Guidelines for Developing Script-Specific Label Generation Rules for Integration into the Root Zone LGR

Version 2015-04-24

Table of Contents

A. Introduction and Background

A.1. Purpose of this Document

This document provides an introduction to the basic tasks of a Generation Panel for a specific script. As such it offers orientation and guidance, rather than detailed and explicit instructions. Such closer details can often be found in the related documents listed at A.2.2

This document does not supersede the [Procedure] as the authoritative definition of the process for developing and maintaining a label generation ruleset for the Root Zone. Instead, it intends to give practical guidance as well as a focused overview of the tasks involved in creating a single-script LGR for integration into the Root Zone LGR.

All examples given in this document are hypothetical.

A.2. Conventions and Background Documents

There is a bibliographic list in section D. Documents are referenced by a short name put in square brackets. Publications in the Request for Comments series are referred by their RFC number, even if they are part of some other series (BCP, STD, and so on). Unicode documents are referred to following the Unicode convention, where the number sign is included in running text but not when used in a reference (so, "UTR#36", but "[UTR36]").

The following are prerequisites for understanding this document:

A.2.1. Background Documents and Terminology

- Procedure to Develop and Maintain the Label Generation Rules for the Root Zone in Respect of IDNA Labels [Procedure]
- "A Study of Issues Related to the Management of IDN Variant TLDs (Integrated Issues Report)" [IIR];
- IDNA2008 [RFC5890] [RFC5891] [RFC5892] [RFC5893] [RFC5894] [RFC5895];
- Unicode [Unicode63];
- "Principles for Unicode Code Point Inclusion in Labels in the DNS." [IABCP].

In addition, the terms defined in Appendix 2 of IIR are incorporated here by reference, and not reproduced. *Some of the terms defined in that Appendix 2 are used in a special or peculiar way*, and the text below is unlikely to be understood completely without having that terminology to hand. (We have not followed the capitalization convention of IIR, because some readers found it confusing, but the terms are otherwise the same.)

A.2.2. Related Documents

The following documents are referred to by these guidelines and are expected to be consulted heavily during a Generation Panel's work.

- Considerations for Designing a Label Generation Ruleset for the Root Zone [LGR-Considerations]

  https://community.icann.org/download/attachments/43989034/Considerations%20for%20LGR.pdf

- Maximal Starting Repertoire (MSR-2) [MSR]

  https://www.icann.org/news/announcement-2-2015-04-27-en

  https://www.icann.org/en/system/files/files/msr-2-overview-14apr15-en.pdf

- Representing Label Generation Rules in XML [XML-LGR]

  https://tools.ietf.org/html/draft-davies-idntables

- Requirements for LGR Proposals [SubmissionRequirements]

  https://community.icann.org/download/attachments/43989034/Requirements%20for%20LGR%20Proposals.pdf

- Variant Rules [VariantRules]

  https://community.icann.org/download/attachments/43989034/Variant%20Rules.pdf

- Whole Label Evaluation Rules [WLERules]

  https://community.icann.org/download/attachments/43989034/WLE-Rules.pdf

A.3. Label Generation Rules

For the purposes of this document, the label generation rules contain four parts:

1. the rules governing the permissibility of Unicode code points (the repertoire),

2. any exchangeable code point variants that follow from those (the variant rules),
3. the status of any resulting label, and
4. a set of optional whole-label evaluation rules that determine whether the output of the previous three portions is still an acceptable label in the root zone.

In parts of this document, we use "LGR" in a more informal sense, to refer to some subset of the total set of LGR. In addition, we sometimes use "LGR" to refer to *proposed* rules, which might be in contention and might not finally become part of the root LGR. The key thing these have in common is that they are all rules intended to govern which labels are permissible in a zone (in this case, the root zone).

A.3.1. How Will Label Generation Rules be Used?

Fundamentally, the Label Generation Rules developed under this process define what IDN labels will be valid for the Root Zone. When a label is applied for, it is for a given script. Each application is confined to the repertoire identified by a single script label. Generally scripts do not overlap, but in some cases, a label may be valid for more than one script. In that case, the variant rules may differ between scripts and the choice of script in the application decides which set will be applied to determine the status of any variant labels.

The LGRs will be expressed in a form that can be used for automated label validation. For some scripts, the only check that needs to be performed is whether the label is formed by using the repertoire specific to that script.

Some scripts' repertoires contain variants. Variants, as this term is used here, are code points or code point sequences that are "the same" as other code points or code point sequences in the minds of users of at least one community.

