WG2 N4889

*** DRAFT ***

A graphetic approach for the Mongolian encoding model

Script Ad Hoc Group Recommendations

31 August 2017

This is the first “separate document” mentioned in the 9 August 2017 document *Script Ad Hoc Group Recommendations on Mongolian Text Model* (which should be read first) as:

> Detailed examples of this [graphetic] model (versus the current approach) will be provided in a separate document.

This document is focused on *Hudum* (ᠬᠤᠳᠤᠮ or худам), ie, the modern Mongolian writing system, as opposed to other writing systems (Hudum Ali Gali, historical Hudum, Todo, Manchu, Manchu Ali Gali, historical Manchu, Sibe, etc) that use the Mongolian script.

1 Introduction

1.1 Characters

The Hudum character set of the graphetic approach is shown in the chart on page 3 and 4. See section 1.2 on page 5 for the graphetic analysis that supports this character set.

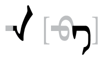
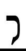
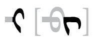
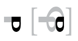
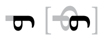
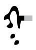
In the chart, every row shows a graphetic character with their name in the "graphetic" column:

- Characters on page 3 are result of the graphetic analysis, denoted with provisional names in lowercase. In order to avoid using existing complex characters, 14 new characters are required, while 3 graphetic characters (`ja` , `ge` , and `ha`) can reuse existing related simple characters (denoted with a code point and character name in parentheses).
- Characters on page 4 are reused simple characters, denoted with their existing code points and character names in uppercase.

Positional variants `isol` , `init` , `medi` , and `fina` These 4 columns in between show each character's positional variants on the 4 cursive joining positions, comforming with the Unicode–OpenType *Arabic cursive joining model.* A character represents only a single grapheme, and its positional variants only differ in cursive joining. Noteworthy context-dependent subgrapheme variants are shown in brackets with their contexts. Since Mongolian fonts often don't clearly show the difference between certain positional variants, the status of cursive joining is emphasized with gray stroke-endings in every glyph.

Note these positional variants are irrelevant to the defination of "word", while the current encoding is a mixture of the Arabic cursive joining positional variants and a glyph's position in an undefined entity of "word" (see section 1.2.1 on page 5).

Current This column shows which current characters are conditionally converted to a graphetic character. Some current characters have been decomposed (see section 1.2.3 on page 8).

CURRENT	isol	init	medi	fina	GRAPHETIC
A, E, NA, aleph, HAA cap, ANG component 1					a
NA	-				na
A, E		×	×	×	a non-joining
I, JA, YA					i
YA	-			-	ya
JA	-	-			ja (U+1854 TODO TSA)
O, U, OE, UE	-				u
O, U, OE, UE, WA		-	-		u tailed
OE, UE	-	-			ue
GA	-				ga
QA, GA	-				qa
QA, GA, ANG component 2	-				ge (U+1889 ALI GALI KA)
TA, DA	-				da
TA, DA	-				ta
DA	-	-			da coda
EE, WA	-				wa
HAA, ZHI, LHA component 2	-				ha (U+1841 ZHI)

CURRENT	**isol**	**init**	**medi**	**fina**	**GRAPHETIC**
NIRUGU	᠊	᠊‍	‍᠊‍	‍᠊	U+180A NIRUGU
BA	-	ᠪ‍	‍ᠪ‍	‍ᠪ	U+182A BA
PA	-	ᠫ‍	‍ᠫ‍	‍ᠫ	U+182B PA
MA	-	ᠮ‍	‍ᠮ‍	‍ᠮ	U+182E MA
LA, LHA component 1	-	ᠯ‍	‍ᠯ‍	‍ᠯ	U+182F LA
SA	-	ᠰ‍	‍ᠰ‍	‍ᠰ	U+1830 SA
SHA	-	ᠱ‍	‍ᠱ‍	‍ᠱ	U+1831 SHA
CHA	-	ᠴ‍	‍ᠴ‍	‍ᠴ	U+1834 CHA
RA	-	ᠷ‍	‍ᠷ‍	‍ᠷ	U+1837 RA
FA	-	ᠹ‍	‍ᠹ‍	‍ᠹ	U+1839 FA
KA, KHA	-	ᠺ‍ (ᠻ‍)	‍ᠺ‍ (‍ᠻ‍)	‍ᠺ (‍ᠻ)	U+183A KA
TSA	-	ᠼ‍	‍ᠼ‍	‍ᠼ	U+183C TSA
ZA	-	ᠽ‍	‍ᠽ‍	‍ᠽ	U+183D ZA
ZRA	-	ᠿ‍	-	-	U+183F ZRA
CHI	-	ᡂ‍	-	-	U+1842 CHI

1.2 Grapheme reanalysis

1.2.1 Correcting the positional mismatches

Positional mismatches must be corrected first before working on the current encoding. Such a fundamental cleanup is not only required by the graphetic approach, but also important to any attempt at improving the current encoding.

