

**ISO/IEC JTC 1/SC 2/WG 2  
PROPOSAL SUMMARY FORM TO ACCOMPANY SUBMISSIONS  
FOR ADDITIONS TO THE REPERTOIRE OF ISO/IEC 10646<sup>1</sup>**

**Please fill all the sections A, B and C below.**

Please read Principles and Procedures Document (P & P) from <http://www.dkuug.dk/JTC1/SC2/WG2/docs/principles.html> for guidelines and details before filling this form.

Please ensure you are using the latest Form from <http://www.dkuug.dk/JTC1/SC2/WG2/docs/summaryform.html>.

See also <http://www.dkuug.dk/JTC1/SC2/WG2/docs/roadmaps.html> for latest *Roadmaps*.

**A. Administrative**

1. **Title:** Proposal to encode additional Tifinagh characters  
2. Requester's name: Lorna A. Priest, Jon Coblentz, Andrew Savage  
3. Requester type (Member body/Liaison/Individual contribution): Individual contribution  
4. Submission date: 13 May 2008  
5. Requester's reference (if applicable): L2/08-198R  
6. Choose one of the following:  
This is a complete proposal: Yes  
or, More information will be provided later: \_\_\_\_\_

**B. Technical – General**

1. Choose one of the following:  
a. This proposal is for a new script (set of characters): No  
Proposed name of script: \_\_\_\_\_  
b. The proposal is for addition of character(s) to an existing block: Yes  
Name of the existing block: Tifinagh  
2. Number of characters in proposal: 2  
3. Proposed category (select one from below - see section 2.2 of P&P document):  
A-Contemporary x B.1-Specialized (small collection) \_\_\_\_\_ B.2-Specialized (large collection) \_\_\_\_\_  
C-Major extinct \_\_\_\_\_ D-Attested extinct \_\_\_\_\_ E-Minor extinct \_\_\_\_\_  
F-Archaic Hieroglyphic or Ideographic \_\_\_\_\_ G-Obscure or questionable usage symbols \_\_\_\_\_  
4. Proposed Level of Implementation (1, 2 or 3) (see Annex K in P&P document): 3 (one joiner)  
Is a rationale provided for the choice? Yes  
If Yes, reference: \_\_\_\_\_  
5. Is a repertoire including character names provided? Yes  
a. If YES, are the names in accordance with the "character naming guidelines"  
in Annex L of P&P document? Yes  
b. Are the character shapes attached in a legible form suitable for review? Yes  
6. Who will provide the appropriate computerized font (ordered preference: True Type, or PostScript format) for publishing the standard? SIL International  
If available now, identify source(s) for the font (include address, e-mail, ftp-site, etc.) and indicate the tools used: \_\_\_\_\_  
7. References:  
a. Are references (to other character sets, dictionaries, descriptive texts etc.) provided? Yes  
b. Are published examples of use (such as samples from newspapers, magazines, or other sources) of proposed characters attached? Yes  
8. Special encoding issues:  
Does the proposal address other aspects of character data processing (if applicable) such as input, presentation, sorting, searching, indexing, transliteration etc. (if yes please enclose information)?  
Yes, suggested character properties are included (see section D)

**9. Additional Information:**

















Submitters are invited to provide any additional information about Properties of the proposed Character(s) or Script that will assist in correct understanding of and correct linguistic processing of the proposed character(s) or script. Examples of such properties are: Casing information, Numeric information, Currency information, Display behaviour information such as line breaks, widths etc., Combining behaviour, Spacing behaviour, Directional behaviour, Default Collation behaviour, relevance in Mark Up contexts, Compatibility equivalence and other Unicode normalization related information. See the Unicode standard at <http://www.unicode.org> for such information on other scripts. Also see <http://www.unicode.org/Public/UNIDATA/UCD.html> and associated Unicode Technical Reports for information needed for consideration by the Unicode Technical Committee for inclusion in the Unicode Standard.

<sup>1</sup> Form number: N2652-F (Original 1994-10-14; Revised 1995-01, 1995-04, 1996-04, 1996-08, 1999-03, 2001-05, 2001-09, 2003-11)

