Integration Panel:

Requirements for LGR Proposals from Generation Panels

REVISION 2015-04-24

1 Overview

This document describes the requirements that LGR Proposals from Generation Panels must meet to be accepted for review by the Integration Panel. For the overall specification of the process to be followed by the Generation Panels see the "Procedure to Develop and Maintain the Label Generation Rules for the Root Zone in Respect of IDNA Labels" [Procedure], as well as the additional instructions in “Setting up and Running a Generation Panel” [Setup] and Section 7 of the overview document for the Maximal Starting Repertoire [MSR].

2 Files and File Formats

Proposals are not single documents; they typically consist, at a minimum, of a *main document* giving background, discussion, and justification and a separate *XML document* with the normative definition of the LGR. Annotated code tables, such as were provided with the MSR, are not required; however, if variants or whole label evaluation rules are specified, the Integration Panel requests that test cases are provided in machine readable form.

3 Script(s) covered

In the *main document*, list the script name(s) and corresponding [ISO15924] script code(s) that the panel is submitting a proposal for.

Attach one *XML document* per script. The script codes in the *main document* should match the script codes in the <language> element in the XML document. Use a single script code in each element; for Japanese use “und-Japn” to cover the mixture of Han, Hiragana and Katakana.

For the normative specification for representing the LGR in XML, see [XML-LGR] and for the specific conventions used for the Root Zone LGR see Section 11 below.

4 Languages using each script

In the *main document*, list the principal languages and, if any, principal minority languages using each script.

Include languages in active, modern use, with a substantial community that uses the script to conduct everyday activities and for which gTLD or ccTLD applications can reasonably be expected at some point.

For some scripts with very diverse user communities (like Arabic, Latin, and Cyrillic), an *exhaustive* list of languages may not be feasible. In such cases, the aim should be to list as many as feasible of the languages (or language groups) that are known to actively use the script for everyday writing. Succinctly discuss the extent to which these languages or language groups are supported by the proposed LGR.

Provide a breakdown of which languages have been investigated and covered, and which could not be investigated for lack of expertise or participation on the Generation Panel.

Where appropriate, list countries or territories where this script has significant user communities.

Indicate whether any language(s) had no community representation on the panel. Describe the panel's outreach efforts to these user communities.

Discuss whether the panel had to "defer" support of any particular languages (or repertoire subsets) to a future version of the LGR. Give details and rationale and discuss the extent to which later inclusion of these languages (or repertoire subsets) will introduce a risk.

5 Related scripts

Are there related scripts (either among the scripts that this panel covers, or across panels)? If so, describe in the *main document*:

1. What are the related scripts, and how are they related?
2. What specific issues did the panel consider?

What approach did the panel undertake to ensure consistent treatment of repertoires and variants across related scripts?

Did the panel subdivide itself into sub-panels for related scripts? If so, how were they structured and work together with each other?

Did the panel coordinate with other panels? If so, provide details.

6 Justification and References

The Process Goals and the Principles in the [Procedure] (§§ A.3.4 and A.3.6) are meant to guide the Generation Panels and the Integration Panel. In the *main document*, discuss the following as they relate to the Process Goals and Principles:

- the process of selecting the repertoire;
- if variants were specified, their choice and disposition;
- if Whole Label Evaluation rules were specified, how they were defined;
- and the overall result of the Generation Panel's process.

For each of these items, indicate the sources or reference used, including the use of outside experts (advisors). Include advisors' reports where available.

A separate document “Considerations for Designing a Label Generation Ruleset for the Root Zone” [Considerations] contains a list of items that a Generation Panel may want to consider while designing an LGR proposal or documenting the justification for particular design choices.

7 Code point repertoire

In the *main document*, describe how the panel arrived at the code point repertoire.

In the Justification for the repertoire, cover every code point included. Summary justification for well known subsets are acceptable, such as “code points in the range XXXX..YYYY form the basic alphabet for language Zzzz, and are in widespread modern use”.

Separately discuss any included code points where the inclusion could be seen as potentially problematic for the root zone, or where the code points are not part of a well-known subset.

Where appropriate, discuss how specific languages investigated relate to specific code points (or rules) in the proposal.

Document how the selected repertoire satisfies the Principles specified in Section A3.6 of the [Procedure].

