Tasks to fulfill

IP Review of Variants for the Proposed Latin LGR dated 2019-11-01

When TBD is done my proposal is to mark it with green

|  |  |  |
| --- | --- | --- |
| Item no | Task description |  |
| 1 | Decision on removal some variants - List of variants between Base/Base+diacritic for further discussion in order to minimize number as much as possible. |  |
| 2 | Decision on removal some variants - List of underlining variants with **Cedilla**, **ogonek**, **comma below**, and **circumflex below** for further discussion in order to minimize or totally exclude such type of variants  In proposed variants C and C WITH CEDILLA are missing |  |
| 3 | Decision for removal of some of the underlining variants(?)- Discuss IP proposal to exclude variants with following characters  þ U+00FE LATIN SMALL LETTER THORN  ŋ U+014B LATIN SMALL LETTER ENG  џ U+045F CYRILLIC SMALL LETTER DZHE  ɣ U+0263 LATIN SMALL LETTER GAMMA |  |
| 4 | Discussion and decision - The case of ỵ U+1EF5 LATIN SMALL LETTER Y WITH DOT BELOW is also peculiar because it is not rendered with a **dot below** but rather ‘dot behind and slightly low’. While an underline could possibly erase the distinction, the main reason the IP would see it as problematic is because it may be confused with ‘y.’. Because of this, its removal from the Latin LGR should be considered. |  |
| 5 | Decision on Unstable Marshallese orthography-IP discussion and suggestion:  using either the **dot below** or the **cedilla** (ref <https://en.wikipedia.org/wiki/Cedilla> , <https://en.wikipedia.org/wiki/Marshallese_language> , and <https://omniglot.com/writing/marshallese.php>) although the cedilla is clearly preferred. This has several consequences on the Latin LGR:  The Latin Proposal claims in section 6.1.3.4.1 page 42: “*First, certain diacritics may be considered conceptually the same as others by significant parts of the user community, such as dot below or a comma below*.” This claim appears related to Marshallese but if so, it should be addressing the confusion between **cedilla** and **dot below** instead. The **comma below** mark is only used in Romanian and it does not appear to be any confusion in the Romanian community for these marks. (Notwithstanding that some fonts may implement **cedilla below** with a comma shape).  However, even if the confusion between the **cedilla** and the **dot below** is accepted for Marshallese, it is clear the usage of the **dot below** was the result of implementation deficiency in rendering the correct sequence and used as a temporary remedy. In the view of the IP, the only two possible outcomes are either:   1. to remove both representation from the LGR (l, m, n, o with **dot below** and l, m, n, o with **cedilla**) as representing an unstable orthography, 2. to keep the preferred representation (l, m, n, o with **cedilla**) and remove l, m, n with dot below from the repertoire (o with dot below is in fact used by other LGR languages and should remain in all cases).   The IP preference would be to keep the preferred representation (option 2), and to remove the alternate representation as deprecated. |  |
| 6 | List of variant sets reviewed section 4.1 with 17 suggestions for removal some variants and adding some other, should be discussed by GP and decisions should be made for each suggestion |  |
| 7 | Decide on -5 Other variants sets included in LGR , two cases |  |
| 8 | 5.2 Possibly mistaken reflexive mapping - The code point U-028B ʋ LATIN SMALL LETTER U WITH HOOK has a reflexive mapping to “out-of-repertoire-var” in the latest XML. This would remove that code point from the repertoire of the Latin LGR. If this was unintentional, the reflexive mapping needs to be removed, otherwise, the whole code point can go. “Out-of-repertoire-var” mappings that go to code points not in *any other* LGR (present or forthcoming) are not meaningful. |  |
| 9 | 5.3 Possibly missing Variant - Appendix D contains a rationale for making U+0066 f a variant of U+0192 ƒ LATIN SMALL LETTER F WITH HOOK, and notes that it was decided by the GP to make these variants. But neither section 6.5 nor the XML follow suit. This appears to be an inconsistency that needs to be resolved one way or another; either by a fix of the Appendix to indicate why these were not made variants after or, or by a fix to the formal definitions.  NOTE: this variant set is not covered by the review above because that review is based on Section 6.5 and the XML. Therefore, there is no IP position on it at this time. |  |