### C. Technical - Justification

1. Has this proposal for addition of character(s) been submitted before? If YES explain _____	<u>No</u>
2. Has contact been made to members of the user community (for example: National Body, user groups of the script or characters, other experts, etc.)? If YES, with whom? <u>script experts: Andrew Savage, Christian Grandoullier</u> If YES, available relevant documents: _____	<u>Yes</u>
3. Information on the user community for the proposed characters (for example: size, demographics, information technology use, or publishing use) is included? Reference: <u>See comments in Section E</u>	_____
4. The context of use for the proposed characters (type of use; common or rare) Reference: <u>See comments in Section E</u>	<u>common</u>
5. Are the proposed characters in current use by the user community? If YES, where? Reference: <u>See comments in Section E</u>	<u>Yes</u>
6. After giving due considerations to the principles in the P&P document must the proposed characters be entirely in the BMP? If YES, is a rationale provided? If YES, reference: <u>should be kept with Tifinagh block</u>	<u>Yes</u>
7. Should the proposed characters be kept together in a contiguous range (rather than being scattered)?	<u>Yes</u>
8. Can any of the proposed characters be considered a presentation form of an existing character or character sequence? If YES, is a rationale for its inclusion provided? If YES, reference: _____	_____
9. Can any of the proposed characters be encoded using a composed character sequence of either existing characters or other proposed characters? If YES, is a rationale for its inclusion provided? If YES, reference: <u>See comments in Section E.</u>	<u>Yes</u> <u>Yes</u>
10. Can any of the proposed character(s) be considered to be similar (in appearance or function) to an existing character? If YES, is a rationale for its inclusion provided? If YES, reference: <u>See comments in Section E.</u>	<u>Yes</u> <u>Yes</u>
11. Does the proposal include use of combining characters and/or use of composite sequences? If YES, is a rationale for such use provided? If YES, reference: _____ Is a list of composite sequences and their corresponding glyph images (graphic symbols) provided? If YES, reference: _____	<u>No</u> <u>Yes</u>
12. Does the proposal contain characters with any special properties such as control function or similar semantics? If YES, describe in detail (include attachment if necessary) <u>See comments in Section E</u>	<u>Yes</u>
13. Does the proposal contain any Ideographic compatibility character(s)? If YES, is the equivalent corresponding unified ideographic character(s) identified? If YES, reference: _____	<u>No</u>

### D.1. Proposed Characters

	2D7
0	
1	
2	
3	
4	
5	
6	
7	
8	
9	
A	
B	
C	
D	
E	
F	

### D.2. Character Names

#### Punctuation

2D70 TIFINAGH SEPARATOR MARK  
=Tazarast

#### Various signs

2D7F TIFINAGH CONSONANT JOINER

- suppresses inherent vowel, functions to indicate the previous character and the following character form a bi-consonant
- shape shown is arbitrary and is not visibly rendered

### D.3. Unicode Character Properties

```
2D70;TIFINAGH SEPARATOR MARK;Po;0;L;;;;;N;;;;;  
2D7F;TIFINAGH CONSONANT JOINER;Mn;0;NSM;;;;;N;;;;;
```

### E. Other Information

#### E.1 Punctuation

Section D.4 (p. 32) of the Unicode proposal for the Tifinagh script (n2739R.pdf) states that “No specific Tifinagh punctuation marks are known.” The footnote (in the original proposal) to this states:

“Prasse, [20] p. 152, however, mentions the presence in Ahaggar of a separator C « à l’intérieur duquel s’écrit la dernière lettre de chaque mot phonétique ». Its usage does not seem systematic; letter 23 (a) in [10] makes use of a similar symbol. Savage [21] also shows a few recent samples of this separator. No trace of this separator was found in recent works and we have thus preferred not to code this character at this stage.”

The document goes on to say:

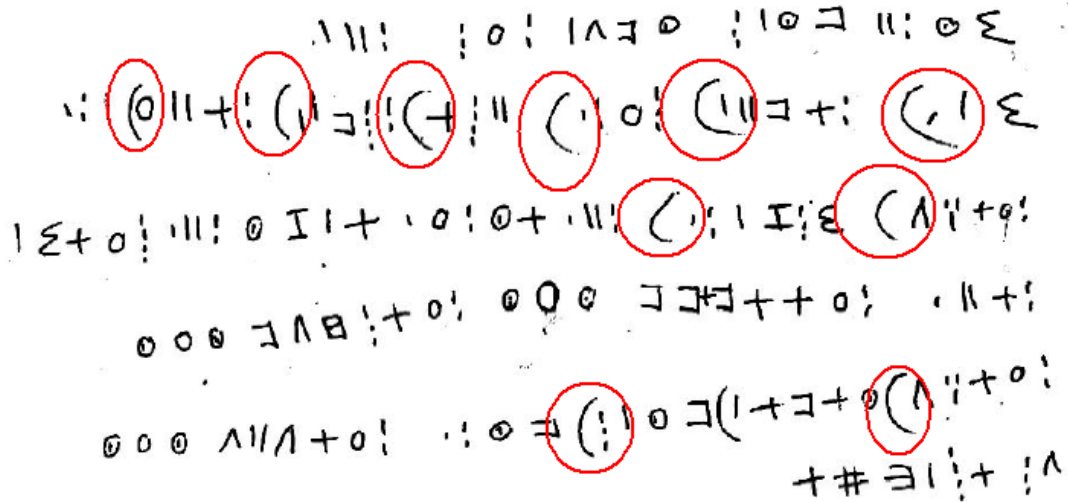
“Ircam has decided that the conventional punctuation signs used in the Latin script shall be used in Tifinagh: “ ” (space), “.”, “,”, “;”, “:”, “?”, “!”, “...”, etc. Consequently, this proposal does not encode any Tifinagh punctuation sign.”

