

“Kim” is a common Korean name, written 김. Unicode says that syllable can be represented by the single “hangul syllable” code point U+AE40, or the sequence of three “conjoining jamo” code points U+1100 U+1175 U+11B7.

In this font (Jieubsida Dodum), there is a precomposed glyph for U+AE40. The conjoining jamo code points have glyphs of their own too, which look respectively like ㄱ ㅡ ㅈ. Those are respectively a zero-width glyph appearing to the right of the reference point, a full-width glyph, and a zero-width glyph appearing to the left of the reference point. Setting them in sequence, they overlay each other and look like 김.

Contextual substitutions in the ljmo and vjmo features replace these glyphs by alternate forms that look like ㄱ ㅡ ㅈ (separately) or 김 (together). Note the change in layout. For some syllables, it is more extreme.

Ligature substitutions in the liga feature then replace that overlaid glyph combination with the precomposed syllable, so if you really use the font as intended, you get the precomposed glyph regardless of whether you used the precomposed or conjoining code points. Setting the overlaid and precomposed syllables next to each other: 김김. Note the difference. In this font, such ligatures exist for 11172 common syllables, but several hundred thousand other syllables can be created with the conjoining jamo.

What happens if we want to change the colour of just part of the syllable?

Suppose we write U+1100 red(U+1175) U+11B7. In naive unmodified XeTeX without any changes to the font or special tricks, we get 김. That actually came out pretty well, but note it is using the less appealing default layout, as if the ljmo and vjmo features had not executed. The colour command prevents the ligature and all the context substitution rules from matching, so we just get the default glyphs with one of them coloured.

Suppose we write U+1100 U+1175 red(U+11B7). Without any special processing, we get 김. That is pretty bad; the red jamo interferes with the black ones. With the colour command hiding the sequence U+1100 U+1175 U+11B7 from the context rules, it has created the shorter sequence U+1100 U+1175, which happens to correspond to a precomposed syllable, and then the U+11B7 negative-bearing glyph gets overlaid on top of that.

It appears that to get correct results, we need to disable some of the features and then (if we want their effects) simulate them by hand, with a lot of font-specific knowledge of how the substitutions work. That seems suboptimal, but colouring just part of something that MAY (depending on the font’s internals) be a single glyph, is perhaps out of the scope of what should be supported anyway.