IP Review of Variants for the Proposed Latin LGR dated 2019-10-11

DATE: 2019-11-01

UPDATED: 2020-01-24

# Overview

This document reviews the variants as defined in the proposed Latin LGR (ICANN-LGP-Proposal V5 merged with Appendix-Repaired-Final.docx), and lgr-4-Latin-Script-2019-10-20-en.xml. Other aspects of the proposal will be reviewed separately.

 UPDATE: additional findings or findings based on revisions of the LGR document as evidenced by the January 08, 2020 document and XML are highlighted like this.

These updates reflect changes in the Latin LGR, some results of the Montreal meeting (although there may be some issues where those have not yet been reflected fully) as well as some new information or understanding available to the IP.

Variant set numbers change if sets are added or deleted. The numbers given here may or may not match the latest proposal, no attempt has been made to ensure a match.

# General Remarks

In analyzing in-script variants, a common case involves cross-language variants. For IDNs it is not safe to assume that labels will be in a given language; malicious registrations may well insert an unfamiliar form used in a language unfamiliar to the reader and presented in context that otherwise looks like a word in that reader’s language. With diacritics generally less prominent than the base form, that would seem to argue for wide-ranging variant sets containing all the marks found with a given base form. What argues against that is the general feature of the Latin script, which is its extensibility to other languages by adding diacritical marks. Unless there are special circumstances, users can be considered aware of this basic feature of the Latin script and expected to treat any unfamiliar marks with the same attention as applied to the distinction between similar letters, like ‘i’ and ‘j’ or ‘n’ and ‘m’. Largely we find that the proposed Latin LGR strikes the correct balance here; for items where we may disagree, see the following sections.

# Variant relationship between Base character and Base + diacritic

These variants create in principle a NULL variant situation, but this is avoided by not allowing the diacritics on their own, as is the case in any of the Latin LGR and correlated LGRs. However, it does create additional variants when the base character is part of a cross-script variant. Some of the variants introduced by transitivity might not have been considered as variants on their own. Therefore variants between Base/Base+diacritic should be minimized as much as possible.

# Variant sets introduced by underlining effects

The Latin GP should be commended for its thorough analysis of the underlining effect on variants. In that aspect, the IP agrees with some of its conclusion.

For example, the IP agrees with confusability between a base character and the same base character associated with a mark consisting of a **dot below**, **macron below**, **line below**, or **underline**. The rationale is that these marks have a single line of “ink” which can be easily confused with an underline style applied to the domain name commonly used to highlight hyperlinks.

Example: i i̱ ị (at 16 pts) i i̱ ị (at 10pts)

(The example shows the effect of underlining U+0069 LATIN SMALLER LETTER I, U+0069+U0331 LATIN SMALL LETTER I WITH MACRON BELOW, and U+1ECB Latin SMALL LETTER I WITH DOT BELOW. The confusability aspect is more pronounced at small point sizes)

UPDATE: In recent informal research by the IP it was noted that underlining that “skips” the ink[[1]](#footnote-1) of descenders and diacritics below appear to have become more common. This somewhat reduces the urgency to address this issue. At the same time, “dot below” may be represented by a single pixel at smaller sizes, which would make a separate argument for treating it, but not macron below, as a variant for the base form, even without an underline.

However, the IP considers that the analysis may have gone too far in some aspects. The IP disagrees with creating variants for marks that are attached and look like larger and more complex descenders and are not merely made of a single point or horizontal line. Diacritics of for which the Latin LGR have proposed variant sets between ‘base character’ and ‘base character with diacritics’ are discussed in the following first in general terms, then by listing each variant set disputed by the IP:

**Cedilla**, **ogonek**, **comma below**, and **circumflex below.**

(It should be noted that the Latin LGR does not propose a variant relation between C and C WITH CEDILLA, which could be seen as not consistent with the current underline policy)

In a similar fashion, the Latin LGR has created variant sets with non-decomposable characters hat have some similarity with base characters but with integral descenders not reduced to a single dot or horizontal line. The IP does not recommend having these included in variant sets. These characters are:

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

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.

Finally, the Marshallese orthography is apparently inconsistent in 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.

## List of variant sets reviewed

The following list enumerates the Latin LGR variant sets affected by the above consideration and suggests possible adjustments. Items marked in color are suggested to be removed from the variant set (that is, unless otherwise indicated, they should lose their variant mappings, turning them into singletons). Items struck out are suggested for possible removal from the repertoire altogether. Items in blue are suggested additions to the variant sets.

UPDATE: Note that variant set w/o issues are not listed.

### Variant Set 1 — 6 Members (remove U+105 and make it a singleton):

a U+0061 LATIN SMALL LETTER A

a̱ U+0061+U+0331 LATIN SMALL LETTER A + COMBINING MACRON BELOW

ą U+0105 LATIN SMALL LETTER A WITH OGONEK

α U+03B1 GREEK SMALL LETTER ALPHA

а U+0430 CYRILLIC SMALL LETTER A

ạ U+1EA1 LATIN SMALL LETTER A WITH DOT BELOW

(The case for making a variant mapping between U+0061 and U+03B1 is still open)

### Variant Set 2 — 2 Members (make them singletons):

b U+0062 LATIN SMALL LETTER B

þ U+00FE LATIN SMALL LETTER THORN

### Variant Set 3 — 2 Members (see variant set 29):

c U+0063 LATIN SMALL LETTER C

с U+0441 CYRILLIC SMALL LETTER ES

င U+1004 MYANMAR LETTER NGA

UPDATE: This set would include U+1004 depending on analysis

### Variant set 4 – 2 Members (make them singletons)

d U+0064 LATIN SMALL LETTER D

ḓ U+1E13 LATIN SMALL LETTER D WITH CIRCUMFLEX BELOW

UPDATE: the case for this is weak, particularly in light of the new findings on skip-ink underline support in recent browsers, but because a circumflex is not fully obscured even by a continuous underline.

