

Background:

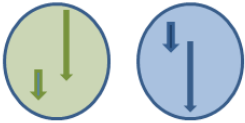
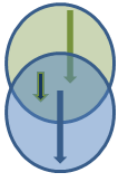
Out of Repertoire Variants in Root-Zone LGR and Proposals

Version 2017-09-25

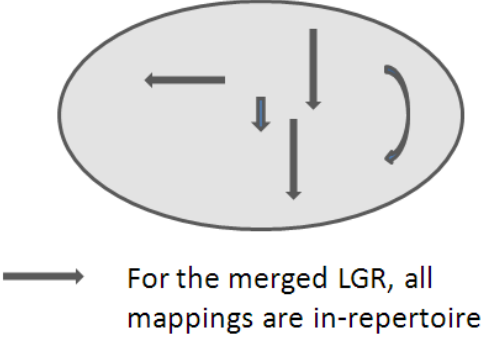
Abstract

This background document summarizes issues related to specifying variants that cross repertoire boundaries.

Overview of Variant categories

In-Repertoire Variants	
<p data-bbox="315 873 570 905">In-Repertoire Variants</p>  <p data-bbox="240 1098 607 1171">—→ In-repertoire for blue —→ In-repertoire for green</p>	<p data-bbox="850 869 1430 1108">The most common scenario for variants is where both code point and variant are always part of the same repertoire, and the repertoires of affected LGRs are distinct. These variants are fully specified in each LGR; all <i>in-repertoire</i> mappings must be symmetric and transitive. See RFC 8228 and RFC 7940 for further details.</p>
<p data-bbox="321 1209 613 1304">In-Repertoire Variants and Overlapping Repertoires</p>  <p data-bbox="220 1577 602 1703">—→ In-repertoire for blue —→ In-repertoire for green —→ In-repertoire for both</p>	<p data-bbox="850 1209 1425 1377">When repertoires overlap, some variants have both code point and variant <i>in-repertoire</i>. Other variants may map between code points that are in-repertoire in one LGR but may be all or partially outside the repertoire in another LGR.</p> <p data-bbox="850 1419 1422 1524">All LGRs always specify all <i>in-repertoire</i> variants. See below for the case of an LGR where a mapping is not <i>in-repertoire</i>.</p>

Out of Repertoire Variants:	
<p>Out-of-Repertoire Variants</p> <p>Cross-script Cross-repertoire (unique to unique) Cross-repertoire (shared to unique)</p>	
<p> Cross script or between unique (non-overlapped) Cross-repertoire (source or target is shared) </p>	
Disjoint Repertoire:	(Partially) Overlapping Repertoire:
<p>Variants defined across disjoint repertoires, including variants between non-shared portions of an overlapped repertoire are never “in-repertoire” for any LGR.</p>	<p>Variants defined between a shared and non-shared portion of an overlapped repertoire are in-repertoire for at least one LGR.</p>
<p>In order for these variants to be defined, they must be defined as “out-of-repertoire” variants by at least one LGR.</p>	<p>Because all in-repertoire variants are supposed to be defined in each LGR, it is possible to compute any missing variants for the LGR in which they would be cross-repertoire.</p>
<p>All GPs must specify any desired variants between disjoint repertoires.</p>	<p>Therefore, it is acceptable for LGRs to omit specifying this type of cross-repertoire variants. This may make such LGRs easier to review. However, these variants will still be defined in the integrated LGR.</p>
<p>Generation Panels are strongly encouraged to review any variants defined between disjoint repertoires, including the affect from the requirement that the total variant set must be transitive. Where the specified variant sets do not mutually agree across affected LGRs, integration will result in the superset.</p>	<p>Generation panels are strongly encouraged to review any shared variants and to come to a joint understanding of what constitutes a variant. In doing so, they need to take into account any side effects from the requirement that the total variant set must be transitive. All variant mappings entirely within the shared repertoire must agree among affected LGRs, or integration will fail.</p>
<p>In computing variants during integration, additional variant mappings will be added if required to make the total set symmetric and transitive. If that results in any additional “in-script” variants, the affected LGR is deemed incomplete and will be rejected.</p>	<p>In computing variants during integration, additional variant mappings will be added if required to make the total set symmetric and transitive. If that results in any additional “in-script” variants, the affected LGR is deemed incomplete and will be rejected.</p>

<p>Example:</p> <p>The Armenian LGR proposal would list which Cyrillic code points are cross-script homoglyphs of corresponding Armenian code points and vice versa for the Cyrillic LGR. If the two GPs did not agree on a matching set of reciprocal mappings that is symmetric and transitive, the IP will compute any mappings needed to make the merged set symmetric and transitive. Should this computation result in additional <i>in-repertoire</i> variants for one or more LGRs, the affected LGRs are deemed incomplete and the integration fails. Otherwise, the integration will proceed with a merged set.</p>	<p>Example:</p> <p>The Korean LGR would list all variants between Han code points that are part of the Korean repertoire, but not between such code points and Chinese-only or Japanese-only code points. The Korean and Chinese GP would agree on a mutually compatible definition of variants. Both LGRs must list their <i>in-repertoire</i> variants, but may omit variants that are <i>in-repertoire</i> in the other LGR. Even if not listed in each LGR, all variant sets apply to all LGRs containing one of the affected code points. If a given Chinese code point is a variant of two Korean code points, these two Korean code points become <i>in-repertoire</i> variants of each other, as required by transitivity. If this required variant relation were missing in the Korean LGR, that LGR would be deemed incomplete.</p>
<p>Merged LGR:</p>	
<p style="text-align: center;">Merged LGR</p>  <p style="text-align: center;">→ For the merged LGR, all mappings are in-repertoire</p>	<p>For the merged LGR, all variants must be in-repertoire, symmetric, and transitive.</p> <p>The merge process will add additional mappings required to make the set symmetric and transitive, as long as that would not introduce variant mappings that are <i>in-repertoire</i> for any element LGR. For details, see the following sections.</p> <p>An LGR proposal that specifies variants that are out of repertoire with respect to the total merged repertoire of the Root Zone LGR will be deferred (waiting for related scripts that would add the missing repertoire) or rejected as out-of-scope as appropriate.</p>
<p>Other cases:</p>	
<p>An LGR that specifies a variant mapping between two out-of-repertoire code points (other than as result of making the variant set transitive) will be considered out-of-scope and rejected.</p>	