The LGRs for these scripts may specify variant mappings that will be used to automatically generate all permutations of the applied for label (together with the original label, these permutations form the set of variant labels).

The type of a variant mapping determines whether

- to ***block*** a variant label resulting from it (only one label in the variant label set can be allocated) or
- to allow ***allocating*** it to the same applicant as original label

As result of integration, blocked variants can exist between script- specific repertoires. Even if a label is applied for in one script, the existence of certain other labels in another script may block it from being delegated. This is because the Root Zone is a shared resource.

For example, both .pax and .pax would be well-formed TLDs in Latin and Cyrillic, respectively, if not equally meaningful as words in both scripts. Assuming the requisite cross-script variants were defined, if one of these two labels is applied for and allocated, the other would be blocked. Otherwise, both could be independently allocated to different applicants causing user confusion.

Finally, the LGR for some scripts may contain additional Whole Label Evaluation Rules (WLE) that are used to automatically suppress certain undesirable code point combinations, even when the code points by themselves are part of the script's repertoire.

B. Developing the LGR

B.1. Summary

As defined in the [Procedure], script-specific LGRs are developed by community-based panels called Generation Panels (GP). They base their work on an initial repertoire created by the Integration Panel (IP), called the Maximal Starting Repertoire [MSR].

The basic steps that a Generation Panel would follow in preparation and submission of a proposed LGR can be summarized as follows:

1. Start with the most recent MSR
2. Create a repertoire based on the selected script
3. Are variants needed for the script? If yes:
   a) define variant relations
   b) decide which variants should lead to ***blocked*** vs. ***allocatable*** labels
4. Must some labels be prohibited via Whole Label Evaluation rules (WLE) ? If yes: define applicable WLE rules.
5. Document the decisions and their rationale
6. Create a XML file for repertoire, variants and WLE
7. Submit for public comment and IP review

B.2. Creating a Repertoire

B.2.1. Root Repertoire

The Repertoire for the Root Zone will be the superset created from the collection of single script repertoires. Each contributing repertoire will be tagged by script, using a [BCP47] language ID with the language set to undefined, as for example in "und-Cyrl" or "und-Jpan".

No cross-repertoire labels will be allowed and the repertoires generally will not overlap, except for any "common" or "inherited" code points, or the Han ideographs (which will be part of both "und-Hani" and "und-Jpan", for example).

Each of the script-specific repertoires will be limited to stable code points from modern, widespread, everyday use as described in the next subsection.

B.2.2. Creating a Script-specific Repertoire

Starting with the latest [MSR] the Generation Panel selects from the MSR all the code points needed for a selected script (the script defined in scope of GP). Acceptable codes will be either those code points that include that script indicated in their Unicode Script Extension Property value [UAX24] or will be certain other code points (such as diacritics) that are used with the script in question and have one of the script property values *Common* or *Inherited*. For convenience, the MSR tags all code points with the applicable script values.

From the intersection of the MSR and the script-based repertoire, the Generation Panel next selects the code points needed for IDN TLDs in that script.

This selection is based on Principles laid out in the [Procedure], which in turn references [IABCP].

If a code point provides any significant systemic risks, or cannot be determined to be in active, common, widespread modern use, it probably lacks sufficient justification to be included.

For typical scripts, the final repertoire selected will be smaller than the starting set of the intersection of MSR and script-use code points. This is because the Integration Panel includes in the MSR many code points for which it was not possible to unambiguously determine that they should be excluded, so that the Generation Panels may make the determination whether they are eligible for the root zone after all.

In selecting the repertoire, the Generation Panel should strive to avoid geographical or language bias, that is, it should attempt to accommodate the needs of all user communities of the script as far as possible under the constraints for the root zone.

The Generation Panel is expected to give clear rationale for inclusion. This rationale should cover each code point or collection of code points. It is not required that the rationale documents ***all*** uses of a code point; what is required is that it can demonstrate that its inclusion is justified because it is being used for everyday, general purpose

writing by at least one of the communities in a stable and widespread manner[1], and that the code point presents no undue risks or other issues.

The [Procedure] adds the requirement that the discussion by the Generation Panel demonstrates how the selections it made ultimately satisfy the Principles.

A few scripts have straightforward repertoires, for example because they are used for a single language, and there are no code points that pose any systemic risks. In these cases, the selection process, but also the documentation of the justification and rationale become much simpler tasks.

B.3. Are Variants Needed?

One important question a Generation Panel has to answer is whether the repertoire contains variants (see A.3.1). Not all repertoires contain code points that are variants of each other or of some code point sequence; therefore, some GPs will answer "no" to this question. In that case, they can skip steps related to variants.