### Variant set 5 – 5 Members (remove U+0119 and make it a singleton)

e U+0065 LATIN SMALL LETTER E

e̱ U+0065+U+0331 LATIN SMALL LETTER E + COMBINING MACRON BELOW

ę U+0119 LATIN SMALL LETTER E WITH OGONEK

е U+0435 CYRILLIC SMALL LETTER IE

ẹ U+1EB9 LATIN SMALL LETTER E WITH DOT BELOW

UPDATE: also remove macron, if the policy wrt to underlining is changed.

### Variant set 9 – 3 Members (remove U+012F and make it a singleton)

j U+006A LATIN SMALL LETTER J

į U+012F LATIN SMALL LETTER I WITH OGONEK

ј U+0458 CYRILLIC SMALL LETTER JE

### Variant set 10 – 2 Members (make them singletons)

k U+006B LATIN SMALL LETTER K

ķ U+0137 LATIN SMALL LETTER K WITH CEDILLA

### Variant set 11 – 5 Members (remove U+013C, U+1E37, and U+1E3D from the variant set, and make U+013C and U+1E3D singleton; U+1E37 should be removed from the LGR altogether)

l U+006C LATIN SMALL LETTER L

ļ U+013C LATIN SMALL LETTER L WITH CEDILLA

ӏ U+04CF CYRILLIC SMALL LETTER PALOCHKA

~~ḷ U+1E37 LATIN SMALL LETTER L WITH DOT BELOW~~

ḽ U+1E3D LATIN SMALL LETTER L WITH CIRCUMFLEX BELOW

### Variant set 12 – 2 Members (for consistency it should have also contained U+006D+U0327; U+1E43 should be removed from the LGR altogether; the end result is that U+006D and U+006D+U+00327 should be singleton)

m U+006D LATIN SMALL LETTER M

m̧ U+006D+U+0327 LATIN SMALL LETTER M + COMBINING CEDILLA

~~ṃ U+1E43 LATIN SMALL LETTER M WITH DOT BELOW~~

### Variant set 13 – 7 Members (remove U+0146, U+014B, U+1E47 and U+1E4B from the variant set, and make U+0146, U+014B and U+1E4B singleton; U+1E47 should be removed from the LGR altogether)

n U+006E LATIN SMALL LETTER N

ņ U+0146 LATIN SMALL LETTER N WITH CEDILLA

ŋ U+014B LATIN SMALL LETTER ENG

ո U+0578 ARMENIAN SMALL LETTER VO

~~ṇ U+1E47 LATIN SMALL LETTER N WITH DOT BELOW~~

ṉ U+1E49 LATIN SMALL LETTER N WITH LINE BELOW

ṋ U+1E4B LATIN SMALL LETTER N WITH CIRCUMFLEX BELOW

### Variant set 14 – 6 Members (for consistency it should have also contained U+006F+U0327, but that should be a singleton; other script mappings to be added shown in blue; U+014D does not belong)

o U+006F LATIN SMALL LETTER O

o̧ U+006F+U+0327 LATIN SMALL LETTER O + COMBINING CEDILLA

o̱ U+006F+U+0331 LATIN SMALL LETTER O + COMBINING MACRON BELOW

~~ō U+014D LATIN SMALL LETTER O WITH MACRON~~

ο U+03BF GREEK SMALL LETTER OMICRON

о U+043E CYRILLIC SMALL LETTER O

օ U+0585 ARMENIAN SMALL LETTER OH

ଠ U+0B20 ORIYA LETTER TTHA

ഠ U+0D20 MALAYALAM LETTER TTHA

ဝ U+101D MYANMAR LETTER WA

UPDATE: the latest version adds U+014D into this set in the XML by mistake. We see no case for the inclusion of this code point into this set; a closer look makes clear that this is because of an incorrect code point in Section 6.5.2. U+006F should have been U+00F5 (o with tilde).

Whether 00F5 should be a variant of 014D is a separate issue.

For a more complete analysis of circle glyphs see section 8 on “Variants based on Generic GlyphsVariants based on Generic Glyphs” below.

### ọ U+1ECD LATIN SMALL LETTER O WITH DOT BELOW

(This set, when integrated to the LGR needs to pick up additional variants based on other scripts with also have a circle glyph: U+0B20, U+0D20 and U+101D. However, that process requires that at least one LGR actually lists the full set; therefore, we are requesting the Latin LGR to formally add these variants.)

### Variant set 18 – 5 Members (U+015F and U+0219 should be removed from that variant set and make into their own 2-member variant set)

s U+0073 LATIN SMALL LETTER S

ş U+015F LATIN SMALL LETTER S WITH CEDILLA

ș U+0219 LATIN SMALL LETTER S WITH COMMA BELOW

ѕ U+0455 CYRILLIC SMALL LETTER DZE

ṣ U+1E63 LATIN SMALL LETTER S WITH DOT BELOW

UPDATE: Evidence collected at the time S with comma below was encoded showed existing print publications at the time “freely” alternating these shapes — even in linguistic environments where only one of the shapes was supposedly native. It is unknown to what extent this ambiguity has survived the last decades and progression into digital text representation; but at the time only expert users cared about the particular shape of the diacritic below; newspaper publishers quite evidently did not. In addition, the distinction was never used contrastively in the same linguistic environment. These facts argue for making them variants of each other.

