OpenType Features in JuniusX/JuniusVF

Features are applied in the following order (see also index):

1. c2sc – Small Capitals from Capitals

Use with smcp for all-small-cap text. All lower- and uppercase pairs have a small cap equivalent. ABCDE → ABCDE.

2. smcp – Small Capitals

Converts lowercase letters to small caps (when available—lowercase letters without matching caps may lack matching small caps); also several symbols and combining marks. fghij → FGHIJ.

3. pcap – Petite Capitals

Produces small caps in a smaller size than smcp. Use these when small caps have to be mixed with lowercase letters. The whole of the basic Latin alphabet is covered, plus several other letters. klmno → KLMNO.

4. case – Case-Sensitive Forms

Produces combining marks that harmonize with capital letters: Ř Ẋ, etc.

5. nalt – Alternate Annotation Forms

Produces letters and numbers circled, in parenthesis, or followed by periods, as follows:

With index 1, circled letters or numbers: ⓐ ⓑ . . . ⓩ; ⓪ ① ② . . . ⑳.
With index 2, letter or numbers in parentheses: (a) . . . (z); (0) (1) . . . (20).
With index 3, double-circled numbers: ⓪ ⓵ . . . ⓾.
With index 4, white numbers in black circles: ⓿ ❶ ❷ ❸ . . . ⓴
With index 5, numbers followed by period: 0. 1. . . . 20.

For enclosed figures 10 and higher, rlig (Required Ligatures) must also be enabled (as it should be by default: see Required Features below).

6. tnum – Tabular Figures

Fixed-width figures: 0123456789 (default or with lnum), 0123456789 (with onum).

7. onum – Oldstyle Figures

Figures that harmonize with lowercase characters: 0123456789 (default or with tnum), 0123456789 (with pnum). When combined with pnum, this feature also affects subscripts and superscripts.

8. pnum – Proportional Figures

Proportionally spaced figures: 0123456789 (default or with lnum), 0123456789 (with onum). When combined with onum, this feature also affects subscripts and superscripts.

9. lnum – Lining Figures

Figures in a uniform height, harmonizing with uppercase letters: 0123456789 (default or with tnum), 0123456789 (with pnum).

10. zero – Slashed Zero

Produces slashed zero in all number styles: 0̸ 0̸ ⌀ ⌀. Includes superscripts and subscripts: ⁰̸ ₀̸ ₀ ⁰.

11. sups – Superscripts

Produces superscript numbers and letters. Only affects lining tabular and oldstyle proportional figures. All lowercase letters of the basic Latin alphabet are covered, and most uppercase letters: ⁰¹²³ ⁴⁵⁶⁷ ᵃᵇᶜᵈᵉ ᴬᴮᴰᴱᴳ.

12. subs – Subscripts

Produces subscript numbers. Only affects lining tabular and oldstyle proportional figures: ₈₉₀₁ ₂₃₄₅.

13. ornm – Ornaments

Produces ornaments (fleurons) in either of two ways: as an indexed variant of the bullet character (U+2022) or as a variant of a-z, A-C (all fleurons are available by either method):

As a variant of •: 1=❀, 2=❁, 3=❃, 4=⚜, etc., up to 29.

As a variant of a-z, A-C: e=❀, f=❁, g=❃, h=⚜, etc.

The method with letters of the alphabet is easier, but the method with bullets will produce a more satisfactory result when text is displayed in an environment where JuniusX is not available or ornm is not implemented.

14. `ss01` – Alternate thorn and eth

Produces Nordic thorn and eth (þðÞ) when the language is English, and English thorn and eth (þðƿ) with any other language. This also affects small caps, crossed thorn (ƿ ƿ), combining mark eth (U+1DD9, ◌ᵭ ◌ᵭ), and enlarged thorn and eth (see `ss06`). This feature depends on `loca` (Localized Forms), which in most applications will always be enabled.

15. `ss02` – Insular Letter-Forms

Produces insular letter-forms, e.g. ðƒɡɲɾƿ. Does not affect capitals (except W), as these are not commonly used in early manuscripts. For these, use the Character Variant (cvNN) features.

16. `hist` – Historical Forms

Changes s to ſ (longs).

17. `ss03` – Long s

Changes s to ſ (duplicating hist). see also `ss08`.