What was not known at the time n2739R was written, is that the word division character is actually in current use, and we believe it should be encoded. U+2D70 TIFINAGH SEPARATOR MARK has traditionally been a "word break" as often "spaces" were not used to separate words - a message was simply a long string of Tifinagh characters, of which the "Tazarast" is used to indicate word separation. In a world more and more influenced by other written languages (French or Arabic), Tifinagh users tend to revert to the convention of blank space to indicate word break. For example, in Algeria a teacher of Tifinagh uses blank spaces and doesn't use this character. However, in Mali, many of the older traditional Tifinagh writers use it, and it is also used in Niger where it can be used in the place of a full stop as well as a comma. In the 03143-tifinagh-samples.pdf (Savage, 2003. pp. 10-12). Savage provided examples of usage in handwritten right-to-left texts. These examples were from Algeria (thus we know of usage in Niger, Mali and Algeria). Figure 1 (from 03143-tifinagh-samples.pdf) illustrates the use of the word division character. Because of the right to left text the character is mirrored. This particular example is from “a couple of decades ago” and illustrates long term usage of this character. Figures 2 and 3 illustrate usage in two languages in Niger where the glyph is oriented for a left to right texts.

In regards to bidi properties, Tifinagh *can* be written right to left. The script has been encoded in Unicode as “strong left to right” (TUS, p 458). In keeping with this, we have suggested a bidi value of “L” for the Tazarast. Regarding mirroring, if the script is used right to left then the Tazarast would need to be mirrored. However, we have not suggested Mirrored=Y because TUS has said it is a left to right script. If this needs to change, we are happy to change it to ‘Y’.

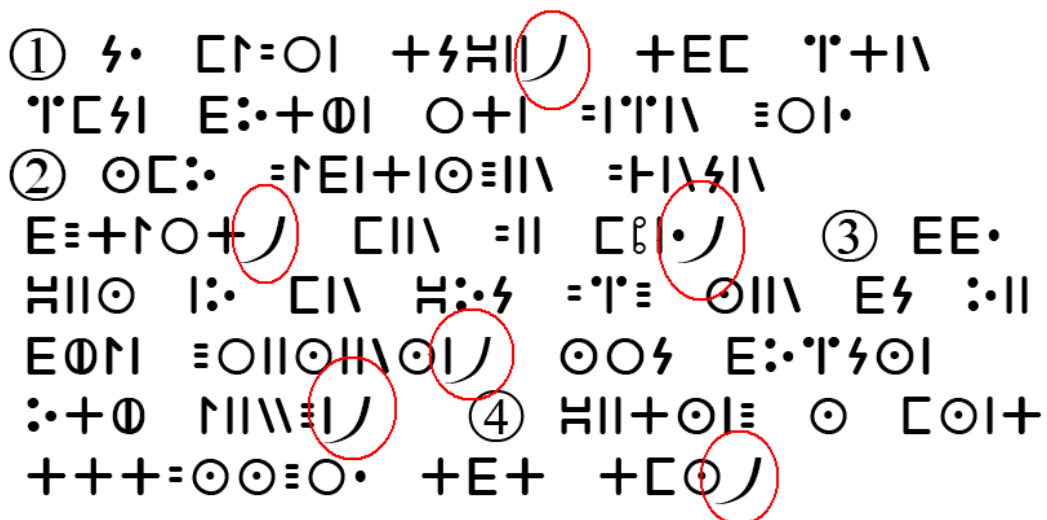
In regards to the alias for this character. Təzərəzt, pl. šizəraz (also written tazərəzt) comes from the root r-z (ərəz: to cut, to stop) (in the Tamajeq dialect in Niger). It is a substantive built on the causative form (i.e. make a cut, make a stop). It means: end, limit. It is something that marks a stop; it could even be something physical marking a limit, an end, like an object, a gate, a property landmark; something that draws limits.

This same name is also used in the Tayərt (Agadez) dialect, the other major dialect in Niger; it is written tezzərəzt, pl. tyəzraz. It is also used in Algeria. We are unaware of whether this term is used in the other main dialect in Mali. It is likely to exist there too as the root is a Təmajəq/Berber root.



