

# Representing Myanmar in Unicode Details and Examples Version 4

*Martin Hosken*<sup>1</sup>

## Table of Contents

Introduction.....	<a href="#">2</a>
Unicode 5.1 Model.....	<a href="#">4</a>
Advanced Issues.....	<a href="#">12</a>
Languages.....	<a href="#">20</a>
Burmese မြန်မာစာ.....	<a href="#">21</a>
Old Burmese ရှေးဟောင်း မြန်မာစာ.....	<a href="#">26</a>
Sanskrit/Pali သက် သက္ကတ/ပါ ပါဠိ.....	<a href="#">28</a>
Rakhine ရခိုင်စာ.....	<a href="#">30</a>
Tavoyan.....	<a href="#">30</a>
Intha.....	<a href="#">30</a>
Mon မန်.....	<a href="#">31</a>
Sgaw Karen စိုက.....	<a href="#">33</a>
Western Pwo Karen ဖျိုး.....	<a href="#">35</a>
Eastern Pwo Karen ပုကညီ.....	<a href="#">36</a>
Pa'o Karen ပအိုဝ်း.....	<a href="#">37</a>
Kayah ကယား.....	<a href="#">38</a>
Asho Chin အရှား.....	<a href="#">40</a>
Shan လိမ်းတီး.....	<a href="#">41</a>
Khamti Shan လိက်တဲးဂမ်းတီး.....	<a href="#">46</a>
Aiton & Phake ကဲတွခ် & လျက်.....	<a href="#">49</a>
Tai Laing - တိုင်းလျမ် ခေါ် ရှမ်းနီ.....	<a href="#">51</a>
Shwe Palaung - ရွှေပလောင်.....	<a href="#">53</a>
Pale Palaung - ပလေး.....	<a href="#">56</a>
Rumai Palaung ရူမာည်းတအာင်း.....	<a href="#">58</a>
References.....	<a href="#">66</a>
Afterward.....	<a href="#">67</a>

---

<sup>1</sup> SIL International and Payap University Linguistics Institute, Chiang Mai, THAILAND

## Introduction

This document aims to give guidance on the encoding of text using the Myanmar script. Since the script is used for a number of orthographies covering different languages, the development of this document is ongoing. It aims to bring together the results of consensus between experts in the encoding of the various orthographies using the script. In terms of the Unicode standard, this document is purely informative since it is concerned with issues not covered by that standard. But within the country, and by developers of the script, this document has been accorded a certain degree of authority. This provides further encouragement to maintain this document and update it as new issues arise.

Readers interested in following the history of the development of this script are recommended to read the different versions of this document, rather than expecting to find this document containing all versions of itself within.

The Myanmar script is used for a number of languages. This means that when considering the script as a whole, care must be taken not to over specify constraints on what character sequences should be considered valid or in error. The temptation is to use script level sequence constraints as a form of spell checking. But spell checking is inherently language specific. The result is that script constraints need to be the lowest common denominator of all the orthographies supported by the script. The orthography list is not closed: we have not described all the existing orthographies yet; languages change and develop and their orthographies with them. As a result, script constraints cannot simply be the intersection of all known writing system constraints, but must take a more intentional approach. The basic principle used here is not to try to constrain what users can generate, but only to ensure that there are no two different valid sequences that look the same, within a writing system. We do this by specifying a valid string as being a sequence of slots. Each slot may be empty or contain a character (or sequence as specified by the slot). Implementations may well add further, language specific, constraints to help their users.

A further concern when reading a developing document such as this is the stability criteria. What can we be sure about going into the future? The approach taken in this document follows the core principle of stability in Unicode: Any valid data today will always remain valid. This requires that any changes to the sequence order, for example, will always be to loosen it. Thus more sequences will be allowed rather than less. This means that invalid data today may not always remain invalid in future versions of this document. It should also be born in mind that while the unity of the script as a whole may well be affected by the addition or changes in a single language, each language stands alone in its encoding and needs its own consistency. Care is taken that any changes that a difference in language may cause on the script as a whole (adding more legal sequences), do not cause any changes in other language encodings. This may result in some decisions made for a particular language, looking different from those for another language and the temptation to try to over unify languages should be avoided.

Following the one time change for Burmese in Unicode 5.1, there will be no more changes to Unicode for Burmese. The extra characters described here are additions for minority and historic languages. This version of UTN#11 brings the specification in line with Unicode 5.2.

## Introduction to Version 2

The first edition of this technical note addressed the issue of how Myanmar text was encoded using the Unicode standard as it stood until version 5.1. With Unicode 5.1 various new characters were added to the Myanmar block which had the effect of simplifying the encoding model considerably. Such a change could only come about with agreement from all implementers and those with existing data because they will need to update and change to the new model. This is nearly impossible to achieve if existing implementations are already in widespread use, which was not the case at the time for the Myanmar block. In addition, such a change was necessary to facilitate the encoding of minority scripts. So with a necessity and a unique opportunity for change, the characters were added and the encoding model simplified.

The author wishes to thank the Myanmar Language Commission, the Myanmar NLP Lab and the Myanmar Computer Federation for reviewing and providing input to this version of the document.

### **Introduction to version 3**

The first two editions of this document were almost exclusively concerned with the needs of the Burmese language. This edition drastically extends the set of allowable sequences and considers the needs of a number of minority languages. It also adds summary descriptions of a number of languages that have Myanmar based writing systems and gives indication on how they are encoded along with other computational issues that these writing systems raise.

### **Introduction to version 4**

This updated edition concentrates on implementation issues and revises the diacritic order slightly. The net effect on real language data is none and all valid strings used in language are still valid with no change. But in order to ease implementation, some minor changes have been made that will affect the order of characters in strings that do not actually occur in real language. This allows the introduction of both a regular expression and a canonical ordering algorithm for the order. The edition also adds information regarding writing systems and characters added after Unicode 6. Some example strings, keyboard layouts and some sort order tailorings have been included for a few languages.

# Unicode 5.1 Model

## Basic Myanmar

The basic consonants and vowels are relatively obvious in how they are encoded, by examining the character charts. Thus:

စာ	1005 102C	letter
----	-----------	--------

Above we show the Myanmar word, the underlying Unicode codes that would be stored to represent this and an English gloss of the word. As this example shows, characters are stored in the order in which they are read.

ခါ	1001 102B	to shake
သိက္ခာ	101E 102D 1000 1039 1001 102C	dignity
သဒ္ဓါ	101E 1012 1039 1013 102B	faith

In this example, we highlight the code of interest. Notice how the ဝါ (U+102B MYANMAR VOWEL SIGN TALL AA) has a different code to the ဝ (U+102C MYANMAR VOWEL SIGN AA). The Myanmar character underlying the two codes is the same, and there are rendering rules that can give the correct form, so why has the tall -aa been given its own code? The primary reason is that Sgaw Karen, among other minority scripts, only has the tall form, and so a rendering system that works for the Myanmar language is not going to work for Sgaw Karen and vice versa. A Myanmar language specific keyboarding implementation could choose to enforce a particular variant of the -aa vowel in the context of certain consonants (in Burmese following စ, ဂ, င, ဒ, ဓ<sup>2</sup>, ဝ, or ဝ), medial combinations and syllable chainings, but this is not required.

ညို	100A 102D 102F	brown
ထိုး	1011 102F 1036 1038	to tie

Notice how the two forms of ဝ (U+102F MYANMAR VOWEL SIGN U) have the same code. It is up to the rendering system to choose which form should be shown and different fonts can have different rules depending on the designer's preference.

### U+1031 –e vowel


We will see later why the vowels are stored in this relative order. But for now it is important to note that the Unicode standard states that vowels are stored after the consonant, according to how they are pronounced, regardless of where they are rendered. This introduces one of the complexities of implementing Myanmar script:

နေ	1014 1031	the sun
ပေါ	1015 1031 102B	plentiful
လှေ	101C 103E 1031	boat
မြေ	1019 103C 103D 1031	snake

The ဝ vowel is rendered in front of the consonant that it is pronounced (and so stored) following. Notice that this says nothing about the relative order for typing, but it does mean that anyone implementing the Myanmar script needs to take special care of this character. In general people are used to and want to type the ဝ vowel in front of the consonant, and so implementers need to address issues of keyboarding as well as rendering.

<sup>2</sup> Some characters may take tall or short forms of -aa based on stylistic preference.

## Medials

The medial characters have their own codes and are always stored after the base consonant and before any vowels. Although the character  has traditionally been typed in non-Unicode fonts before the consonant, it is consistent with normal spelling to store U+103C MYANMAR CONSONANT SIGN MEDIAL RA after the consonant.


ဖျာ့	1016	<b>103B</b>	102C	1038	fever
ကြဲ	1000	<b>103C</b>	1031	1038	grime
မွဲ	1019	<b>103D</b>	1031	1038	give birth
မူ	1019	<b>103E</b>	102F		regard important

## Syllable Chaining



In the case of syllable chaining, subjoined characters are not given their own codes. Instead a virama character is used to indicate that the following character is subjoined and should take a subjoined form.

ပတ္တ	1015	1010	<b>1039</b>	1010	102C	hinge
------	------	------	-------------	------	------	-------

## Devoweliser


There are two ways of representing the devowelising process. The first is by creating a syllable chained form, using U+1039 to mark the devowelising (as shown above). The second is to use the visible virama character  U+103A MYANMAR SIGN ASAT in conjunction with a base consonant.

ထင်	1011	1004	<b>103A</b>		think
ကြည်	1000	103C	1009	<b>103A</b>	avoid
ကော်	1000	1031	102C	<b>103A</b>	glue

The second example also illustrates that  is encoded with U+1009 followed by U+103A even though the glyph shape closely resembles the independent vowel  U+1025 MYANMAR LETTER U. Keyboard implementers may wish to enforce this.

The third example is not a true devowelising, but it shows that U+103A can also be used as a vowel in combination with U+102B and U+102C.


## Kinzi

The remaining issue regarding representation needed for the modern Myanmar language is how kinzi is represented in Unicode. Glyph based encodings give the kinzi its own code. But linguistically, the kinzi is merely a special form of a devowelised nga  U+1004 MYANMAR LETTER NGA. We encode kinzi as a devowelised nga with the following letter underneath, subjoined. But the difference is that when rendered, the devowelised nga changes shape and the subjoined base character remains a full character. Thus we use U+1004 U+103A U+1039.

ပကြဲ	1005	<b>1004</b>	<b>103A</b>	<b>1039</b>	1000	path
					103C 1036	
သဘော်	101E	<b>1004</b>	<b>103A</b>	<b>1039</b>	1018	ship
					1031 102C	

Like the –e vowel, kinzi is particularly problematic to implement since people want to type it following the base consonant and it also needs careful handling during rendering.

## Diacritic storage order

It is possible for a Myanmar syllable to have a number of diacritics surrounding a base consonant, independent vowel or digit. Since all these diacritics are not spacing, how do we know in which order they should be stored? For example,  can be stored as U+1004 U+102D U+102F or as U+1004 U+102F

U+102D. But what happens if one person stores it one way and then someone searches for that word spelled the other way? It is important that there is a consistent way of storing strings so that applications can work consistently.

The following list gives the relative order that each diacritic should be stored, if it occurs, following a base consonant. The specification of each slot is a sequence of characters. Where there is a list of characters enclosed in [], only one of them may occur in that position. x .. y implies an inclusive range of characters. The Id column contains a possible canonical combining class for the characters listed. Rows with a grey background containing spacing characters and so may have a zero canonical combining class.

Name	Id	Specification	Sample	Constraints
Kinzi	-1	[U+1004, U+101B, U+105A] U+103A U+1039	Ⴄ	
Consonant	0	[U+1000 .. U+102A, U+103F, U+1041 .. U+1049, U+104E, U+1050 .. U+1055, U+105A .. U+105D, U+1061, U+1065, U+1066, U+106E .. U+1070, U+1075 .. U+1081, U+108E, U+109F, U+A9E0 .. U+A9E4, U+A9E7 .. U+A9EF, U+A9FA .. U+A9FF, U+AA60 .. U+AA6F, U+AA71 .. U+AA76, U+AA7A, U+AA7E, U+AA7F]	Ⴄ	required
Stacked	1	U+1039 [U+1000 .. U+101C, U+101E, U+1020, U+1021, U+1050, U+1051, U+105A .. U+105D]	Ⴄ	
Stacked2	1	U+1039 [U+1000 .. U+101C, U+101E, U+1020, U+1021, U+1050, U+1051, U+105A .. U+105D]	Ⴄ	
Asat	2	U+103A	Ⴄ	_ [^U+103E, U+1082, U+1037]
Medial Y	3	[U+103B, U+105E, U+105F]	Ⴄ	
Medial R	4	U+103C	Ⴄ	
Medial W	5	[U+103D, U+1082]	Ⴄ	
Medial H	6	[U+103E, U+1060]	Ⴄ	
Mon Asat	2	U+103A	Ⴄ	[U+103E, U+1082] _ [^U+1037]
E vowel	7	[U+1031, U+1084]	Ⴄ	
Shan E vowel	7	U+1031	Ⴄ	U+1031 _
Upper Vowel	8	[U+102D, U+102E, U+1032 .. U+1036, U+1071 .. U+1074, U+1085, U+109D, U+A9E5]	Ⴄ	[U+1032, U+1036] [^U+102F, U+1030, U+1086] <sup>3</sup>
Lower Vowel	9	[U+102F, U+1030, U+1058, U1059]	Ⴄ	
Karen Vowel	0	U+1062	Ⴄ	[^U+102F, U+1030] _ [^U+1032]
Shan Vowel	10	U+1086	Ⴄ	
Dot Vowel	11	U+1037	Ⴄ	[^U+102F, U+1030] _ [^U+1032, U+1036]
A Vowel	0	[U+102B, U+102C, U+1056, U+1057, U+1062, U+1063, U+1067, U+1068, U+1083]	Ⴄ	
Mon h	6	U+103E	Ⴄ	U+102C _ U+103A
A Asat	2	U+103A	Ⴄ	[A Vowel, U+103E <sup>4</sup> ] _ [^U+1037]

<sup>3</sup> Also disallow the sequence U+1036 U+102B. Not listed in the chart for brevity.

<sup>4</sup> Where this follows U+102C as in the Mon h slot.



## Stacked

The normal stack is made up from a VIRAMA U+1039 followed by a consonant. Not every consonant can be stacked, and while theoretically any consonant can take a subjoined form, not all implementations will necessarily need to support all subjoined forms. The ones listed here are the ones known to exist.

## Stacked2

In some languages, particularly Sanskrit in rare situations, complex conjuncts involving three non medial consonants may occur. For example: tsna in Sanskrit would be encoded U+1010 U+1039 U+101E U+1039 U+1014.

## Asat

This slot position is used for all cases where an asat is rendered over a consonant, unless that consonant is followed by a MEDIAL HA U+103E which is used as a contraction in Mon, or a SHAN MEDIAL WA U+1082 which is used with asat to form a vowel in Shan. See the section on Mon for more details.

When data is normalized, if U+1037 directly follows U+1039 or U+103A then it is reordered before it. So an additional constraint is that U+103A may not occur immediately before U+1037.

## Medial Y

This slot also includes any specific medials that do not correspond to other medials. In this case we include the Mon -m and -n medials.

## Medial R

Notice that medial ra is stored after the consonant even though it may be considered to be rendered before it.

## Medial W

This slot also includes the Shan medial w. This results in an encoding of ဝ as U+1082 U+103A with the asat being in the Mon Asat slot.

## Medial H

This slot includes MON MEDIAL LA U+1060 since it is used as a medial h in Karen.

## Mon Asat

This only occurs in Mon, where it immediately follows a MEDIAL HA U+103E or a SHAN MEDIAL WA U+1082.

## E Vowel

All pre-vowels go into this slot.

## Shan E Vowel

The Common Shan script encodes the vowel that is more usually encoded as ဝ U+1084 using two E vowels, following Thai. Since this only occurs in a historic script, the easiest solution is to allow two E vowels as in ဝဝ U+1031 U+1031. This slot is for the second U+1031.