The variant set used in the following analysis is based on Shen Yilei (沈逸磊)'s document that has thoroughly discussed the mismatch issue, *Comments on positional mismatches in Mongolian encoding*[1], in particular:

- Chart A, "Full chart of the present Mongolian variant specifications in TUS without editorial errors", from the bottom of page 8 to page 9.
- The chart in secton 5, "An excerpt of the resultant chart of Mongolian variants", on page 8.

1.2.2 Setting the scope and grouping variants

The variant set (now without editorial errors and positional mismatches) are arranged to the chart on page 6 and 7. All positional variants of the same grapheme are grouped together in a single line.

Notes on alignment:

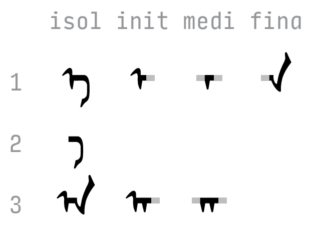
- 3-way contrast of rounded vowel fina and neutralization after B and on init/medi.
- TA, DA - etc, see the original graphetic discussion.

The following graphemes are excluded from the analysis for Hudum:

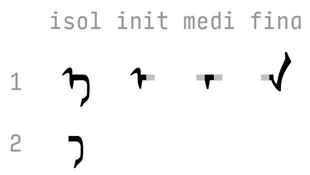
- `A ali gali`
- `E ali gali`, `E historical`
- `NA historical`, `NA todo enclitic`
- `BA stylistic`
- `QA historical masculine`, `QA historical feminine`
- `GA historical`
- `SA historical`, `SA manchu ali gali`
- `ZRA unattested`

[1]https://lists.w3.org/Archives/Public/public-i18n-mongolian/2017JulSep/0001.html

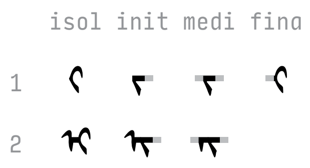
U+1820 A



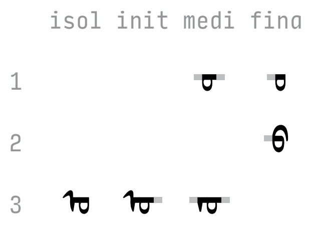
U+1821 E



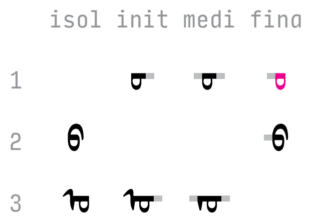
U+1822 I



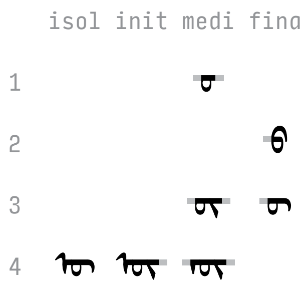
U+1823 O



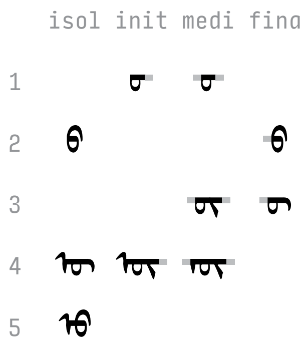
U+1824 U



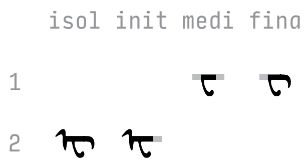
U+1825 OE



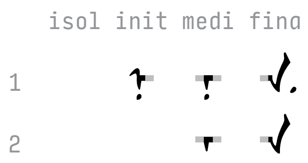
U+1826 UE



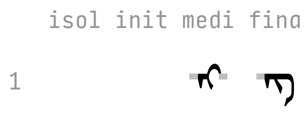
U+1827 EE



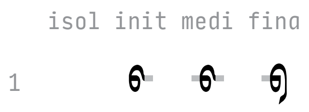
U+1828 NA



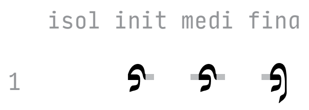
U+1829 ANG



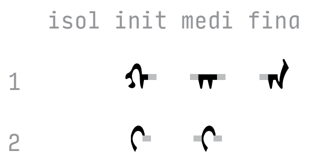
U+182A BA



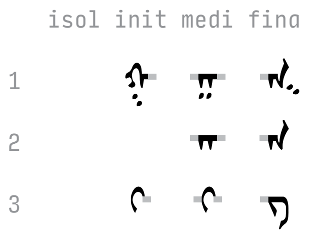
U+182B PA



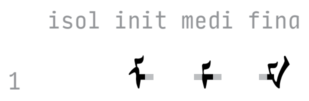
U+182C QA



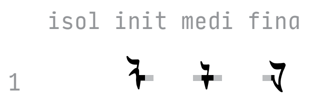
U+182D GA



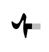
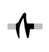
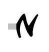
U+182E MA



