L2/24-133

On the future of MANDAIC LETTER KAD

Ben Joeng (Yang) 楊偉堅 *

02024-05-17

1 Current Situation

On 2020-12-20, David Corbett sent the following comment regarding Mandaic:

> Chapter 9 says “There are two ways to represent kad in Mandaic: U+0857 MANDAIC LETTER KAD or the sequence <U+084A MANDAIC LETTER AK, U+0856 MANDAIC LETTER DUSHENNA>.” Do these two ways mean the same thing? Are they rendered identically? If they are identical, which one should people use?

David has noticed that we’ve basically painted ourselves into a corner with the most recent changes to the joining groups of two Mandaic characters that happened in Unicode 13.0.0. For some background and as a reminder to all of us, there are two different ways of visually representing the Mandaic word “kad” (meaning “when”, “as”, or “like”):

1. With the standard Mandaic letter *ak* followed by the Mandaic letter *dushenna*, without any joining between them.

2. With a ligature of *ak* + *dushenna*.

In Everson’s original Mandaic proposal L2/08-270R, scenario 1 is encoded as [U+084A MANDAIC LETTER AK + U+0856 MANDAIC LETTER DUSHENNA]. Because DUSHENNA was originally Non_Joining, this code point sequence did not have joining behavior between the characters. Scenario 2 is encoded as an atomic ligature, [U+0857 MANDAIC LETTER KAD].
Everson originally thought that *dushenna* never joined, so it was classified as Non_Joining. However, Ardwan Al-Sabti demonstrated in L2/20-043 that *dushenna* is right-joining in Neo-Mandaic (the use of the Mandaic script for modern languages). As such, the UTC switched the joining behavior of MANDAIC LETTER DUSHENNA to Right_Joining[1] in Unicode 13.0.0.
This created a conundrum, as AK + DUSHENNA would now have a join between them, and thus would be visually identical to KAD.
There would be one remaining use case for KAD, though, because KAD originally was Non_Joining. Thus, if a user wanted the *ak* + *dushenna* ligature that right-joined, they could use AK + DUSHENNA, but if they wanted the ligature that didn’t join to the right, they could use KAD.
However, due to Al-Sabti’s document L2/20-049, which showed that the “ak + dushenna” ligature is right-joining in Neo-Mandaic, the UTC decided to switch KAD’s joining group to Right_Joining as well.

This means that the sequence AK + DUSHENNA and the atomic character KAD no longer have **any** distinction, visually or behaviorally.

*yang@unicode.org
[1] n.b., MANDAIC LETTER AK (U+084A) has always been Dual_Joining

2 Solutions

I see the following solution as ideal:

> Recommend that KAD (U+0857) be declared DO NOT EMIT, and recommend the “kad” sequence always be encoded as AK (U+084A) + DUSHENNA (U+0856), with fine-grain joining control being done with ZWNJs as necessary.

This would mean that all instances of joining-blocking behavior in Mandaic would henceforth be controlled by ZWNJ. This is, in my opinion, what should have been done in the first place.

The 3rd and 4th paragraph of the “Structure” section of the Mandaic chapter of the Core Spec (starting with “Two additional Mandaic characters...”) should be replaced to the following:

> Two additional Mandaic characters are encoded in the Unicode Standard: U+0858 MANDAIC LETTER AIN is a borrowing from U+0639 ARABIC LETTER AIN. The second additional character, U+0857 MANDAIC LETTER KAD, is a digraph originally encoded to write the word *kd*, which means “when, as, like”, but is no longer recommended for use. New documents should encode *kd* as the sequence <U+084A MANDAIC LETTER AK, U+0856 MANDAIC LETTER DUSHENNA>. Implementations should treat U+0857 MANDAIC LETTER KAD in existing text as equivalent to the recommended representation sequence <U+084A MANDAIC LETTER AK, U+0856 MANDAIC LETTER DUSHENNA>.
>
> The Joining_Type values for U+0856 MANDAIC LETTER DUSHENNA, U+0857 MANDAIC LETTER KAD, and U+0858 MANDAIC LETTER AIN were changed in Unicode Version 13.0 from Non_Joining to Right_Joining. In cases where the isolated form of *dushenna*, *ain*, or *kad* following a right join-causing character is desired, a U+200C ZERO WIDTH NON-JOINER should be employed to prevent joining with the previous character. (See *Table 9-4* for the definition of a right join-causing character.)

