants. It is not clear. The result is that Korea and China, you see these two as variants.

In other words, KGP accepts CGP position. And scenario two, CGP accepts KGP's position. That is, CGP variant could be modified, and usually one character is removed from CGP variant group, and that character becomes an independent character.

So, for example, Korea says [C4 and C5?] as independent, and China says [C4 C5 and C6] as variants, and [C6] is not Korean character. The result is that Korea says [C4 and C5] as independent, and China says [C4] is independent. It has changed. And [C5 and C6] remain as variants.

And there is special [inaudible] of variant groups in Chinese variant groups. There are about 56 [inaudible] characters. That

is point from simplified and traditional character. Currently, the character is simplified character in China, however, the character has been used for a long time in Korea, China, etc. before [inaudible] announced simplified characters in 1964.

On the example that is shown here, in China this character is a simplified character, meaning [inaudible], and simplified from traditional [inaudible]. In Korea, these two characters are distinct. The first one, simple one, no, the less complicated one is desk. And traditional character is machine. So, it is very hard for [inaudible] to accept most variant groups containing one of those 56 [inaudible] characters.

The number 56 might be decreased by 10 or so, but still, we have 40 something. And here is an example, showing the place. And I will not explain details. CGK coordination meeting was held in Taipei, Taiwan in September this year. And the KGP and CGP tried to reduce the number of unaccepted variant groups.

It used to be 256. Now, we hope that it is less than 50 or so, but it is not finalized yet. Okay, this activity [inaudible] I'll not quote it. Thank you.

UNKNOWN SPEAKER: Thank you Professor Kim. So, we are running a bit late on the session, but happy to take one or two questions before we end.

UNKNOWN SPEAKER: Yeah. This is [inaudible]. I just wanted to ask [inaudible] or Michele, regarding one of the points that was there [inaudible] about variants, should not help context. So, I just wanted to know what exactly you mean by that.

UNKNOWN SPEAKER: This is [inaudible] replying. The RFC 7940 has the provision that you can apply a context rule to a variant mapping. That is, if a code point, for instance, is at the end of the word, you can say, or at the end of the label, you can say at that location, it is a variant of another code point. But if they occur anywhere else in the label, they're not variants of each other.

That is a provision that it's in RFC 7940. It turns out that there are quite a number of complications in designing a LGR that uses context rules and variants, that are not very easy to get right. And as a result, the integration panel strongly discourages anybody from trying to design a LGR, that uses context rules and variants.

This is probably a situation where if somebody comes and can demonstrate one, that it is impossible to write a reasonable LGR without that feature, and two, that the proposed solution is

actually a stable and secure and whatever, we might at least discuss it.

But absent those two factors, we wish you would not even think of that as an option. So, that... To reiterate, those are context rules on whether the variant mapping exists in the particular location on the label, which is different from context rules on code points, which define whether a code point, in the applied for label, may occur at a certain location or not.

That's a much more common case, we understand it much better in the evaluation is much less dependent on subtle effects. So it's much easier to write a LGR that has context on code points. And that's the normal case, and we would certainly expect that scripts from India have many context rules on code points, but not on variance.

UNKNOWN SPEAKER: So, actually this room is needed. So, let's close this meeting, thank you very much, and sorry for starting a bit late due to technical issues. We will have, we can take the rest of the discussion maybe outside the room. Okay. Thank you.

[END OF TRANSCRIPTION]