Because both are taller than 1 pixel and with the advent of more support for skip-ink underlining, there’s considerably less pressure on making these variants with the base form. A beneficial effect of removing the mapping to the bare “s” is that it greatly simplifies the situation with overlapped variants for the sequence “ss”. Not only do these two add to the combinatorial explosion of permuted variants for “ss”, but these permuted sequences would also have to become variants of sharp-s and beta (as well as Cyrillic “ѕѕ” by transitivity, something that is not altogether satisfactory.

The open question then is whether s-dot below would be better with “s” or with the new variant set. There are arguments for either. This may be a good example where it is not possible to use the variant mechanism for all related pairs, even though there may be arguments in favor of each pair on its own. It may be preferable to make the split and then note some issues for considerations in manual post processing (string similarity review).

### Variant set 20 – 4 Members (U+021B and U+1E71 should be removed from that variant set and made singletons)

t U+0074 LATIN SMALL LETTER T

ț U+021B LATIN SMALL LETTER T WITH COMMA BELOW

ṭ U+1E6D LATIN SMALL LETTER T WITH DOT BELOW

ṱ U+1E71 LATIN SMALL LETTER T WITH CIRCUMFLEX BELOW

(note that U+0163 LATIN SMALL LETTER T WITH CEDILLA is not part of the Latin LGR)

### Variant set 21 – 7 Members (U+0173 and U+045F should be removed from that variant set and made singletons)

u U+0075 LATIN SMALL LETTER U

ų U+0173 LATIN SMALL LETTER U WITH OGONEK

ʋ U+028B LATIN SMALL LETTER V WITH HOOK[[2]](#footnote-2)

υ U+03C5 GREEK SMALL LETTER UPSILON

џ U+045F CYRILLIC SMALL LETTER DZHE

ս U+057D ARMENIAN SMALL LETTER SEH

ụ U+1EE5 LATIN SMALL LETTER U WITH DOT BELOW

### Variant set 24 – 6 Members (despite its name U+0263 is not confusable with its namesake Greek character and should be a singleton, U+1EF5 has issue of its own, but should not be part of the underlining logic and should be either removed from the LGR or made a singleton)

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~~

(This set introduces an in-script variant with Cyrillic due to inclusion of U+04AF; this should be subject of a separate discussion).

UPDATE: The XML contains 0263 in this variant set. It is listed in the Word document only as a repertoire item, not a variant. If it was truly intended to be a variant, it needs to be added to some table.

In Montreal, the Greek GP was strongly in favor of mapping 0263 based on the fact that the distinction between the two is an arbitrary convention: the range of generally encountered shapes for Greek gamma encompasses also the form chosen for Latin gamma, even if user-interface fonts tend to keep these two distinct. A Greek users may accept 0263 as an unusual font style, because of lack of awareness that a Latin clone of this letter even exists.

### Variant set 29 – 2 Members (for consistency with the current underlining policy, U+0063 and U+0441 should have been part of this set, but IP recommend to keep them in their own variant set (#3))

c U+0063 LATIN SMALL LETTER C

ç U+00E7 LATIN SMALL LETTER C WITH CEDILLA

с U+0441 CYRILLIC SMALL LETTER ES

ҫ U+04AB CYRILLIC SMALL LETTER ES WITH DESCENDER

### Variant Set 30 — 2 Members (resolution depends on the case for U+0061 and U+03B1)

á U+00E1 LATIN SMALL LETTER A WITH ACUTE
ά U+03AC GREEK SMALL LETTER ALPHA WITH TONOS

UPDATE: For variant set 1 we had noted: “The case for making a variant mapping between U+0061 and U+03B1 is still open”. The same applies to this set.

### Variant Set 50 — 6 Members (U+1EA5, U+1EA7, U+1EA9 should be removed and made singletons)

ă U+0103 LATIN SMALL LETTER A WITH

ǎ U+01CE LATIN SMALL LETTER A WITH

ӑ U+04D1 LATIN SMALL LETTER A WITH

ấ U+1EA5 LATIN SMALL LETTER A WITH

ầ U+1EA7 LATIN SMALL LETTER A WITH

ẩ U+1EA9 LATIN SMALL LETTER A WITH

UPDATE: While there may be a reasonable case for treating breve and caron as highly confusable (not least because they occur in different linguistic contexts) the same cannot be said for the Vietnamese stacked accents. We think that none of them is confusable with the single accents in this set (if anything they might possibly be confusable with a-circumflex), but they are all three limited to a single linguistic context where they are mutually distinct and used contrastively. That makes it unattractive in our view to designate them as variants.

## Variant sets added for diacritics above

UPDATE: this section is new

The latest version of the proposal document adds several subsections introducing variant sets based on diacritics above. Generally, the IP considers these at best confusable, because, even more so than diacritics below, they are generally distinguishable: in many languages there are many minimal pairs of words that are only distinct by the selection of a specific diacritic, and not merely the presence/absence of that diacritic. For example in French:

 adhérent    - adhèrent
  afférent    - affèrent
  compétent   - compètent
  excédent    - excèdent
  indifférent - indiffèrent
  précédent   - précèdent

Because native readers of the language need to distinguish these, it would make little sense to make them blocked variants of each other. The same applies to the cases of stacked diacritics for use in Vietnamese. Even though they may look quite similar to a non-native reader, the native reader will be trained to look for that small difference.

In principle, a case could be made for situations where a diacritic used in one language is substituted for one used in another language and where the two may be close enough in appearance that at least one of the user population could be expected to accept the substitution (presumably because they are not aware of the existence of the other diacritic).