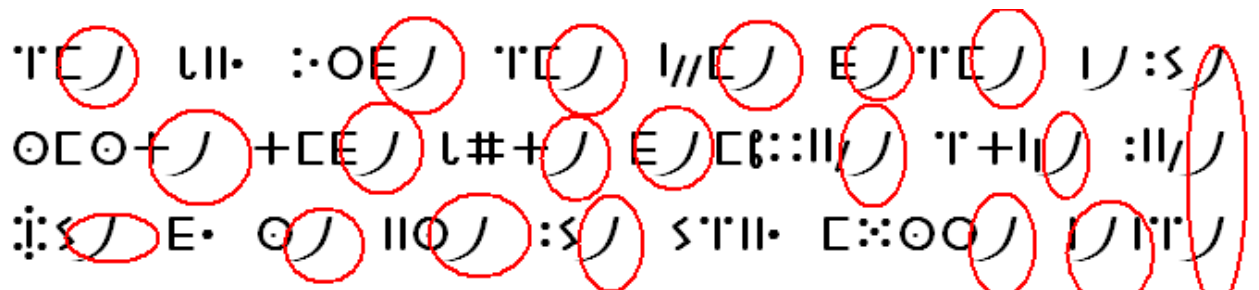
A block of handwritten Tifinagh script. Several characters are circled in red, representing separator marks. The characters are arranged in several lines, with some appearing to be part of larger words or phrases.

Figure 1. TIFINAGH SEPARATOR MARK (Savage, 2003. p. 11)



A block of printed Tifinagh script. The text is organized into four numbered sections (①, ②, ③, ④). Red circles highlight specific characters within each section, which are identified as separator marks. The characters are a mix of letters and symbols, including the plus sign and the circle.

Figure 2. TIFINAGH SEPARATOR MARK (SIM, 2004. p. 1)



A block of printed Tifinagh script. Red circles highlight various characters throughout the text, identifying them as separator marks. The text consists of several lines of characters, including letters, symbols, and punctuation.

Figure 3. TIFINAGH SEPARATOR MARK (Mission Baptiste. p. 1)

## E.2 Various signs

### E.2.1 The Use of Tifinagh Bi-consonantal Characters

Tifinagh is the ancient and original script of the Tuareg people of the Sahara desert. The Tuareg language is related to the Semitic languages, and like those languages (including Arabic and Hebrew) it lends itself to a “consonants-only” script.

The majority of the languages of the world could not be written conveniently with a consonantal (minimum vowel representation) script, however due to the unique syllabic structure of the Semitic and related languages, this is not only possible, but preferable.

One of the features of the syllabic structure of the Tuareg language is that there are never more than two consonants in a cluster eg. *Tefert, temarwelt, ilsa, talamt, enta, tajrist* etc, as opposed to multiple consonant clusters in other languages, such as in English, eg. *Stride, clumps, thirsts* etc. Another feature is that there are no vowel clusters.

These two features lend themselves perfectly to the use of bi-consonantal characters (sometimes called bi-graphs). The bi-consonantal characters are not widely used in writing the Tuareg language, and their absence could be partly responsible for poor literacy levels and consequently a certain decline in the use of the script itself.

There are many different variations in the way Tifinagh is written. Some of the basic variations include character shape and character orientation (left to right, or right to left). Other, more ‘specialised’ variations include vowel representation (which we are not addressing in this proposal as we do not believe there is any general consensus, as yet) and usage (total, partial or non-usage) of bi-consonantal characters. Even if additional vowel characters are encoded in the future, this would not have any affect on how the TIFINAGH CONSONANT JOINER is used.

The character shape and character orientation variations tend to be region specific, having developed in relatively isolated environments in much the same way as spoken dialect variation develops. However the degree of vowel representation and usage of bi-consonantal characters tend to be related to the degree of ‘proficiency’ of the individual writers themselves.

The usage of bi-consonantal characters is a device to help disambiguate words that are written with limited vowel representation. A similar device is used in the Arabic writing system which helps to understand the value of the Tuareg bi-consonantal characters. This device is called the ‘sukun’ (a small circle) which is placed over a consonant, as in the second word [ənkər] in Table 1, which looks like this: . That sukun signifies that there is *no* vowel following that consonant (the “n”), ie. that the “n” and the “k” form a consonant cluster. If that ‘sukun’ was not shown, there would be several additional possible renderings for that word, ie. you could pronounce it with different vocalic possibilities. As you can imagine, the sukun greatly limits the number of possible renderings of that string of three consonants “nkr” and hence makes deduction of the correct meaning readily accessible.

	Meaning	Phonetic	Arabic script	Tifinagh
1.	‘get up’ IMP. habitual	[nâk:ər]	ناكر /nagr/	l:O /nkr/
2.	‘get up’ IMP non hab.	[ənkər]	انكر /'nkr/	·j:O /nkr/

Table 1. Two Tuareg words: written in both the Arabic script and the Tifinagh script

The bi-consonantal characters in Tifinagh perform exactly the same function, ie. they indicate where there is a consonant cluster, and therefore where there is *not* a vowel to be pronounced between the two consonants.