## Upper Vowel

This slot contains anything that can go on top of a consonant. Notice that only one upper vowel can occur. U+1036 may only occur in this slot if there is nothing in the Lower Vowel slot or there is a following spacing component and the anusvara is to be rendered over the consonant. There is one exception to this and that is in Mon where the sequence U+102B U+1036 is rendered ဝ်. The constraint listed in the chart only applies to the characters U+1032 and U+1036, hence there is no \_ placeholder.



## Lower Vowel

These are the standard Burmese lower vowels. This also specifies the order of ဝ as being U+102D U+102F.

## Karen Vowel

This slot does not occur with the previous Lower Vowel slot. It contains characters that are used as vowels in other languages. Notice that in Sgaw Karen one can have two occurrences of ဝ U+1062 as in ကာ် U+1000 ဝ U+1062 ဝ U+1062 ဝ U+103A.

## Shan Vowel

This upper diacritic may either occur above a consonant, or above a following Shan a vowel ဝ U+1062. The position depends on which of the various Shan scripts is being written. As a result, this slot position is optimal since it occurs between two slots containing ဝ U+1062.

## Dot Vowel

### A Vowel

Unlike other slots which may or may not include spacing characters, the A vowel slot always contains a spacing character. This is not to say that the A Vowel slot always has to be filled.

### Mon H

Mon has the concept of contracting final consonants using diacritics. One such is using medial h followed by an asat to represent a final h. Since the medial h may occur under a ဝ U+102C it is listed here before the visible virama which will also occur. This slot is only filled if there is a ဝ U+102C and a following visible virama.

### A Asat

### Anusvara

In Mon ဝ U+1032 acts as a final character and so may occur over a ဝ U+102C. In the situation where it occurs after a ဝ U+102D, it is still rendered as a visual ligature with the ဝ U+1032 occurring first as in: ဝ. Different languages use ဝ U+1036 in different ways. ဝ U+1036 here is acting as a final character, in contrast to the same character in the Upper Vowel slot where it is acting as a vowel.

There is one language in which this approach may result in a possible invisible ambiguity and that is Mon. Mon treats anusvara ဝ U+1036 as a final nasal and as such it may follow a ဝ U+102C. In Mon, though, anusvara may also follow ဝ U+102B. But when that happens, it is rendered above the preceding consonant. This may result in two valid sequences ဝ U+1036 ဝ U+102B and ဝ U+102B ဝ U+1036, according to the above table, rendering the same, hence the constraint on the Upper Vowel. Likewise for ဝ U+102F U+1032. The visually identical sequence ဝ U+1032 followed by a Lower Vowel (U+102F or U+1032) is illegal. For more details see the section on Mon.

### Pwo Tone

These are all spacing and may take ဝ U+1037.

### Lower Dot

This lower dot slot position may only be filled when either of the A Vowel or Pwo Tone, spacing slots are filled. It is possible for two ဝ U+1037 to occur. For example, in Pwo Karen: ကာ် U+1000 ဝ U+1060 ဝ U+1037 ဝ U+106B ဝ U+1037. In addition, lower dot may occur after a lower vowel, since lower dot cannot occur in the Karen vowel slot in that context.

### Visible Virama

This is only used if there is a spacing character after the consonant on which the asat is rendered (I.e. something in any of the A Vowel or Pwo Tone slots), or immediately following ဝ U+1037.

## Visarga

The visarga slot not only includes visarga U+1038 but also Shan tone letters. The sequence U+104B U+1038 is also valid and is used as a question mark.

## Reduplication

The reduplication character is found in Khamti Shan. In addition it may ligate with some other characters, but regardless of this ligation, it occurs at the end of the sequence.

## Symbols

The following characters classes do not take part in the diacritic order and as such may not be followed by a diacritic.

Symbol	[U+104A .. U+104D, U+104F, U+109E, U+109F, U+AA77 .. U+AA79]	ၵ
Digits	[U+1090 .. U+1099, U+A9E0 .. U+A9E9]	၀

There are some exceptions to the rule of symbols not being able to take diacritics:

ၵ
104B 1038

## Normalization

The chart shown in this document differs from what one might expect with regard to the relative order of visible virama and lower dot. The normal typing order of these two characters is the visible virama first as part of the final and then the tone mark. But due to an oversight in the standard checking, the combining orders of visible virama and lower dot were set<sup>6</sup> such that any normalization process will order them with the lower dot first, but only when they are stored directly after each other. Thus U+103A U+1037 will always be normalized to U+1037 U+103A.

This makes no difference to keyboard entry and people should still be able to type visible virama before lower dot. But it impacts rendering, searching and sorting. It is best if such processes can handle both orders of encoding U+103A U+1037 and U+1037 U+103A, recognising that after normalization the order will be U+1037 U+103A regardless of the order text was entered.

A common question is whether the uu independent vowel is spelled U+1026 or U+1025 U+102E. According to the Unicode standard, the answer to this question is simple: either. Since the two sequences are canonically equivalent, a process needs to treat them identically.

There are other characters that might be expected to be canonically equivalent to sequences, but that are not. In the following, the two cells in a row are not canonically equivalent. Therefore, users should only use the left hand character (except where the right hand side looks different and you need that particular sequence).

ၵ U+103F ≠ ၵ U+101E U+1039 U+101E  
ၵ U+1029 ≠ ၵ U+101E U+103C  
ၵ U+102A ≠ ၵ U+101E U+103C U+1031 U+102C U+103A  
ၵ U+102A ≠ ၵ U+1029 U+1031 U+102C U+103A

<sup>6</sup> Due to the stability criteria of the Unicode standard, once a combining order is set in the standard, it is impossible to change it for that character. In addition, there is no requirement that normalized order must mirror linguistic order.

## Use of U+104E

One of the significant changes between Unicode 4 and Unicode 5.1 was the change in spelling of lagaun ၵ: changed from U+104E to U+104E U+1004 U+103A U+1038. This is to facilitate an alternative spelling of lagaun of ၵ U+1004 U+103A U+1039 U+104E. This change results in a subtle change of behaviour for U+104E ၵ from being a complete punctuation symbol with corresponding predefined line breaking behaviour, to being just another character that needs algorithmic analysis both for segmentation and for sorting.

## Fractions

A number of legacy fonts have special glyphs for particular fractions. Rather than encoding these with special codes, they can be marked using the U+2044 FRACTIONAL SLASH which is used to build fractions.

## Keyboarding

There are many keyboard layouts for the various languages using Myanmar script. But as yet, there is little standardisation. All that can be said is that certain layouts are becoming de facto standards. Even then, the implementation quality of most keyboards is low. This arises because of the issue that with the complexity of the diacritic order, expecting a user to type in the correct order is unreasonable. Thus there are three particular technical issues that need to be addressed by a keyboard:

- People want to type the prevowels, that are rendered before the initial consonant, before the initial consonant.
- People want to type kinzi after the initial consonant, just as another diacritic, but have it stored before the initial consonant.
- Diacritics typed in the wrong order should be reordered appropriately.

Reordering as people type raises the issue of what happens when one pressed backspace. If pressing backspace merely deletes the last code in the string, one might type one diacritic, have it reordered with another and then press backspace and find the computer deleting a completely different diacritic to the last one typed.

## Advanced Issues

So far we have covered what is explained in the Unicode Standard<sup>7</sup>. In this section we examine some of the more difficult areas of the Myanmar language including some implementation details regarding line breaking, sorting and rendering; further examination of the kinzi question; contractions and some issues with respect to Old Myanmar.

### Line breaking

Burmese does not have inter-word spaces like English. Instead spaces are used to mark phrases. Some phrases are relatively short (two or three syllables, 1.5em, or 2.3 times the width of U+1000 က) while others can be quite long (8.5em or 13 times the width of U+1000 က). A common approach to addressing line breaking issues is to adjust the phrase spacing so that a line breaks at a phrase break. If line breaking is required within a phrase then there are a number of possible approaches. What is presented here is a sliding scale of quality of line breaking approaches, starting with the simplest.

### Insert Zero-Width Spaces

The simplest approach is to insert a U+200B ZERO WIDTH SPACE (ZWSP) between words in a phrase. This would allow line breaks between words in a phrase. The problem is, though, ensuring that ZWSP characters are only inserted between words. The standard approach is to insert ZWSP between syllables, since most words in the languages using the script are monosyllabic. But the problems occur when ZWSP is erroneously inserted into the middle of a polysyllabic word. Such insertions cause problems for searching and indexing. Thus ZWSP should only be used where there is certainty that there is a word break.

### Automated Syllable Breaking

A better approach uses a purely algorithmic approach to line breaking based on syllable breaks. The outline algorithm described here should work for any of the languages using the Myanmar script. A syllable break may occur before any cluster (as described in the diacritic ordering section) so long as the kinzi, asat and stacked slots remain empty in the cluster following the possible break point<sup>8</sup>. In reality such an algorithm requires refinement, but it still requires no dictionary. For example, sequences of digits should be kept together and visible virama needs more complex analysis.

These same syllable breaking rules are used for sorting purposes, with the addition of non-line breaking syllable breaks, such as those occurring between the two characters in a syllable chain. For example these phrases show possible inter-syllable line breaks.

ကောင်လေးတွေကျော	1000 1031 102C 1004 103A   101C 1031 1038	
င်းကိုသွားကြတယ်။	1010 103D 1031   1000 103B 1031 102C	the kids are
	1004 103A 1038   1000 102D 102F   101E	going to
	103D 102C 1038   1000 103C   1010 101A	school
	103A 104B	
အိပ်ခန်းတံခါးကို	1021 102D 1015 103A   1001 1014 103A 1038	to the
	1010 1036   1001 102B 1038   1000 102D	bedroom
	102F	door

Notice how in the second example the word 1010 1036 | 1001 102B 1038 is a single word with multiple syllables. Is there some way, without a dictionary, that we can ensure that the word is not line broken? There is a Unicode character : U+2060 WORD JOINER. The role of this character is to indicate a non-breaking point in a text. Lines should not be broken at that point. Therefore, if we want to ensure that no line-break occurs at the syllable boundary within our polysyllabic word, we can insert a U+2060 into our data stream between the two syllables and a rendering engine should not break a line at that point. Thus:

<sup>7</sup> Version 5.1

<sup>8</sup> This is made more complicated when U+1037 is normalized before U+103A, but a syllable break should still not be allowed.

အိပ်ခန်းတံခါးကို

1021 102D 1015 103A | 1001 1014 103A 1038  
| 1010 1036 2060 1001 102B 1038 | 1000  
102D 102F

to the  
bedroom  
door

The problem with inserting Word Joiners is that it makes searching for polysyllabic words much harder since the searching engine must be able to recognise the Word Joiner characters and ignore them. This is unlikely to happen. Therefore it is advisable not to use Word Joiner characters if at all possible.

### Dictionary Based Line Breaking

The next level of sophistication builds on the previous by adding the ability for the line breaker to identify polysyllabic words. Such words are usually held in a dictionary. Thankfully, such a dictionary only need contain polysyllabic words which are far fewer than a complete word list for a language. The main weakness of this approach is where new words are used that are not in the dictionary. For this, one may need to fall back to ZWSP or WJ approaches. The complexity of this approach is that users are not generally aware of the contents of such dictionaries and so cannot predict when they will have difficulties and when not.

Notice that at each level of sophistication, it is necessary for the line breaking approach to be able to handle data that has been generated for a less sophisticated line breaking approach and to handle that appropriately. For example, if a text contains ZWSP characters, they should be honoured.

### Sorting

Sorting Myanmar strings is a complex process involving significant string transformation and four levels of comparison. The string transformation is a syllable based operation for which the identification of syllable boundaries (but not word boundaries) are required. The same techniques that are used for line-breaking, therefore, may be used for sorting.

The basic principle used in sorting most Myanmar based languages, in the script, is to treat a syllable as consisting of one or more of the following components in order:

*Consonant Medials Vowels Finals Tone*

There are two primary approaches to sorting. The thinbongyi approach is the current national standard and reorders the components so that the Finals occur before the Vowel:

*Consonant Medials Finals Vowels Tone*

The Pali sort uses a different reordering:

*Consonant Medials Vowels Tone Finals*

Then sorting proceeds simply, taking each component as having a primary sort relationship to the other components. It should be noted that where there is more than one medial character, they may interact to produce a single sort key. This is also true for sequences of vowels.

### Sort Tailoring

Where a sort tailoring for a specific language is given, the tailoring may be expressed in two parts. The ordering part, listed second, gives the sort tailoring in terms of ICU rules. The first part, if present, specifies a set of replacements that are conceptually apply to a string before the ordering rules are applied, These replacements form a set of non-feeding substitutions. That is the regular expression for each rule is tested for a string at a given point. The first regular expression that matches has its substitution applied and the search restarts at the beginning of the rules for the character after the last character matched by the regular expression. In reality though, such a rule set may be combined with the sort order itself to create a more complex sort order that does everything needed without preprocessing any strings.

## Contractions

The Myanmar language has a system of double acting consonants, where a consonant acts as both the final of a syllable and the initial of a following syllable. These are significant for sorting purposes. Double acting consonants are rare, but occur in two common words.

ယောက်ျား	101A 1031 102C 1000 <b>103A</b> 103B 102C 1038	man, husband
ကျွန်ုပ်	1000 103B 103D 1014 <b>103A</b> 102F 1015 103A	I (1 <sup>st</sup> person singular)

This storage approach also affects syllable breaking since a devowelised consonant with a vowel acts like a normal base consonant with its preceding syllable break.

There are also words with double acting consonants which are unmarked. Since these are unmarked, it has been decided that despite their etymology, these words should be sorted as if there were no double acting consonant.

ဝါကျ	101D 102B <b>1000</b> 103B	sentence
ဂိဗ္ဗာန်	1002 102D <b>1019</b> 103E 102C 1014 103A	summer

## Contextual Shaping

There are a number of situations in which characters change shape to accommodate diacritics and to avoid glyphs clashing.

န + lower diacritic or medial ra → န့

ည + lower diacritic → ည့

ရ + lower vowel or medial other than medial h → ရ့ (short tail)

ရ + medial h → ရှ (long tail with no hook)

၉ changes width according to the base character being wrapped. It also truncates its top arm if an upper diacritic would clash with it.

၉ + ဝ → ၉

့ and ိ if they would clash with anything under the base character or a tail → ိ and ိ

၉ + medial or asat → ၉. Which means that you never use U+1025 for U+1009

## Diacritic Ordering Implementation

In this section we consider implementation issues regarding the diacritic ordering table. As stated earlier, the diacritic ordering table is not an attempt to specify only those valid sequences that may occur in a writing system. It is much wider than that, allowing many sequences that make no linguistic sense in any writing system. An additional complexity in these implementations is that the invariant required canonical order laid down by Unicode is awkward to work with given that it requires U+1037 MYANMAR SIGN DOT BELOW after U+103A MYANMAR SIGN ASAT, if the two occur together. A more natural order is for the U+103A to occur first. Ensuring the Unicode canonical order adds noticeably to the complexity of implementation.