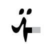
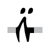
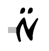
U+182F LA



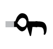
U+1830 SA

	isol	init	medi	fina
1				

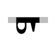
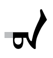
U+1831 SHA

	isol	init	medi	fina
1				

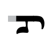
U+1832 TA

	isol	init	medi	fina
1				
2				

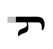
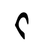
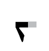
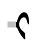
U+1833 DA

	isol	init	medi	fina
1				
2				
3				

U+1834 CHA

	isol	init	medi	fina
1				

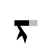
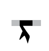
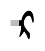
U+1835 JA

	isol	init	medi	fina
1				
2				

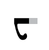
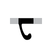
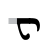
U+1836 YA

	isol	init	medi	fina
1				
2				

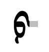
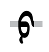
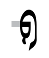
U+1837 RA

	isol	init	medi	fina
1				

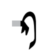
U+1838 WA

	isol	init	medi	fina
1				
2				

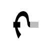
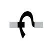
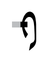
U+1839 FA

	isol	init	medi	fina
1				

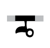
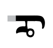
U+183A KA

	isol	init	medi	fina
1				

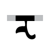
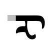
U+183B KHA

	isol	init	medi	fina
1				

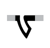
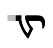
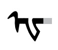
U+183C TSA

	isol	init	medi	fina
1				

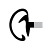
U+183D ZA

	isol	init	medi	fina
1				

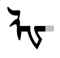
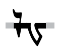
U+183E HAA

	isol	init	medi	fina
1				
2				

U+183F ZRA

	isol	init	medi	fina
1				

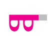
U+1840 LHA

	isol	init	medi	fina
1				

U+1841 ZHI

	isol	init	medi	fina
1				

U+1842 CHI

	isol	init	medi	fina
1				

Note the following variants are added and marked magenta:

- `U.1.fina` : Apparently missed in the standard.
- `ZHI.1.init` , `CHI.1.init` : Positional variants not explicitly listed in the standard in addition to their presentative glyphs.

A longer version of `E.1.init` , namely `E long` , might also be a required variant.

1.2.3 Decomposing

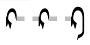
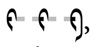
Certain graphemes are decomposed to sequences, with 2 additional graphemes introduced by the decomposition: `aleph` and `HAA cap` :

- Stem-beginning vowel graphemes decomposed into an `aleph` and a normal vowel grapheme:
  - `A.3` ⇒ `aleph` + `A.1`
  - `I.2` ⇒ `aleph` + `I.1`
  - `O.3` ⇒ `aleph` + `O.1`
  - `U.3` ⇒ `aleph` + `U.1`
  - `OE.4` ⇒ `aleph` + `OE.3`
  - `UE.4` ⇒ `aleph` + `UE.3`
  - `UE.5` ⇒ `aleph` + `UE.2`
  - `EE.2` ⇒ `aleph` + `EE.1`
- `ANG.1` ⇒ `NA.2` + `GA.3`
- `HAA.2` ⇒ `HAA cap` + `HAA.1`
- `LHA.1` ⇒ `LA.1` + `HAA.1`

1.2.4 Merging

Graphemes duplicated in multiple characters are merged into a single one:

- `A.1` = `E.1` = `NA.2` = `aleph` = `HAA cap` — Debatable.
- `A.2` = `E.2`

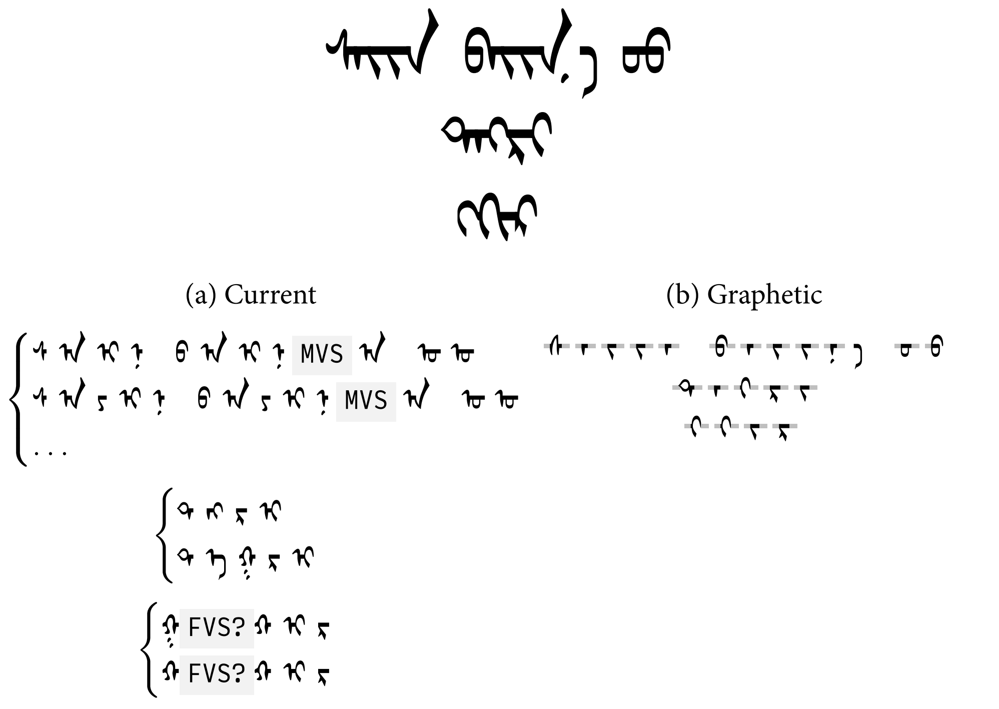
- `I.1` = `JA.2` = `YA.2`
- `O.1` = `U.1` = `OE.1` = `UE.1`
- `O.2` = `U.2` = `OE.2` = `UE.2` = `WA.2`
- `OE.3` = `UE.3`
- `EE.1` = `WA.1` — Debatable.
- `QA.1` = `GA.2`
- `QA.2` = `GA.3` = (`U+1889 ALI GALI KA`)
- `TA.1` = `DA.1`
- `TA.2` = `DA.2`
- `JA.1` = (`U+1854 TODO TSA`)
- `KA.1` = `KHA.1` — Debatable. Not a typical merger. In Hudum this pair of graphemes,  and , are considered a pair of stylistic variants (generally the former is preferred in the Inner Mongolia and the latter is preferred in Mongolia). But they're distinguished in Hudum Ali Gali text as two letters, for the Sanskrit–Tibetan sounds /ɡ/ and /k^h/, respectively.
- `HAA.1` = `ZHI.1` — Debatable.

Now all the resulted graphemes are considered graphetic characters, see the chart of graphetic characters on page 3 and 4.

2 Examples for Hudum

The graphetic approach only requires cursive joining to be handled contextually according to character joining types. (The standard joining-control format characters `ZWJ` and `ZWNJ` are still available for requesting a positional variant not produced by the current context.) All other grapheme-level shaping processes are intuitively handled by inputting appropriate simple characters. No complicated contextual rules involved. See figure 1. (Graphetic characters are represented with their `medi` variants be default.)

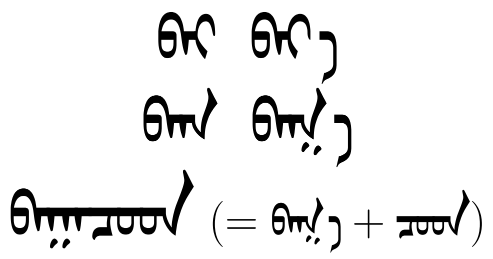
Figure 1: An intuitive approach.



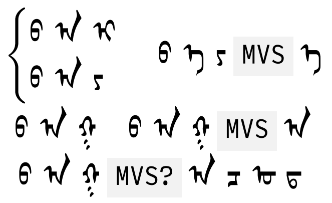
2.1 The disjointed tail ([illegible])

With the dedicated non-joining character `a non-joining` (apparently friendlier than an invisible format character), the disjointed tail doesn't need `MVS` to separate it from the preceding letter anymore. For the letter preceding a disjointed tail, an appropriate character is chosen according to the desired grapheme. When a suffix is appended, a normal `a` character should be used instead. See figure 2.

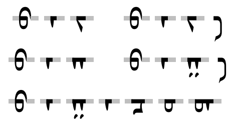
Figure 2: The disjointed tail.



(a) Current



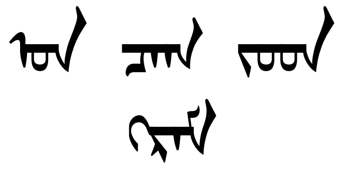
(b) Graphetic



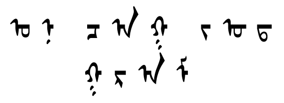
2.2 Coda and stray consonants

Characters are simply chosen according to the desired graphemes. Inputting no longer depends on a particular font's shaping logic. See figure 3.