18. `ss04` – High Overline

Produces a high overline over letters used as roman numbers, e.g. $\overline{\text{cdij}}$ $\overline{\text{LMVX}}$.

19. `ss05` – Medium-High Overline

Produces a medium-high overline over (or through the ascenders of) letters used as roman numbers, e.g. $\overline{\text{cd̄ıȷ}}$.

20. `ss06` – Enlarged Minuscules

Lowercase letters that match the height of normal ones, but with a higher x-height, e.g. abcdefg. Covers the whole of the basic Latin alphabet and several other letters: consult the MUFI recommendation for more details.

21. `ss07` – Underdotted Text

Produces underdotted text (indicating deletion in many medieval manuscript) for many letters (including the whole of the basic Latin alphabet and a number of other letters), e.g. ạḅc̣ḍẹf̣g̣ Ḥḷ̣ḵ̣Ḳ̣Ḷ̣Ṃ. This also affects small caps, e.g. ABCDEF → ẠḄC̣Ḍ̣Ẹ̣F̣. For letters without corresponding underdotted forms (e.g. U+A751, ꝑ), use U+0323, combining dot below (ꝑ̣).

22. `ss08` – Contextual Long s

In English and French text only, varies s and ſ according to rules followed by many early printers: ſports, eſſence, ſtormy, diſheveled, transfuſions, ſyneſs, cliffside. For this feature to work properly, `calt` (Contextual Alternates) must also be enabled (as it should be by default: see Required Features below).

23. `ss11` – r Rotunda

In lowercase and small caps, substitutes r rotunda (ꝛ) for r. See also `ss16`

24. `ss12` – Early English Futhorc

Changes Latin letters to their equivalents in the early English futhorc. Because of the variability of the runic alphabet, this method of transliteration may not produce the result you want. In that case, it may be necessary to manually edit the result. fisc flodu ahof → ᚠᛁᛋᚳ ᚠᛚᚩᛞᚢ ᚪᚻᚩᚠ.

25. `ss13` – Elder Futhark

Changes Latin letters to their equivalents in the Elder Futhark. Because of the variability of the runic alphabet, this method of transliteration may not produce the result you want. In that case, it may be necessary to manually edit the result. ABCDEFG → ᚨᛒᚲᛞᛖᚠᚷ.

26. `ss14` – Younger Futhark

Changes Latin letters to their equivalents in the Younger Futhark. Because of the variability of the runic alphabet, this method of transliteration may not produce the result you want. In that case, it may be necessary to manually edit the result. ABCDEFG → ᚼᛒᚴᛏᛁᚠᚴ.

27. `ss15` – Long Branch to Short Twig

In combination with `ss14`, converts long branch to short twig runes: ᚼᛒᚴᛏᛁᚠᚴ → ᚻᚠᚴᛚᛁᚠᚴ.

28. `ss16` – Contextual r Rotunda

Converts r to ꝛ following the most common rules of medieval manuscripts: pꝛiest, firmer, frost, oꝛnament. For this feature to work properly, `calt` (Contextual Alternates) must also be enabled (as it should be by default: see Required Features below).

29. ss18 – Old-Style Punctuation Spacing

Colons, semicolons, parentheses, quotation marks and several other glyphs are spaced as in early printed books.

30. ss19 – Latin to Gothic Transliteration

Produces Gothic letters from Latin: Warþ þan in dagans jainans → ϒλϰψ ψλɴ ɪɴ ⱔλɾλɴs ϩλɪɴλɴs.

31. cv01 – Variants of aA

1=ⱔⱅ, 2=ⱅⱔ, 3=ⱅ, 4=ⱔ, 5=ⱔ.

32. cv02 – Variants of ąĄ

1=ąⱔⱔ, 2=ąⱔ, 3=Ąⱔ

cv03

There are no variants of bB.

33. cv04 – Variant of C

1=Ⱅ

34. cv05 – Variants of dD

1=ⱔⱔⱔ, 2=ⱔ.

35. cv06 – Variant of đ (U+0111, d with stroke)

1=ⱔ

36. cv07 – Variants of eE

1=ϵⱔϵ, 2=eⱔ, 3=e.

37. cv08 – Variants of ęĘ

1=ęĘęęęęęę (etc.), 2=ęĘę.

38. cv09 – Variants of f F

1=ƒƒƒƒ, 2=ƒ, 3=ƒ, 4=ƒ, 5=f.