If variants are needed, the task of the Generation Panel becomes to define which code points or sequences are variations of each other, and to create a list of variant mappings and to identify their type. As described earlier, there are two types of variant mappings. Some lead to labels that are blocked, which means that two labels that are so closely equivalent as to be functionally the same cannot be allocated simultaneously – only the first one can be allocated. The others lead to labels that can both be allocated, but only to the same applicant.

B.4. Defining Variants

The [Variant Rules] document describes variants and their definitions in more detail. The following is highly abbreviated summary and only gives the highlights of the process of defining variants.

1. Define which two code points are variants: **A** ~ **B**
2. Create the variant code point mappings. They
   a) must be ***symmetric***:
      if we have **A** → **B,** we must also include **B** → **A**

[1] The Integration Panel has used the EGIDS scale of language vitality, published in the [Ethnologue] as a proxy to make an initial cut of code points in the MSR (see [MSR] for details). Generation Panels are normally expected to have more direct sources of information about language use.

   b) and ***transitive***:
      if we have **A → B** and **B → C,** we must also include **A → C**
3. Assign a type to each mapping, such as
   a) **A → B** (blocked)
   b) **B → A** (allocatable)

Note that the requirements in 2 above simply state that the variant mappings defined in the LGR must not have gaps: if a variant mapping is present, its symmetric correlate, for example, must also be present.

While the mappings must be symmetric and transitive, the types assigned to them do not have to be symmetric or transitive. Indeed, where **A → B** (allocatable) and **B → C** (allocatable), it is not uncommon to find **A → C** (blocked).

The values "blocked" and "allocatable" shown in (3) are the default type values for variant mappings. The default types work with predefined default actions provided in the [MSR].

Using more specific type values than "allocatable", may allow a Generation Panel to define an LGR that significantly reduces the number of allocatable variant labels that can be derived from variant code point mappings. Use of such non-default values requires the definition of specific actions that evaluate finally to either "blocked" or "allocatable" variant labels, so that ultimately there are only those two dispositions for the labels. An example of this is the use of simplified and traditional types for Han characters in Chinese, with actions defined to restrict co-occurrence of the two types in a single allocatable label.

As mentioned above, blocked variants can exist between different script- specific repertoires as result of integration. There is a provision that allows the proposed LGR to contain variant mappings to code points that are not in the repertoire, to facilitate the specification of shared variants in a symmetric and transitive manner, even where the repertoires differ (see [SubmissionRequirements] for details).

B.5. Constraints on Variants

The [Procedure] calls for Generation Panels to: "maximize the number of blocked variants, and minimize the number of allocatable variants".

There is a steep complexity cost to allocatable variants. If naively defined, they could easily generate dozens of allocatable labels, especially for labels that are more than a few code points long.

Therefore, it is imperative to find ways to limit to what is truly required in order to accommodate the fundamental needs of the writing system.

On the other hand, blocked variants can make LGR more robust, by restricting labels that are too closely equivalent to other labels. There is a commonly voiced concern about a potential cost of "too many" blocked variant code points. This may be misplaced, because blocked variant code points only affect labels that are otherwise the same. The longer an applied for label is, the more likely it is to differ in at least one position from all variants of an existing label (which would make their variant label sets distinct so both can be allocated). Even if a considerable percentage of code points in a repertoire were to have blocked variants, the total space of permissible labels may not be unduly impacted.

B.6. What, Why and When of WLE Rules

The final question a Generation Panel has to answer is whether some sequences of code points should be generally prohibited within a script.

Whole Label Evaluation (WLE) Rules are intended primarily to prevent labels that cannot be processed or rendered correctly, or to limit the application of combining marks to those code points required for the support of the orthography of one or more supported languages.

When should a Generation Panel consider adding a WLE rule?

WLE rules are not intended to enforce mere orthographic "spelling rules". These are irrelevant to labels, since their purpose is to be useful mnemonics and they are not required to be well-formed words in any specific language. Rather, WLE rules identify contexts where certain code points cannot be rendered properly, where they are redundant, or lead to labels that are readily confusable with other sequences of code points. Possible examples include combining marks, including combining vowel marks at the start of a label or following other vowel marks, markers of syllable structure that are out of place, and special digraphs in languages where they are usually rendered indistinguishably from sequences of code points.