This argument might be made for a variant between u-umlaut (German) and u with double acute accent (Hungarian). Native readers of German may be unaware of the double acute and may treat it as a font variation for an umlaut. If such a variant is introduced for non-overlapping use of diacritics across languages it would be notionally similar to the case of a cross-script variant: use of a label in a different linguistic context may block some putative label, but two labels that are distinct to the native reader wouldn’t be blocked from each other (as in the case of the Vietnamese stacked accents).

In the case of macron and tilde: users of some language using the tilde can be expected to be somewhat familiar with the concept of a macron, even if it isn’t used for everyday writing. Such familiarity may exist wrt to other accents more or less regularly encountered in neighboring or classical languages.

We recommend to the Latin GP to amend the in-script variants policy to add weight against making a pair of diacritics a variant when they (primarily) occur contrastively in the same language; and to add a small weight in favor of making them variants, if they are strongly similar, but occur in different linguistic environments, particularly if one of the populations can be expect to be unfamiliar with the other language.

In particular, the effects of transitivity need to be considered, so as to avoid unwieldy variant sets or resulting transitive variants among diacritics that are, or should be, considered distinct.

For example, collectively the policies inherent in the sections titled “Acute and Dot Above”, “Grave and Dot above”, “Acute and Hook Above”, and “Grave and Hook Above” would imply a transitive closure over Acute, Dot Above, Grave and Hook occurring on the same base letter.

However, in the actual table the same diacritics aren’t always mapped, so that for certain base characters, Acute and Grave aren’t made variants of each other, for example; which arguably is correct for a and e because of French. However, that makes the remaining entries in these sections somewhat ad-hoc (particularly the entire table in “Acute and Hook Above”).

The following sections present a detailed analysis of the remaining variant sets in this category. The comments and suggested ways of analyzing are only given for a sample of each kind of variant set. The IP expects the GP to review all of them using a similar methodology to the IP examples.

## Macron and Tilde

UPDATE: this section is new

### \*Variant set Macron and tilde: o – 2 members

\*This set is missing because of an error in the code point listed in the corresponding table.

Once that is corrected, the question whether 00F5 should be a variant of 014D should be reviewed as for the other sets here: the IP finds these two diacritics rather well distinguished in print, but notes that this pair is used in non-overlapping linguistic contexts.

### Variant set Macron and tilde: a – 2 members

ã U+00E3 LATIN SMALL LETTER A WITH TILDE
ā U+0101 LATIN SMALL LETTER A WITH MACRON

These seem to be used in non-overlapping linguistic contexts.

###  Variant Set 52 Tilde and Macron — 2 Members

ē U+0113 LATIN SMALL LETTER E WITH MACRON
ẽ U+1EBD LATIN SMALL LETTER E WITH TILDE

### Macron and tilde: g – 2 members (make singletons? or keep?)

g̃ U+0067 U+0303 LATIN SMALL LETTER G + COMBINING TILDE

ḡ U+1E21 LATIN SMALL LETTER N WITH MACRON

These seem to be used in non-overlapping linguistic contexts.

### Variant Set 56 Tilde and Macron — 2 Members

ĩ U+0129 LATIN SMALL LETTER I WITH TILDE
ī U+012B LATIN SMALL LETTER I WITH MACRON

### Macron and tilde: n – 2 members (make singletons)

n̄ U+006E U+0304 LATIN SMALL LETTER N + COMBINING MACRON

ñ U+00F1 LATIN SMALL LETTER N WITH TILDE

IP is not persuaded by the argument that handwritten macron substituted for tilde is enough of a case for this variant set: in print these are readily distinct. Many users of n-tilde can arguably be expected to know what a macron is, although that may not be the case for all of them. Generally, the user community for n-macron as part of an everyday orthography is very small and linguistically not overlapped; that could argue for making an exception and defining the variant anyway. IP expects GP to define an explicit rationale for this pair.

### Variant Set 60 Tilde and Macron — 2 Members

ũ U+0169 LATIN SMALL LETTER U WITH TILDE

ū U+016B LATIN SMALL LETTER U WITH MACRON

Non-overlapping linguistic context, but IP doesn’t find the tilde that similar to a macron.

### Variant Set - Macron and tilde: o – 2 members

õ U+00F5 LATIN SMALL LETTER O WITH TILDE
ō U+014D LATIN SMALL LETTER O WITH MACRON

In the XML, U+014D is mistakenly mapped to 006F based on a wrong code point in Section 6.5.2. Arguably the set listed here was the intended one.

As to evaluating the merits of this pair, similar arguments to the case of n-tilde/macron apply.

### Variant set - Macron and tilde: r – 2 members

r̃ U+0072 U+0303 LATIN SMALL LETTER R WITH TILDE

ř U+0159 LATIN SMALL LETTER R WITH CARON

There’s a mistaken U+0071 U+0303 in some version of the XML documentation which needs to be fixed.

These seem to be used in non-overlapping linguistic contexts, but, contrary to the section heading in the document, these are not examples of TILDE and MACRON, but instead, TILDE and CARON.

## Grave and Hook

UPDATE: this section is new

### Variant Set - Grave and Hook: a — 2 Members (make singletons?)

à U+00E0 LATIN SMALL LETTE A WITH GRAVE
ả U+1EA3 LATIN SMALL LETTER A WITH HOOK ABOVE

We note that a-grave is used in French and in Vietnamese. Neither of these languages is cited for U+00E0 in the XML (although French has been added in the Word document).[[3]](#footnote-3) The use of Grave and other accents (acute, circumflex, tilde, hook, dot below) for tone in Vietnamese should be cited explicitly in the repertoire table in Section 5 of the proposal (as well as the XML), even when they occur singly and not part of a stacked combination. At the moment, the listing is incomplete.