39. cv10 – Variants of Gg

1=ʒʒʒ, 2=gⱅ, 3=g, 4=g, 5=g.

40. cv11 – Variants of ʒ3 (Yogh)
1=ʒ3ʒʒ̣ʒ̣.

41. cv12 – Variants of hH
1=ɧƕ.

42. cv13 – Variants of iI
1=ɪɪ, 2=ɪ◌ⁱ.

43. cv14 – Variants of j
1=ɟ◌ʲ

44. cv15 – Variants of k
1=k, 2=ƙ, 3=ƙ.

45. cv16 – Variant of l
1=ɭ.

46. cv17 – Variant of ƚ (U+A749, l with high stroke)
1=ƚ.

47. cv18 – Variants of mM
1=ɯɱɯ◌ᵐ, 2=ɱɱɱ, 3=ɱᴍ.

48. cv19 – Variants of nN
1=ŋɴɴ, 2=ɴɴ.

cv20
There are no variants of oO.

49. cv21 – Variants of ø
1=ơ, 2=ø, 3=o, 4=ơ.

50. cv22 – Variant of P
1=ꟼ.

51. cv23 – Variant of Q
1=ꟼ.

52. cv24 – Variants of rR
1=ɿ2ɿ2̊, 2=ɲƝɲ.

53. cv25 – Variants of sS
1=ɾƔɾ, 2=ѕȣ, 3=ſ, 4=ʃ.

54. cv26 – Variants of tT
1=ʈƦʈ̊.

cv27
There are no variants of uU.

55. cv28 – Variants of vV
1=ƿƔƿ, 2=ʋƱ.

cv29
There are no variants of wW.

56. cv30 – Variants of x
1=ᵡ, 2=ᵡ.

57. cv31 – Variants of y
1=ᵧ, 2=ᵧ.

58. cv32 – Variant of Z
1=Ʒ.

59. cv33 – A to a
l=a. This features reverts small cap A to a, enabling it to ligature with small cap N or R via `hlig`: aN, aR. Be sure to apply `smcp`, `cv33` and `hlig` to both components of the ligature.

60. cv34 – Variant of aa
1=aa.

61. cv35 – Variants of æÆ
1=æƆ, 2=æ.

62. cv36 – Variants of ꝏꝎ

1=ꝏꝎꝏ.

63. cv37 – Variant of ꝋ

1=ꝋ.

cv38–cv39

Reserved.

64. cv40 – Variants of ⁊ (U+204A, Tironian nota)

1=ꝛ, 2=ꝛ.

65. cv41 – Variant of ⁊ (U+A75D, rum abbreviation)

1=ꝝ.

66. cv42 – Variants of ◌꞉ (U+035B, combining zigzag above)

1=◌꞉, 2=◌꞉, 3=◌꞉. Positioning of the zigzag can differ from that of other combining marks, e.g. b̓, f̓, d̓. If `calt` "Contextual Alternates" is enabled (as it should be in most apps), variant forms of alternate 2 will be used with several letters, e.g. d̓, f̓, k̓. Enable `case` for forms that harmonize with capitals (Ǻ B̓ C̓ D̓), `smcp` for forms that harmonize with small caps (Ḗ F̓ G̓ H̓).

67. cv43 – Variant of ◌ꞇ (U+1DD3, combining open a)

1=◌ꞇ.

68. cv44 – Variant of ◌ꞇ (U+1DE3, combining r rotunda)

1=◌ꞇ.

69. cv45 – Variant of ◌◌ (U+0305, two-letter overline)

1=◌◌.

70. cv46 – Variant of ◌ (U+0303, combining tilde)

1=◌.

cv47–cv49

Reserved.

71. `cv50` – Variant of ʔ (U+0294, glottal stop)

1=ʔ.

72. `cv51` – Variant of ? (question mark)

1=ʕ.

73. `cv53` – Variants of short horizontal stroke (U+0335)

1= ⊖, 2= ⊖, 3= ⊖

This character can be used with letters with ascenders or descenders, e.g. đ ƀ þ ƿ. `cv53` with index=1 (the default) widens the stroke, and indexes 2 and 3 offset the stroke to the right or left. Via `calt`, this offset is performed automatically for certain characters with ascenders, e.g. ƀ đ þ ƿ. Thus it should rarely be necessary to use an index with `cv53`.