WLE Rules are also not an appropriate method for filtering unacceptable or prohibited words. Such specific exclusions will be political, social or commercial in nature, and as such are better handled by some layer of registry policy, which can be more sensitive, and flexible, than by WLE rules, whose outcomes will end up built permanently into the root system.

Both the [MSR] and the [Procedure] suggest specific examples that Generation Panels for the relevant scripts should investigate. Additional examples and details of how to design WLE rules can be found in [WLERules].

In determining whether a WLE is needed, a key consideration is whether such a prohibition would satisfy the Principles (See [IABCP] and [Procedure]). For example, would the increase in complexity be offset by reduction in risk?

In practice, it is likely that most Generation Panels would come to the conclusion that no WLE rules are needed for their script. But in case a Generation Panel decides to include a WLE rule, advance discussion with the Integration Panel is strongly recommended.

Unlike the disposition of variant labels, WLE rules must be common for the root and do not depend on the script. Nevertheless, any rules that act only on a non-overlapping script repertoire would effectively be script specific. Even in that case, the shared nature of the root makes it desirable that WLE rules for related scripts are functionally equivalent.

B.7. Defining WLE Rules

Whole Label Evaluation rules are in a way akin to regular expressions in that they specify a set of contexts which must occur (or are prohibited) in a label. WLE rules are used to identify labels that must not be allocated; whether the rule is formulated so as to require a context or to prohibit one is immaterial. That choice simply depends on which leads to a simpler statement of the rule.

Creating a WLE rule requires the definition of one or more “rules”. A rule is as follows:

- Optionally, a definition of “classes” or sets of code points, which may be explicitly listed or based on Unicode properties
- Definition of a structural context
- Defining an “action” that prevents the label from being allocated when any of the rule’s target code points are found in a prohibited, or not found in a required context.

The MSR contains a Default WLE rule which serves as an example, but further details can be found in the [WLERules] document.

B.8. Documenting the LGR

When the Generation Panel has decided on the Repertoire, defined any variants and/or WLE the Generation Panel documents its decisions and rationale for

- Choice of repertoire, coverage and contents
- Necessity, choice and type of variants
- Necessity and design of WLEs

Finally, the [Procedure] requires that the Generation Panel evaluate these design choices in light of the Principles and document the results of this evaluation.

The overview and rationale part of the Proposal is provided as an ordinary text document in English, but the formal definition of the content of the LGR is required to be submitted in a specific format using XML.

B.9. Formal Definition for LGR

The formal definition of the LGR is an XML file. The format for this is defined in [XML-LGR]. In understanding the XML schema for LGRs it is worthwhile to keep in mind two facts. One is that the [Procedure] was written before [XML-LGR] was completed and therefore the former does not reference any of the syntax elements defined in the latter. In fact, the procedure uses terms ("rule", "element") in ways that differ from how they would be used in an XML environment or in the [XML-LGR] schema in particular. Second, [XML-LGR], while used for the Root Zone LGR, is not limited in its design to features supported in the Root Zone. The use of the XML format for the Root Zone LGR process is described in [Requirements for LGR].

In addition to the XML file for the content of the LGR the Integration Panel requires some data that help it review the LGR proposals. These include examples of labels, variant labels and labels blocked by WLEs. These example labels are only needed if the LGR contains variants or WLEs

Finally, a Generation Panel may optionally create informative charts or code point tables of the LGR repertoire, as was done by the Integration Panel with annotated code tables that are part of the [MSR].

B.10. Submission for Review

When a Generation Panel has completed a proposed LGR it will be submitted to ICANN for release for public comment and to the Integration Panel for initial review. The format for the submission must match that described in the [Requirement for LGR] document.

In order to streamline the process, it is anticipated that upon submission there will be a quick review of the proposed LGR for any obvious issues, such as completeness of submission or errors in submission format. This will allow the Generation Panel to address these issues before the proposal is released for public comment.

After public comments have been resolved by the Generation Panel, the Integration Panel will attempt to integrate the LGR. As part of that process, the Integration Panel will review the contents and rationale presented and make sure there are no unresolved issues. This process requires a unanimous decision by the Integration Panel to accept the proposed LGR.

If the integration is successful, the Integration Panel will submit an integrated LGR for public comment (consolidating several scripts). After any public comments have been resolved, that integrated LGR will take effect.

If the integration is not successful, and the Integration Panel cannot reach unanimous agreement on accepting the proposed LGR, the Integration Panel will reject the proposal and tell the Generation Panel which features prevented integration.