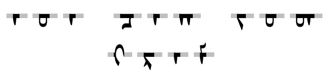
Figure 3: Coda and stray consonants.



(a) Current



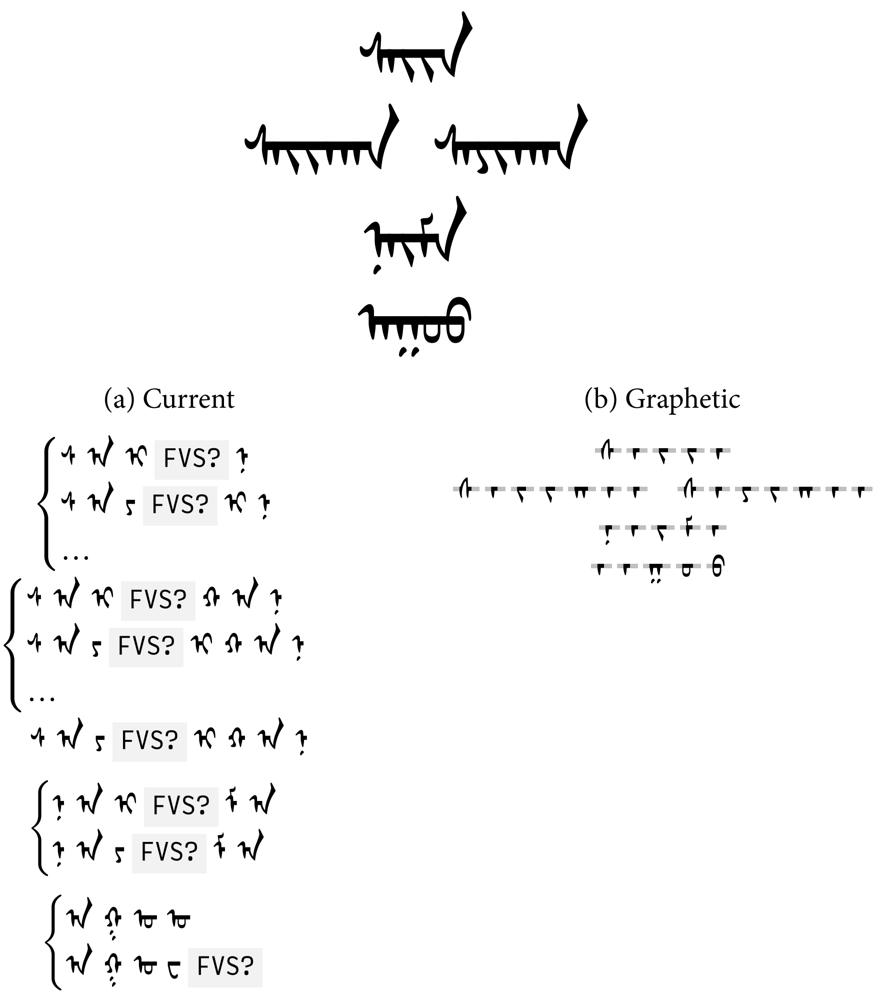
(b) Graphetic



2.3 Postvocalic vowels

Dipthongs don't have to suffer from the debates on the underlying phonetic sequences. Type what you see. See figure 4. See also the first word in figure 2.

Figure 4: Postvocalic /i/ and related cases.



2.4 Stem boundary behaviors

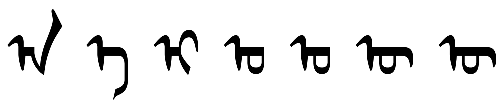
The graphemic features in *stem-beginning* vowels (the prepended aleph, see figure 5) and *first-in-stem* vowels (see figure 6) in the current encoding are generally considered part of `isol` and `init` forms, and contextually triggered on the `medi` position.

This seems to be a straightforward logic, but actually doesn't have the first-in-stem vowel behavior well defined on `fina` and makes second-stem (see figure 7) and enclitic (see figure 9) situations complicated. The graphetic approach has the aleph part decomposed and treat the three `fina` forms of round vowels (phonetic vowel 4, 5, 6, 7) equally.

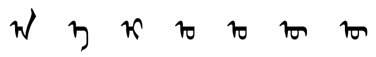
Figure 6 is actually a part of *the first syllabary*, the de facto standard of Hudum's basic writing logic. But it is poorly supported by the current encoding due to the failure to recognize the first-in-stem behavior and abusing `FVS` -less cases to simplify certain commonly used words' encoding.