Without using the bi-consonantal characters, each of the two example words in the above table would be written identically in Tifinagh, ie. l:⊙, and there would be potential confusion regarding the correct ‘reading’ of the word. However, by using the bi-consonantal character ḥ, ie. ”nk”, there can only be one ‘reading’ of that word. And consequently, if we consistently use the bi-consonantal characters, we can deduce that there *is* a vowel to be pronounced between the “n” and the “k” of our first example word since we did *not* make use of a bi-consonantal character in that instance.

One can see that there is not an automatic combination of two consonantal characters to give a bi-consonantal character: the two consonants can be represented separately (side by side) or as a single bi-consonantal character depending on the presence or not of a vowel.

The following chart lists contrastive pairs of vocalic and non-vocalic clusters. An examination of these examples clearly shows why bi-consonants are needed in the script.

Non-vocalic clusters (using bi-consonants)				Vocalic clusters (no bi-consonant use)			
Script		Examples		Script		Examples	
Tifinagh	Latin	Tifinagh	Latin	Tifinagh	Latin	Tifinagh	Latin
⊖	mb	⊖	imbal	⊖⊙	m <sub>v</sub> /b	⊖⊙⊙	šemeber
⊕	rt	+⊖⊕	təməddurt	⊙+	r <sub>v</sub> /t	⊕⊙+	əstizarat
χ	lt	+E:χ	tədhəlt	+	l <sub>v</sub> /t	⊖  +	šilat
⊖	št	⊖⊙+	aštərot	⊖+	š <sub>v</sub> /t	+:⊖+	tawšetən
⊖	mt	+⊙:⊖	taryəmt	⊖+	m <sub>v</sub> /t	+⊖+	tamattay
⊖	ft	⊖:	annaftay	⊖+	f <sub>v</sub> /t	+⊙⊖+	təbbəffət
⊖	ms	+⊖	tamsay	⊖⊙	m <sub>v</sub> /s	⊖⊙	əmosan
⊖	nd	⊖	iyyanda	⊖	n <sub>v</sub> /d	⊖ ⊖	šinad
⊖	nf	⊖⊙	infas	⊖	n <sub>v</sub> /f	+ ⊖+	tənafut
ḥ	ng	ḥ	əngəḥ	ḥ	n <sub>v</sub> /g	ḥ⊙	əḥḥmar
ḥ	ny	ḥ	inyal	:	n <sub>v</sub> /y	+ :	ilatanay
⊖	nj	⊖⊙+	šinjar-net	#	n <sub>v</sub> /j	E: #:⊙	dak-za-najjawwab
ḥ	nk	#:ḥ	jiwanken	:	n <sub>v</sub> /k	⊖ :	əməḥokal
⊙	ns	+⊙⊙:	təsənsəḥ	⊙	n <sub>v</sub> /s	+⊖⊙	təmənesən
⊖	nt	⊖	əntanay	+	n <sub>v</sub> /t	E +	əḥḥnet
⊖	nz	:⊙	izakkanzar		n <sub>v</sub> /z	⊙	əḥzər
⊕	st	⊕⊙+	əstizarat	⊙+	s <sub>v</sub> /t	+⊖⊙+	təmosat

Table 2. Tifinagh bi-consonant example chart (Tawallammat dialect)



## E.2.2 ZWJ vs TIFINAGH CONSONANT JOINER

Section D.8 (pp.34-36) of the Unicode proposal for the Tifinagh script (n2739R.pdf) discusses the “vast repertoire<sup>2</sup> of Tuareg ligatures (sometimes called “biconsonants”)” (p. 34). It concludes “This proposal neither encodes nor prescribes any ligature. It leaves the choice of ligatures to the font chosen at rendering time. Some fonts may be deprived of any ligature; others may include only some ligatures for a given geographical variant. Users may use the usual convention of inserting a U+200D ZERO WIDTH JOINER (resp. U+200C ZERO WIDTH NON-JOINER) to encourage (resp. discourage) the formation of a “biconsonant” at rendering time.” (p. 36).