B.11. Throughout the Process

Throughout the process, Generation Panels are encouraged to keep the Integration Panel informed and allow it to raise issues and questions well before the public review. The benefit of such a two-way interaction before the release of the LGR proposal is that the chance of rejection over an unanticipated issue is reduced sharply, and that the Generation Panels can improve the documentation and justification supporting their choices. In turn, this makes it easier for the Integration Panel to achieve the required unanimous support on all aspects of the proposal.

Generation Panels are expected to follow the prescriptions laid out in the [Procedure]. Despite the existence of these guidelines and other supporting documents, the Procedure remains the authoritative prescription.

The proposed LGRs created by the Generation panes are expected to be compatible with the Principles described in [IABCP] and the [Procedure], and the Generation Panels are expected to document the extent to which their proposal satisfies these principles. (There is some tension between the principles, so not all will be able to be fully satisfied simultaneously).

The list of Considerations for Designing an LGR for the Root [Considerations] attempts to help Generation Panels in weighing the factors for their decisions and evaluating them against the principles. Essentially, they provide a helpful list of items for GPs to consider during the process. Generation Panels are encouraged to refer to them, but are not required to do so.

B.11.1. Coordinate with GPs for Related Scripts

Where scripts are related, in accordance with the [Procedure] and the Principles, the Integration Panel will look for consistent and compatible treatment of repertoire, variants and whole label evaluation rules.

If Generation Panels coordinate among themselves, there is less of a chance that the Integration Panel will run into issues of inconsistencies or outright conflict between LGR proposals during Integration.

The ultimate goal of such coordination is to arrive at a consistent user experience in the heterogeneous linguistic environment represented by the Root Zone.

B.11.2. What should be Coordinated?

The following gives some brief suggestions and a few examples of issues that would benefit from coordination

- Repertoire

  Examples for the consistent treatment of similar repertoires might include a consistent approach to defining the repertoire for the neo-Brahmi scripts of India, since all have traditionally been arranged on a common (phonetically based) plan.

- Variants

  Example for the consistent definition of variants might include the mutually compatible definition of variants for the various script LGRs sharing the Han repertoire. This would include agreement on what is a variant, but does not include agreement on whether such variants would lead to blocked or allocatable labels.

- Cross-script homoglyphs

  The Cyrillic, Greek and Latin scripts provide an example of related scripts where a large number of cross script homoglyphs exist. A consistent treatment of these across scripts might be achieved by coordination.

- WLE

  Examples of consistent treatment of structurally similar scripts might include a consistent definition and handling of the *akshara* structure of Indic scripts.

B.12. Need, Limitations and Mechanisms for the Root Zone LGR

Any LGR for the Root Zone must succeed in defining and supporting the use of meaningful subset of all the possible words and word-like mnemonics (those not limited to actual words) writable in the given script.

If code points or their combinations are judged likely to promote confusion or present a risk to security in the way labels are rendered they must be excluded. This may lead to the exclusion from the set of possible labels of some words which are well-formed within some languages written with this script.

The possibility that some well-formed words are excluded is not – in itself – a valid argument against the decisions that led to these exclusions, as long as the exclusions also have beneficial consequences such as rendering the use of the script for TLDs as more transparent, easier to understand or more secure. For example, both English and French use the apostrophe in many common words, but its exclusion is not contested.

B.13. Limitations of the LGR

According to the [IABCP] TLDs are intended for "*unambiguous labels with good mnemonic value*". They are not intended to capture all facets of a writing system. The [Procedure] makes this very explicit:

> "This … may on occasion... result in the ... exclusion of ... possibly useful labels. It is nevertheless appropriate in the root zone, where the goal is not to maximize the number of possible labels but to minimize the confusion possible in a shared environment supporting heterogeneous linguistic communities."

The design of the Root Zone LGR, and of all the script-specific LGRs that are integrated into it should focus on modern, every day use. Therefore, it is not a failure if the LGR does not support some particular conventions, even if they are not actually rare.

Some limits necessary to reduce systemic risks, but they prevent complete coverage even of common languages. For example, by disallowing the apostrophe it is not possible to support the 's ending for names of businesses common in English. Likewise, by disallowing the hyphen it is not possible to spell hyphenated words or names. This had been the case even before IDNs were added to the root.

C. Generation Panels' Use of the MSR[2]

As stated in the [Procedure], the Integration Panel is" *tasked with establishing the* ***maximal set of code points*** *(see Section B.5.3.2 of the Procedure) and* ***default whole label variant evaluation rules*** *(see Section B.5.5. of the Procedure) for the root zone,* ***which serve as starting point for the generation panels***" (emphasis added). These constitute the MSR. The MSR and the [Procedure] are used by the GPs as the starting point for their work.