Because a-hook is used in Vietnamese contrasting with a-grave, this argues against making them variants. IP feels that visually, a-hook is probably distinct enough to allow it to be treated as merely confusable by users of languages other than Vietnamese.

### Variant Set 74 Grave and Hook Above — 2 Members

ừ 1EEB LATIN SMALL LETTER U WITH HORN AND GRAVE
ử 1EED LATIN SMALL LETTER U WITH HORN AND HOOK ABOVE

Contrasting use in Vietnamese only.

### Variant Set - Grave and Hook Above: y — 4\* Members (The entry U+1EF9 is mistaken; the XML should contain an entry for U+00FD, but the relevant entry in the proposal document repeated U+1EF3, collapsing the entries)

ý U+00FD LATIN SMALL LETTER Y WITH ACUTE

ỳ U+1EF3 LATIN SMALL LETTER Y WITH GRAVE

ỷ U+1EF7 LATIN SMALL LETTER Y WITH HOOK

~~ỹ U+1EF9 LATIN SMALL LETTER Y WITH TILDE~~

U+1EF9 is a mistaken code point (y-tilde); 1EF7 is intended. The comment “Acute and Hook Above” in the XML is an indication that U+00FD was also intended to map to U+1EF7.

## Acute and Dot Above

### Variant Set 51 Acute and Dot Above — 2 Members

ć U+0107 LATIN SMALL LETTER C WITH ACUTE
ċ U+010B LATIN SMALL LETTER C WITH DOT ABOVE

Non-overlapping linguistic contexts (if attribution in repertoire table is exhaustive).

### Variant Set 54 Cedilla and Dot Above — 2 Members

ġ U+0121 LATIN SMALL LETTER G WITH DOT ABOVE
ģ U+0123 LATIN SMALL LETTER G WITH CEDILLA

Non-overlapping linguistic contexts (if attribution in repertoire table is exhaustive).

### Variant Set 58 Acute and Dot Above — 2 Members

ń U+0144 LATIN SMALL LETTER N WITH ACUTE
ṅ U+1E45 LATIN SMALL LETTER N WITH DOT ABOVE

Non-overlapping linguistic contexts (if attribution in repertoire table is exhaustive).

### Variant Set 62 Acute and Dot Above — 2 Members (make singletons?)

ź U+017A LATIN SMALL LETTER Z WITH ACUTE

ż U+017C LATIN SMALL LETTER Z WITH DOT ABOVE

These are used contrastively in Polish (1); but also, non-contrastively in some much smaller communities. A question of judgment, but the IP would tend to argue against a variant relation here.

## Breve and Caron

UPDATE: this section is new

### Variant Set 53 Breve and Caron — 2 Members

ğ U+011F LATIN SMALL LETTER G WITH BREVE
ǧ U+01E7 LATIN SMALL LETTER G WITH CARON

### Variant Set 61 Breve and Caron — 2 Members

ŭ U+016D LATIN SMALL LETTER U WITH TILDE

ǔ U+01D4 LATIN SMALL LETTER U WITH CARON

Breve and caron appear hard to tell apart and are not used in the same linguistic context.

## Vietnamese accents and tone marks

UPDATE: this section is new

### Variant Set 71 Vietnamese Accents on a — 3 Members (these should be singletons)

ắ U+1EAF LATIN SMALL LETTER A WITH BREVE AND ACUTE

ằ U+1EB1 LATIN SMALL LETTER A WITH BREVE AND GRAVE

ẳ U+1EB3 LATIN SMALL LETTER A WITH BREVE AND HOOK ABOVE

### Variant Set – Vietnamese Accents: e — 3 Members (all of these should be singletons)

ế U+1EBF LATIN SMALL LETTER O WITH CIRCUMFLEX AND ACUTE

ề U+1EC1 LATIN SMALL LETTER O WITH CIRCUMFLEX AND GRAVE

ể U+1EC3 LATIN SMALL LETTER O WITH CIRCUMFLEX AND HOOK ABOVE

(see discussion below)

### Variant Set - Vietnamese Accents: o — 6 Members (all of these should be singletons)

ố U+1ED1 LATIN SMALL LETTER O WITH CIRCUMFLEX AND ACUTE

ồ U+1ED3 LATIN SMALL LETTER O WITH CIRCUMFLEX AND GRAVE

ổ U+1ED5 LATIN SMALL LETTER O WITH CIRCUMFLEX AND HOOK ABOVE

ớ U+1EDB LATIN SMALL LETTER O WITH HORN AND ACUTE

ờ U+1EDD LATIN SMALL LETTER O WITH HORN AND GRAVE

ở U+1EDF LATIN SMALL LETTER O WITH HORN AND HOOK ABOVE

The entire set should be removed. These are all in use in a single linguistic context (Vietnamese) where they are used contrastively and users can be expected to look for the distinction, even if it isn’t very apparent to non-native users. None of the proposed mappings would alleviate any spoofing of non-Vietnamese labels.

# Other variant sets included in the LGR

The IP is not convinced that the all member of the two following variant sets are confusable.

### Variant set 43 – 3 Members (U+1EA7 looks different from the others and could be a singleton)

ă U+0103 LATIN SMALL LETTER A WITH BREVE

ӑ U+04D1 CYRILLIC SMALL LETTER A WITH BREVE

ầ U+1EA7 LATIN SMALL LETTER A WITH CIRCUMFLEX AND GRAVE

### Variant set 47 – 3 Members (U+03C3 looks quite different from the others and could be a singleton)

ơ U+01A1 LATIN SMALL LETTER O WITH HORN

σ U+03C3 GREEK SMALL LETTER SIGMA

ợ U+1EE3 LATIN SMALL LETTER O WITH HORN AND DOT BELOW

## Possibly mistaken reflexive mappings

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.