Did any controversy / contention or points of disagreement exist regarding any of the code points and rules (including those that did not make it into the proposed repertoire and rule set) contemplated by the panel? If so, provide a summary of the issues and how the panel reached its conclusion. Where appropriate provide a link to any archived discussions.

In the *XML document*, list the repertoire, as discussed below in Section 11.3 “Data element”. The use of references (“ref” attributes) may be useful in documenting the source of the repertoire.

The repertoire *must not* contain any code points not defined in the Maximal Starting Repertoire [MSR]. It should not contain any code points outside the repertoire of the covered script, except as required for symmetry and transitivity of variant mappings (see Section 8 below)

8 Variant mappings and dispositions

The proposed LGR might contain rules represented by action elements in the XML format (other than the default rules) that transform these disposition values to the final dispositions for variant labels. If so, explain what those rules intend to accomplish and why they are appropriate for the root zone; also explain their corresponding disposition.

Provide several example labels (test cases) along with expected results of running the variant rules.

If the proposal contains code point variant rules that lead to allocatable or blocked labels, include examples of such labels as well as of labels without variants.

In the *XML document*, specify the variant mappings, dispositions (and optionally actions) as discussed below in Section 11.3 “Data element”. See also the separate [Variant Rules] document.

The use of references ("ref" attributes) may be useful in documenting the source of the variant information.

Mappings *must* be symmetric and transitive as described in [Variant Rules].

Any out-of-repertoire variants must be specified as described in [Variant Rules] or [Packaging].

9 Whole label evaluation rules (WLE)

If there are WLEs in your proposed LGR, provide an explanation in the *main document* of what they are intended to accomplish and why they are appropriate for the Root Zone. In particular, any proposed WLE rules must be evaluated against the *Simplicity Principle*, which states "Overly complex rules are to be avoided, in favor of rules easily understood by users with only some background. In particular, in the root, rules should not require deep familiarity with a particular script or language." (See [Procedure]).

Provide examples of labels that pass these rules and labels that fail (test cases).

In the *XML document*, list any WLE rules and associated actions as discussed below in Section 11.4 Rules element.

10 Format for test cases

For test cases, use one or more plain text files with test cases in UTF-8, with one label per line. Lines starting with # may be used for comments. Use comments to indicate, for example, which labels should pass or fail the WLE rules.

11 XML Format Features Required or Permitted for LGR Proposals

This section describes features required or permitted for use in the XML files that accompany LGR Proposals, and where applicable, specifies required settings.

11.1 File format and the <lgr> element

For the formal specification of the XML format see [XML-LGR].

The file must contain one <lgr> element, containing a <meta>, <data> and <rules> element in that order. The file begins with the following lines:

```
<?xml version="1.0"?>
<lgr xmlns="http://www.iana.org/lgr/0.1">
```

The XML name space in the "xmlns" attribute given here is temporary, so check for updates to the internet draft before submitting the file.

The use of XML comments in the file is discouraged, because they are not preserved by the tools used in the integration process. The use of the "comment" attribute in the XML format is preferred. If an element does not allow a comment attribute, its containing element usually does, or the information can be provided in the <description> element or as part of the main documentation for the proposal.

ICANN IDN team can assist with creating and validating the XML prior to submission.

11.2 The <meta> element

- A meta element must be present in each XML file and must include these elements:

  <version>
  <date>
  <language>
  <scope>
  <unicode-version>
  <description>

- The version element is set to “1” for the first submission, and increased by 1 for each re-submission, any comment attribute is ignored.
- The date element gives the date of submission.
- The single <language> element must contain the script code for the LGR. Since the root zone LGR operates on a script level, the language subtag should be set to "und-", e.g. “und-Cyrl” or “und-Japn”.
- The single scope element has a “type” attribute of “domain” and is set to “.” to indicate the root zone.
- The Unicode version element is set to the same value as in the [MSR] on which the LGR proposal is based, for example “6.3.0” for MSR-2.
- The <description> element should give some summary information, but is not intended to contain the full rationale or justification text for the LGR proposal. It should contain any information relevant to understanding the file itself. If the file uses non-default values for the “type” attributes on variants, these must be summarized in the description. A brief description of any <action> or <rule> element defined in the file (for WLE rules) must also be provided.
- A <references> element may optionally be provided. References may be used to give information about characters or rules, for example by the source of code point (either by Unicode version, or by some other character collection). References may also cite sections of the main proposal document. Note that the “ref” attribute on a code point is multi-valued, allowing several ids that are separated by space.
- A <validity-start> or <validity-end> element **must not** be used.