Although we have listed some bi-consonants in the chart above, there are ever increasing numbers of them as is referenced by the original Tifinagh proposal – “vast repertoire...” If one were to encode each bi-consonant there would likely be an ever growing number that needed encoding. Rather than proposing the addition of each bi-consonant, we are proposing the addition of U+2D7F TIFINAGH CONSONANT JOINER. This would be different than the ZWJ in that it would not be default ignorable. It would have the Mn (Mark, Non-Spacing) property rather than Cf (Other, Format). We are aware of at least one font implementation which uses the ZWJ to form bi-consonants. We do not believe that using the ZWJ is the right decision. ZWJ is default ignorable and bi-consonants in certain languages are only formed when there is no inherent vowel. When searching and when sorting, these bi-consonants are considered totally separate from when two characters appear together. If the ZWJ is used, they are processed as the same characters, because the ZWJ is default ignorable. In fact, they should be processed separately. TUS (p. 537) states that the ZWJ is “meant only for overriding the normal behavior of the text.” Because the bi-consonants are not used just as “normal behavior”, but indicate *different meaning to the text* we believe that the ZWJ is inappropriate. For example ⵎ: “nk” (with inherent vowel) has a different meaning than ⵎ̣: “nk” (no vowel). For these reasons we believe the TIFINAGH CONSONANT JOINER should be encoded so that the bi-consonants will be processed separately in searching and in sorting.

Although the TIFINAGH CONSONANT JOINER does not exist in the script, there is precedent for introducing such a character in the script in the Coeng encoding model (Everson).

## E.2.3 Rendering of TIFINAGH CONSONANT JOINER

It is clear that if U+2D7F is used in text, the bi-consonant should be rendered. Since TIFINAGH CONSONANT JOINER is part of the spelling of a word, it must always impact rendering either by creating a bi-consonant or by being rendered as a mark of some kind to show that a TIFINAGH CONSONANT JOINER is part of the character sequence. If a font does not implement the bi-consonant, there should be a visible display of 2D7F. For example, 2D54 + 2D7F + 2D5C should render as:  if the font has implemented the bi-consonant and as  if the font has not implemented the bi-consonant.

U+2D7F cannot occur before *or* after anything but a Tifinagh consonant. These are:

(⊖ 2D31) (⊕ 2D32) (⊗ 2D33) (⊘ 2D34) (⊙ 2D35) (⊚ 2D36) (⊛ 2D37) (⊜ 2D38) (⊝ 2D39) (⊞ 2D3A) (⊟ 2D3B) (⊠ 2D3C) (⊡ 2D3C 2D7F 2D5C) (⊢ 2D3D) (: 2D3E) (⊣ 2D3F) (⊤ 2D40) (⊥ 2D41) (⊦ 2D42) (⊧ 2D43) (⊨ 2D44) (⊩ 2D45) (: 2D46) (⊬ 2D47) (⋯ 2D48) (⊞ 2D49) (⊠ 2D4A) (⊡ 2D4B) (⊣ 2D4C) (⊥ 2D4D) (⊦ 2D4D 2D7F 2D5C) (⊧ 2D4E) (⊩ 2D4E 2D7F 2D40) (⊫ 2D4E 2D7F 2D59) (⊭ 2D4E 2D7F 2D5C) (⊮ 2D4F) (⊯ 2D4F 2D7F 2D36) (⊱ 2D4F 2D7F 2D39) (⊲ 2D4F 2D7F 2D3C) (⊳ 2D4F 2D7F 2D3E) (⊴ 2D4F 2D7F 2D4C) (⊵ 2D4F 2D7F 2D57) (⊶ 2D4F 2D7F 2D59) (⊷ 2D4F 2D7F 2D5C) (⊸ 2D4F 2D7F 2D64) (≠ 2D50) (⊩ 2D51) (⊫ 2D52) (: 2D53) (⊭ 2D54) (⊯ 2D54 2D7F 2D5C) (⊱ 2D55) (⊲ 2D56) (: 2D57) (: 2D58) (⊭ 2D59) (⊯ 2D59 2D7F 2D5C) (⊱ 2D5A) (⊲ 2D5B) (⊳ 2D5B 2D7F 2D5C) (+ 2D5C) (⊡ 2D5D) (⊣ 2D5E) (⊥ 2D5F) (⊦ 2D60) (⊨ 2D61) (⊬ 2D62) (⊰ 2D63) (⊲ 2D64) (⊴ 2D65)

## E.2.4 Input (keyboarding) of Tifinagh script

Regarding the keyboarding of consonant clusters as bi-consonantal characters, the choice needs to remain with the keyboarder himself for each individual word. It simply does not work, for example, to

<sup>2</sup> Savage (2000, p. 94) says that Aghali-Khalil (1994, p. 118) has documented 82 bi-consonants.



automatically convert keystroke “n” followed by keystroke “k” to the bi-consonantal character “nk”, as often those two Tifinagh characters “n” and “k” will need to remain separate, indicating that there is a vowel between them (though not written) that needs to be pronounced. Refer to the Tifinagh rendering of the two words in Table 1.

Regarding the bi-consonants themselves, it has been said there are two types of consonant clusters that they represent, i.e. those clusters with “n” as the first consonant and those with “t” as the final consonant. This is generally true, but there are a considerable number of exceptions, some of which can be seen in Table 2 above.