`cv53–cv99`

Reserved.

74. `rtlm` – Right to Left Mirrored Forms

Produces mirrored runes, e.g. ᚠᛒᚳᛞᛖᚠᚷ → ᛉᛒᛅᛞᛖᛉᚷ.

75. `hlig` – Historic Ligatures

Produces ligatures for combinations that should not ordinarily be rendered as ligatures in modern text. This feature does not produce digraphs (which have a phonetic value): see ss17 for these. The ligatures: Hr→ħ O2→ꝏ; O2̧→ꝏ̨; PP→ꝑ; UE→Ꝺ; UU→Ꝺ; af→ꝺ; aꝼ→ꝼ; ag→ꝺ; al→ꝺ; an→ꝺ; aɴ→ꝺɴ (for ꝺɴ and ꝺʀ see cv33 above); ap→ꝺ; ar→ꝺ; aʀ→ꝺʀ; aþ→ꝺþ; bb→ƀƀ bg→ƀ; ch→ꝺ; ck→ꝺk; ðð→ðð; ey→ey; fä→fü; gd→ꝺ; gð→ꝺ; gð→ꝺ; gg→ꝺ; ꝺꝺ→ꝺ; go→ꝺ; gp→ꝺ; gr→ꝺ; hr→ħ; kr→ƙ; ll→ll; ɴſ→ɴſ; (U+017F ſ must be entered directly for this ligature to work properly); oc→ꝏ; oꝛ→oꝛ; oꝝ→oꝝ; pp→ꝑ; ꝓp→ꝓ; qv→ꝗ; qꝫ→ꝗ; qꝛ→ꝗ; ſä→ſü; ſcɦ→ſcɦ; ſtr→ſtr; ſv→ſ; ꝛꝛ→ꝛꝛ; ue→ꝺe; uu→ꝺ; þr→þ.

76. `dlig` – Discretionary Ligatures

Produces lesser-used ligatures, but also roman numbers, e.g. ii, II, xi, XI. The lesser-used ligatures: ct, ſp, ſtr, st, tr, tt, ty.

77. `ss17` – Rare Digraphs

By "digraph" we mean conjoined letters that represent a phonetic value: the most common examples for western languages are **æ** and **œ** (though these, because they are so common, are not included in this feature). Use of this feature in web pages enables easier searches: for example, producing **þaɩ** from **þau** allows the word to be searched as "þau." The digraphs covered by this feature are aɩ, ɔɔ, aɩ, ɑ̃, ɑ̃, ɔɔ, ɯ, plus capital and small cap equivalents and digraph + diacritic combinations anticipated in the MUFI recommendation. To produce such a digraph + diacritic combination, either type a letter + diacritic combination as the second element of the digraph or type the diacritic after the second element. For example, a + ú yields **áɩ**, and so does a + u + U+0301 (combining acute accent). To produce a digraph + diacritic combination not covered by MUFI (e.g. **ɔ̈ɔ**), you may have to enter the digraph directly and not via `ss17` (rare diacritics, however, are usually safe to use with this feature—e.g. **ɑ̊ɑ̊**). Note that `ss17` must be applied to both elements of the digraph *and* any following diacritic.

78. Required Features

Required features, which provide some of the font's most basic functionality—ligatures, support for other features, kerning, and more—include `ccmp` (Glyph Composition/Decomposition), `calt` (Contextual Alternates), `liga` (Standard Ligatures), `loca` (Localized Forms), `rlig` (Required Ligatures), `kern` (Horizontal Kerning), and `mark`/`mkmk` (Mark Positioning). In MS Word these features have to be explicitly enabled on the Advanced tab of the Font dialog (Ctrl-D or Cmd-D: enable Kerning, Standard Ligatures, and Contextual Alternates, and the others will be enabled automatically), but in most other applications they are enabled by default.

79. Appendix: Codes for Special Characters

The following mnemonic codes, which are included in JuniusX/JuniusVF as a convenience, are resolved before all other features, which can therefore act upon them. These codes affect the *appearance* of the text, but not the underlying text: thus if you use the sequence {(wn)} to produce the "wynn" (ƿ), it will not be searchable as its Unicode value (U+01BF), but rather as "{(wn)}." Further, if you change the font, the code will reappear instead of the intended character. Therefore, if you are sending a document to a publisher or publishing it on the web, you should first replace the codes with the corresponding Unicode characters.