11.3 The <data> element

The <data> element contains all information on repertoire, and, where present, variants.

All features defined in [XML-LGR] for <char> and <range> elements are permissible. Any intended use of “when” and “not-when” attributes may pose potential problems in integration and should be discussed with the Integration Panel beforehand.

11.3.1 Tags Attributes

In the MSR all code points are tagged with tag attributes containing one or more space separated values based on the Unicode Script property (e.g. “sc:Latn”). Because the MSR is used for all scripts, if a code point is used with multiple scripts, it may have more than one script attribute in its tag. For ease of review, the LGR proposal may retain the script tag from the MSR for each code point.

Some whole label evaluation rules require specific tag values that are not script property values. If a whole label evaluation rule is added that makes use of specific tag values, they may be added in addition to any script property values. Note that the “tag” attribute is multi-valued: it consists of a list of space-separated tag values. The use of a colon delimited prefix (such as “sc:”) is restricted to tag values matching the corresponding Unicode property, such as “sc:” for “Script”.

11.3.2 Variants

The mapping rules for variants, if any are specified, must form a set closed under symmetry and transitivity. Hence, if

```
     <char cp=”XXXX”> <var cp=”YYYY” /></char>
```

is specified, symmetry requires that

```
     <char cp=”YYYY”> <var cp=”XXXX” /></char>
```

is specified also. If

```
     <char cp=”XXXX”> <var cp=”YYYY” /></char>
     <char cp=”YYYY”> <var cp=”ZZZZ” /></char>
```

are specified, transitivity requires that

```
     <char cp=”ZZZZ”> <var cp=”XXXX” /></char>
```

is specified also. (For clarity, the tag attributes are not shown in the examples).

For each <var> element specifying a mapping, a “type” attribute (not shown in the simplified example above) **must** be supplied, to set the disposition value for that variant code point mapping. These “disp” attributes do not have to be symmetric or transitive. For example:

```
<char cp=”ZZZZ”> <var cp=”XXXX” type=”blocked” /></char>
```

Typically, the **type** value is one of “blocked” or “allocatable”. The default rules are sufficient to evaluate these into the corresponding variant label dispositions.

Other values for the “type” attribute on var elements are only permitted if corresponding action elements are defined in the rules section that evaluate these into either “blocked” or “allocatable” dispositions for variant labels (see [VariantRules] for more information). The specification of reflexive variants (where cp element on <char> and <var> have the same value)

```
     <char cp=”XXXX”> <var cp=”XXXX” /></char>
```

is permitted. The intent to use reflexive mappings or special “type” values on <var> elements should be discussed with the Integration panel prior to submission.

In case the target of a variant mapping is outside the repertoire, symmetry requires adding the target to the repertoire. If such a code point is given a reflexive mapping as shown, the default rules will treat it correctly as an out-of-repertoire code point.

The following example assumes that code point YYYY is in-repertoire and maps to a code point XXXX outside the repertoire.

```
<char cp=”YYYY”>
     <var cp=”XXXX” type=”blocked” />
</char>

<char cp=”XXXX”>
     <var cp=”XXXX” type=”out-of-repertoire-var” />
     <var cp=”YYYY” type=”blocked” />
</char>
```

Note that in the case of out-of-repertoire variants the non-reflexive mappings are by necessity all of type “blocked”. For more information on specifying variants, see [Variant-Rules].

ICANN IDN team can assist with validating these requirements before submission.

> **Note:** It may be easier to edit the repertoire and variant information as a plain text file, for example in an existing IDN table format, and then use a simple script to convert that to the XML format for the <data> section. This is how the Integration Panel created the XML for the MSR.
>
> Alternatively, for a script not using variants or WLE rules, the data section could simply be created by copying only the desired entries from the MSR.

11.4 The <rules> element

All script LGRs require a rules element. The entire contents of the *default* <rules> element from the MSR MUST be included in all LGRs. There’s no requirement for an LGR to add to the default rules, and in fact it is likely that few Generation Panels will need to add additional rules.

As any intended additions to the rules may pose potential issues in integration, they should be discussed with the Integration Panel prior to submission.

Permitted final disposition values for labels as result of any added actions are limited to “invalid”, “blocked”, and “allocatable”. For more information, see [WLE-Rules].