This section gives additional guidance and direction for the GPs when evaluating the MSR. It assumes that the reader is familiar with the [Procedure].

C.1. Repertoire

As stated in the [Procedure], "*The generation panel's starting point is a subset of the maximal set of code points for the root zone. From that maximal set, the generation panel picks the set of Unicode characters used in the writing systems in question."*

The MSR is the fixed outer limit of the code point repertoire potentially available for the Root Zone LGR. Following the Inclusion Principle, the Generation Panels are expected to build their proposed repertoire "from the ground up" — positively affirming each and every code point in their LGR proposals. Code points that are not part of the MSR must not be included as part of the repertoire in an LGR proposal.

As stated in the [Procedure], LGR proposals for the root zone will be created on a per-script basis, with normally no script mixing, and, in particular, no mixture of Latin (ASCII) letters with other scripts. Therefore, the repertoire for any LGR proposal from a given GP is expected to be the intersection of the MSR and the set of code points associated with the script in question. There are some exceptions based on the shared use of, for example, the Han script; including the way the Japanese writing system uses a mix of Hiragana, Katakana and Han code points while being treated as the script "Jpan", based on the script code defined in the ISO 15924 registry.

Finally, some code points in the MSR formally have INHERITED (Zinh) as their Unicode script property value; these code points will normally be eligible to be part of the repertoire of any script for which their use is required.

For convenience of the Generation Panels, the XML file containing the normative definition of the MSR identifies the script of each code point. A limited number of code points may be used with multiple scripts.

[2] The text of this section has been excerpted from the MSR

As required by the Inclusion Principle in the [Procedure], the Integration Panel expects the Generation Panels to justify the inclusion of every single code point in their proposed repertoire. While the Integration Panel may accept a summary justification for the core alphabet(s) in a script, the less common characters and sequences will have to be documented individually.

Adherence to these guidelines has the effect that the Inclusion Principle and Conservatism Principle from the [Procedure] may be fully applied to the LGR; nevertheless, even though the MSR (being an interim step) will include code points that, after further review by the Generation Panel, or after final review by the Integration Panel, are found not to satisfy these principles and therefore will not be part of the final, integrated LGR.

Some code points included in the MSR have ambiguous status or are potentially problematic for the root zone, but were included in the MSR expressly for the purpose of allowing the proper Generation Panel to research them. These include, but are not limited to the code points mentioned as problematic or ambiguous in Section 5 of [MSR]. Generation Panels are advised that while inclusion of any code point into the LGR requires an affirmative decision under the Inclusion Principle, any potentially problematic code points are expected to meet particularly high standards of justification before they would be acceptable to the Integration Panel for inclusion in the integrated Root Zone LGR. Generation Panels that intend to submit such code points in their LGR proposals are encouraged to discuss this choice with the Integration Panel before submission.

C.2. Variants

In addition to deciding on a repertoire, the Generation Panels must decide whether any variant relationships between code points exist, and if so, must specify them. For purposes of the Root Zone LGR, each code point variant must have exactly one disposition value; from these the disposition of any variant label containing them is calculated. How variants are specified in the XML format [XML-LGR] is beyond the scope of this document.

For each variant, the Generation Panel must make a determination about whether the presence of one variant character in a label will block another label that has the other variant code point (blocked variant), or whether the second label could be allocated later (allocatable). Note that assigning a disposition of “allocatable” does not mean that the second label will actually be delegated in the root zone, only that such a delegation may happen; as indicated in the [Procedure], ICANN is currently in the process of determining how “allocatable” labels will be handled.

In contrast, the effect of blocked variants is completely predictable. Because that effect prevents delegations, it can be argued that blocked variants tend to make the DNS more, not less robust - and are thus in many cases the more conservative alternative, even compared to not defining a variant relation at all. On the other hand, allocatable variants (to the degree they are delegated) do impact the DNS and its users and the conservative choice is to minimize the number of delegated variant labels. Generation Panels should consider how the Conservatism principle applies and how this affects the decision to define variant code points as allocatable.

Generation Panels considering defining variants should carefully review all sections of the [Procedure] that concern variants. Appendixes E and F of the [Procedure] give useful, non-normative examples of how variants might be assigned. Finally, in Section A.3.3, the [Procedure] states:

> "It may be argued that the LGR process should be set up to minimize the number of variants defined. The benefits of a strictly minimal variant set apply only to those variants for which the returned disposition would be "allocatable". From the Conservatism Principle (if not others), it follows that the number of allocatable variants should be minimized. But the LGR process is also a way to identify all those variants that should be unambiguously blocked from allocation. Instead of minimizing the set of blocked variants, it would appear possible to simplify the evaluation of new candidate labels by maximizing the generation of such labels, thus removing them from the set that must be subject to case-by-case analysis. In other words, the output of this procedure should aim to maximize the number of blocked variants, and to minimize the number of allocatable variants."