The following note should be added to the Names List for U+0857 KAD:

> * use of this character is discouraged
> : 084A 0856

The following line should be added to DoNotEmit.txt:

```
0857; 084A 0856; Discouraged # MANDAIC LETTER KAD; MANDAIC LETTER AK, MANDAIC
LETTER DUSHENNA
```

The following line should be added to confusables.txt:

```
0857 ; 084A 0856 ; MA # ( □ → □□ ) MANDAIC LETTER KAD → MANDAIC LETTER AK,
MANDAIC LETTER DUSHENNA #
```

3 Alternative Solutions for Unicode

- Switch the joining groups of KAD and DUSHENNA back to Non_Joining, and encode a new NEO-MANDAIC DUSHENNA that looks identical to DUSHENNA but is Right_Joining. Then, to get the right-joining “ak + dushenna” ligature, it would be encoded as AK + N-M DUSHENNA. For the non-joining ligature, you’d use the original KAD code point.

  This is pretty awful because you now have two code points that render identically but have different joining behavior. You’d also still have the same problem of AK + N-M DUSHENNA rendering identically to KAD except in contexts where it’d right join.

- Switch just KAD back to Non_Joining, and recommend that to get a right-joining “ak + dushenna” ligature, use AK + DUSHENNA. To get the “ak” + “dushenna” sequence without a join, use AK + ZWNJ + DUSHENNA.

  This is pretty bad because you still have AK + DUSHENNA and KAD looking identical (except in right-joining contexts), and KAD only exists as an alternative to ZWNJ + AK + DUSHENNA.

4 Alternative Solutions for users

- Always use U+0857 KAD to represent the “ak” + “dushenna” sequence, and recommend the sequence <U+084A AK + U+0856 DUSHENNA> not be used. If an unjoined sequence is desired, <U+084A AK + ZWNJ + U+0856 DUSHENNA> would need to be used.

  Pro: this solution would keep continuity with existing data, as U+0857 KAD has been recommended for representing this ligature in the past, there exists some existing documents using it (although not much, there’s not a lot of Unicode-encoded Mandaic).

  Cons: it would mean this specific sequence of two graphemes would be special-cased, it would be difficult for implementers to forbid a sequence from being emitted (as opposed to not emitting a single codepoint), and if a user wants an unconnected “ak” + “dushenna”, they’d still need to use the sequence (albeit with a ZWNJ in between).

- Leave it up to users to use either U+0857 KAD or <U+084A AK + U+0856 DUSHENNA>. If an unjoined sequence is desired, <U+084A AK + ZWNJ + U+0856 DUSHENNA> would need to be used.

  Pro: this solution would allow continuity with existing data and would also allow the sequence to be used.

  Cons: having two different ways to encode the same visual output results in data fragmentation, making operations such as searching become significantly more difficult for users. If it would be possible to normalize between the sequence and the ligature, this would ameliorate this problem somewhat, but adding a decomposition to U+0857 KAD isn’t allowed by the Unicode stability policy.

5 Extra Background Information

Dushenna itself was originally a ligature of *ad* (𐡄) + *aksa* (𐡀) (the “i” vowel). However, it’s always very visually distinct from the *ad* + *aksa* (𐡀𐡄) sequence and thus its atomic encoding is entirely reasonable. It’s also traditionally considered a separate letter in the Mandaic script, for what it’s worth.

The dushenna in the “ak + dushenna” ligature takes a very slightly different form from its appearance in other contexts (the right “tooth” bends to the left as opposed to the right). This might imply that there could be a visual distinction between AK + DUSHENNA (normal AK joined to normal DUSHENNA) and KAD (proper ligature). However, none of the sources in Everson, Al-Sabti, nor any of my personal Mandaic sources show evidence of AK + DUSHENNA ever occurring without ligating.

Al-Sabti also proposed (and the UTC accepted) that U+0858 MANDAIC LETTER AIN also be switched to Right_Joining. This has no bearing on the above conundrum, but it should be noted that if the use of ZWNJ is completely standardized as the only way to break joining behavior for Mandaic, getting the non-joining *ain* would fall completely into this regular pattern.

6 Additional document links

- TUS section on Mandaic (p. 401)
- Relevant SAH report (p. 7)
- UTC minutes where joining groups were changed (§ C.11.1 and § C.11.3)

All other relevant documents are linked inline.