However, the Medieval Unicode Font Initiative (MUFI) codes for combining marks in the Private Use Area (highlighted below) are a special case. Most software will not

recognize these characters as combining marks, and so will not position them correctly over the preceding characters—unless you enter them via these codes.

There are sometimes other methods than those outlined here for producing combining marks. Any alphabetic mark with a small cap equivalent can be produced by applying `smcp` to the mark, e.g ◌ᶜ from U+1DDA ◌ᵍ. Variants can also be produced via Character Variant features: for example, just as dotless i can be produced via `cv13` with index 2, so combining dotless i can be produced by applying the same feature to U+0365: ◌ᵢ; and as `cv26` changes t to insular ꞇ, so the same feature changes U+036D ◌ᵗ to ◌ᵗ. Such methods, where available, have a good chance of producing acceptable results where JuniusX or another MUFI font is not available; but as features must be applied to both the combining mark and the base character, it may not always be practical to use them (you can produce **gᵏ** with "g{^ksc}" but applying `smcp` to the sequence g + U+1DDC (**gᵏ**) will produce **Gᵏ**).

{(aa)} → ꜳ (U+A733)
{(AA)} → Ꜳ (U+A732)
{(ae)} → æ (U+00E6)
{(AE)} → Æ (U+00C6)
{(ao)} → ꜳ (U+A735)
{(AO)} → Ꜳ (U+A734)
{(au)} → ꜳ (U+A737)
{(AU)} → Ꜳ (U+A736)
{(av)} → ꜳ (U+A739)
{(AV)} → Ꜳ (U+A738)
{(ay)} → ꜳ (U+A73D)
{(AY)} → Ꜳ (U+A73C)
{(dh)} → ð (U+00F0)
{(DH)} → Ð (U+00D0)
{(yo)} → ꜳ (U+021D)
{(YO)} → Ꜳ (U+021C)
{(kl)} → ꜳ (U+A741)
{(OO)} → Ꜳ (U+A74E)
{(oo)} → ꜳ (U+A74F)
{(ob)} → ꜳ (U+A74B)
{(OB)} → Ꜳ (U+A74A)
{(pr)} → ꜳ (U+A751)
{(po)} → ꜳ (U+A753)
{(q1)} → ꜳ (U+A757)
{(q2)} → ꜳ (U+A759)
{(rr)} → ꜳ (U+A75B)
{(ru)} → ꜳ (U+A75D)
{(sd)} → ꜳ (U+1E9C)
{(wy)} → ꜳ (U+A761)
{(WY)} → Ꜳ (U+A760)
{(wn)} → ꜳ (U+01BF)
{(WN)} → Ꜳ (U+01F7)
{(th)} → þ (U+00FE)
{(TH)} → Þ (U+00DE)
{(ct)} → ꜳ (U+A765)
{(co)} → ꜳ (U+A76F)
{(et)} → ꜳ (U+A76B)
{(ti)} → ꜳ (U+204A)
{(is)} → ꜳ (U+A76D)
{(US)} → ꜳ (U+A770)
{^a} → ◌ᵃ (U+0363)
{^oa} → ◌ᵘ (U+1DD3)
{^æ} → ◌ᵆ (U+1DD4)
{^an} → ◌ᵃⁿ (U+F036)
{^ansc} → ◌ᵃⁿ (U+F03A)
{^ao} → ◌ᵃᵒ (U+1DD5)
{^ao} → ◌ᵃᵒ (U+1DD5)
{^ar} → ◌ᵃʳ (U+F038)
{^arsc} → ◌ᵃʳ (U+F130)
{^av} → ◌ᵃᵛ (U+1DD6)
{^b} → ◌ᵇ (U+F012)
{^bsc} → ◌ᵇ (U+F013)
{^c} → ◌ᶜ (U+0368)
{^ç} → ◌ᶜ (U+1DD7)
{^d} → ◌ᵈ (U+0369)
{^ð} → ◌ᵈ (U+1DD8)
{^dsc} → ◌ᵈ (U+F016)
{^ð} → ◌ᵈ (U+1DD9)
{^e} → ◌ᵉ (U+0364)
{^ę} → ◌ᵉ (U+F135)
{^ē} → ◌ᵉ (U+F136)
{^f} → ◌ᶠ (U+F017)
{^g} → ◌ᵍ (U+1DDA)
{^gsc} → ◌ᶜ (U+1DDB)
{^h} → ◌ʰ (U+036A)
{^i} → ◌ⁱ (U+0365)

