

TO: Unicode Technical Committee
FROM: Deborah Anderson, SEI, UC Berkeley
SUBJECT: Request for a decision on encoding model for Mende numbers
DATE: 20 July 2012

The Mende script was approved for encoding at the February 2012 UTC, but because no decision was made on the encoding model for Mende numbers, the Mende numbers were not approved. I request the topic be raised again, and a decision made (if possible).

The last revised proposal for Mende is L2/12-023 (N4167). A listing of four model options is contained in L2/12-049. However, since only two options were deemed true candidates during the discussion, only those two are repeated here.

Note: The “pros” and “cons” below are not fleshed out, so a fuller discussion of the options should take place during the UTC.

The two main options are:

1 Atomic encoding

This approach encodes the numbers atomically, requiring 72 code points. This is the model currently reflected in the Mende proposal, L2/12-023, and is the proposal authors’ preferred method for handling Mende numbers.

Pros:

- By encoding the Mende numbers atomically, the character properties can have the correct values.
- Rendering will be simple, as is true for the main syllabary - no special ligation or OpenType behavior would be required.
- “Pre-composed” complex numbers have already been encoded for Cuneiform, Egyptian hieroglyphs, and the Aegean scripts, and many of these could, in principle, be “composed”. The method described below under **2** is not used elsewhere.

Cons:

- The Mende system is a *de novo* one, which works differently than Cuneiform, etc., so the argument which relies on the encoding models of Cuneiform, Aegean scripts, etc., is not analogous.
- This approach requires a total of 72 code points (filling one row, 1E8D0-1E8FF, plus two columns in the 1E900 row in the SMP). The alternative described under **2** below will require far fewer characters and be easier to type and map to a keyboard.

2 Atomically encode digits and units but represent stacks with ligated glyphs.

This method would encode 16 characters (9 digits and 7 units) as atomic characters, and represent the elements which stack in Mende as sequences of a digit character followed by a unit character.

The presentation of the two-character sequences, <base, base>, would be as a ligated glyph. The font would require 63 ligature definitions, with the same number of display elements, but fewer characters than the atomic option described in 1.

Pros:

- Easier for users to type.
- Easier to define a keyboard than the approach in 1, and with fewer characters.

Cons:

- This encoding model is not used elsewhere.
- Typographic ligatures are essentially optional, so it must be stipulated that legibility must not be compromised if the ligatures are broken.
- Use of ligatures forces complex rendering requirements on Mende, which otherwise does not need it.

EXAMPLE:

9,999 would be typed as <9, 1000s-character, 9, 100s-character, 9>, with the display ligating the 1000s-character to 9, and 100s-character to 9.