UPDATE: code point sequence 0071 0303 has an unknown variant type of “out-of-repertoire” (missing “-var”). It appears to be a typo for 0072 0303 anyway, but it needs to be corrected.

## 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 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.

UPDATE: This variant is listed in the LGR now, but has not been finally reviewed by IP.

## Possibly extraneous code point

UPDATE: Code point U+0115 shows up as variant in some tables, but is not a part of the repertoire.

Code point 0251 shows up in the research notes, but is not part of the repertoire. Some reviewers requested that it be explicitly marked as excluded, to avoid confusion.[[4]](#footnote-4)

# Allocatable variants

We note that the spelling variation between “ss” and “ß“ between standard Swiss and other German-language orthographies has been suggested as allocatable in the proposal by the Latin GP by adding an allocatable variant from ‘ß’ to ‘ss’ (but not in the reverse direction). This appears based on the fact that Swiss users normally use only the “ss” form with no “ß“ on their keyboards while not all German words spelled with “ss” have a possible equivalent spelling with “ß“. In the proposal, there is no formal restriction on the number of possible allocatable variant labels for any label containing “ß“ although there may be an implicit limit on the number of “ß“ in any labels other than some rather fanciful ones.

 However, it seems sensible to adopt a restriction requiring that either all “ß“ are mapped to “ss” or none. This cannot be addressed by a “no-mix” rule in the style used in the Arabic LGR, because “ss” and “ß“ can and do coexist in German words, but it could be achieved by using a subtype for the variant type with a corresponding <action> defined that resolves only labels where all mappings are taken as “allocatable”.

## Addendum: Minimizing allocatable variants

As defined the variant set can lead to **multiple allocatable variants**. With German labels (especially with German compound nouns) there can easily be multiple "ß". A non-contrived example is "Außenmaß" (outer dimension). As a label, this would have 4 allocatable variants, but only two are needed (the Swiss would write all "ß" with "ss", not just the first or the last). However, it is possible to mix "ss" and "ß" in German, as in "Außeninteresse" (outside interest). That label would have only one allocatable variant, which is fine; however, the alternation between "ss" and "ß" makes it impossible to define a "no-mix" rule between "ss" and "ß".

The situation of multiple orthographies sharing a code space looks a bit like Chinese variants, where simplified and traditional ways of writing coexist with those that are used in both. In that scenario, various forms of “subtyping” of the “allocatable” variant type are used. For a generic discussion see for example RFC 8228.

What if we made "ß" a reflexive "r-german" and gave the mapping "ß" --> "ss" the type "swiss"?

Then we could define these actions:

<action disp="allocatable" all-variants="swiss" />

<action disp="allocatable" all-variants="r-german" />

Any mixed variant label that contains both an original "ß" but also some that have been mapped to "ss" would fail the "all-variants" test, but one that either has all the original "ß" or only those mapped to "ss" would pass. (Any original "ss" wouldn't have a typed variant at all and therefore be processed normally).

For those not familiar with reflexive variants: these map the code point to itself and are “applied” to the original label, or any code position that that is unchanged in a given variant label permutation.

UPDATE: The “all-variants” test above makes sure that all variant mappings used in a particular permuted label are from the same set. (Any code point left original would be represented by its reflexive mapping where it exists). The latest iteration of the LGR now contains these mappings, but lacks the definition of the corresponding <action> elements in the XML; however they are noted in the document.

(There are some edge cases that this scheme may not support, such as any word spelled with two "ß" in the old orthography, but only one in the modern one: you wouldn't be able to support both old and modern orthography in that case - in the view of the IP, the need to reduce multiple allocatable variants would tend to outweigh that level of linguistic fidelity - and the number of non-contrived examples should be rather low in any case: dual "ß" are not that frequent to begin with).

Finally, let's check a triple "sss" to see whether this would get into any trouble:

ѕѕѕ --> sss ->  --> sß  / ßs --> ṣṣṣ --> ѕѕѕ

It looks like both partitions {s}{ss} and {ss}{s} are well-behaved. Because we only had an overlap, not a null-variant, the sequences cannot get longer. As long as we handle the overlap by adding all the other variants for "ss" into the variant set with "ß" as suggested above, the resulting variant set should be well-behaved.

NOTE: it is true that "sß"  is an impossible combination in a German word (since "ß" has to follow a vowel, so "ßß" is likewise impossible). Should such combination be explicitly banned in labels, or is non-existence in German text to be considered a mere spelling rule?

To answer this, consider “xkcd” which is not a word in any language written in the Latin script and yet, there's <https://xkcd.com/>.

In the Latin script, whether these can occur as words or not is based on spelling rules which are not enforced in the LGR. (Other than for combining marks, all combinations go).

## Addendum: Overlapped variants

There is an **overlapped** v**ariant** between the variants defined for “ss” and “s”. (See RZ-LGR-3 section 6.6 for a general discussion). In the Latin LGR, there is a cross-script variant for s (and the Latin GP also maps it to some combinations of s+diacritic as well). That "s" overlaps with the sequence "ss". As a result, the "ss" in a label like "pass" (passport) would have some variants which themselves would not have a "ß" as a variant. Therefore, transitivity among variant labels is broken.[[5]](#footnote-5)

E.g. paṣṣ ---/---> paß ---> pass --->  paṣṣ and paṣṣ <---/-- paß <--- pass <---  paṣṣ

(In this example "paß " isn't a spelling that is currently correct, but until recently, in the old pre-1996 orthography that many users are still familiar with, it used to be the standard one.)