C.3. Restrictions on Combining Sequences

Some combining marks are used properly only in a very small number code point sequences for a particular script. A GP for such a script needs to evaluate the utility of each combining mark. Limiting the acceptable combinations of a combining mark to a small subset of characters is likely to be justified by the Conservatism Principle. Such limitations need also to be considered in light of the Simplicity and Predictability principles:

Simplicity: Overly complex rules are to be avoided, in favor of rules easily understood by users with only some background.

Predictability: People with reasonable knowledge of the topic should, by and large, reach the same conclusions about which code points should be included.

If a combining mark can be used sensibly with only a few characters, the Generation Panel may decide to add only the allowed combinations to the LGR, which would limit

the use of the combining mark. On the other hand, if a combining mark is used with a wide variety of characters, the Generation Panel may decide to add the combining mark by itself to the repertoire but then needs to provide proper justification for allowing arbitrary combinations.

Complex rules that would allow a combining mark based on complicated context (other than fixed sequences) would likely run afoul of the Simplicity Principle; although something like a requirement for well-formed syllables might be appropriate for some scripts[3] in light of the adverse effects of such ill-formed syllables. Nevertheless, the least complex formulation of such rules should be aimed at, even at the loss of some linguistic fidelity (see [WLERules]). Any intention along those lines should be discussed with the Integration Panel ahead of time.

C.4. Whole Label Evaluation Rules

All LGR proposals by Generation Panels must include the default WLE rules from the MSR. They may include additional WLE rules (expressed in the XML representation) as long as they satisfy the principles in the [Procedure] and are appropriate for the root zone. If the same label can be formed using different scripts' LGRs, all WLE rules affecting it must lead to the same result. Generation Panels are advised to discuss any tentative WLE rules with the Integration Panel before submitting them as part of an LGR proposal.

C.5. Coordination between GPs

To allow the Integration Panel to create an integrated LGR for the root zone requires that proposed LGRs for related scripts are available so they can be reviewed together. Attempts to integrate each proposal in isolation would create unacceptable risks of incompatibilities and risks violating the *Stability Principle and* the *Least Astonishment Principle*. This has some straightforward consequences for the work of GPs covering related scripts. As stated in the [Procedure]:

> *"Panels for* ***related or structurally similar scripts*** *are encouraged to communicate or cooperate in the interest of arriving at a more consistent treatment of repertoires and variants for the root zone."* (Emphasis added).

Ideally, GPs for related scripts would be active at a similar phase of development and coordinate their efforts, so as to resolve any issues arising out of the relationship between the scripts in question. To facilitate the procedure-mandated dialogue

[3] These scripts include the neo-Brahmi scripts of India, as well as Sinhalese, Tamil, Khmer, Lao, Thai, Myanmar and Tibetan.

between the panels, GPs are encouraged to keep the IP advised of their plans for coordination and progress of such effort.

Each Generation Panel still submits a separate LGR per script. Even in cases of significant overlap (as between Chinese and Japanese use of the Han script) the coordinated repertoires may differ. (For example, the Chinese LGR would not be expected to include Japanese only ideographs in its repertoire).

Where there is an overlap between the repertoires, any variant mappings specified must be the same. However, whether a particular variant mapping (by its assignment of type) results in a blocked variant label or in an allocatable one may be different for each LGR.

There is a provision that allows the proposed LGR to contain variant mappings to code points that are not in the repertoire, to facilitate the specification of shared variants in a symmetric and transitive manner, even where the repertoires differ.[4]

Where a requirement for coordination between GPs may exist, a GP may submit a preliminary, not yet coordinated LGR if they would like ICANN to perform a review of the contents to determine issues that need to be addressed in such coordination.

D. References

D.1. Resources

[Ethnologue] Lewis, M. Paul, Gary F. Simons, and Charles D. Fennig (eds.). 2014. Ethnologue: Languages of the World, Seventeenth edition. Dallas, Texas: SIL International. Online version: http://www.ethnologue.com. Visited 2014-11-18

[IABCP] Sullivan, A., Thaler, D. , Klensin, J. , and O. Kolkman, "Principles for Unicode Code Point Inclusion in Labels in the DNS." draft-iab-dns-zone-codepoint-pples-00.txt. Work in progress. Available at http://wiki.tools.ietf.org/html/draft-iab-dns-zone-codepoint-pples-00. Visited 2012-09-21.