11.5 Example XML file for LGR Proposal

The following is a rather minimal example of an XML file for a LGR proposal. See [Sample-LGR-Greek] and [Sample-LGR-Thaana] for more realistic examples.

```
<?xml version="1.0"?>
```

```
<lgr xmlns="http://www.iana.org/lgr/0.1">
      <meta>
            <version>1</version>
            <date>2014-03-01</date>
            <language>und-Arab</language>
            <scope type=”domain”>.</scope>
            <description>
                  Example specification for an Arabic script
                  LGR Proposal. For Justification and Notes
                  see the main document.
                  This example contains no variants or added
                  whole label evaluation rules. A dummy code
                  point is provided so this example can be
                  validated.
                  The rules element contains the default
                  rules from MSR-1.
                  Reference 0 – The Unicode Standard, 6.3.0
            </description>
            <unicode-version>6.3.0</unicode-version>

            <references>
                  <reference id="0">See description</reference>
            </references>
      </meta>
      <data>
            <char cp="0641" tag=”sc:Arab” ref=”0”/>
      </data>
      <rules>
            <!--Character class definitions (if any) go here-->

            <!--Whole label evaluation and context rules go here-->
            <rule name="leading-combining-mark">
                  <start />
                  <union>
                        <class property="gc:Mn" />
                        <class property="gc:Mc" />
                  </union>
            </rule>

            <!--Action elements go here - order defines precedence-->
            <action disp="invalid" match="leading-combining-mark" />
            <action disp="invalid" any-variant="out-of-repertoire-var"
comment="any variant label with a code point out of repertoire is invalid" />
            <action disp="blocked" any-variant="blocked" />
            <action disp="allocatable" any-variant="allocatable" />
            <action disp="valid" comment="catch all" />
      </rules>
</lgr>
```

While additional rules and actions may be added, the <rules> element must always contain a copy of the default rules supplied in the MSR.

12 References

[Considerations] Integration Panel "Considerations for Designing a Label Generation Ruleset for the Root Zone" available online as https://www.icann.org/en/system/files/files/lgr-design-considerations-23sep14-en.pdf

[ISO15924] ISO 15924 Registration Authority, "ISO 15924 Code Lists: Codes for the representation of names of scripts", available online as http://www.unicode.org/iso15924/iso15924-codes.html. Visited 2014-08-25

[MSR] Internet Corporation for Assigned Names and Numbers, "Maxima Starting Repertoire – MSR-2", (Los Angeles, California: ICANN, April 2015) available online as
(Overview and Rationale) https://www.icann.org/en/system/files/files/msr-2-overview-14apr15-en.pdf;
(Annotated Han tables) https://www.icann.org/en/system/files/files/msr-2-han-13apr15-en.pdf;
(Annotated Hangul tables) https://www.icann.org/en/system/files/files/msr-2-hangul-13apr15-en.pdf;
(Annotated Non-CJK tables) https://www.icann.org/en/system/files/files/msr-2-non-cjk-13apr15-en.pdf;
(Repertoire and WLE Rules) https://www.icann.org/en/system/files/files/msr-2-wle-rules-13apr15-en.xml.

[Packaging] Integration Panel, "Packaging the MSR and LGR", available online as https://community.icann.org/download/attachments/43989034/Packaging-MSR-LGR.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) http://www.icann.org/en/resources/idn/variant-tlds/draft-lgr-procedure-20mar13-en.pdf

[Sample-LGR-Greek] Michel Suignard, "Label Generation Rules for Greek: Sample LGR for Greek" available online as https://github.com/kjd/lgr/blob/master/resources/Sample-LGR-Greek.xml

[Sample-LGR-Thaana] Michel Suignard, "Label Generation Rules for Thaana: Sample LGR for Thaana" available online as https://github.com/kjd/lgr/blob/master/resources/Sample-LGR-Thaana.xml

[Setup] Internet Corporation for Assigned Names and Numbers, "Setting up and Running a Generation Panel" ((Los Angeles, California: ICANN, November, 2013) https://community.icann.org/download/attachments/43989034/Setting%20up%20and%20Running%20a%20Generation%20Panel.pdf

[VariantRules] Integration Panel, “Variant Rules”, available online as https://community.icann.org/download/attachments/43989034/Variant%20Rules.pdf

[WLE-Rules] Integration Panel, “Whole Label Evaluation(WLE) Rules”, available online as 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 2014-07-21.