Since stem boundaries inside compound words are also syllable boundaries, coda consonants are also triggered and rely on complicated `FVS` manipualtion in the current encoding. See figure 7.[2]

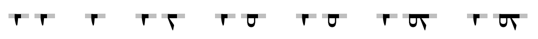
Figure 5: Stem-beginning vowel alephs.



(a) Current



(b) Graphetic



2.5 Vowel harmony affected consonants

The major source of long-distance effect in the current encoding is vowel-harmony-affected consonants. The long-distance effect might seem convenient in simple words but it quickly gets complicated in non-harmonious compound words and loanwords. With the graphetic approach, again, users simply type the desired graphemes. See figure 8.

[2]Some sample words from Siqinbilige:
https://lists.w3.org/Archives/Public/public-i18n-mongolian/2015JulSep/0355.html

Figure 6: First-in-stem vowels.

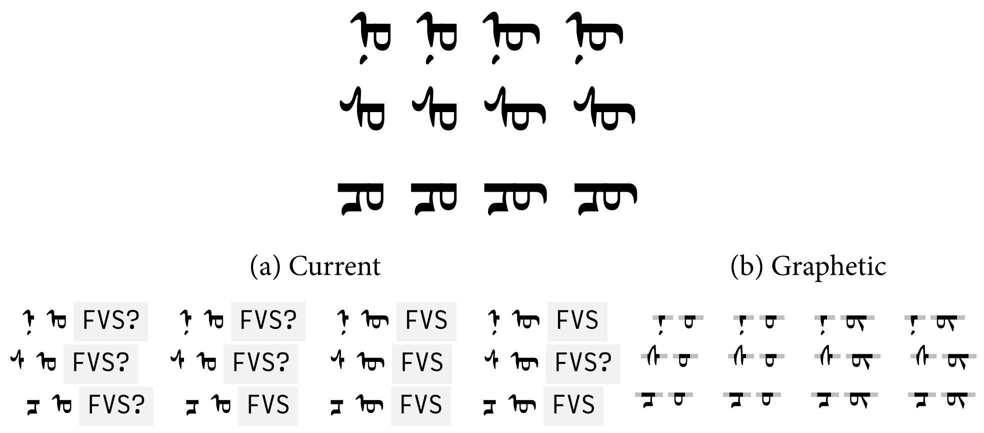


Figure 7: Compound words.

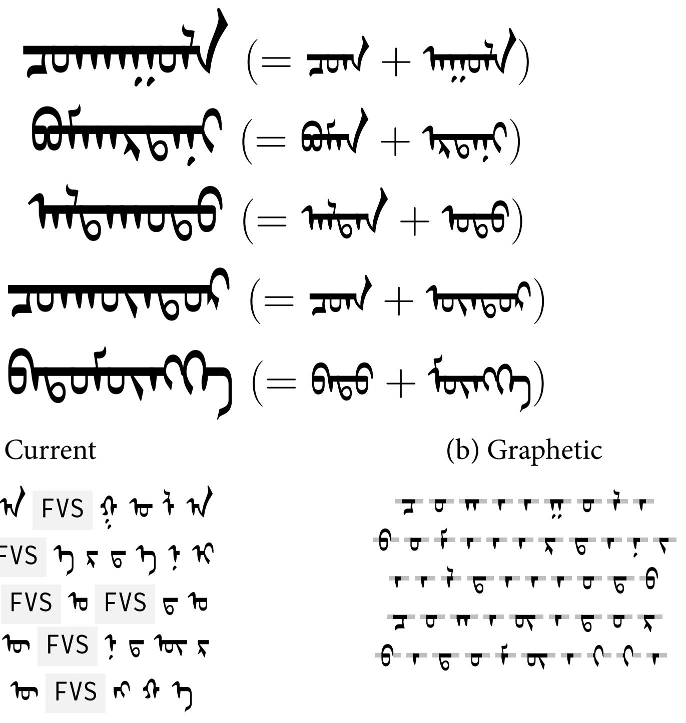
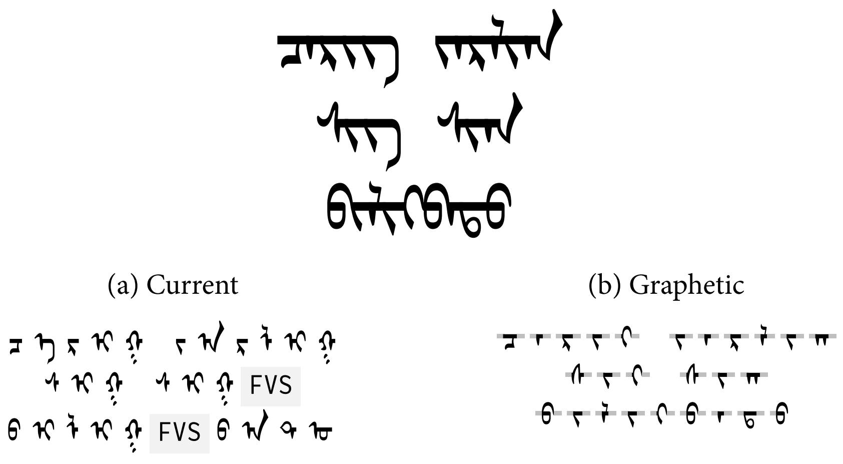


Figure 8: Vowel harmony affected consonants.



