

# Proposal to encode ADLaM NASALIZATION MARK for ADLaM script

Abdoulaye Barry, Ibrahima Barry, Peter Constable, Andrew Glass

September 17, 2018



## Summary

ADLaM script was added to Unicode in version 9.0 (2016) based on Michael Everson’s proposal documents submitted in 2013 and 2014 (L2/13-191 & L2/14-219R). The present document addresses a gap in the ADLaM encoding by proposing a new mark character as an addition to the existing ADLaM block:

Glyph	Code	Character name
'	1E94B	ADLaM NASALIZATION MARK

## Usage requirements

ADLaM script uses an apostrophe-like sign called Nyondal to nasalize a succeeding consonant. For example:

	/hin.du/	‘ancient’
	/hi. <sup>ⁿ</sup> du/	‘it’

This sign was known at the time the original proposals for ADLaM were submitted, but at that time, the use of European punctuation signs at their standard code points was expected to suffice (Everson 2013: 2, Everson 2014: 2). While implementing and using the ADLaM Unicode encoding, it has become clear that using ' (U+0027) or similar signs to represent ADLaM Nyondal does not work. Specific typographical features of Nyondal make it necessary to encode a distinct sign. These requirements are:

1. ADLaM Nyondal is a letter modifier not a punctuation sign and occurs in word internal position
2. As a letter, ADLaM Nyondal should have strong right-to-left directionality
3. Typographically, ADLaM Nyondal is straight, not hooked
4. ADLaM Nyondal is transparent to joining behavior, i.e., adjacent letters connect under it, but it doesn’t cause joining by itself.

## Comparable signs

Alternative characters already encoded in Unicode have been considered and rejected as follows:

- U+0027 APOSTROPHE

This character is functionally a punctuation character. It is subject to being replaced by a smart quote (U+2018 and U+2019) in many applications resulting in inconsistent spelling across applications. Not all

applications have the option to disable smart quotes and many users are not aware that the feature is enabled. Many applications give the apostrophe special logic in text-selection, etc. When used with joining scripts, it blocks joining behavior between adjacent signs.

- U+02BC MODIFIER LETTER APOSTROPHE

The default form is curved to work with LTR scripts. This character is used in several Latin-based orthography for African languages. As a result, modifying the design for ADLaM specifically will cause inconsistent appearance in mixed, script/language fonts. It blocks joining between adjacent signs. It has a strong-left directionality.

- U+A78B LATIN CAPITAL LETTER SALTILLO
- U+A78C LATIN SMALL LETTER SALTILLO

The default form of these characters are straight; but they have a strong-left directionality and will create the same inconsistent text ordering issues as U+02BC.

- U+05F3 HEBREW PUNCTUATION GERESH

The default form is not vertical, and it blocks joining behavior. Though it is strong RTL, it's association with Hebrew script will cause complications with text itemization and font fallback behaviors.

Therefore, the right approach to encoding this sign is to encode as a dedicated character, ADLAM NASALIZATION MARK. The suggestion here is to encode it at the first unassigned code point in the ADLAM block.

## Joining behavior

ADLAM Nyondal is spacing and transparent to joining. That is, it has the following characteristics:

Position	With Nyondal	Without Nyondal
Initial, before ADLAM letter	ɣ'	ɣ
Medial position, between ADLAM letters	ɣ'a	ɣa
Final position, after ADLAM letter	'a	a
Isolated position, not adjacent to ADLAM letters	'	

## Properties

It should have the following properties:

**Character properties** In the format of UnicodeData.txt:

```
1E94B;ADLAM NASALIZATION MARK;Lm;0;R;;;;;N;;;;;
```

**Linebreaking** In the format of LineBreak.txt:

```
1E94B;AL # Lm [1] ADLAM NASALIZATION MARK
```

**Joining categories** In the format of ArabicShaping.txt:

```
1E94B; ADLAM NASALIZATION MARK; T; No_Joining_Group
```

## References

Everson, Michael, 2013 “Preliminary proposal for encoding the Adlam script in the SMP of the UCS”. N4488 L2/13-191. <http://www.unicode.org/L2/L2013/13191-n4488-adlam.pdf>

Everson, Michael, 2014. “Revised proposal for encoding the Adlam script in the SMP of the UCS”. N4628R L2/14-219R. <http://www.unicode.org/L2/L2014/14219r-n4628-adlam.pdf>

---

1E900

Adlam

1E95F

	1E90	1E91	1E92	1E93	1E94	1E95
0	𐤀 1E900	𐤁 1E910	𐤂 1E920	𐤃 1E930	𐤄 1E940	𐤅 1E950
1	𐤆 1E901	𐤇 1E911	𐤈 1E921	𐤉 1E931	𐤊 1E941	𐤋 1E951
2	𐤌 1E902	𐤍 1E912	𐤎 1E922	𐤏 1E932	𐤐 1E942	𐤑 1E952
3	𐤒 1E903	𐤓 1E913	𐤔 1E923	𐤕 1E933	𐤖 1E943	𐤗 1E953
4	𐤘 1E904	𐤙 1E914	𐤚 1E924	𐤛 1E934	𐤜 1E944	𐤝 1E954
5	𐤞 1E905	𐤟 1E915	𐤠 1E925	𐤡 1E935	𐤢 1E945	𐤣 1E955
6	𐤤 1E906	𐤥 1E916	𐤦 1E926	𐤧 1E936	𐤨 1E946	𐤩 1E956
7	𐤪 1E907	𐤫 1E917	𐤬 1E927	𐤭 1E937	𐤮 1E947	𐤯 1E957
8	𐤰 1E908	𐤱 1E918	𐤲 1E928	𐤳 1E938	𐤴 1E948	𐤵 1E958
9	𐤶 1E909	𐤷 1E919	𐤸 1E929	𐤹 1E939	𐤺 1E949	𐤻 1E959
A	𐤼 1E90A	𐤽 1E91A	𐤾 1E92A	𐤿 1E93A	𐥀 1E94A	
B	𐥁 1E90B	𐥂 1E91B	𐥃 1E92B	𐥄 1E93B	𐥅 1E94B	
C	𐥆 1E90C	𐥇 1E91C	𐥈 1E92C	𐥉 1E93C		
D	𐥊 1E90D	𐥋 1E91D	𐥌 1E92D	𐥍 1E93D		
E	𐥎 1E90E	𐥏 1E91E	𐥐 1E92E	𐥑 1E93E		𐥒 1E95E
F	𐥓 1E90F	𐥔 1E91F	𐥕 1E92F	𐥖 1E93F		𐥗 1E95F