### Regular Expression

A regular expression that is both minimal<sup>9</sup>, and that covers the diacritic ordering chart is not a trivial beast. The way it is presented here is broken into components which are assembled into a string that can

<sup>9</sup> Minimal here means that for any given string, there is only one path through the regular expression, or when expanded, no string is generated twice.

be compiled into a regular expression. The code is written in Python but could easily be converted to another language. Only the most basic regular expression concepts are used with () not being capturing. There is no use of kline star (\*) or (+) and as such the regular expression can be flattened to a set of strings corresponding to all the strings the expression would completely match. The fact that there are some  $3.5 \times 10^{12}$  strings so generated, is clear evidence that this regular expression is not intended as a poor man's spell checker, but to ensure data order integrity only.

```

1 def e(x) :
2     '''Expand $var in a string'''
3     return re.sub(r'\$([a-z]+)', lambda m: globals()[m.group(1)], x)
4
5 # Basic character classes
6 kinzi = e(ur'([\u1004\u101B\u105A]\u103A\u1039)')
7 cons = e(ur'[\u1000-\u102A\u103F\u1041-\u1049\u104E\u1050-\u1055\u105A-\u105D\u1061\u1065
8 \u1066\u106E\u1070\u1075-\u1081\u108E\u109F\uA9E0-\uA9E4\uA9E7-\uA9EF\uA9FA-\uA9FE\uAA60-
9 \uAA6F\uAA71-\uAA76\uAA7A\uAA7E\uAA7F]')
10 stack = e(ur'(\u1039[\u1000-\u101C\u101E\u1020\u1021\u1050\u1051\u105A-\u105D])')
11 asat = e(ur'\u103A')
12 my = e(ur'[\u103B\u105E\u105F]')
13 mr = e(ur'\u103C')
14 mw = e(ur'[\u103D\u1082]')
15 mwa = e(ur'\u103D')
16 mwb = e(ur'\u1082')
17 mh = e(ur'[\u103E\u1060]')
18 mha = e(ur'\u103E')
19 mhb = e(ur'\u1060')
20 uvowelna = e(ur'[\u102D\u102E\u1033-\u1035\u1071-\u1074\u1085\u109D\uA9E5]')
21 uvowel = e(ur'[\u102D\u102E\u1032-\u1036\u1071-\u1074\u1085\u109D\uA9E5]')
22 lvowel = e(ur'[\u102F\u1030\u1058\u1059]')
23 shan = e(ur'\u1086')
24 avowel = e(ur'[\u102B\u102C\u1056\u1057\u1062\u1063\u1067\u1068\u1083]')
25 anusvara = e(ur'[\u1036\u1032]')
26 pwo = e(ur'[\u1064\u1069-\u106D]')
27 ldot = e(ur'\u1037')
28 visarga = e(ur'[\u1038\u1087-\u108D\u108F\u109A-\u109C\uAA7B-\uAA7D]')
29 redup = e(ur'[\uA9E6\uAA70]')
30 symbol = e(ur'[\u104A-\u104D\u104F\u109E\u109F\uAA77-\uAA79]')
31 digit = e(ur'[\u1090-\u1099\uA9F0-\uA9F9]')
32
33 # Complex Expansions
34 tail = e(ur'($visarga?$redup?)')
35 finals = e(ur'($ldot?$tail)')
36 avowels = e(ur'(' # handle -a and all that follows it
37     ($avowel(($asat?$anusvara$finals)|$ldot?$asat?$tail)) # normal a vowel
38     |(\u102C\u103E\u103A$visarga?$redup?) # mon contraction
39     |((($avowel$anusvara)?$pwo$ldot?$asat?$tail))') # pwo tone
40 ending = e(ur'($tail|$avowels)')
41 uvowels = e(ur'(' # upper vowel sequences
42     (($uvowel\u1062$shan?|$uvowelna$lvowel$shan?)($anusvara?$finals|$avowels))
43     |($anusvara$ldot?$ending) # anusvara acting as vowel
44     |($uvowelna$shan?$anusvara?$finals) # anusvara always before ldot
45     |($uvowelna$shan?$ldot?$avowels))') # ldot occuring early
46 lvowels = e(ur'(($lvowel|\u1062)$shan?$anusvara?($finals|$avowels))')
47 nuvowels = e(ur'(($lvowels|(($shan?|$shan$ldot)$ending))')
48 asatmed = e(ur'(($my$mr?$mw?$mh?|$mr$mw?$mh?|$mw$mh?|$mh))')
49 asats = e(ur'(' # handle medials and asats
50     ($asat$asatmed?) # initial asat
51     |((($mwb$mha?|$mha)$asat) # mon contractions
52     |($my$mr?$mw?$mh?|$mr$mw?$mh?|$mw$mh?|$mh))') # no asat + medial (non-empty)
53 evowels = e(ur'(\u1031{0,2}|\u1084)')
54 myregex = e(ur'(' # syllable start
55     ($asats$evowels($uvowels|$nuvowels)) # no ldot directly after asats
56     |($asat?$asatmed$evowels$ldot$ending) # asat + medials + ldot
57     |($evowels($evowels|$nuvowels|$ldot$ending)) # empty, no medials
58     |($ldot$asat$ending) # ldot + asat
59     |$symbol|$digit|(\u104B\u1038))') # other 'syllables'

```

This expands out to the following regular expression, which is much longer and far less readable than the code to build it:

```

1 ([\u1004\u101b\u105a]\u103a\u1039)?
2 [\u1000-\u102a\u103f\u1041-\u1049\u104e\u1050-\u1055\u105a-\u105d\u1061\u1065\u1066\u106e\u1070\u1075-\u1081\u108e\u109f\ua9e0-\ua9e4\ua9e7-

```





```

61 | |(((\u102b\u102c\u1056\u1057\u1062\u1063\u1067\u1068\u1083][\u1036\u1032]?)?[\u1064\u1069-\u106d]\u1037?\u103a?
62 | (([\u1038\u1087-\u108d\u108f\u109a-\u109c\uaa7b-\uaa7d]?[\ua9e6\uaa70]?) # pwo tone
63 | )))|\u1037(([\u1038\u1087-\u108d\u108f\u109a-\u109c\uaa7b-\uaa7d]?[\ua9e6\uaa70]?)|( # handle -a and all that follows it
64 | ([\u102b\u102c\u1056\u1057\u1062\u1063\u1067\u1068\u1083]((\u103a?[\u1036\u1032]?\u1037?
65 | ([\u1038\u1087-\u108d\u108f\u109a-\u109c\uaa7b-\uaa7d]?[\ua9e6\uaa70]?)?)|\u1037?\u103a?
66 | ([\u1038\u1087-\u108d\u108f\u109a-\u109c\uaa7b-\uaa7d]?[\ua9e6\uaa70]?)?) # normal a vowel
67 | |([\u102c\u103e\u103a][\u1038\u1087-\u108d\u108f\u109a-\u109c\uaa7b-\uaa7d]?[\ua9e6\uaa70]?) # mon contraction
68 | |(((\u102b\u102c\u1056\u1057\u1062\u1063\u1067\u1068\u1083][\u1036\u1032]?)?[\u1064\u1069-\u106d]\u1037?\u103a?
69 | ([\u1038\u1087-\u108d\u108f\u109a-\u109c\uaa7b-\uaa7d]?[\ua9e6\uaa70]?) # pwo tone
70 | ))) # empty, no medials
71 | |([\u1037\u103a]([\u1038\u1087-\u108d\u108f\u109a-\u109c\uaa7b-\uaa7d]?[\ua9e6\uaa70]?)|( # handle -a and all that follows it
72 | ([\u102b\u102c\u1056\u1057\u1062\u1063\u1067\u1068\u1083]((\u103a?[\u1036\u1032]?\u1037?
([\u1038\u1087-\u108d\u108f\u109a-\u109c\uaa7b-\uaa7d]?[\ua9e6\uaa70]?)?)|\u1037?\u103a?
([\u1038\u1087-\u108d\u108f\u109a-\u109c\uaa7b-\uaa7d]?[\ua9e6\uaa70]?)?) # normal a vowel
([\u102c\u103e\u103a][\u1038\u1087-\u108d\u108f\u109a-\u109c\uaa7b-\uaa7d]?[\ua9e6\uaa70]?) # mon contraction
|(((\u102b\u102c\u1056\u1057\u1062\u1063\u1067\u1068\u1083][\u1036\u1032]?)?[\u1064\u1069-\u106d]\u1037?\u103a?
([\u1038\u1087-\u108d\u108f\u109a-\u109c\uaa7b-\uaa7d]?[\ua9e6\uaa70]?) # pwo tone
|))) # ldot + asat
|([\u104a-\u104d\u104f\u109e\u109f\uaa77-\uaa79]|\u1090-\u1099\ua9f0-\ua9f9)|([\u104b\u1038) # other 'syllables'

```

### Canonical Ordering

In addition to testing whether a string is canonical, it would also help to be able to take a string with diacritics stored in some arbitrary order and canonicalise it by reordering the diacritics into the canonical order. This algorithm does this using the following approach.

- Allocate a canonical order based on the character code. Take into account that some sequences may give their components a different ordering to what the individual codes would normally be assigned if not in such a sequence.
- Sort the diacritics following a character with order of 0, by canonical order with lowest first.
- Apply reordering swapping to shift some diacritics based on context.

The only limitation of this algorithm, is that diacritics will not be ordered before characters with a canonical order of 0. This is because if the algorithm is called more than once, and each time a diacritic is moved before a previous base character, that diacritic will walk backwards through the string into completely the wrong syllable cluster. This only affects the kinzi character, which instead of being given a very low diacritic order, is given a very high one to push it away from its preceding base, and just before the following base character where it should be positioned.

Again, the algorithm is presented in Python as being a relatively simple language to understand and translate from.

```

1 | class unitable(object) :
2 |     reorder_class = 3
3 |     reorder = 12
4 |     extending = 16
5 |     seqflag = 32
6 |     orders = ((0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
7 |               0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
8 |               0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 8, 8, 9,
9 |               9, 7, 8, 8, 8, 8, 8, 11, 12, 1, 2, 3, 4, 5, 6, 0,
10 |              0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
11 |              0, 0, 0, 0, 0, 0, 0, 0, 9, 9, 0, 0, 0, 0, 3, 3,
12 |              6, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
13 |              0, 8, 8, 8, 8, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
14 |              0, 0, 5, 0, 7, 8, 10, 12, 12, 12, 12, 12, 12, 0, 12,
15 |              0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 12, 12, 12, 8, 0, 0),
16 |              (0, 0, 0, 0, 0, 8, 13, 0, 0, 0, 0, 0, 0, 0, 0, 0,
17 |              13, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0),
18 |              (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
19 |              0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 12, 12, 12, 0, 0))
20 |     flags = (0, 0, 0, 0, 32, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
21 |              0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 32, 0, 0, 0, 0,
22 |              0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2,
23 |              2, 0, 8, 0, 0, 0, 8, 1, 0, 16, 4, 0, 0, 0, 1, 0,
24 |              0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
25 |              0, 0, 0, 0, 0, 0, 0, 0, 2, 2, 32, 0, 0, 0, 0, 0,
26 |              0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
27 |              0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
28 |              0, 0, 1, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0,
29 |              0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
30 |     seqs = {0x1004: [[2, 0xFF, 0x103A, 0x1039]],
31 |             0x101B: [[2, 0xFF, 0x103A, 0x1039]],
32 |             0x105A: [[2, 0xFF, 0x103A, 0x1039]]}

```

```

33
34 def order(self, num) :
35     if 0x1000 <= num < 0x10A0 :
36         return self.orders[0][num - 0x1000]
37     elif 0xAA60 <= num < 0xAA80 :
38         return self.orders[1][num - 0xAA60]
39     elif 0xA9E0 <= num < 0xAA00 :
40         return self.orders[2][num - 0xA9E0]
41     else :
42         return 0
43
44 def flag(self, num) :
45     if 0x1000 <= num < 0x10A0 :
46         return self.flags[num - 0x1000]
47     return 0
48
49 def get_vals(table, text, index) :
50     num = ord(text[index])
51     order = table.order(num)
52     flags = table.flag(num)
53     length = 1
54     if flags & table.extending :
55         length = 2
56     elif flags & table.seqflag :
57         for r in table.seqs[num] :           # we use a generic lookup here10
58             if r[0] + index > len(text) : continue
59             hit = True
60             for i in range(r[0]) :
61                 if ord(text[index + 1 + i]) != r[2 + i] :
62                     hit = False
63                     break
64             if hit :
65                 length = r[0] + 1
66                 order = r[1]
67     return (order, flags, length)
68
69 def canon_subsort(table, text, orders, flags, start, end) :
70     def canon_cmp(x, y) :
71         if orders[x] == orders[y] :
72             return cmp(x, y)
73         else :
74             return cmp(orders[x], orders[y])
75
76     indices = sorted(range(end - start), cmp=canon_cmp)
77     final = len(indices) - 1
78     i = 0
79     while i < final :
80         f = (flags[indices[i]] & table.reorder) >> 2
81         if f :
82             j = i + 1
83             num = ord(text[start + indices[j]])
84             if j + 1 <= final and text[start + indices[j]] == u'\u1082' and text[start +
indices[j+1]] == u'\u1060' :
85                 i = j + 2
86                 continue
87             while j <= final and f & flags[indices[j]] :
88                 (indices[j-1], indices[j]) = (indices[j], indices[j-1])
89                 j += 1
90             if j > i + 1 and i > 0 :
91                 i -= 2
92             i += 1
93         substr = map(lambda x: text[start + x], indices)
94         return text[:start] + u"".join(substr) + text[end:]
95
96 def canon(table, text) :
97     index = 0
98     while index < len(text) :
99         (order, f, length) = get_vals(table, text, index)
100        if order :

```

<sup>10</sup>A direct test of: if `index + 2 < len(text)` and `text[index+1:index+3] == u'\u103A\u1039'` : `length = 3; order = 0xFF` saves code and complexity in exchange for genericity. A similar tradeoff is made in the opposite direction in line 84.

```

101     start = index
102     flags = [f] * length
103     keys = [order] * length
104     index += length
105     while index < len(text) and order :
106         (order, f, length) = get_vals(table, text, index)
107         keys.extend([order] * length)
108         flags.extend([f] * length)
109         if order : index += length
110     text = canon_subsort(table, text, keys, flags, start, index)
111     index += 1
112     return text

```

The `unitable` class corresponds to a database of information regarding all the Unicode characters. We only consider the ones of interest in this code. The flags associated with each character constitute a bit field. Bit 0 indicates whether this character should be reordered if preceded by a character with bit 2 set. In this case, U+103A has bit 2 set. Likewise bit 1 says whether the character should be reordered if following a character with bit 3 set (U+1032, U+1036). Bit 4 specifies that this character gives both itself and the next character its order and flags (U+1039). Bit 5 says that this character should be tested to see if it is part of a sequence (in this case a kinzi sequence) whereby if the sequence is found, all the characters receive the same order, as specified by the sequence. The `seqs` attribute contains a small database of such strings keyed by initial character codepoint. The database has two query methods to lookup values. All values outside those specified are considered to be 0.

The `get_vals` function at line 49 is used to get the flags and order for a character at a position in the string. It implements the first requirement in that it checks to see if the particular character (U+1039) gives its properties to the following character (lines 54-55). It also checks to see if there is a sequence at this point and if so gives all the characters in the sequence the same values (lines 56-66). So rather than returning the values for one character it returns the two values and the number of characters these values apply to.

The main `canon` function (line 96) goes through the text string collecting contiguous sequences of characters with a non-zero ordering. Each sequence is then passed off to `canon_subsort` (line 69) for reordering. The `canon_subsort` function first declares a comparator function for use with the sort algorithm. This function (lines 70-74) compares the canonical order value with the lowest occurring first. If the values are the same, then the relative order in the original string is compared with lowest occurring first. A list of indices into the string, corresponding to the characters to be reordered, is sorted according to the comparator (line 76). The next loop (lines 79-92) walk through the reordered string looking for characters that may be reordered. If it finds one (line 81), it tests the following characters (lines 87-89), if the next character should reorder with this character, then the two are reordered and the next character is tried. A test is done to see if a particular blocking sequence occurs, for the one sequence where reordering should not occur (line 84). If reordering does occur, the current character we are considering as the primary candidate for reordering is reset to the first reordered character (lines 90-91), so that it too can be retested. This allows characters to propagate to earlier in the string by moving many things after them. Finally the list of indices is mapped against the string to create a reordered substring and the overall string is assembled from the part before the reordered section, the reordered section and the part after (lines 93-94).

Analysis of the regular expression and canonical algorithm reveals some other useful statistics regarding string lengths. The maximum string length matched is 25 characters and the maximum string of diacritics with a non-zero canonical value is 18, which is within the 32 character limit laid down by the Unicode normalization algorithm.

## Languages

This section gives summary descriptions of a number of writing systems that are based on the Myanmar Unicode block. Each description consists of:

A language tag identifying the particular writing system

- Summary of characters in the alphabet, given in alphabetical order
- Unicode encoding for all characters
- Rendering information including standard ligatures and shaping

Where information is omitted about a particular feature of a writing system, it is assumed that the writing system follows Burmese in that respect.

Since there are no standardised keyboard layouts, none are included.

# Burmese မြန်မာစာ

The Burmese language is the primary language that uses the Myanmar script. All other languages are described in terms of it. So where another language does not describe something, it should be assumed to be the same as the Burmese language in that respect.