The same applies to the cross-script variant to the Cyrillic Dze (U+0455) which looks like "ѕ":

раѕѕ <---/---> paß <---> pass <---> раѕѕ

In order to make the variant label set well-behaved, the solution is to add all the other variants for "ss" into the variant set with "ß". Those variants would be derived from mapping “ss” to all possible combinations of pairs of variants for “s” (including mixed pairs).

UPDATE: “All possible” turns out to be quite a large number, and with transitive closure the number of line items in a full listing of the variant set quickly exceeds 300. That’s arguably not defensible. So we need to attack this issue on two fronts.

One is to limit the variants of “s”. See other parts of this document on that issue.

The other is to limit the permutations listed to those that are actually needed. A quick analysis will show that any mixed-script variant sequences for “ss” (those that contain Cyrillic U+0455 “ѕ” with Latin “s” and any of the other variants of “s”) can never occur in any valid label. Therefore, it is never necessary to compute an index variant for them. At the same time, none of them can be an index variant, because any index variant for a sequence of type AA with partition {A}{A} would have its smallest variant when each {A} maps to the ***same*** smallest variant for A, therefore the variant for AA would be in a single script.

As these considerations show, we can safely disregard any mixed-script variants for “ss” for the purpose of creating transitive closure over the ***label*** variant set for any label containing “ss” or one of its variants.

There are good arguments in favor of also eliminating some of the variants of “s” by splitting the variant set. The resulting variant set then becomes rather small. This is particularly important because this set involves cross-script variants and therefore affects two other LGRs. (It would be even simpler, if s-dot below were not treated as a variant of “s”, see above in the discussion of the relevant variant set.)

|  |  |  |
| --- | --- | --- |
| Glyph | Code Point | Name |
| ss | U+0073 U+0073 | LATIN SMALL LETTER S + LATIN SMALL LETTER S |
| sṣ | U+0073 U+1E63 | LATIN SMALL LETTER S + LATIN SMALL LETTER S WITH DOT BELOW |
| ß | U+00DF | LATIN SMALL LETTER SHARP S |
| β | U+03B2 | GREEK SMALL LETTER BETA |
| ѕѕ | U+0455 U+0455 | CYRILLIC SMALL LETTER DZE + CYRILLIC SMALL LETTER DZE |
| ṣs | U+1E63 U+0073 | LATIN SMALL LETTER S WITH DOT BELOW + LATIN SMALL LETTER S |
| ṣṣ | U+1E63 U+1E63 | LATIN SMALL LETTER S WITH DOT BELOW + LATIN SMALL LETTER S WITH DOT BELOW |

# Other characters not included in LGR variant sets

Other characters that are part of the Latin LGR have confusability issues that have not been explored in the proposal. These are:

ď U+010F LATIN SMALL LETTER D WITH CARON

ľ U+013E LATIN SMALL LETTER L WITH CARON

Check for example the French word ‘aujourd’hui’ (correct spelling with an apostrophe or better a smart quote) ‘aujourďhui’ (using U+010F); or ‘l’ornière’ (correct spelling), versus ‘ľornière’ (using U+013E). The confusion is more pronounced with serif fonts than with sans serif fonts, but it still is a concern that should be addressed by the Latin GP, including possible removal of these two characters from the LGR repertoire.

Such a restriction might seem, at first blush, as too limiting; however, it should be considered in the context of a similar exclusion for 02BC MODIFIER LETTER APOSTROPHE.

UPDATE: some French web sites (like https://lefigaro.fr) use extreme font kerning for the display of L or D followed by apostrophe, meaning that the apostrophe does not use any advance width and looks exactly like a L or D Caron. It is especially true on iPhone Safari, not so much on Windows browsers, but may occur even there: “…” (The site may serve different pages to different platforms.) Even if one assumes that the kerning only happens where the mark is medial, it's still possible to spoof labels. Take the example:

    .aujourďhui  ≈ .aujourd + ' + hui

where ď is d-caron on the left, '  is an apostrophe and "hui" is not part of the label on the right, but would look like it is. Unless some linkification process clearly marks that the “’hui” are not part of the label, the user will be unaware that they will reach “.aujourd” instead.

At minimum, the Latin GP needs to be aware of that issue and explicitly discuss it.

# Variants based on Generic Glyphs

UPDATE: this is a new section, but it incorporates material previously submitted under separate cover as “Partial IP Response to Latin LGR Update”, but with some further edits in detail (not tracked).

In December, the IP reviewed the set of putative variants based on generic glyphs. (These are now listed in Section 6.3.4 of the Latin LGR proposal document, where they are explicitly defined as variants). In cases where suggested variants appeared doubtful, the IP decided to test them by constructing a possible label and entering it into a variety of address bars (where the user has generally no control over the font selection for each script). This properly evaluates not simply the outline (ink) of a glyph, but its placement and relative size as well, all factors that affect the possible equivalence of simple shapes.

We expect the Latin GP to review these observations and to make suggested changes to the table in Section 6.3.4 and the variant sets defined in the XML. This should include providing explicit reference to sample labels in the manner discussed below (probably in an appendix) as part of the rationale for the decision to add specific code point variants.

## Detailed observations

(1) **Cyrillic Palochka**  -- see our discussion of the Cyrillic LGR in the LGR-3 overview

(2) **Arabic Letter Alef** – the IP discussed this at length. The tendency seems to be for fonts to show this letter as somewhat floating above the common baseline:

   

 (Firefox on Windows) (Chrome on Mac)

Thus the shape is similar to, but not indistinguishable from Latin small L or dotless i. In the IP’s estimation this would speak against a variant relation, but is perhaps worth calling out as potential confusable.