{^ɪ} → ◌ᵢ (U+F02F)
{^j} → ◌ⱼ (U+F030)
{^ɟ} → ◌ᶨ (U+F031)
{^k} → ◌ᵏ (U+1DDC)
{^ksc} → ◌ᴷ (U+F01C)
{^l} → ◌ˡ (U+1DDD)
{^lsc} → ◌ᴸ (U+1DDE)
{^m} → ◌ᵐ (U+036B)
{^msc} → ◌ᴹ (U+1DDF)
{^munc} → ◌ᵐ (U+F01F)
{^n} → ◌ⁿ (U+1DE0)
{^nsc} → ◌ᴺ (U+1DE1)
{^o} → ◌ᵒ (U+0366)
{^ǫ} → ◌ᵓ (U+F13E)
{^ø} → ◌ᵊ (U+F032)
{^ō} → ◌ᵒ̄ (U+F13F)
{^oɔ} → ◌ᵒᵓ (U+F03E)
{^or} → ◌ᵒᵓ (U+F03E)
{^orr} → ◌ᵒᵓ (U+F03F)
{^oʐ} → ◌ᵒᵓ (U+F03F)
{^p} → ◌ᵖ (U+F025)
{^q} → ◌ᵠ (U+F033)
{^r} → ◌ʳ (U+036C)
{^rsc} → ◌ᴿ (U+1DE2)
{^ɔ} → ◌ᵓ (U+1DE3)
{^ʐ} → ◌ᵓ (U+F040)
{^2} → ◌ᵔ (U+1DD1)
{^ur} → ◌ᵔ (U+1DD1)
{^s} → ◌ˢ (U+1DE4)
{^ſ} → ◌ᶠ (U+1DE5)
{^t} → ◌ᵗ (U+036D)
{^tsc} → ◌ᵀ (U+F02A)
{^ʈ} → ◌ᵗ (U+F03B)
{^u} → ◌ᵘ (U+0367)
{^v} → ◌ᵛ (U+036E)
{^w} → ◌ʷ (U+F03C)
{^x} → ◌ˣ (U+036F)
{^y} → ◌ʸ (U+F02B)
{^z} → ◌ᶻ (U+1DE6)
{^þ} → ◌ᵇ (U+F03D)
{^ol} → ◌ᵔ̄ (U+0305)
{^us} → ◌ᵓ (U+1DD2)
{^zz} → ◌ᶻ (U+035B)
{^ZZ)} → ◌{^ZZ} (U+1DCF)

Index

a, combining open, variant of. 67.
ᴀ to a. 59.
aA, variants of. 31.
ąĄ, variants of. 32.
ꜳ, variant of. 60.
ꜳꜴ, variants of. 62.
ꜵ, variant of. 63.
æÆ, variants of. 61.
C, variant of. 33.
c2sc – Small Capitals from Capitals. 1.
case – Case-Sensitive Forms. 4.
cv01 – Variants of aA. 31.
cv02 – Variants of ąĄ. 32.
cv04 – Variant of C. 33.
cv05 – Variants of dD. 34.
cv06 – Variant of đ (U+0111, d with stroke). 35.
cv07 – Variants of eE. 36.
cv08 – Variants of ęĘ. 37.
cv09 – Variants of f F. 38.
cv10 – Variants of Gg. 39.
cv11 – Variants of ʒȝ (Yogh). 40.
cv12 – Variants of hH. 41.
cv13 – Variants of iI. 42.
cv14 – Variants of j. 43.
cv15 – Variants of k. 44.
cv16 – Variant of l. 45.
cv17 – Variant of ł (U+A749, l with high stroke). 46.
cv18 – Variants of mM. 47.
cv19 – Variants of nN. 48.
cv21 – Variants of ø. 49.
cv22 – Variant of P. 50.
cv23 – Variant of Q. 51.
cv24 – Variants of rR. 52.
cv25 – Variants of sS. 53.
cv26 – Variants of tT. 54.