*E.2.5 Sorting (allkeys.txt)*

TIFINAGH SEPARATOR MARK should be sorted after 2D6F TIFINAGH MODIFIER LETTER LABIALIZATION MARK. The primary weight for 2D7F (TIFINAGH CONSONANT JOINER ) should have the affect that the bi-consonants are sorted after the consonant. So, for example, using the above bi-consonants, they would occur this way (where the bi-consonants are in red and underlined).

This is the current sort order:

(• 2D30) (⊖ 2D31) (⊕ 2D32) (⌘ 2D33) (⌘ 2D34) (⌘ 2D35) († 2D36) (∧ 2D37) (∨ 2D38) (⊖ 2D39) (⊖ 2D3A) (⊖ 2D3B) (∩ 2D3C) (∩ 2D3D) (∩ 2D3E) (∩ 2D3F) (∩ 2D40) (∅ 2D41) (∩ 2D42) (∩ 2D43) (∩ 2D44) (X 2D45) (∩ 2D46) (∩ 2D47) (∩ 2D48) (∩ 2D49) (∩ 2D4A) (X 2D4B) (∩ 2D4C) (∩ 2D4D) (∩ 2D4E) (∩ 2D4F) (∩ 2D50) (∩ 2D51) (∩ 2D52) (∩ 2D53) (∩ 2D54) (∩ 2D55) (∩ 2D56) (∩ 2D57) (∩ 2D58) (∩ 2D59) (∩ 2D5A) (∩ 2D5B) (∩ 2D5C) (X 2D5D) (∩ 2D5E) (∩ 2D5F) (∩ 2D60) (∩ 2D61) (∩ 2D62) (X 2D63) (∩ 2D64) (∩ 2D65) (∩ 2D6F)

**E.3 Changes to 13.4 in TUS.**

The documentation on pp.457-458 of the TUS 5.0 should be updated. We would suggest the following.

*E.3.1 Bi-consonants*

The bi-consonantal characters in Tifinagh indicate where there is a consonant cluster, and therefore where there is *not* a vowel to be pronounced between the two consonants (There is the possibility of tri-consonants, however they have traditionally been called bi-consonants which is why we are using this term). The Unicode Standard does not assign independent code points for the bi-consonants. Instead, bi-consonants can be achieved by placing the TIFINAGH CONSONANT JOINER between the two consonants. The glyph for U+2D7F TIFINAGH CONSONANT JOINER shown in the code charts is arbitrary and is not actually rendered directly; the dotted box around the glyph indicates that special rendering is required. While the Unicode encoding for a bi-consonant is thus three codepoints, the sequence functions as if it had been encoded as a single character. Figure 13.x illustrates the use of the bi-consonants. Note that this is by no means a comprehensive list of bi-consonants. Note also that there are glyph variants for these bi-consonants depending on the region.

Figure 13.x Using TIFINAGH CONSONANT JOINER to create bi-consonants

∩ 2D4E	+	∩TCJ∩ 2D7F	+	∩ 2D5C	➔	∩∩ 2D5C
 2D4F	+	∩TCJ∩ 2D7F	+	∩ 2D3E	➔	∩∩ 2D3E
 2D4F	+	∩TCJ∩ 2D7F	+	∩ 2D5C	➔	∩∩ 2D5C
○ 2D54	+	∩TCJ∩ 2D7F	+	∩ 2D5C	➔	∩∩ 2D5C
⊙ 2D59	+	∩TCJ∩ 2D7F	+	∩ 2D5C	➔	∩∩ 2D5C

### E.3.2 Contextual shaping

Besides bi-consonants, there is sometimes contextual shaping of characters. This occurs when "ll" (lll), "ln" (lll), "nl" (lll) or "nn" (ll) appear together. This contextual shaping can occur in order to distinguish between characters when like pairs appear next to each other. To write this, the 2nd character is shifted vertically or it can be slanted. Figure 13.y illustrates the use of contextual shaping.

Figure 13.y Contextual shaping

ll 2D4D	+	ll 2D4D	➔	ll\
ll 2D4D	+	l 2D4F	➔	ll\
l 2D4F	+	ll 2D4D	➔	l\
l 2D4F	+	l 2D4F	➔	l\