## Language Tag

my – မြန်မာစာ (မြန်) Burmese

## Alphabet

### Consonants

က	ခ	ဂ	ဃ	င	စ	ဆ	ဇ	ဈ	ည	ဋ	ဌ	ဍ	ဎ	ဏ	
1000	1001	1002	1003	1004	1005	1006	1007	1008	1009	100A	100B	100C	100D	100E	100F

တ	ထ	ဒ	ဓ	န	ပ	ဖ	ဗ	ဘ	မ	ယ	ရ	လ	ဝ	သ	ဟ
1010	1011	1012	1013	1014	1015	1016	1017	1018	1019	101A	101B	101C	101D	101E	101F

ဇ	အ
1020	1021

### Independent Vowels

These sort as if they are အ followed by the corresponding dependent vowel.

က	ဤ	ဥ	ဦ	ဧ	ဩ	ဪ
1023	1024	1025	1026	1027	1029	102A
အိ	အီ	အု	အူ	အေ	အော	အော်
1021 102D	1021 102E	1021 102F	1021 1030	1021 1031	1021 1031 102C	1021 1031 102C 103A

### Medials

ချ	ြ	ဝ်	ု်
103B	103C	103D	103E

In addition to the basic medials, the following is the relative sort order for medial sequences:

ချ	ြ	ချ	ြ	ဝ်	ချ	ြ
103B 103D	103C 103D	103B 103E	103C 103E	103D 103E	103B 103D 103E	103C 103D 103E

### Dependent Vowels

ာ, ဝါ	ိ	ီ	ု	ူ	ေ	ဲ	ော	ို	ော်
102C, 102B	102D	102E	102F	1030	1031	1032	1031 102C	102D 102F	1031 102C 103A

ော်
1031 102B 103A

The relative sort order for ເ is ເ, ເ, ເ. ເ (U+102B MYANMAR VOWEL SIGN TALL AA) is typically used following: ခ, ဂ, င, ဒ, ဓ, ဖ, ဘ, ဟ, ဝ but may also occur following other characters, for example: လါးရှိုးသိန်းအောင်.

### Tones

◌်	◌း
1037	1038

### Final Consonants

Final consonants are those that are marked as having their inherent vowel killed. That is they are consonants that are either followed by a U+103A MYANMAR SIGN ASAT ၵ or they are in a stacking relationship with a following subjoined full consonant, in which case they are followed by U+1039 MYANMAR LETTER VIRAMA. The kinzi character U+1004 MYANMAR LETTER NGA U+103A MYANMAR LETTER ASAT U+1039 MYANMAR LETTER VIRAMA ၵ is stored before the base character it occurs over and is treated as a final consonant of the previous syllable to that base character.

Note that the final ၵ is encoded U+1009 U+1039 and not using U+1025 MYANMAR LETTER U.

### Symbols

The charts show various symbols, how they are encoded and their corresponding sort equivalent sequences.

သ	ဌ	၍	၎း	၏	ံ
103F	104C	104D	104E 1004 103A 1038	104F	1036
ဋ	ဋ်	၎့	လည်းကောင်း	အံ	မ်
101E 1039 101E	1014 103E 102D 102F 1000 103A	101B 103D 1031 1037	101C 100A 103A 1038 1000 1031 102C 1004 103A 1038	1021 102D	1019 103A

Note that U+1036 only acts as a final consonant for sorting purposes in combination with another vowel: ၵ U+102D U+1036 or ၵ U+102F U+1036.

### Sequences

There are a few words involving contractions which ideally sort differently from how they are spelled. A complete list is not included here and processes may sort such words using default character sorting as though they were not special.

ောက်ျ	နံ	လကျ	သို့	ထွင်း	လွက်
1031 102C 1000 103A 103B	1014 103A 102F 1015 103A	101C 1000 103A 103B	101E 1039 1019 102E	1011 1039 1019 1004 103A 1038	101C 1039 1018 1000 103A
ောက်ကျ	နံနံ	လက်ယာ	သမိ	ထမင်း	လက်ဘက်
1031 102C 1000 103A 1000 103B	1014 103A 1014 102F 1015 103A	101C 1000 103A 101A 102C	101E 1019 102E	1011 1019 1004 103A 1038	101C 1000 103A 1018 1000 103A

### Punctuation

I	II
104A	104B

## Rendering

### Subjoined Consonants

Not all consonants have a corresponding subjoined form. In some cases the corresponding medial character is used since a subjoined consonant indicates a new syllable.

က	ခ	ဂ	ဃ	င	စ	ဆ	ဇ	ဈ	ည	ဋ	ဌ	ဍ	ဎ	တ	ထ	ဒ
1039 1000	1039 1001	1039 1002	1039 1003	1039 1004	1039 1005	1039 1006	1039 1007	1039 1008	1039 100A	1039 100C	1039 100D	1039 100E	1039 100F	1039 1010	1039 1011	1039 1012

ဓ	န	ပ	ဖ	ဗ	ဘ	ဗ	ရ	လ	သ	ဒ
1039 1013	1039 1014	1039 1015	1039 1016	1039 1017	1039 1018	1039 1019	1039 101B	1039 101C	1039 101E	1039 1021

### Ligatures

Burmese uses a number of standard ligatures.

ဗု	သ	ဗု	ဗု	ာ်	ဇ
100D 1039 100E	103F	100F 1039 100B	100F 1039 100D	102B 103A	1020 1039 1020

ဗု	ံ
100B 1039 100C	1004 103A 1039

### Variants

Alternate forms of some characters exist:

င
100B

### Examples

Text	Unicode
ကျံ	1000 103B 1036 1037
ကျင့်	1000 103B 1004 1037 103A
ခြင်္သေ့	1001 103C 1004 103A 1039 101E 1031 1037
ညောင်	100A 103E 1031 102C 1004 1037 103A
တိ	1010 102D 1036
နသီ	1014 102F 103F 102E
ဖြေ	1016 103C 102F 1036 1037
ဖြေ	1019 103C 103D 103E 102C
သင်္ဂြိုဟ်	101E 1004 103A 1039 1002 103C 102D 102F 101F 103A
သံ	101E 102F 1036 1037

# Sorting

## Preprocessing

	Regular Expression Match	Replacement
1	(\u1029) (\u1031\u102C)? ((\u1004\u103A [\u1000-\u1021]) ([\u1039\u103A ]))?	\u1021\4\5\u1031\u102C
2	(\u102A) ((\u1004\u103A [\u1000-\u1021]) ([\u1039\u103A]))?	\u1021\3\4\u1031 \u102C\u103A
3	(\u1023) ((\u1004\u103A [\u1000-\u1021]) ([\u1039\u103A]))?	\u1021\3\4\u102D
4	(\u1024) ((\u1004\u103A [\u1000-\u1021]) ([\u1039\u103A]))?	\u1021\3\4\u102E
5	(\u1025) ((\u1004\u103A [\u1000-\u1021]) ([\u1039\u103A]))?	\u1021\3\4\u102F
6	(\u1026) ((\u1004\u103A [\u1000-\u1021]) ([\u1039\u103A]))?	\u1021\3\4\u1030
7	([\u1027\u1028]) ((\u1004\u103A [\u1000-\u1021]) ([\u1039\u103A]))?	\u1021\3\4\u1031
8	(\u1031\u102C \u1031\u102B \u102D\u102F [\u102B-\u102D\u102F\u1031]) (\u1004\u103A [\u1000-\u1021]) ([\u1039\u103A])	\2\3\1

The first 7 replacements are concerned with breaking apart independent vowels when they are followed by a final consonant of some kind. In each case the independent vowel is split into a U+1021 and the corresponding dependent vowel and reordered for sorting such that the final consonant occurs before the vowel.

The final replacement handles the general case and reorders a final consonant to before the vowel sequence.

## Ordering

```

1 &\u1021 < \u108C < \u1037 < \u1038 < \u1037\u1038
2 &\u102C << \u102B
3 &\u1032 < \u1031\u102C( ဝေ ) << \u1031\u102B( ဝေါ ) < \u1031\u102C\u103A( ဝော် ) <<
4 \u1031\u102B\u103A( ဝေါ )
5 &\u1034 < \u1036 < \u102D
6 < \u1000\u1039( က ) << \u1000\u103A( ကံ )
7 < \u1001\u1039( ခ ) << \u1001\u103A( ခံ )
8 < \u1002\u1039( ဂ ) << \u1002\u103A( ဂံ )
9 < \u1003\u1039( ဃ ) << \u1003\u103A( ဃံ )
10 < \u1004\u1039( င ) << \u1004\u103A( ငံ )
11 < \u1005\u1039( စ ) << \u1005\u103A( စံ )
12 < \u1006\u1039( ဆ ) << \u1006\u103A( ဆံ )
13 < \u1007\u1039( ဇ ) << \u1007\u103A( ဇံ )
14 < \u1008\u1039( ဈ ) << \u1008\u103A( ဈံ )
15 < \u1009\u1039( ည ) << \u1009\u103A( ညံ )
16 < \u100A\u1039( သ ) << \u100A\u103A( သံ )
17 < \u100B\u1039( ဋ ) << \u100B\u103A( ဋံ )
18 < \u100C\u1039( ဌ ) << \u100C\u103A( ဌံ )
19 < \u100D\u1039( ဍ ) << \u100D\u103A( ဍံ )
20 < \u100E\u1039( ပ ) << \u100E\u103A( ပံ )
21 < \u100F\u1039( ဖ ) << \u100F\u103A( ဖံ )
22 < \u1010\u1039( တ ) << \u1010\u103A( တံ )
23 < \u1011\u1039( ထ ) << \u1011\u103A( ထံ )
24 < \u1012\u1039( ဒ ) << \u1012\u103A( ဒံ )
25 < \u1013\u1039( ဓ ) << \u1013\u103A( ဓံ )
26 < \u1014\u1039( န ) << \u1014\u103A( နံ )
27 < \u1015\u1039( ဏ ) << \u1015\u103A( ဏံ )
28 < \u1016\u1039( ဖ ) << \u1016\u103A( ဖံ )
29 < \u1017\u1039( ဗ ) << \u1017\u103A( ဗံ )
30 < \u1018\u1039( ဘ ) << \u1018\u103A( ဘံ )
31 < \u1019\u1039( မ ) << \u1019\u103A( မံ )
32 < \u101A\u1039( ယ ) << \u101A\u103A( ယံ )
33 < \u101B\u1039( ရ ) << \u101B\u103A( ရံ )
34 < \u101C\u1039( လ ) << \u101C\u103A( လံ )
35 < \u101D\u1039( ဝ ) << \u101D\u103A( ဝံ )
36 < \u101E\u1039( သ ) << \u101E\u103A( သံ )
37 < \u101F\u1039( ဟ ) << \u101F\u103A( ဟံ )

```



```

37 < \u1020\u1039(င) << \u1020\u103A(ဋ)
38 < \u1021\u1039(အ) << \u1021\u103A(အိ)
39 < \u105E < \u105F < \u103B < \u103C < \u1060 < \u106D < \u1082 < \u103E
40 < \u103B\u103D < \u103C\u103D < \u103B\u103E < \u103C\u103E < \u103D\u103E
41 < \u103B\u103D\u103E < \u103C\u103D\u103E
42
43 # Special finals
44 &\u1019\u103A(မ်) < \u102F\u1036(ံ) / \u102F < \u1019\u103A\u102F(မ်) / \u102F <
\u102F\u1036\u1037(ံ) / \u102F < \u1019\u103A\u102F\u1038(မ်) / \u102F < \u102F\u1036\u1038(ံ) /
\u102F
45 &\u1019\u103A(မ်) < \u102D\u1036(ံ) / \u102D < \u1019\u103A\u102D\u1037(မ်) / \u102D <
\u102D\u1036\u1037(ံ) / \u102D < \u1019\u103A\u102D\u1038(မ်) / \u102D < \u102D\u1036\u1038(ံ) /
\u102D
46 &\u1021(အ) <<< \u1021\u102F\u102F\u1036(အံ) / \u102F\u1036\u102F(ံ)
47 # Contractions
48 &\u103A = \u1000\u103A\u1031\u102C\u103B(ကော) / \u1031\u102C\u1000\u103B(ော)
49 &\u103A = \u1014\u103A\u1015\u103A\u102F(န) /
\u1014\u1015\u103A\u102F\u1014\u1015\u103A\u102F(န) = \u1014\u103A\u102F\u1015\u103A(န) /
\u1014\u1015\u103A\u102F(န)
50 &\u101E\u1039\u101E(ဝ) = \u103F(ဝ)
51 &\u1014\u103E\u1000\u103A\u102D\u102F(န) << \u104C(ွ)
52 &\u101B\u103D\u1031\u1037(ရွ) = \u104D(ွ)
53 &\u101E\u1019\u102E(သ) = \u101E\u1039\u1019\u102E(သ)
54 &\u1011\u1019\u1004\u103A\u1038(စ) = \u101C\u1039\u1018\u1000\u103A(ဝ)
55 &\u101C\u1000\u103A\u1018\u1000\u103A(လ) = \u101C\u1039\u1018\u1000\u103A(ဝ)

```

The starting point for the rules is that the DUCET has the basic consonant and vowel orders already. So this is just a tailoring, if rather a complex one. The rules are written in terms of the output of the preprocessing pass. Thus final consonants are placed before vowels in the input text.

Vowels are sorted before final consonants (lines 1-4) so that syllables without final consonants sort before those with them, There are two types of final consonant: stacked and visibly marked. They are basically the same but stacks store before marked when they both occur for a given consonant (lines 5-38). Medials sort after final consonants so that syllables without them sort before those with them.  $\overset{\circ}{\text{U}}+1036$  ANUSVARA acts like a final  $\text{m}$   $\overset{\circ}{\text{U}}+1019$   $\text{U}+103A$ . The rules (lines 44-46) are more complex than might be expected since the final  $\text{m}$  has been sorted in front of the vowel but the anusvara has not. Various contractions (lines 48-55) sort exactly as their expansions would.

### Keyboarding

The Myanmar3 keyboard layout seems to be emerging as a de facto standard.

~ ဃ	! ည	@ ဟ	# ဋ	\$ ဗ	% ဈ	^ ဧ	& ခ	* *	( (	) )	- -	+ +	← Backspace	
Tab	Q ချ	W ဝ	E က	R င	T ဤ	Y ဋ	U ဉ	I ဤ	O သ	P ဏ	{ ဧ }	[ ဟ ]	↵ Enter	
Caps Lock	A ဃ	S ဟ	D ဝ	F ဝ	G ဝ	H ဝ	J ဝ	K ဝ	L ဝ	: ဝ	" "	' '	↵ Enter	
Shift	Z ဃ	X ဟ	C ဃ	V ဟ	B ဃ	N ဟ	M ဟ	<	>	? ?	/ /	↵ Shift		
Ctrl	Win Key	Alt	SPACE					Alt	Win Key	Menu	Ctrl			

# Old Burmese ရှေးဟောင်း မြန်မာစာ

## Language Tag

obr-Mymr – ရှေးဟောင်း မြန်မာစာ Old Burmese

## Alphabet

The Old Burmese alphabet is identical to that of Burmese for the most part. The only difference occurs in some ligatures and what characters can be subjoined.

### Consonants

က	ခ	ဂ	ဃ	င	စ	ဆ	ဇ	ဈ	ည	ဋ	ဌ	ဍ	ဎ	ဏ
1000	1001	1002	1003	1004	1005	1006	1007	1008	1009	100A	100B	100C	100D	100E 100F

တ	ထ	ဒ	ဓ	န	ပ	ဖ	ဗ	ဘ	မ	ယ	ရ	လ	ဝ	သ	ဟ
1010	1011	1012	1013	1014	1015	1016	1017	1018	1019	101A	101B	101C	101D	101E 101F	

ဇ	အ
1020	1021

### Independent Vowels

These sort as if they are အ followed by the corresponding dependent vowel.

က	က	ဂြ	ဉ	ဦ	ဧ	ဉ	ဩ	ဩ
1022	1023	1024	1025	1026	1027	1028	1029	102A

### Medials

ဗ	ဇ	ဍ	ဎ
103B	103C	103D	103E

Notice that ဗ U+103B ဇ U+103C exists as a sequence, as does ဍ U+103D ဎ U+103E.