What is the effect of out-of-repertoire variants on the integrated LGR?

The integrated LGR itself will not have variant mappings that are out-of-repertoire with respect to the merged root zone repertoire. Because the integrated Root Zone LGR is complete, nothing can be specified about code points that are outside its repertoire.

The Root Zone LGR is specified via a collection of files that together represent the integrated LGR. These files have different purposes. There are Element LGRs (one for each script) that contain all the data necessary to

- a) determine whether a label is valid for that script
- b) determine the set of allocatable variants for that label

There is a single common LGR file that contains the merged repertoire and all variant mappings between code points, no matter which Element LGR's repertoire or variant set they originate from.

This merged file contains all the data necessary to

- c) determine whether two labels (of the same or different scripts) collide with each other

This test is performed only for labels already determined to be valid in a given LGR, therefore there is no need to consider possible collisions with code points outside the merged repertoire. Therefore, the merged file never contains out-of-repertoire variant specifications.

What need is there to specify out-of-repertoire variants in any Element LGR?

For determining validity and allocatable variants, out-of-repertoire variants are not used. Therefore, the element LGRs do not need to contain these mappings for those purposes.

However, the set of variant mappings in the merged file is intended to be the result of merging the variant sets from the element LGRs.

For the case of cross-script variants, at least one of the element LGRs must contain an out-of-repertoire mapping between it and some other script for that mapping to be included in the merged LGR. Ideally, the other script would contain the corresponding mappings. This would serve as a cross check, and the IP strongly encourages the GPs involved to arrive at mutually agreed sets. However, per the procedure the merge may proceed even if there's no corresponding mapping supplied; in that case the IP will compute any missing mappings needed to make the merge set symmetric and transitive.

For overlapping repertoires there are two possible scenarios. Variant mappings that are not "out-of-repertoire" for all of the involved LGRs (labeled "disjoint repertoire" in the table above); and the more common case of variant mappings that are "in-repertoire" for at least one of the LGRs.

In principle, each of two LGRs that overlap a common repertoire may have some code points that are unique (not shared with the other LGR). If any variants are defined between the unique (non-shared) part of the repertoire of these two LGRs, then the situation is similar to a cross-script case: such

mappings must be listed in at least one LGR as out-of-repertoire variants, otherwise they will not be merged or become part of the Root Zone LGR. (This case is expected to be uncommon).

For the more common case, where all variant mappings involve code points that are in-repertoire to one or both of the LGRs sharing a repertoire, there is no need for any of the element LGRs to define anything other than their repertoire-internal mappings. When all of these repertoire-internal variant mappings are merged, the correct Root Zone level merged variant set will be created. GPs are strongly encouraged to resolve any conflicts between repertoire-internal mappings affecting their shared repertoire.

Especially in the case of large repertoires, listing all out-of-repertoire variants (whether strictly required or not) could obscure the more important in-repertoire variants for that LGR.

In summary, in an LGR it is not required to list variants from the *common* shared set for two LGRs to the *unique* part of the repertoire of the other LGR. The only variants that must strictly be specified are those internal to the repertoire of that LGR, plus any that establish a mapping between a *unique* part of the repertoire of that LGR to the *unique* part of the repertoire of another LGR.

Note, that even if some variants are not strictly required to be listed in some of the Element LGRs, their presence in the merged set is what determines which labels in one script may block which labels in another script. For that reason, GPs are strongly encouraged to mutually review their variant sets whenever they involve shared repertoires, or cross-script scenarios.

Root Zone LGR

The merged file will not contain any out-of-repertoire variants.

The element LGRs will contain any variants defined in the corresponding LGR proposals, including out-of-repertoire variants, whether required to be listed or optional.

The merged LGR will contain a merged set of all variant mappings, including computed mappings needed to make the merged set symmetric and transitive.

If any LGR specifies variants that are out-of-repertoire with respect to the merged repertoire, that LGR will be deferred (in case where the mappings involve other pending scripts) or rejected.

How to specify an out-of-repertoire variant in XML

An out-of-repertoire variant is specified as:

```
<char cp="0000">
  < var cp="1111" type="blocked" />
</char>

<char cp="1111">
  <var cp="1111" type="out-of-repertoire-var" />
```

```
<var cp="0000" type="blocked" />  
</char>
```

Here "0000" is a code point that is in the repertoire, and has a blocked mapping to code point 1111 that is outside the repertoire. Code point 1111 MUST have a reflexive mapping (to itself) of type "out-of-repertoire-var", in addition to the reciprocal mapping back to "0000". Mappings between in and out of repertoire code points must be of type "blocked".

An actual LGR file would contain "ref" and perhaps "comment" attributes as well, these are not shown. For more details, see RFC 8228 and RFC 7940.