### F. References

- Andries, P. 2004-06-06. *Proposal to add the Tifinagh Script*. N2739R ISO/IEC JTC 1/SC 2/WG 2 N2739. <http://anubis.dkuug.dk/jtc1/sc2/wg2/docs/n2739.pdf>.
- Everson, Michael. 2002-03-31. *On the suitability of the COENG encoding model for Khmer*. ISO/IEC JTC1/SC2/WG2 N2412. <http://std.dkuug.dk/jtc1/sc2/wg2/docs/n2412.pdf>
- Mission Baptiste – Agadez. In press. *Ayuba*. Mission Baptiste: Agadez, Niger
- Savage, Andrew. 2000. *Writing Tuareg Vowels. Advantages and Disadvantages of the Three Script Options: Arabic, Tifinagh and Roman*. Submitted in partial fulfillment of the requirements for the degree of Master of Letters in Linguistics from the University of New England, New South Wales, Australia.
- \_\_\_\_\_. 2003. Tifinagh Unicode Propositions, Stanthorpe, <http://www.unicode.org/L2/L2003/03143-tifinagh-samples.pdf>
- The Unicode Consortium. *The Unicode Standard*, Version 5.0.0, defined by: The Unicode Standard, Version 5.0 (Boston, MA, Addison-Wesley, 2007. ISBN 0-321-48091-0)
- SIM. 1997, 2004. *Isālan n-Ālkher win iktāb Luk* (The Gospel of Luke in the Tamajeq Language). SIM, Niamey, Niger. (<http://www.westafricanscripts.com/> accessed 2008-01-25)
- Aghali-Zakara, M. 1994. "Graphies berbères et dilemme de diffusion: interactions des alphabets latin, ajami et tifinagh, *Etudes et Documents berbères*, 11, 1994 : 107-121.

**G. Chart of some Bi-Consonants**

	transcription	Bi- tri- consonants	CGr	DE	AS	PA document on inventory of oails (Prasse, p. 157 et Aghali-Zakara)							
						0	1	2	3	D	E	F	
1.	bt	ⵜⵓ				ⵜⵓ	ⵜⵓ						
2.	dj	ⵈⵓ		ⵈⵓ									
3.	ḍt	ⵉⵓ				ⵉⵓ							
4.	ft	ⵏⵓ	ⵏⵓ		ⵏⵓ	ⵏⵓ							
5.	gt	ⵔⵓ				ⵔⵓ							
6.	ḡt	ⵓⵓ				ⵓⵓ							
7.	lt	ⵙⵓ	ⵙⵓ	ⵙⵓ	ⵙⵓ	ⵙⵓ	ⵙⵓ	ⵙⵓ					
8.	mb	ⵎⵓ	ⵎⵓ	ⵎⵓ						ⵎⵓ	ⵎⵓ	ⵎⵓ	
9.	ms	ⵎⵓ	ⵎⵓ										
10.	mt	ⵎⵓ	ⵎⵓ	ⵎⵓ	ⵎⵓ	ⵎⵓ	ⵎⵓ	ⵎⵓ					
11.	nb	ⵎⵓ				ⵎⵓ							
12.	nd	ⵎⵓ	ⵎⵓ	ⵎⵓ	ⵎⵓ	ⵎⵓ	ⵎⵓ	ⵎⵓ	ⵎⵓ				
13.	nḍ	ⵎⵓ				ⵎⵓ							
14.	nf	ⵎⵓ	ⵎⵓ	ⵎⵓ	ⵎⵓ								ⵎⵓ
15.	nft	ⵎⵓ	ⵎⵓ										
16.	ng	ⵎⵓ	ⵎⵓ	ⵎⵓ		ⵎⵓ	ⵎⵓ						
17.	nḡ	ⵎⵓ				ⵎⵓ							
18.	ny	ⵎⵓ	ⵎⵓ			ⵎⵓ	ⵎⵓ						
19.	ngh	ⵎⵓ		ⵎⵓ									
20.	nj	ⵎⵓ	ⵎⵓ	ⵎⵓ									ⵎⵓ
21.	nk	ⵎⵓ	ⵎⵓ	ⵎⵓ	ⵎⵓ	ⵎⵓ	ⵎⵓ						
22.	nkn	ⵎⵓ											
23.	ns	ⵎⵓ	ⵎⵓ										
24.	nš	ⵎⵓ											ⵎⵓ

25.	ntš	I+G		±								
26.	nt	I+	┆	┆	T	+	T				┆	┆
27.	ntn	I+I										
28.	nz	Iʀ	˘	˘							˘	˘
29.	rd	OE		⊕								
30.	rf	O]C		⊕								
31.	rh	O∴		⊕								
32.	rk	O∴		⊕								⊕
33.	rg	Oʀ		⊕								
34.	rgħ	O∴		⊕								
35.	rm	OC		⊕								
36.	rn	O		⊕								O'
37.	ry	O∴		⊕								
38.	rš	OG		⊕								
39.	rt	O+	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕		
40.	rtš	O+G		⊕								
41.	rw	O∴		⊕								
42.	rz	OX		⊕								
43.	sn	⊙		⊕	⊕							
44.	st	⊙+	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕		
45.	št	G+	⊕			⊕	⊕	⊕				
46.	tš	+G		⊕								
47.	wt	:+		⊕	⊕							
48.	yt	˘+		˘	˘							
49.	zt	#+			⊕	⊕	⊕	⊕				