### Dependent Vowels

တ/ါ	ိ	ီ	ု	ူ	ေ	ဲ	ေ	ိ
102C/102B	102D	102E	102F	1030	1031	1032	1031 102C	102D 102F

### Tones

့	း
1037	1038

## Rendering

Old Burmese has extra characters that can be part of stacked sequences, than are found in modern Burmese:

ယ	ရ
1039 101A	1039 101B

Old Burmese has a few unique ligatures:

ယ	ယ	ယ	ယ
1051 1039 100C	1051 1039 100D	101B 103A 1039	1039 100B

### Stacked Ya

There are occasions where a medial ya (U+103B) representation is used for a stacking ya. What is needed is a syllable break between the base consonant and the ya.

ဥယျာန်
 
 1025 101A 200C 103B 102C  
 1014
 

 ဥယျာဉ်
 
 garden/orchard

The use of U+200C ZERO WIDTH NON-JOINER indicates the break in the syllable. It makes no difference to rendering and is only used in Pali sorting. U+2060 WORD JOINER cannot be used since it is functionally identical to U+FEFF ZERO WIDTH NON-BREAKING SPACE and so acts as a space character. This would cause a rendering problem with the following diacritic.

# Sanskrit/Pali သက် သက္ကတ/ပါ ပါဠိ

The writing system described here is used for both Sanskrit and Pali.

## Language Tag

pli-Mymr, san-Mymr – ပါ ပါဠိ Pali, သက် သက္ကတ Sanskrit

## Alphabet

### Consonants

က	ခ	ဂ	ဃ	င	စ	ဆ	ဇ	ဈ	ည	ဋ	ဌ	ဍ	ဎ	ဏ
1000	1001	1002	1003	1004	1005	1006	1007	1008	1009	100B	100C	100D	100E	100F

တ	ထ	ဒ	ဓ	န	ပ	ဖ	ဗ	ဘ	မ	ယ	ရ	လ	ဝ	ဓ	ဗ
1010	1011	1012	1013	1014	1015	1016	1017	1018	1019	101A	101B	101C	101D	1050	1051

သ	ဟ	အံ	အာ
101E	101F	1021 1036	1021 1038

### Independent Vowels

အ	အာ	အိ	အု	ဥ	ဋီ	ဣ	ဵ	ဥ	ဧ	ဩ	အဲ	အေ	အော်
1021	1021 102C	1023	1024	1025	1026	1052	1053	1054	1055	1027	1021 1032	1029	102A

### Dependent Vowels

ာ	ိ	ီ	ု	ူ	ာ	ါ	ိ	ီ	ေ	ဲ	ော	ော်
102C	102D	102E	102F	1030	1056	1057	1058	1059	1031	1032	1031 102C	1031 102C 103A

ံ	း
1036	1038

### Conjuncts

Rather than a true medial mechanism, Sanskrit follows the Indic script tradition with the use of conjuncts. Here we list some of them, showing some of the complexities of rendering, but also showing how such conjuncts fit the encoding model naturally.

ကြံ	က္ခ	က္ခိ	က္ခိ	
1000 1039 1010 103C 102D	1000 1039 1010 103D	1004 103A 1039 1000 1039 1010 102D	1005 1039 1005 1032	101B 103A 1039 1010 1039 1010

ကြံ	က္ခဲ	က္ခိ
101B 103A 1039 101F 103C 102E	1000 1039 1051 1032	101B 103A 1039 1050 102D

## Finals

In addition to normal final consonants, Sanskrit has final conjuncts including those that are made up of kinzi or rapha.

ꣳ	ꣳꣳ	ꣳꣳ	ꣳꣳꣳ
1019 1039 1017 103A	1014 1039 1010 103A 103C	1004 103A 1039 1002 103A	101B 103A 1039 1015 103A 103B

## Examples

Text	Unicode
ꣳꣳꣳ	1004 103A 1039 1000 1039 1051 102D
ꣳꣳꣳꣳ	101B 103A 1039 1002 103C
ꣳꣳꣳꣳꣳ	101B 103A 1039 1011 102E
ꣳꣳꣳꣳꣳꣳ	101C 1039 1002 102D
ꣳꣳꣳꣳꣳꣳꣳ	1050 1039 1005 102E
ꣳꣳꣳꣳꣳꣳꣳꣳ	1014 1039 1010 103A
ꣳꣳꣳꣳꣳꣳꣳꣳꣳ	103F 103A

This is a collection of Burmese dialects that have only minor differences from Burmese language and identical writing systems to Burmese:

## **Rakhine** ရခိုင်စာ

Also known as Arakenese. With regard to the writing system, Rakhine follows Burmese very closely. It is also said to be 75% cognate with Burmese. It has all the same features and the same character sequences.

### **Language Tag**

rki-Mymr – (ရ) ရခိုင်စာ Rakhine

## **Tavoyan**

A dialect of non standard Burmese. But it has an identical writing system to Burmese.

### **Language Tag**

tvn-Mymr

## **Intha**

A well known dialect of nonstandard Burmese but using an identical writing system to Burmese.

### **Language Tag**

int-Mymr

# Mon မန်

## Language Tag

mnw-Mymr – မန် (မန်) Mon

## Alphabet

### Consonants

က	ခ	ဂ	ဃ	င	စ	ဆ	ဇ	ဈ	ည	ဋ	ဌ	ဍ	ဎ	ဏ
1000	1001	1002	1003	105A	1005	1006	1007	105B	1009	100A	100B	100C	100D	100E 100F

တ	ထ	ဒ	ဓ	န	ပ	ဖ	ဗ	ဘ	မ	ယ	ရ	လ	ဝ	သ	ဟ
1010	1011	1012	1013	1014	1015	1016	1017	1018	1019	101A	101B	101C	101D	101E 101F	

ဇ	အ	ဓ	ဗ
1020	1021	105C	105D

The nga letter in Mon is encoded U+105A င and not U+1004 င as in Burmese. Independently, these characters look very different. But in the context of something occurring below the character, the Mon nga (U+105A) loses its tail. Thus a Mon kinzi is encoded using U+105A U+103A U+1039. Since this is visually identical to the Burmese kinzi, where confusability is a concern (for example in International Domain Names), the Mon kinzi will not be usable and a misspelling may be necessary to get the desired name. In addition, the medial form of Mon nga is simply the tail: င (U+1039 U+105A).

Mon has a character 'great nya' which is encoded ည U+100A U+1039 U+100A. But this is stylistic and the same sequence may also be rendered ည U+100A U+1039 U+100A.

### Medials

Mon has a number of medial forms even where the characters are not linguistic medials. The specific forms in Mon are:

င	န	မ	ယ	ရ	လ	ဝ	ဟ	ည	ဋ	ဌ
1039 105A	105E 105F	103B 103C	1060 103D	103E 1039 100A	1039 100D	1039 105C				

### Independent Vowels

အ	အ	က	က	ဉ	ဉ	ဉ	ဉ	ဉ	ဉ
1021	1021 102C	1023 1023 1033	1025 1025 102F	1028 1029	102A				

### Dependent Vowels

တ	ဝိ	ဝိ	ု	ု	ေ	ဲ	ေ	ံ	ိ	ိ	ိ	း
102C	102D 1033	102F 1030	1031 1032	1031 102C	1034 102D 102F	1036 1035 102F	1038					

Mon has a sequence U+102C U+1036 ဝံ and correspondingly U+102B U+1036, but here the dot is rendered over the previous consonant: ဝံ and for consistency this is encoded with the dot after the vowel.

ကံ	U+1000 <b>U+1036</b> U+102C U+103A
ကံ	U+1000 U+102C <b>U+1036</b>
ဂံ	U+1002 U+102B <b>U+1036</b>

The ordering of U+1036 U+102C U+103A follows the default encoding order and keeps consistency across the script.

### Contractions

Mon has the concept of final character contractions. One of these is where ဝ် becomes ဝ် on the final character of the syllable. Thus one can have ဝ်. The natural order for these would be U+102C U+103E U+103A following the order of the characters being contracted. But a contraction may also occur before 102C. Thus the following examples are all possible: ဝ် ဝ် ဝ်.

ဝ်	1005 <b>103E 103A</b>
ဝ်	1005 <b>103E 103A</b> 1031
ဝ်	1005 <b>103E</b> 1031 102C <b>103A</b>
ဝ်	1005 <b>103E 103A</b> 1031 102C
ဝ်	1005 1031 102C <b>103E 103A</b>

Likewise with the sequence ဝ်ဝ် which can contract to ဝ်. For example:

ဝ်	105D 102D 102F <b>1032</b>
ဝ်	1013 101D 102F <b>1032</b>

### Rendering

Mon has some extra complex stacking:

ဝ်	1039 1010 103D
----	----------------

### Examples

Text	Unicode
ဝ်	1001 1039 100D 102D 102F 1000 103A
ဝ်	1005 1000 1039 1001 1033
ဝ်	1005 1039 105A 102D 1014 103A
ဝ်	1007 100A 1039 100A 1010 102D
ဝ်	1007 105A 103A 1039 1019 103E 1032 102C
ဝ်	1015 1039 1010 1031 102B 1036
ဝ်	1019 105F 1036
ဝ်	101C 103E 103A
ဝ်	101E 1039 1021 1031 1032
ဝ်	105A 1031 105A 103A
ဝ်	105C 1033
ဝ်	105C 1039 105C



# Sgaw Karen နီ

The Sgaw Karen language is the primary language in the Karen language group. Other languages in the group often base their writing system on this Sgaw Karen writing system. Karen languages have no final consonants. Thus while they may be sorted as any other Myanmar script based language, there is actually no reordering required.

## Language Tag

ksw-Mymr – နီ (နီ) Sgaw Karen

## Alphabet

### Consonants

က	ခ	ဂ	ဃ	င	စ	ဆ	ရှ	ည	တ	ထ	ဒ	န	ပ	ဖ	ဘ
1000	1001	1002	1003	1004	1005	1006	1061	100A	1010	1011	1012	1014	1015	1016	1018

မ	ယ	ရ	လ	ဝ	သ	ဟ	အ	ဇ
1019	101A	101B	101C	101D	101E	101F	1021	1027

U+1061 looks as if it could be encoded as U+101B ရ U+103E န. But since this character occurs as an independent consonant in Sgaw Karen, it has its own code. The two spellings are not equivalent.

### Medials

Sgaw Karen medials have different linguistic values and styling to Burmese. The third row gives the base consonant that the medial represents.

ၚ	ၛ	ၜ	ၝ	ၞ
103E	1060	103B	103C	103D
ဂ	ယ	လ	ရ	ဝ

### Vowels

ါ	ိ	ိ	ု	ူ	ု	ဲ	ိ	ိ
102B	1036	1062	102F	1030	1037	1032	102D	102E

Notice that there is no short form (U+102C) of the -aa vowel.

### Tones

ံ	ံ	း	ံ	ံ
1062 103A	102C 103A	1038	1063 103A	1064

## Ligatures

The following contractions expand as listed and sort according to their expansion:

၆	၆
1012 103A	1019 103A
၆	၆
1012 1036	1019 102E 1064

## Rendering

Sgaw Karen has no subjoined consonants.

One stylistic positioning preference is that U+1037 renders to the left of any lower diacritic. Thus ၆ renders as ၆

U+103E is styled differently to Burmese in that the main stem is angled and the foot is horizontal ၆.

Some older readers of Sgaw Karen like to always use the full height forms of U+102F and U+1030.

## Examples

Text	Unicode
၆	1014 1062 1062 103A
၆	1014 1062 103A
၆	1014 1062 1063 103A
၆	1014 1062 1064
၆	1015 103D 1037 1064
၆	1021 1015 103E 1036 1062 103A

## Sorting

Sgaw Karen has no final consonants, so there is no need for any reordering when sorting.

### Order

```
1 &\u1021 < \u1027
2 < \u1062\u103A < \u102C\u103A < \u1038 < \u1063\u103A < \u1064
3 < \u102B < \u1036 < \u1062 < \u102F < \u1030 < \u1037 < \u1032 < \u102D < \u102E
4 < \u103E < \u1060 < \u103B < \u103C < \u103D
5 &\u1012 < \u1012\u103A / \u1036
6 &\u1019 < \u1019\u103A / \u102E\u1064
```

The basic order is tones follow consonants; vowels follow tones; medials follow vowels (lines 1-4). These are all in primary relation. The two ligatures are again in primary relation with their expansions (lines 5-6).

# Western Pwo Karen ဖျိး

Pwo Karen is based on Sgaw Karen and has many similar behaviours.

## Language Tag

pwo-Mymr – ဖျိး (ဖျိး) Western Pwo Karen

## Alphabet

### Consonants

က	ခ	ဂ	ဃ	င	စ	ဆ	ဇ	ည	ရှ	တ	ထ	ဒ	န	ပ	ဖ
1000	1001	1002	100E	1004	1005	1006	1007	100A	1061	1010	1011	1012	1014	1015	1016

ဆ	မ	ယ	ရ	လ	ဝ	၁	ဟ	အ	ဧ	ပျ
1018	1019	101A	101B	101C	101D	1065	101F	1021	1027	1066

Notice that ပျ has its own code U+1066 and the sequence U+1015 U+103E is not used and constitutes a spelling error.

### Medials

၉	၉	၉	၉	၉
1060	103B	103C	103D	103E

### Vowels

ါ	ံ	့	ဲ	့	ံ	့	ံ	ံ	ံ
102B	1036	1037	1032	1067	1068	102F	1030	102D	102E

### Tones

း	့	း	း	း	း	း	း	း
1069	106A	106B	106C	106D	1069 1037	106B 1037	106A 1037	1038

## Examples

Text	Unicode
ကျ်း	1000 103B 1037 106D
ကြ်း	1000 103C 106B 1037
ကွ်း	1000 103D 1037 106D
ခကျ်း	1001 1067 106A 1037
ပွ်း	100E 103D 1037 1069 1037
ဒွ်း	1012 1060 106B 1037

## Eastern Pwo Karen ပုၤကညီ

This is known as the monastic script and is based on the Mon script. Tone is not marked.

### Language Tag

kjp-Mymr – ပုၤကညီ (ပုၤ) Eastern Pwo Karen

### Alphabet

The sort order for this writing system is unknown.

#### Consonants

The consonants are listed in the corresponding order to pwo-Mymr.

က	ခ	င	စ	ဆ	ည	တ	ထ	ဗ	န	က	ပ	ဖ	ဓ	မ	ယ
1000	1001	1004	1005	1006	100A	1010	1011	100D	1014	106E	1015	1016	105C	1019	101A

ရ	လ	ဝ	ဟ	အ
101B	101C	101D	101F	1021

#### Medials

ဝ်	ၤ	ဝ်	ၤ	ၤ	ၤ	ၤ
103D	1060	103C	103B	103E	1039 1012	1039 101A
ဝ	လ	ရ	ယ	ဟ	ဒ	ယ

#### Vowels

ေ	း	ဲ	တ	်	ိ	ိ	ိ	ိ	ိ	ိ	ိ
1031	1038	1032	102C	103A 102F	105C	102D	102E	1030	102D 102F	102F	1036

#### Finals

The Monastic script follows Mon in supporting some final contractions. Details of how these are encoded is the same as in Mon.

# Pa'o Karen ပအိုဝ်း

## Language Tag

blk-Mymr – ပအိုဝ်း (ပ) Pa'o Karen

## Alphabet

### Consonants

က	ခ	ဂ	ဃ	င	စ	ဆ	ဇ	ဈ	ည	ဋ	ဌ	ဍ	ဎ	တ	
1000	1001	1002	1003	1004	1005	1006	1007	1008	100A	100B	100C	100D	100E	100F	1010

ထ	ဒ	ဓ	န	ပ	ဖ	ဗ	ဘ	မ	ယ	ရ	လ	ဝ	သ	ဟ	ဌ
1011	1012	1013	1014	1015	1016	1017	1018	1019	101A	101B	101C	101D	101E	101F	1020

အ
1021

Pa'o also has stacking consonants and kinzi as in Burmese.