[IIR] Internet Corporation for Assigned Names and Numbers, "The IDN Variant Issues Project: A Study of Issues Related to Management of IDN Variant TLDs (Integrated Issues Report)." (Marina del Rey, California: ICANN, February, 2012). http://www.icann.org/en/topics/idn/idn-vip-integrated-issues-final-clean-20feb12-en.pdf.

[4] see [SubmissionRequirements] for details.

[ISO15924] *Codes for the representation of names of scripts*, ISO 15924:2004. Available from http://www.unicode.org/iso15924/. Visited 2012-09-21.

[LGR-Considerations]Integration Panel, "Considerations for Designing a Label Generation Ruleset for the Root Zone" https://community.icann.org/download/attachments/43989034/Considerations%20for%20LGR.pdf

[MSR] Internet Corporation for Assigned Names and Numbers, "Maximal Starting Repertoire (MSR-2)" (Los Angeles, California, April 2015). Announcement: https://www.icann.org/news/announcement-2-2015-04-27-en Overview: https://www.icann.org/en/system/files/files/msr-2-overview-14apr15-en.pdf

[Procedure] Internet Corporation for Assigned Names and Numbers, "Procedure to Develop and Maintain the Label Generation Rules for the Root Zone in Respect of IDNA Labels" (Los Angeles, California: ICANN, March, 2013). https://www.icann.org/en/system/files/files/draft-lgr-procedure-20mar13-en.pdf

[SubmissionRequirements] Integration Panel, "Requirements for LGR Proposals" https://community.icann.org/download/attachments/43989034/Requirements%20for%20LGR%20Proposals.pdf

[RFC3743] Konishi, K., Huang, K., Qian, H., and Y. Ko, "Joint Engineering Team (JET) Guidelines for Internationalized Domain Names (IDN) Registration and Administration for Chinese, Japanese, and Korean", RFC 3743, April 2004.

[RFC4290] Klensin, J., "Suggested Practices for Registration of Internationalized Domain Names (IDN)", RFC 4290, December 2005.

[RFC5646] Phillips, A. and M. Davis, Eds., "Tags for Identifying Languages", RFC 5646, BCP 47, September 2009.

[RFC5890] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework", RFC 5890, August 2010.

[RFC5891] Klensin, J., "Internationalized Domain Names in Applications (IDNA): Protocol", RFC 5891, August 2010.

[RFC5892] Faltstrom, P., Ed., "The Unicode Code Points and Internationalized Domain Names for Applications (IDNA)", RFC 5892, August 2010.

[RFC5893] Alvestrand, H., Ed., and C. Karp, "Right-to-Left Scripts for Internationalized Domain Names for Applications (IDNA)", RFC 5893, August 2010.

[RFC5894] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Background, Explanation, and Rationale", RFC 5894, August 2010.

[RFC5895] Resnick, P. and P. Hoffman, "Mapping Characters for Internationalized Domain Names in Applications (IDNA) 2008", RFC 5895, September 2010.

[UAX24] UAX #24: *Unicode Script Property.* An integral part of The Unicode Standard. Most recent version available from http://www.unicode.org/reports/tr24/. Visited 2012-09-21.

[Unicode63] The Unicode Consortium. The Unicode Standard, Version 6.3.0, defined by: "The Unicode Standard, Version 6.3.0", (Mountain View, CA: The Unicode Consortium, 2012. ISBN978-1-936213-08-5). http://www.unicode.org/versions/Unicode6.3.0/.

[VariantRules] Integration Panel, "Variant Rules", https://community.icann.org/download/attachments/43989034/Variant%20Rules.pdf

[WLERules] Integration Panel, "Whole Label Evaluation (WLE) Rules" https://community.icann.org/download/attachments/43989034/WLE-Rules.pdf

[XML-LGR] Davies, K. and A. Freytag, "Representing Label Generation Rulesets using XML", http://tools.ietf.org/html/draft-davies-idntables/.Visited 2014-06-06.

Appendix A Contributors

The guidelines in this document were created by members of the Integration Panel in collaboration with ICANN staff

1. Integration Panel

Marc Blanchet
Asmus Freytag
Nicholas Ostler
Michel Suignard
Wil Tan

2. Staff

Sarmad Hussain
Nicoleta Munteanu