| 10 | 6 Allocatable variants – needs decision and actions for 2 suggested cases for discussion |  |
| 11 | Characters not included in LGR – cases confusabilty issues  ď U+010F LATIN SMALL LETTER D WITH CARON  ľ U+013E LATIN SMALL LETTER L WITH CARON |  |
| 12 | Item 1: Cross-script variants causing in-script variant for Cyrillic The following variant set proposed for the Latin LGR would cause an in-script variant for Cyrillic which would require review by the Cyrillic GP.  y U+0079 LATIN SMALL LETTER Y  ɣ U+0263 LATIN SMALL LETTER GAMMA  γ U+03B3 GREEK SMALL LETTER GAMMA  у U+0443 CYRILLIC SMALL LETTER U  ү U+04AF CYRILLIC SMALL LETTER STRAIGHT U  ỵ U+1EF5 LATIN SMALL LETTER Y WITH DOT BELOW |  |
| 13 | Items 2-3: Cross-Script variants causing in-script variants for ASCII The following variant set proposed for the Greek LGR contains one letter U+1E7F not found in the Latin LGR proposal, which should be removed. It would also pick up transitive mappings, if integrated (bold red). One of them would introduce a variant between ASCII letters V and Y. That is not permissible (because it affects ASCII-only labels, which are outside the IDN process).  v U+0076 LATIN SMALL LETER V  **y U+0079 LATIN SMALL LETTER Y**  ɣ U+0263 LATIN SMALL LETTER GAMMA  **γ U+03B3 GREEK SMALL LETTER GAMMA**  **ν U+03BD GREEK SMALL LETTER NU**  **у U+0443 CYRILLIC SMALL LETTER U**  ү U+04AF CYRILLIC SMALL LETTER STRAIGHT U  ұ U+04B1 GREEK SMALL LETTER  ṿ U+1E7F LATIN SMALL LETTER V WITH DOT BELOW **(not in Latin LGR)** The same set contains a proposed in-script variant for Greek. (In the proposal text they map separately to U+04AF, but transitivity requires that they also map to each other) |  |
| 14 | Item 4: Mappings for Latin Sharp S The following variant set maps “ss” to “ß” as well as to Greek beta.  ss U+0073 U+0073 LATIN LETTER SMALL S + LATIN LETTER SMALL S  ß U+00DF LATIN SMALL LETTER SHARP S  β U+03B2 GREEK SMALL LETTER BETA  The Greek LGR proposal does not contain a cross-script mapping for beta. Normally, for assigning any cross-script variant, it may be sufficient if it is clearly seen as a variant by the users of one script, in which case the conservative approach may be to allow it.  However, this case is interesting because the in-script variant in Latin is a “semantic” one: transitivity would introduce a mapping between β U+03B2 and ss U+0073 U+0073.  It is clear that the transitive mapping would be surprising to Greek users. But it may well be motivated, because users of the “ss” variant may not natively use “ß”. While they would generally know of the equivalents, they may be more likely to accept “β” as a substitute. The situation is made more complex, by the fact that *ß* is a typical *italic* form of “ß”. This may motivate the variant mapping from the perspective of Latin users, even if it isn’t as obvious a variant for Greek users.  Required handling of overlapped variants would introduce U+0455 U+0455 (Cyrillic ѕ) as well as any in-script variants of ‘s’, such as U+015F (ş) |  |
| 15 | Check results from meeting-material-ACGL-at-icann66, snapshot of working report |  |

ADDITIONAL INFORMATIONS

## Cross-script variant process

There are limits to what can be defined as variants. Generally, there has to be some equivalence relation, because symmetry and transitivity must apply to variant definitions. In some cases, the perspective from users of different may need to be reconciled; in other cases, the conservative approach supports assigning a variant, even if only one script can make a strong case. Here are some discussion points:

1. The integration process applies the superset of all cross-script variants
2. The integration process extends this superset to be symmetric and transitive
3. Users of two different scripts may not see the same pair as variants; in that case, the more conservative route is to allow the variant.
4. Some variants may be seen to adversely impact another script; in that case, the variant should be allowed only if there is a solid case for it in the proposing script.
5. Some cross-script variants would lead to in-script variants in another LGR. This case generally requires a chance for the other GP to decide whether such “imposed” in-script variants should be treated as allocatable or blocked (inside the script). (See also 3).