### Medials

ချ	ငြ	ဝံ
103B	103C	103D

### Vowels

တ, ဝါ	ဝိ	ဝီ	ု	ူ	ေ့	ေ	ဲ	ဲ
102C, 102B	102D	102E	102F	1030	1031 0137	1031	1032 1037	1032

ဲငြ	ဲငြ	ေ့	ေ	ိ	ိ
102F 1032 1004 1037 103A	102F 1032 1004 103A	1031 102C 1037	1031 102C 103A	102D 102F 1037	102D 102F

ိ	ိ	ိ	ိ	ိ
1036 1037	1036	102F 1036 1037	102F 1036	102F 1032

U+102F U+1032 and U+102F U+1036 have their orderings because this in order to be consistent across writing systems, we need to follow Mon here. Notice that due to normalization, the order of ဝံ 1037 and ဝိ 103A following the final င U+1004 is counter intuitive.

### Tones

း	း	း
AA7B	1038	108F

## Kayah ကယား

There are no final consonants in Kayah.

## Language Tag

kyu-Mymr – ကယား (ယား) Kayah Li

## Alphabet

Each of the sets of characters are in alphabetic order.

### Consonants

က	ခ	ဃ	င	စ	ဆ	ဇ	ည	တ	ထ	ဒ	န	ပ	ဖ	ဗ	ဘ
1000	1001	1003	1004	1005	1006	1007	100A	1010	1011	1012	1014	1015	1016	1017	1018

မ	ယ	ရ	လ	ဝ	သ	ဟ	အ
1019	101A	101B	101C	101D	101E	101F	1021

### Medials

Kayah uses the two lower vowel characters (U+102F, U+1030) as medials. Following the script order, these two characters are stored following the vowel (excepting U+1032 and U+1036). While this is linguistically inaccurate, it only causes problems during keying and sorting.

့	့	့	့	့	့
102F	1030	103C	103B	103D	103E

### Vowels

The sequence order for the two vowels: U+1032 and U+1036 are that they follow U+102F and U+1030.

ံ	ိ	ိ	ံ	ဲ	ဲ	့	့
1072	102E	102D	1036	1032	1073	1074	1034

### Tones

ံ	း
1064	1038

## Examples

Text	Unicode
ကိကျဲကိး	1000 102E 1064 1000 103B 1074 1030 1000 102E 1064 1004 1038
နဲဆဲး	1005 1072 102F 1006 1073 1064 1012 1030 1038
နဲထွဲ	1005 1072 102F 1011 103D 103E 102D 1038
နဲထဲ	1005 103E 1034 1038 1011 1074 1064
နဲ	1007 103E 1073 1015 103E 1074
တိးတိး	1010 1072 1064 1005 1072 1064 1010 1072 102F 1038
တိး	1010 1036 1012 1036 1064

Text	Unicode
တိုရ်	1010 103E 1072 102F 101B 103E 1074
တဲအို:	1010 103E 1032 1064 1021 1074 1030 1038
ဒံတံ:	1012 1036 1010 103E 1036 1038
ပြု	1015 103C 1072 102F
မံပြုရံပြုတဲ	1019 103E 1074 1064 1015 103C 103E 1064 101B 1036 1015 103C 103E 1064 1010 1032
အိကံကံ:	1021 102D 1064 1000 103E 1072 102F 1000 103E 1072 102F 1038

## Sorting

Being a Karennic language, Kayah has no final consonants and therefore needs no reordering.

### Order

```

1 &\u1021 < \u1064 < \u1038
2 < \u1072 < \u102E < \u102D < \u1036 < \u1032 < \u1073 < \u1074 < \u1034
3 < \u102F < \u1030 < \u103C < \u103B < \u103D < \u103E

```

The basic order of consonants, tones, vowels, medials is used. This ensures appropriate ordering when any of them is missing in a syllable.

## Keyboarding

~	!	@ ]	# ^	\$ %	^ *	& ရ	* ဂ	( )	-	+	←	Backspace	
Tab	Q [	W ဝ	E သ	R ဣ	T အ	Y ဝ	U ဝ	I ဝ	O ဝ	P ဝ	{ }	\	X ÷
Caps Lock	A ဝ	S ဝ	D ဝ	F ဝ	G ဝ	H ဝ	J ဝ	K ဝ	L ဝ	:	"	Enter	
Shift	Z ဝ	X ဝ	C ဝ	V ဝ	B ဝ	N ဝ	M ဝ	< ,	>	?	Shift		
Ctrl	Win Key	Alt	SPACE						Alt	Win Key	Menu	Ctrl	

# Asho Chin အရှား

## Language Tag

csh-Mymr – အရှား (ရှား) Asho Chin

## Alphabet

### Consonants

က	ခ	ဂ	င	စ	ဆ	ဇ	ည	တ	ထ	ဒ	ဓ	န	ပ	ဖ	ဗ
1000	1001	1002	1004	1005	1006	1007	100A	1010	1011	1012	1013	1014	1015	1016	1017

ဘ	မ	ယ	ရ	ရှ	လ	ဝ	ဟ	အ	ဇ
1018	1019	101A	101B	1061	101C	101D	101F	1021	1027

### Medials

့	့	့	ျ
103E	1060	103D	103B

### Vowels

ံ	့	ံ	ဲ	့	ံ	ံ	ံ	ံ	ံ	ံ
1036	1037	1034	1032	1067	1068	102F	1030	102D	102E	1033

### Tones

ံ	့	ံ	ဲ	ံ	ံ
1069	106A	106D	106C	1069 1037	106A 1037

Notice that ဝ U+1037 may occur twice. It may occur as a vowel and also as a tone modifier.

### Digraphs

ံ	ံ	ံ
102D 102B	102E 102B	1038



# Shan လိၵ်းတံး

## Language Tag

shn-Mymr – လိၵ်းတံး (တံး) modern Shan script

## Alphabet

### Consonants

ဂ	ခ	ၣ်	င	လ	သ	ဃ	ဆ	တ	ထ	တဲ	ဆ	ပ	ဖ	ဃ	ဗ
1075	1076	1077	1004	1078	101E	107A	1079	1010	1011	107B	107C	1015	107D	107E	107F

မ	ယ	ရ	လ	ဝ	ဆ	ဂ	က
1019	101A	101B	101C	101D	1080	1081	1022

The character ဂ 1081 is not represented using the sequence U+1002 U+103E and often takes a different visual form. Likewise ဃ 107E is not represented by U+107D U+103E. U+103E does not occur in Shan.

The consonants are listed in alphabetical order. But typically characters: U+1077, U+1079, U+107B, U+107F, U+1080 are not included in the alphabet when it is taught since they are only used for loan words.

### Medials

ချ	ၼ	ဗ
103B	103C	1082

### Vowels

The following are used in open syllables and are listed in alphabetical order.

ၤ	ိ	ီ	ေ	ေ	ု	ူ	ု်	ူ်	ေ်
1083	102D	102E	1031	1084	102F	1030	102F 101D 103A	1030 101D 103A	1031 1083 103A

ေ	ိ်	ိ်	ု်	ု်
1031 1083	102D 102F 101D 103A	102D 1030 101D 103A	1086	107A 103A

ံ	ံၤ	ံၤ	ံၤ	ံၤ	ံၤ
1062 1086	1062 107A 103A	102F 107A 103A	1030 107A 103A	103D 1086	103D 107A 103A

ိ်	ိ်	ါ	ံ	ံၤ
102D 102F 107A 103A	102D 1030 107A 103A	109F	101D 103A	1062 101D 103A

ိ်	ိ်	ိ်	ိ်
102D 101D 103A	1035 101D 103A	1085 101D 103A	102D 102F 101D 103A 101D 103A

ဝိုဝိုဝို	ဝို
102D 1030 101D 103A 101D 103A	1082 103A

The sequence U+1082 U+103A uses the visible virama U+103A to mark the medial wa U+1082 as a final.

ဝိုဝို has the encoding U+103A U+1031 which is counterintuitive, but is necessary because of the complexities surrounding such a sequence in other languages. In Burmese there is the issue of contractions and in Mon there is the issue of final contractions. The conclusion is that Shan integrates best by having this order. Note that U+103A comes before the medial. For example, ဝိုဝိုဝို U+1075 U+103A U+103C U+1031.

The following are used in closed syllables:

ဝို	ဝို	ဝို	ဝို	ဝို	ဝို	ဝို	ဝို	ဝို
1062	102D	1035	1085	102F	1030	103D	102D 102F	102D 1030

Notice that different codes are used for the -a vowel when in an open syllable (U+1083) and a closed syllable (U+1062). If a single key is desired for the one vowel then it is up to the keyboard to make the contextual change rather than storing a single code and using rendering to change shape.

### Finals

All initial consonants (except U+101D and U+107A) may be used as conventional finals if followed by U+103A. In addition, the following consonants may also take a medial U+103B with U+103A. In such cases they follow their unadorned counterparts in the sort order.

ဝို	ဝို	ဝို	ဝို
1075 103A 103B	1076 103A 103B	1077 103A 103B	101E 103A 103B

Notice that while the linguistic order would imply the sequence U+103B U+103A, in order to resolve the ambiguity in the diacritic sequence order that such an order would introduce, it is necessary to store these sequences as U+103A U+103B. This is not a problem so long as keyboard implementations can handle the typing order being reversed and analytical processes make suitable allowance.

### Tones

ဝို	ဝို	ဝို	ဝို	ဝို
1087	1088	1038	1089	108A

### Digits

Shan has its own set of digits, although they are rarely used.

၀	၁	၂	၃	၄	၅	၆	၇	၈	၉
1090	1091	1092	1093	1094	1095	1096	1097	1098	1099

### Symbols

There are two symbols in Shan that are used for standalone words.

ဝို	ဝို
109E	109F

## Old Shan

There are a number of old Shan orthographies, but the traditional script adds one extra character: U+1036 as a final -m. In the modern script this has been replaced by a full final ၵ U+1019 U+103A. The use of the character also introduces one extra sequence:

◌်
103A 1036

which corresponds to ◌် U+102D U+1019 U+103A in the modern script.

## Examples

Text	Unicode
ဂါ,ရှမ်း	1075 1062 1086 1087 1075 1082 1062 1019 103A 1038
ရွှ်ဂိင်း	1075 103D 1086 1075 1035 1004 103A 1088
ဂိခ်ဂူဂ်,တၢ	1075 1085 107C 103A 1081 1030 107A 103A 1087 1010 1083
ဂျ,သၢခ်	1075 103B 1083 1087 101E 1062 107C 103A 1087
ဂျ,ဝိ,မိဝိ	1075 1082 1062 1086 1089 101D 1086 1089 1019 1062 1086 101D 1086 1089
ဗခဗခ	1076 1084 107D 1031 1038
သါလ်	101E 1062 1086 1078 1082 103A
မိခ်ဂိင်း	1019 1085 107C 103A 1038 1081 1085 1004 103A 1038
ဂူဂ်	1081 103D 1086 1004 1030 101D 103A 1089
ဂိတ်	1081 1085 1010 103A 1088
ဂ်းပျး	1081 1082 103A 1088 1015 1031 1083 1038

## Sorting

Rather than using preprocessing and reordering, the Shan collation order is listed in its entirety with every rhyme (vowel final consonant sequence) given its own key.

### Order

```

4 &\u1075 < \u1076 < \u1077 < \u1004 < \u1078 < \u101E < \u107A < \u1079 < \u1010 < \u107B < \u107C
  < \u1015 < \u107D < \u107E < \u107F < \u1019 < \u101A < \u101B < \u101C < \u101D < \u1080 <
  \u1081
5 < \u1083 < \u102D < \u102E < \u1031\u102C\u103A
6 < \u1031 < \u1064 < \u102F < \u1030 < \u102F\u101D\u103A
7 < \u1030\u101D\u103A < \u1031\u1083\u103A < \u1031\u1083 < \u102D\u102F\u101D\u103A
8 < \u102D\u1030\u101D\u103A < \u1086 < \u107A\u103A < \u1062\u1086 < \u1062\u107A\u103A
9 < \u1030\u107A\u103A < \u103D\u1086 < \u103D\u107A\u103A
10 < \u102D\u102F\u107A\u103A < \u102D\u1030\u107A\u103A < \u109F
11 < \u101D\u103A < \u1062\u101D\u103A < \u102D\u101D\u103A < \u1035\u101D\u103A
12 < \u1085\u101D\u103A < \u102D\u102F\u101D\u103A\u101D\u103A
13 < \u102D\u1030\u101D\u103A\u101D\u103A < \u103D\u103A
14
15 < \u1075\u103A < \u1062\u1075\u103A < \u102D\u1075\u103A < \u1035\u1075\u103A
16 < \u1085\u1075\u103A < \u102F\u1075\u103A < \u1030\u1075\u103A < \u103D\u1075 \u103A <
  \u102D\u102F\u1075\u103A < \u102D\u1030\u1075\u103A
17 < \u1075\u103A\u103B < \u1062\u1075\u103A\u103B < \u102D\u1075\u103A\u103B
18 < \u1035\u1075\u103A\u103B < \u1085\u1075\u103A\u103B < \u102F\u1075\u103A\u103B
19 < \u1030\u1075\u103A\u103B < \u103D\u1075\u103A\u103B < \u102D\u102F\u1075\u103A\u103B <
  \u102D\u1030\u1075\u103A\u103B
20 < \u1076\u103A < \u1062\u1076\u103A < \u102D\u1076\u103A < \u1035\u1076\u103A
21 < \u1085\u1076\u103A < \u102F\u1076\u103A < \u1030\u1076\u103A < \u103D\u1076 \u103A <
  \u102D\u102F\u1076\u103A < \u102D\u1030\u1076\u103A
22 < \u1076\u103A\u103B < \u1062\u1076\u103A\u103B < \u102D\u1076\u103A\u103B <
  \u1035\u1076\u103A\u103B < \u1085\u1076\u103A\u103B < \u102F\u1076\u103A\u103B
23 < \u1030\u1076\u103A\u103B < \u103D\u1076\u103A\u103B < \u102D\u102F\u1076\u103A\u103B <
  \u102D\u1030\u1076\u103A\u103B