2.6 Enclitics

The complicated situation for enclitics in the current encoding is the result of analyzing all enclitics (while the categorization work is incomplete in the first place) as special cases trigered by `NNBSP` instead of productive cases. With the intuitive encoding logic, the graphetic approach doesn't rely on any special mechanism like `NNBSP`, but simply forms correct shapes of enclitics from simple character sequences not so different from those of normal words. See figure 9 on page 16.

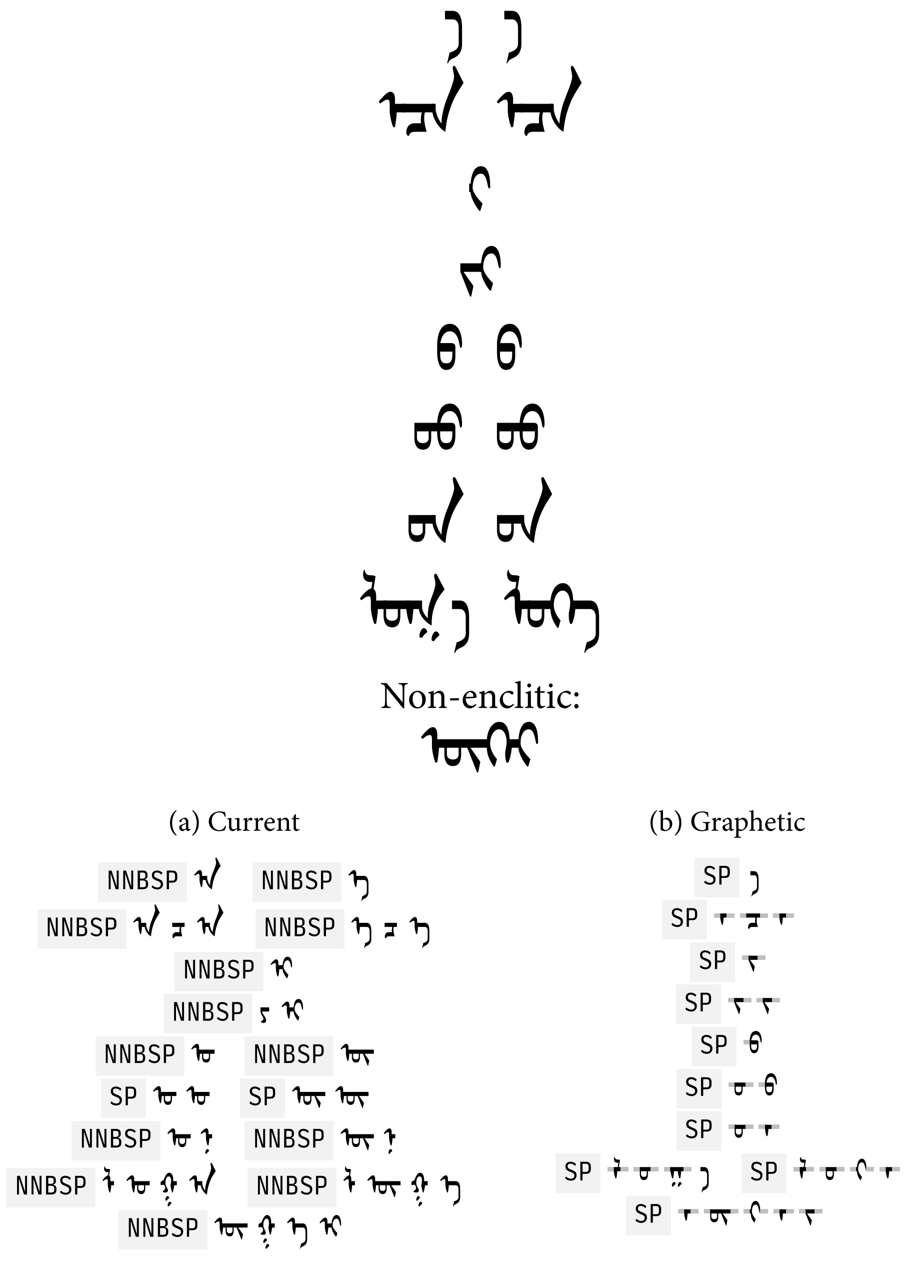
Although the preceding white space in the graphetic approach should be a normal `SP` (`U+0020 SPACE`) by default, any white space character can be used instead to satisfy various typesetting preferences — such as `U+00A0 NO-BREAK SPACE` (to prevent line breaking), `U+2009 THIN SPACE` (to have a narrower gap), and even `NNBSP` (but only as a thin space that disallows line breaking and *doesn't* have any shaping effect).