(3) **Oriya vowel sign Aa** -- as a combining mark it requires the presence of another Oriya letter. Presumably the case is made that alternated with Oriya letter Ttha it mimics Latin ololo.. -- however the Oriya label lacks the alternation in height: ଠାଠାଠ… (We would expect cases like to be analyzed in this fashion, i.e. on the basis of the possible conflicting labels, with the conclusion that full variant status is not needed for vowel sign Aa; in contrast, the case for Tha is a bit stronger: an Oriya sequence ଠଠଠ, if presented in any FQDN without another Latin component, could be more easily read as ooo ).

(4) **Oriya Letter Ttha and Malayalam Letter Tha** -- these appear to take up the full full-height. Should this affect their variant set? Or can an argument be made that this particular shape is "high risk", esp. if presented with a FQDN not containing European x-height letters for comparison (e.g. a Chinese 2nd level domain name combined with .ଠଠଠ as variant for the same with a TLD of .ooo?) An explicit argument, e.g. in a footnote, may be required one way or the other.

Current IP thinking favors to treat the significant and consistent difference in height argues for these two code points 0D20/0B20 as a rationale for them to be in their own variant set, and distinct from Latin o (and its mappings).

(5) **Myanmar Vowel Sign Sgaw Karen** Eu -- the case here is perhaps stronger than for the Oriya vowel sign, as there isn't a height difference to Myanmar letter Wa.

Here is the comparison of the full label: Latin =  ooıoıoı, Myanmar =  (note use of a screenshot for the second one resulting in a bit of a vertical displacement). However, if the little hook to the left is in fact something users would ignore, the Myanmar label might also be close to Armenian = օօւօւօւ, which may or may not be desired from the perspective of transitivity.

While there can be a size difference, at smaller font sizes, in browser UIs the only difference remaining is the hook.

 

(firefox on Windows) (Wikipedia search field, Firefox, Windows)

(Left: Latin small o and dotless i)

The IP considers the hook to be a deal-breaker when it comes to making a variant mapping to dotless I, other than via transitivity.

(6) **Hebrew letter vav** -- the hook at the top is rather heavy in any serifed font and would show up well, if the letter is repeated. (If would have to be repeated, if not alternated with the only other candidate Samekh, in any whole-script confusable label. Note that Samekh shows even more distinct signs of a typical Hebrew ductus.) However, we find that some common user interface fonts reduce these characteristics to the point where the code point becomes indistinguishable from lowercase letter dotless I (not L).

  

(Calibri) (Lucida) (FF on windows) (Chrome, Firefox, opera and Safari on Mac)

In light of the choices made by modern sans-serif fonts (used by default in browser user interfaces) identifying vav as a variant of dotless I would seem motivated.

ıoıoıo 

(Latin dotless I and small letter o for comparison) (Latin on the right)

The equivalence isn’t perfect given that in some cases the Hebrew letters are shown smaller than the Latin ones, but given that ICANN delegated a .ooo TLD already, the risk seems there. If added, requires explicit discussion.

(7) **Hebrew letter Samekh** -- like vav, this one appears to have a rather identifiable script ductus in serifed fonts, but lacks any distinction from small letter o in sans-serif.

(See item 6)

(8) **Ethiopic Syllable Pharyngeal A** -- like Hebrew letter Samekh, this one appears to have a rather identifiable script ductus in serifed fonts, and an identifiable shape even in sans-serifed fonts; the IP notes that this was discussed already in the context of LGR-2 and rejected (in the context of a putative variant with Armenian small letter Oh U+0585). We think it should be considered a confusable (at worst), and then probably more with digit zero (e.g. on the second level). Worth documenting in an appendix.



(9) **Cyrillic small letter ES** -- the glyph shown is uppercase. Please fix.
(Reviewers noted that there appeared to be issues like that in other parts of the text)

 (10) **Lao vowel sign E** -- like Lao letter Wo, the IP considers the presence of a tell-tale circle at the head of the stroke a strong characteristic of the script ductus for the Lao script. This feature is clearly visible in browser address bars as well. The code point is also a combining mark and can only occur in alternation with Lao Letter Wo. The latter has a different aspect ratio then the putative Latin counterpart, and an even more pronounced leading circle. (We would expect this case to be analyzed in terms of a putative complete label, and, if that is done, it becomes obvious - at least to us - that this is at best a confusable). Latin = ɔcɔc   Lao = ວເ ວເ

(9) **Myanmar Letter Nga** -- while this is more circular than Latin C, there are enough Latin fonts where c is fairly circular so that Latin ococ and Mynamar =  (screenshot) look fairly identical

(10) **Myanmar** Vowel sign Aa: Arguably the labels Latin = oɔoɔ and Myanmar =  (screenshot) look nearly identical (because there's a slight difference in connectedness, this proposed variant relation is a bit more marginal than for (9).)

The LGR should contain samples of these labels as part of any rationale for adding variants.

1. <https://css-tricks.com/almanac/properties/t/text-decoration-skip-ink/> [↑](#footnote-ref-1)
2. See section 5.2 [↑](#footnote-ref-2)
3. Note that this comment relates to the listing of the repertoire. We would like to call attention to this, so it gets fixed in the appropriate place. Because variant analysis for diacritics properly depends on knowing where a mark is used, it is important for the attribution by language to be exhaustive and complete. [↑](#footnote-ref-3)
4. Note that these comments affect the listing of the repertoire. [↑](#footnote-ref-4)
5. The ability to shortcut the match for blocked variants depends on well-behaved sets for variant labels. As also discussed in RFC 8228, having well-behaved sets of code point variants is not by itself a guarantee for well-behaved sets of variant labels if sequences are involved. (Additional guidance can be found in RZ-LGR-3 Section 6). [↑](#footnote-ref-5)