```

```

24 < \u1077\u103A < \u1062\u1077\u103A < \u102D\u1077\u103A < \u1035\u1077\u103A
25 < \u1085\u1077\u103A < \u102F\u1077\u103A < \u1030\u1077\u103A < \u103D\u1077 \u103A <
\u102D\u102F\u1077\u103A < \u102D\u1030\u1077\u103A
26 < \u1077\u103A\u103B < \u1062\u1077\u103A\u103B < \u102D\u1077\u103A\u103B
27 < \u1035\u1077\u103A\u103B < \u1085\u1077\u103A\u103B < \u102F\u1077\u103A\u103B
28 < \u1030\u1077\u103A\u103B < \u103D\u1077\u103A\u103B < \u102D\u102F\u1077\u103A\u103B <
\u102D\u1030\u1077\u103A\u103B
29 < \u1004\u103A < \u1062\u1004\u103A < \u102D\u1004\u103A < \u1035\u1004\u103A <
\u1085\u1004\u103A < \u102F\u1004\u103A < \u1030\u1004\u103A < \u103D\u1004\u103A <
\u102D\u102F\u1004\u103A < \u102D\u1030\u1004\u103A
30 < \u1078\u103A < \u1062\u1078\u103A < \u102D\u1078\u103A < \u1035\u1078\u103A <
\u1085\u1078\u103A < \u102F\u1078\u103A < \u1030\u1078\u103A < \u103D\u1078\u103A <
\u102D\u102F\u1078\u103A < \u102D\u1030\u1078\u103A
31 < \u101E\u103A < \u1062\u101E\u103A < \u102D\u101E\u103A < \u1035\u101E\u103A <
\u1085\u101E\u103A < \u102F\u101E\u103A < \u1030\u101E\u103A < \u103D\u101E\u103A <
\u102D\u102F\u101E\u103A < \u102D\u1030\u101E\u103A
32 < \u101E\u103A\u103B < \u1062\u101E\u103A\u103B < \u102D\u101E\u103A\u103B <
\u1035\u101E\u103A\u103B < \u1085\u101E\u103A\u103B < \u102F\u101E\u103A\u103B <
\u1030\u101E\u103A\u103B < \u103D\u101E\u103A\u103B < \u102D\u102F\u101E\u103A\u103B <
\u102D\u1030\u101E\u103A\u103B
33 < \u107A\u103A < \u1062\u107A\u103A < \u102D\u107A\u103A < \u1035\u107A\u103A <
\u1085\u107A\u103A < \u102F\u107A\u103A < \u1030\u107A\u103A < \u103D\u107A\u103A <
\u102D\u102F\u107A\u103A < \u102D\u1030\u107A\u103A
34 < \u1079\u103A < \u1062\u1079\u103A < \u102D\u1079\u103A < \u1035\u1079\u103A <
\u1085\u1079\u103A < \u102F\u1079\u103A < \u1030\u1079\u103A < \u103D\u1079\u103A <
\u102D\u102F\u1079\u103A < \u102D\u1030\u1079\u103A
35 < \u1010\u103A < \u1062\u1010\u103A < \u102D\u1010\u103A < \u1035\u1010\u103A <
\u1085\u1010\u103A < \u102F\u1010\u103A < \u1030\u1010\u103A < \u103D\u1010\u103A <
\u102D\u102F\u1010\u103A < \u102D\u1030\u1010\u103A
36 < \u107B\u103A < \u1062\u107B\u103A < \u102D\u107B\u103A < \u1035\u107B\u103A <
\u1085\u107B\u103A < \u102F\u107B\u103A < \u1030\u107B\u103A < \u103D\u107B\u103A <
\u102D\u102F\u107B\u103A < \u102D\u1030\u107B\u103A
37 < \u107C\u103A < \u1062\u107C\u103A < \u102D\u107C\u103A < \u1035\u107C\u103A <
\u1085\u107C\u103A < \u102F\u107C\u103A < \u1030\u107C\u103A < \u103D\u107C\u103A <
\u102D\u102F\u107C\u103A < \u102D\u1030\u107C\u103A
38 < \u1015\u103A < \u1062\u1015\u103A < \u102D\u1015\u103A < \u1035\u1015\u103A <
\u1085\u1015\u103A < \u102F\u1015\u103A < \u1030\u1015\u103A < \u103D\u1015\u103A <
\u102D\u102F\u1015\u103A < \u102D\u1030\u1015\u103A
39 < \u107D\u103A < \u1062\u107D\u103A < \u102D\u107D\u103A < \u1035\u107D\u103A <
\u1085\u107D\u103A < \u102F\u107D\u103A < \u1030\u107D\u103A < \u103D\u107D\u103A <
\u102D\u102F\u107D\u103A < \u102D\u1030\u107D\u103A
40 < \u107E\u103A < \u1062\u107E\u103A < \u102D\u107E\u103A < \u1035\u107E\u103A <
\u1085\u107E\u103A < \u102F\u107E\u103A < \u1030\u107E\u103A < \u103D\u107E\u103A <
\u102D\u102F\u107E\u103A < \u102D\u1030\u107E\u103A
41 < \u107F\u103A < \u1062\u107F\u103A < \u102D\u107F\u103A < \u1035\u107F\u103A <
\u1085\u107F\u103A < \u102F\u107F\u103A < \u1030\u107F\u103A < \u103D\u107F\u103A <
\u102D\u102F\u107F\u103A < \u102D\u1030\u107F\u103A
42 < \u1019\u103A < \u1062\u1019\u103A < \u102D\u1019\u103A < \u1035\u1019\u103A <
\u1085\u1019\u103A < \u102F\u1019\u103A < \u1030\u1019\u103A < \u103D\u1019\u103A <
\u102D\u102F\u1019\u103A < \u102D\u1030\u1019\u103A
43 < \u101A\u103A < \u1062\u101A\u103A < \u102D\u101A\u103A < \u1035\u101A\u103A <
\u1085\u101A\u103A < \u102F\u101A\u103A < \u1030\u101A\u103A < \u103D\u101A\u103A <
\u102D\u102F\u101A\u103A < \u102D\u1030\u101A\u103A
44 < \u101B\u103A < \u1062\u101B\u103A < \u102D\u101B\u103A < \u1035\u101B\u103A <
\u1085\u101B\u103A < \u102F\u101B\u103A < \u1030\u101B\u103A < \u103D\u101B\u103A <
\u102D\u102F\u101B\u103A < \u102D\u1030\u101B\u103A
45 < \u101C\u103A < \u1062\u101C\u103A < \u102D\u101C\u103A < \u1035\u101C\u103A <
\u1085\u101C\u103A < \u102F\u101C\u103A < \u1030\u101C\u103A < \u103D\u101C\u103A <
\u102D\u102F\u101C\u103A < \u102D\u1030\u101C\u103A
46 < \u1080\u103A < \u1062\u1080\u103A < \u102D\u1080\u103A < \u1035\u1080\u103A <
\u1085\u1080\u103A < \u102F\u1080\u103A < \u1030\u1080\u103A < \u103D\u1080\u103A <
\u102D\u102F\u1080\u103A < \u102D\u1030\u1080\u103A
47 < \u1081\u103A < \u1062\u1081\u103A < \u102D\u1081\u103A < \u1035\u1081\u103A <
\u1085\u1081\u103A < \u102F\u1081\u103A < \u1030\u1081\u103A < \u103D\u1081\u103A <
\u102D\u102F\u1081\u103A < \u102D\u1030\u1081\u103A
48 < \u103B < \u103C < \u1082
49 &[last secondary ignorable] << \u1087 << \u1088 << \u1038 << \u1089 << \u108A
50 &\u102D\u107C\u103A < \u102D\u107A\u103A
51 &[before]\u1019\u103A < \u1036
52 &[before]\u1062\u1019\u103A < \u1062\u1036
53 &[before]\u102D\u1019\u103A < \u103A\u1036
54 &[before]\u102F\u1019\u103A < \u102F\u1036
55 &[before]\u1030\u1019\u103A < \u1030\u1036
56 &[before]\u103D\u1019\u103A < \u103D\u1036

```

The sort order begins with the consonant order (line 1). Then follow the open vowels (lines 5-13). Then follow all the final consonants. Shan allows all initial consonants as finals, and it also allows for 4 special

final sequences. It is the interleaving of these special sequences into the general consonant order that has meant that it was easier to generate the complete set of rhymes. For each final consonant we list all the closed vowels in sequence (lines 15-47). Following the rhymes we list the medials (line 48).

Tones are primarily ignorable and so only come into play if all the consonants and vowels are the same. The secondary ordering is given in line 49. Finally we list all the contractions such that the anusvara ၵ U+1036 sorts before the corresponding final m ၶ U+1019 U+103A.

## Keyboarding

There is a growing agreement over a standard layout for a keyboard for typing Shan:

~ ~	! !	@ @	# #	\$ \$	% %	^ ^	& &	* *	( (	) )	- -	+ +	← Backspace	
1 1	2 2	3 3	4 4	5 5	6 6	7 7	8 8	9 9	0 0	- -	= =			
Tab	Q ၵ ၶ	W ၷ ၸ	E ၹ ၺ	R ၻ ၼ	T ၽ ၾ	Y ၿ ႀ	U ႁ ႂ	I ႃ ႄ	O ႅ ႆ	P ႇ ႈ	{ {	} }		
	ၵ ၶ	ၷ ၸ	ၹ ၺ	ၻ ၼ	ၽ ၾ	ၿ ႀ	ႁ ႂ	ႃ ႄ	ႅ ႆ	ႇ ႈ	[ [	] ]	\ \	
Caps Lock	A ႉ ႊ	S ႋ ႌ	D ႍ ႎ	F ႏ ႐	G ႑ ႒	H ႓ ႔	J ႕ ႖	K ႗ ႘	L ႙ ႚ	: ႛ ႜ	" ႝ ႞	Enter	↵	
	ႉ ႊ	ႋ ႌ	ႍ ႎ	ႏ ႐	႑ ႒	႓ ႔	႕ ႖	႗ ႘	႙ ႚ	; ႛ ႜ	' ႝ ႞	↵		
Shift	Z ႛ ႜ	X ႝ ႞	C ႟ Ⴀ	V Ⴁ Ⴂ	B Ⴃ Ⴄ	N Ⴅ Ⴆ	M Ⴇ Ⴈ	< Ⴉ Ⴊ	> Ⴋ Ⴌ	? Ⴍ Ⴎ	Shift	↵		
	ႛ ႜ	ႝ ႞	႟ Ⴀ	Ⴁ Ⴂ	Ⴃ Ⴄ	Ⴅ Ⴆ	Ⴇ Ⴈ	, Ⴉ Ⴊ	. Ⴋ Ⴌ	/ Ⴍ Ⴎ	↵			
Ctrl	Win Key	Alt	SPACE					Alt	Win Key	Menu	Ctrl			

# Khamti Shan လိက်.တဲးဂမ်းတီး

Support for Khamti Shan is added in Unicode 5.2. The most noticeable feature of the writing system is that many characters have a stylistic dot added to them. This dot does not necessarily make them a different character since the dot is only considered to be stylistic rather than a normative part of the character.

## Language Tag

kht-Mymr – လိက်.တဲးဂမ်းတီး (ဂမ်း) Khamti Shan

## Alphabet

### Consonants

က	ခ	ဂ	ဂ	င	ဆ	ဗ	ဆ	ယ	ဗျ	တ	ထ	တ	ဆ	ဆ
1000	1075	AA71	1002	1004	AA61	AA62	AA63	AA64	AA65	AA66	AA67	AA68	AA69	107C

တ	ထ	တ	ထ	ဂ	ပ	လ	ပ	ဆ	မ	ယ	ရ	လ	ဝ	ယ
1010	1011	107B	AA6A	AA6B	1015	1078	107F	1079	1019	101A	101B	101C	101D	AA6C

ဗျ	လ	ဂ	ဝ	လ
AA6D	AA6E	1022	AA6F	1080

### Medials

ချ	ငြ	ဝွ
103B	103C	103D

### Vowels

There are no independent vowels in Khamti Shan

၀	၀	၀	၀	၀	၀	၀	၀	၀	၀	၀	၀	၀
1062	1083	102D	102E	1085	1032	102F	1030	1031	1084	1082	103A	1036

### Tones

၀	၀	၀	၀	၀	၀	၀
109A	1089	109B	1087	1088	1038	108A

Notice that the unmarked tone is in fact tone 7 and is sorted before tone 8 U+108A.

### Digits

Khamti Shan uses the Shan digits.

### Logograms

Khamti Shan has 3 characters which can each take tone but that represent complete syllables

၀	၀	၀
AA74	AA75	AA76

**Finals**

The following consonants may occur with an asat at the end of a syllable. Khamti does not chain syllables.

က	င	ဗျ	တ	န	လ	မ	ဝ
1000	1004	AA65	1010	AA6B	1015	1019	101D

**Reduplication**

The reduplication character is functionally similar to the corresponding character in Thai ๓ U+0E46 THAI CHARACTER MAIMAYOK. It is a spacing character.

◌်
AA70

The reduplication character ligates with two diacritics. This ligation may also occur across a tone mark.

◌်◌်	◌်◌်
1032 AA70	103A AA70

**Historic Khamti Shan**

The Khamti Shan script has undergone script development and as such there are some transition characters that are no longer used. But due to the existence of documents using these characters, they are included in Unicode and sort at the end of the list of consonants in this order.

က	က	က
AA60	AA72	AA73

**Examples**

<b>Text</b>	<b>Unicode</b>
ကိုထွက်.	1000 1082 103A 109B 1011 103D 1000 103A 1089 AA70
ကုမ္ပဏီ.	1000 1083 109B AA70 1019 1085 1015 103A 1089 AA70
ကုပ္ပာနိးနီ	1000 1030 AA65 103A 109B AA62 102E 1038 AA6B 102E
နံ,	1075 1032 1087 AA70
ဆူဝ်း	AA61 1030 101D 103A 1038 AA70
ဆတ်	AA61 101D 103A 109B
မာ်နိတ်	AA62 1062 1032 AA6B 1085 1010 103A
ဗျပ်.	AA65 103D 1015 103A 1089
တပ်.၉၅,	1010 1062 1015 103A 1089 AA6D 1031 1087
နာ်ခင်း	AA6B 1062 1032 AA70 1075 1004 103A 1038 AA70
နု	AA6B 1083
လိမန်း	1078 1082 103A AA70 1019 AA6B 103A 1038 AA70
ဂြိုင်းခွမ်	AA6D 102D 1030 101D 103A 1038 1075 103D 1019 103A 109B
ဖျိး	AA74 108A
ဖျိး	AA74 1038

# Sorting

## PreProcessing

### Regular Expression Match

### Replacement

```
1 ((?:[\u108A\u1062\u1083\u102D\u102E\u1085\u1032\u102F
  \u1030\u1031\u1084\u1036]|\u1082\u103A)?) ([\u1000\u1004
  \uAA65\u1010\uAA6B\u1015\u1019\u101D]) (\u103A)
```

Here we reorder the final consonants before the vowel. Notice that we reorder the final marker before the consonant. This enables the sorting to not have to list explicit final consonants in the order.

## Order

```
1 &\u1000 < \u1075 < \uAA60 < \u1002 < \u1004 < \uAA61 < \uAA62 < \uAA63 < \uAA64 < \uAA65
2 < \uAA66 < \uAA67 < \uAA68 < \uAA69 < \u107C < \u1010 < \u1011 < \u107B < \uAA6A < \uAA6B
3 < \u1015 < \u1078 < \u107F < \u1019 < \u101A < \u101B < \u101C < \u101D < \uAA6C < \uAA6D
4 < \uAA6E < \u1022 < \uAA6F < \u1080 < \uAA74 < \uAA75 < \uAA76
5 &\u108A < \u1062 < \u1083 < \u102D < \u102E < \u1085 < \u1032 < \u102F < \u1030 < \u1031
6 < \u1084 < \u103A\u1082 < \u1036
7 < \u103A
8 < \u103B < \u103C < \u103D
9 &\u109A < \u1089 < \u109B < \u1087 < \u1088 < \u1038
```

Since the basic consonant order is so different for Khamti Shan, we specify it in its entirety (lines 1-4). Vowels are specified following the first in the sequence (lines 5-6). Then we place the final consonant marker (line 7). This has the effect of forcing any following consonant to sort after the vowel rather than before it. The medials come next (line 8) so that their absence sorts earlier. The tones are ordered after their first (line 9).

## Keyboarding

~ ∞	! !	@ @	# #	\$ \$	% %	^ ^	& &	* *	( (	) )	_ ∞	+ +	← Backspace
· ∞	1 1	2 2	3 3	4 4	5 5	6 6	7 7	8 8	9 9	0 0	- -	= =	Tab
Q	W	E	R	T	Y	U	I	O	P	{ [	}	]	\
W	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
Caps Lock	A	S	D	F	G	H	J	K	L	:	"	∞	Enter
↑	GO	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	←
Shift	Z	X	C	V	B	N	M	<	>	?	?	∞	Shift
↑	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
Ctrl	Win Key	Alt	SPACE					Alt	Win Key	Menu	Ctrl		



# Aiton & Phake ကဲတွံင် & လျဲကဲ

Aiton and Phake are closely related languages with nearly identical orthographies. The differences are purely stylistic. Aiton and Phake have their own font styles that are related to Khamti Shan but different. The style used here is Khamti Shan and not Aiton or Phake.

## Language Tag

aio-Mymr, phk-Mymr – ကဲတွံင် (ကဲ) Aiton, လျဲကဲ (လျဲ) Phake

## Alphabet

### Consonants

က	ခ	င	ဆ	ဇ	တ	ထ	န	ပ	လ	မ	ယ	ဝ	လ	ဝ
1000	1075	1004	AA61	107A	1010	1011	AA6B	1015	1078	1019	101A	AA7A	101C	101D

၂	က
AA6D	1022

### Medials

၂	၂	၂
103B	103C	105E

### Subjoined Consonants

Aiton follows Burmese in using subjoined consonants to chain syllables in a polysyllabic word. The following subjoined characters exist:

က	ခ	င	ဆ	ဇ	တ	ထ	န	ပ	လ	မ	ယ	ဝ
1039 1000	1039 AA60	1039 1010	1039 1011	1039 1015	1039 101A	1039 101C						

### Vowels

These are final vowels that have no following consonant.

၂	၂	၂	၂	၂	၂	၂	၂
1083	109C	102E	1030	1031	1031 1083	102F 101D 103A	102D 102F 101D 103A

These vowels are followed by a final consonant.