See figure 9 and note how and in the current encoding breaks the general encoding model for the disjointed tail (see section 2.1).

2.7 Disambiguating and alternative forms

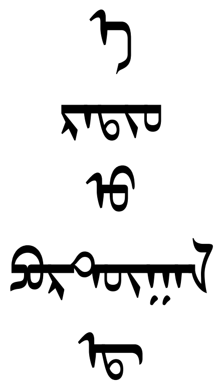
Hudum employs a lot of disambiguating forms in various words, native or loan. Some of them are currently handled by independent characters, such as `EE` (loanword E), `HAA` (loanword QA feminine), `ZHI` and `CHI` (JA and CHA in Mandarin

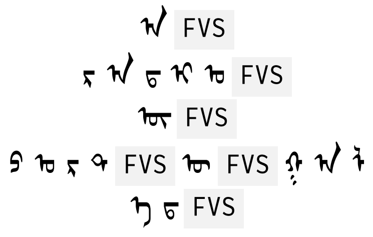
Figure 9: Enclitics.



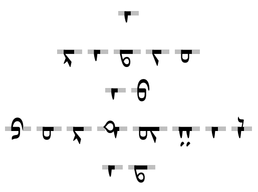
syllable zhi and chi, respectively). However other cases (such as loanword O, Mandarin U, loanword UE, Mandarin UE, loanword TA, native or loanword DA) all have to be requested with `FVS` in complex contexts that are hard to predict and behavior of which varies from font to font. The graphetic approach directly encodes graphemes instead of fiddling with complex contexts and `FVS`es. See figure 10 on 17 for some examples.

Figure 10: Disambiguating and alternative forms.





(a) Current



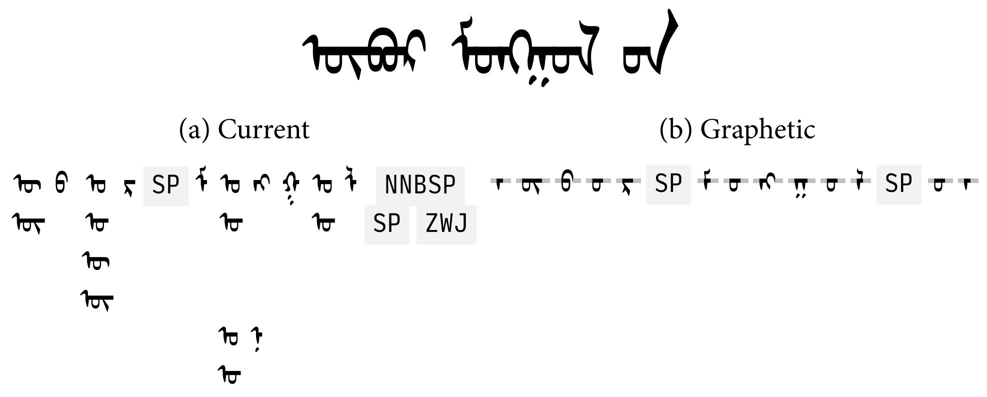
(b) Graphetic

3 Improvements on confusables

3.1 Character-to-character confusables

In the current encoding, exactly which variant is shown is decided by a complicated shaping logic according to the context, but theoretically or ideally every variant can be requested explicitly with a certain format character, therefore the context is not taken into consideration in this analysis.

Figure 11: Character-to-character confusables.



(a) Current

(b) Graphetic

(To be finished.)

3.2 Character-to-sequence confusables

- ? `ue` ↔ `u` + `i`
- ? `qa` ↔ `a` + `a`
- ? `ga` ↔ `na` + `na`
- `da coda` ↔ `u` + `a`
- `CHI` ↔ `u` + `u`

(To be finished.)

4 Examples for Hudum Ali Gali and historical Hudum

(To be finished.)

4.1 Reused Hudum graphemes

(To be finished.)

4.2 Potentially additional graphemes

Such graphemes actually require separate characters or other encoding-level manipulations.

(To be finished.)

4.3 Stylistic variants of graphemes

(To be finished.)