cv28 – Variants of vV. 55.
cv30 – Variants of x. 56.
cv31 – Variants of y. 57.
cv32 – Variant of Z. 58.
cv33 – ᴀ to a. 59.
cv34 – Variant of ɑ. 60.
cv35 – Variants of æÆ. 61.
cv36 – Variants of ɔƆ. 62.
cv37 – Variant of ɤ. 63.
cv40 – Variants of ⟆ (U+204A, Tironian nota). 64.
cv41 – Variant of ꝶ (U+A75D, rum abbreviation). 65.
cv42 – Variants of ◌ͤ (U+035B, combining zigzag above). 66.
cv43 – Variant of ◌ᷓ (U+1DD3, combining open a). 67.
cv44 – Variant of ◌ᷛ (U+1DE3, combining r rotunda). 68.
cv45 – Variant of ◌◌ (U+0305, two-character overline). 69.
cv46 – Variant of ◌̃ (U+0303, combining tilde). 70.
cv50 – Variant of ʔ (U+0294, glottal stop). 71.
cv51 – Variant of ? (question mark). 72.
cv53 – Variants of short horizontal stroke. 73. (U+0335)
dD, variants of. 34.
đ, variant of. 35.
digraphs, 77.
eE, variants of. 36.
ęĘ, variants of. 37.
dlig – Discretionary Ligatures. 76.
f F, variants of. 38.
figures – see numbers
fleurons. 13.
futhark, Elder. 25.
futhark, Younger. 26.
futhorc, Early English. 24.
hist – Historical Forms. 16.
Gg, variants of. 39.
glottal stop, variant of. 71.
ȝȜ, variants of. 40.
hH, variants of. 41.
hlig – Historic Ligatures. 75.
iI, variants of. 42.
j, variants of. 43.
k, variants of. 44.
l, variant of. 45.
letters, circled. 5.
letters, superscript. 11.
ligatures, discretionary. 76.
ligatures, historic. 75.
ligatures, required. 78.
ligatures, standard. 78.
lnum – Lining Figures. 9.
long branch runes. 27.
loca – Localized forms. 78.
mM, variants of. 47.
Mirrored runes. 74.
nN, variants of. 48.
nalt – Alternate Annotation Forms. 5.
number, circled. 5.
numbers, lining. 9.
numbers, oldstyle. 7.
numbers, proportional. 8.
numbers, subscript. 12.
numbers, superscript. 11.
numbers, tabular. 6.
øØ, variants of. 49.
onum – Oldstyle Figures. 7.
ornm – Ornaments. 13.

overline, two-character, variant of. 69.
P, variants of. 50.
pcap – Petite Capitals. 3.
pnum – Proportional Figures. 8.
Q, variant of. 51.
question mark, variant of. 72.
rR, variants of. 52.
r rotunda. 23.
r rotunda, contextual. 28.
r rotunda, combining. 68.
Required Features. 78.
rtlm – Right to Left Mirrored Forms. 74.
rum abbreviation. 65.
runes, Early English. 24.
runes, Elder Futhark. 25.
runes, Long Branch. 27.
runes, mirrored. 74.
runes, Short Twig. 27.
runes, Younger Futhark. 26.
sS, variants of. 53.
s, long. 16, 17.
s, contextual long. 22.
short twig runes. 27.
smcp – Small Capitals. 2.
ss01 – Alternate thorn and eth. 14.
ss02 – Insular Letter-Forms. 15.
ss03 – Long s. 17.
ss04 – High Overline. 18.
ss05 – Medium-High Overline. 19.
ss06 – Enlarged Minuscules. 20.
ss07 – Underdotted Text. 21.
ss08 – Contextual Long s. 22.
ss11 – r Rotunda. 23.
ss12 – Early English Futhorc. 24.
ss13 – Elder Futhark. 25.
ss14 – Younger Futhark. 26.
ss15 – Long Branch to Short Twig. 27.
ss16 – Contextual r Rotunda. 28.
ss18 – Old-Style Punctuation Spacing. 29.
ss19 – Latin to Gothic Transliteration. 30.
stroke, horizontal, variants of. 73.
subs – Subscripts. 12.
sups – Superscripts. 11.
tT, variants of. 54.
tilde, combining, variants of. 70.
Tironian nota. 64.
tnum – Tabular Figures. 9.
vV, variants of. 55.
Yogh, variants of. 40.
Z, variant of. 58.
zero – Slashed Zero. 10.
zigzag, combining, variants of. 66.