၂	၂	၂	၂
102D	102F	103D	102D 102F

### Diphthongs

၂	၂	၂	၂	၂	၂	၂	၂
1036	103A 1036	109D	103D 109D	103D 1031	102D 102F 109C	103A 103D	103A 105E

## Ligatures

The following ligatures do not take diacritics, but are considered as words.

ꨆ	ꨇ	ꨈ
AA77	AA78	AA79

# Tai Laing - တိုင်းလျမ် ခေါ်ရှမ်းနီ

## Language Tag

tle-Mymr

## Alphabet

### Consonants

က	ဂ	ဣ	ဣ	င	ဇ	ဇ	ဇ	ဇ	ဇ	ဇ	ဇ	ဇ	ဇ	ဇ
1000	1075	A9E9	A9EA	1004	1078	AA6C	A9EB	A9EC	A9E7	AA66	AA67	A9ED	A9EE	A9EF

တ	ထ	တ	ထ	ဂ	ပ	င	မ	င	င	မ	ယ	ဇ	လ	လ
1010	1011	A9FB	A9FC	AA6B	1015	A9E4	A9FD	A9FE	A9E8	1019	101A	AA7A	A9FA	101C

ဝ	ဇ	ဟ	က
101D	A9EC	106F	1022

Cells marked in grey indicate historical characters, primarily used for Pali. An additional final -m U+1036 MYANMAR SIGN ANUSVARA also occurs in historic texts.

### Medials

ဂ	ဇ	ဇ
103B	103C	1082

### Vowels

၀	၀	၀	၀	၀	၀	၀	၀	၀	၀	၀	၀					
1083	102D	102E	1031	1031	A9E5	102F	1030	1030	101D	103A	1031	1083	102D	102F	101D	103A

၀	၀	၀				
102D	1030	101D	103A	1086	1082	A9E5

### Tones

၀	၀	၀	၀	၀
108D	AA7C	1038	1089	AA7D

When AA7C MYANMAR SIGN TAI LAING TONE-2 occurs following an upper diacritic vowel mark, the tone is rendered to the left of the vowel. Likewise if 108D MYANMAR SIGN SHAN COUNCIL EMPHATIC TONE occurs following a lower diacritic (usually a medial), the tone mark renders to the left of the diacritic.

## Examples

Text	Unicode
ကိုင်းကိုင်း	1022 102D 102F 101D 1037 103A 1022 102D 1030 108D 101D 103A
လှိုင်မိုးမိုး-	101C 102D 1030 101D 1037 103A 1019 1030 AA7C 1019 102D 1030 101D 103A AA7D
ကိစ္စတီ-ထံ	1000 1086 1077 1086 1010 102E AA7D 1011 1086 108D
ပိ-မုဇ်လှိုင်-	1015 102E AA7D 1019 1086 108D A9E8 1086 108D 101C 1030 AA7D
ခေါ်	1022 1031 A9E5 AA7C

## Keyboarding

~ ~	! !	@ @	# #	\$ \$	% %	^ ^	& ရ	* ဂ	( (	) )	- -	+ -	←	Backspace	
1 ရှိ	2 ဗ	3 ဃ	4 င	5 ဖ	6 ဖ	7 ရ	8 င	9 ယ	0 ဝ	- -	= ဝ				
Tab	Q ခ	W ဝ	E ရာ	R ဓာ	T ဂ	Y ဂ	U ဃ	I ဃ	O ဝ	P ဖ	{ {	} }	' '		
	ခ	တ	ခ	မ	အ	ပ	က	င	သ	လ	[ ဝ	] }	' '		
Caps Lock	A ခ	S ဂ	D ဝ	F ဝ	G ဝ	H ဝ	J ဝ	K ဝ	L ဝ	: ဝ	" ဝ	Enter			
	ခ	ဂ	ဝ	ဝ	ဝ	ဝ	ဝ	ဝ	ဝ	: ဝ	" ဝ	↵			
Shift	Z ခ	X ဝ	C ဝ	V ဝ	B ဝ	N ဝ	M ဝ	< ဝ	> ဝ	? ဝ	Shift				
	ခ	ဝ	ခ	လ	ဝ	ဝ	ခ	, ဝ	. ဝ	/	↵				
Ctrl	Win Key	Alt	SPACE					Alt	Win Key	Menu	Ctrl				

# Shwe Palaung - ရွှေပလောင်

## Language Tag

pll-Mymr

## Alphabet

### Consonants

က	ခ	ချ	ဂ	င	စ	ဆ	ဆျ	ဇ	ဈ	ည	တ	ထ	ဒ	န
1000	1001	AA7E	1002	1004	1005	1006	AA7F	1007	1008	100A	1010	1011	1012	1014

ပ	ဖ	ဘ	မ	ယ	ရ	လ	ဝ	ဓ	ဓ့	သ	ဟ	အ
1015	1016	1018	1019	101A	101B	101C	101D	108E	108E 103E	101E	101F	1021

### Medials

ဈ	ဇြ	ဝ္	ဂ္	လ္
103B	103C	103D	103E	1039 101C

The following medial sequences also occur: U+103B U+103C, U+1039 U+101C U+103B, U+1039 U+101C U+103D, U+103B U+103D. Note that the consonant U+108E U+103E may not be followed by a medial, and that the consonants U+AA7E and U+AA7F may only be followed by U+103D.

### Finals

ကံ	ငံ	ငံး	ညံ	တံ	နံ	ပ်
1000 103A	1004 103A	1004 103A 1038	100A 103A	1010 103A	1014 103A	1015 103A

မ်	ယ်	ရံ	ရံး	ဝံ
1019 103A	101A 103A	101B 103A	101B 103A 1088	101D 103A

Notice that both U+100A U+103A and U+101D U+103A are linguistically open syllables. But for textual purposes they are considered closed. U+101D U+103A also occurs in the open syllable vowel list due to those contexts taking different tones.

### Vowels

The following chart lists all open syllable vowel and tone sequences.

ဝ	တ	ဝါ	တ,	ဝါ,	တး	ဝါး	တး	ဝါး
	102C	102B	102C 1087	102B 1087	102C 1038	102B 1038	102C 1088	102B 1088

တး	ဝါး	ဝိ	ဝီ	ဝီ,	ဝီး	ဝီး	ဝီး	ု
102c 108F	102B 108F	102D	102E	102E 1087	102E 1038	102E 1088	102E 108F	102F

၀	၀,	၀:	၀:	၀:	့	့	့,	့:
1030	1030 1087	1030 1038	1030 1088	1030 108F	1031 1037	1031	1031 1087	1031 1038

့:	့:	့	့	့,	့:	့:	့:
1031 1088	1031 108F	1032 1037	1032	1032 1087	1032 1038	1032 1088	1032 108F

့	့	့	့,	့:
1031 102C 1037	1031 102C	1031 102C 103A	1031 102C 1087	1031 102C 1088

့:	့	့	့,	့:
1031 102C 108F	102D 102F 1037	102D 102F	102D 102F 1087	102D 102F 1038

့:	့:	့	့
102D 102F 1088	102D 102F 108F	102E 102F 101D 103A 1037	102E 102F 101D 103A

့,	့:	့
102E 102F 101D 103A 1087	102E 102F 101D 103A 1088	102D 1030 101D 103A 1037

့	့,	့:
102D 1030 101D 103A	102D 1030 101D 103A 1087	102D 1030 101D 103A 1088

့	့	့,	့:
103A 102C 1037	103A 102C	103A 102C 1087	103A 102C 1088

The following lists all the vowel sequences that can occur within a closed syllable, preceding one of the final consonants.

့	့	့/့	့	့	့	့	့	့	့	့
	1036	102C/102B	102D	102E	102F	1030	1031	1032	1031 1032	1031 102C

့	့	့	့	့	့	့
102D 102F	102E 102F	102E 102F 1037	102D 1030	103A 102C	103A 102F	1034

One ordering issue revolves around the vowel ိ which may be broken by a medial U+103E. See the ordering rule regarding Mon Asat.

# Keyboarding

~ `	1 ! ၁	2 @ ၂	3 # ၃	4 \$ ၄	5 % ၅	6 ^ ၆	7 & ၇	8 * ၈	9 ( ၉	0 ) ၀	- _ -	+ =	Backspace
Tab	Q ၁	W ၂	E ၃	R ၄	T ၅	Y ၆	U ၇	I ၈	O ၉	P ၀	[ ၁	] ၂	' ၃   ၄
Caps Lock	A ၁	S ၂	D ၃	F ၄	G ၅	H ၆	J ၇	K ၈	L ၉	: ၀	" ၁	Enter	
Shift	Z ၁	X ၂	C ၃	V ၄	B ၅	N ၆	M ၇	< ၈	> ၉	? ၀	/ ၁	Shift	
Ctrl	Win Key	Alt	SPACE						Alt	Win Key	Menu	Ctrl	

# Pale Palaung - ပလေး

## Language Tag

pce-Mymr

## Alphabet

### Consonants

က	ခ	ဂ	င	စ	ဇ	ဆ	ဇ	ည	တ	ထ	ဒ	န	ပ	ဖ
1000	1001	1002	1004	1005	1005 103E	1006	1007	100A	1010	1011	1012	1014	1015	1016

ဘ	မ	ယ	ရ	လ	ဝ	ဟ	အ	၎
1018	1019	101A	101B	101C	101D	101F	1021	101D 103E

### Medials

ချ	ငြ	ဝွ	၎	လ
103B	103C	103D	103E	101C

Note that လ is fully spacing.

### Vowels

The following is a list of vowels used in open syllables, not followed by a final consonant.

၎	ထ	ဝီ	ူ	၎	ေ	ဲ	ေ	ွ
103A 102F	102C	102E	1030	102F 101D 103A	1031	1032	1031 102C	102D 102F 101D 103A

ွ	ူ	ွ	ွ	ွ:
102D 1030 101D 103A	103B 102E	103D 102C	101D 103A	101D 103A 1038

ွ	ွ	ွ	ွ
101D 1037 103A	103D 101A 1037 103A	103D 1004 1037 103A	103D 101A 103A

ွ	ွ:	ွ	ွ
102D 102F 101D 1037 103A	102D 102F 101D 1038	1030 1000 1037 103A	1030 1004 1037 103A

The following is a list of vowels used in closed syllables and require a following final consonant.

ဝ	ထ	ဝီ	၎	ူ	ေ	ဲ	ဝွ	ဝီ	ဝီ
	102C	102D	102F	1030	1031	1032	103D	102D 102F	102D 1030



ꨀ	ꨁ	ꨂ
103B 102E	103D 102F	103D 102D 102F

**Final Consonants**

က	င	တ	တံ	န	နံ	ပ
1000 103A	1004 103A	1010 103A	1010 1037 103A	1014 103A	1014 1037 103A	1015 103A

မ	ယ	ဝ	့	း
1019 103A	101A 103A	101D 103A	1037	1038

**PreSyllables**

There are two presyllabic nasals each marked with ꨀ U+1037. We list each presyllabic nasal followed by its possible main syllable initial consonant.

န့	န့	န့	န့	န့
1014 1037 1005	1014 1037 1005 103E	1014 1037 1006	1014 1037 1007	1014 1037 1010

န့	န့	မ့	မ့	မ့
1014 1037 1011	1014 1037 1012	1019 1037 1015	1019 1037 1016	1019 1037 1018

**Examples**

Text	Unicode
ကာလံ	1000 102C 101C 101D 103A 1037
ကုရေန်	1000 1030 101B 1031 1014 103A 1037
ကွဲဟူမ်	1000 103D 1032 1037 101F 1030 1019 103A
ခြံ	1001 103B 103C 102E 1037
ဒီရိုင်	1012 102E 101B 103B 102E 1004 103A
ညွတ်	100A 103D 103E 1010 103A
န့တေန်	1014 1037 1010 1031 1014 103A 1037
မ့ဖိုင်	1019 1037 1016 103D 102F 1014 103A

# Rumai Palaung ရှုမည်းတအာင်း

## Language Tag

rbb-Mymr – ရှုမည်းတအာင်း (တ) Rumai Palaung

## Alphabet

### Consonants

က	ခ	ဂ	င	စ	ဆ	ဇ	ည	တ	ထ	ဒ	န	ပ	ဖ	ဘ	မ
1000	1001	1002	1004	1005	1006	1007	100A	1010	1011	1012	1014	1015	1016	1018	1019

ယ	ရ	လ	ဓ	ဝ	ဟ	အ
101A	101B	101C	108E	101D	101F	1021

### Medials

ချ	ြ	ဝ်	ရွ	့	ျ	လ်
103B	103C	103D	103B 103D	103E	103B 103E	1039 101C

### Vowels

တ	ဝိ	ဝီ	ု	ူ	ေ	ဲ	ဲ	ေ	ိ	ိ
102C	102D	102E	102F	1030	1031	1032	1031 1032	1031 102C	102D 102F	102C 103A

### Tones

း	း	း	း
1038	1089	1088	108F

	100	101	102	103	104	105	106	107	108	109
0	က	တ	ဇ	ူ	ဝ	ဓ	ံ	ဃ	ဆ	ဝ
1	ခ	ထ	အ	ေ	၁	မ	ရ	ံ	၂	၁
2	ဂ	ဒ	က	ံ	၂	ဖ	ံ	ံ	ံ	၃
3	ဃ	စ	ဒ	ံ	၃	ဗ	ံ	ံ	ံ	၄
4	င	န	ြ	ံ	၄	ဇ	ံ	ံ	ံ	၅
5	စ	ပ	လ	ံ	၅	ဇ	၁	၈	ံ	၆
6	ဆ	ဖ	လ	ံ	၆	၈	၂	ခ	ံ	၇
7	ဇ	ဗ	ဇ	ံ	၇	၈	ံ	၈	ံ	၈
8	ဈ	ဘ	ဇ	ံ	၈	ံ	ံ	ံ	ံ	၉
9	ည	မ	ြ	ံ	၉	ံ	ံ	ံ	ံ	၉
A	ည	ယ	ြ	ံ	၁	ံ	ံ	ံ	ံ	ံ
B	ဋ	ရ	ံ	ံ	၂	ံ	ံ	ံ	ံ	ံ
C	ဌ	လ	ံ	ံ	၃	ံ	ံ	ံ	ံ	ံ
D	ဍ	ဝ	ံ	ံ	၄	ံ	ံ	ံ	ံ	ံ
E	ဎ	သ	ံ	ံ	၅	ံ	ံ	ံ	ံ	ံ
F	ဏ	ဟ	ံ	ံ	၆	ံ	ံ	ံ	ံ	ံ

## Consonants

1000	MYANMAR LETTER KA
1001	MYANMAR LETTER KHA
1002	MYANMAR LETTER GA
1003	MYANMAR LETTER GHA
1004	MYANMAR LETTER NGA
1005	MYANMAR LETTER CA
1006	MYANMAR LETTER CHA
1007	MYANMAR LETTER JA
1008	MYANMAR LETTER JHA
1009	MYANMAR LETTER NYA
100A	MYANMAR LETTER NNYA
100B	MYANMAR LETTER TTA
100C	MYANMAR LETTER TTHA
100D	MYANMAR LETTER DDA
100E	MYANMAR LETTER DDHA
100F	MYANMAR LETTER NNA
1010	MYANMAR LETTER TA
1011	MYANMAR LETTER THA
1012	MYANMAR LETTER DA
1013	MYANMAR LETTER DHA
1014	MYANMAR LETTER NA
1015	MYANMAR LETTER PA
1016	MYANMAR LETTER PHA
1017	MYANMAR LETTER BA
1018	MYANMAR LETTER BHA
1019	MYANMAR LETTER MA
101A	MYANMAR LETTER YA
101B	MYANMAR LETTER RA
101C	MYANMAR LETTER LA
101D	MYANMAR LETTER WA
101E	MYANMAR LETTER SA
101F	MYANMAR LETTER HA
1020	MYANMAR LETTER LLA

## Independent vowels

1021	MYANMAR LETTER A • also represents the glottal stop as a consonant
1022	MYANMAR LETTER SHAN A
1023	MYANMAR LETTER I
1024	MYANMAR LETTER II
1025	MYANMAR LETTER U
1026	MYANMAR LETTER UU ≡ 1025 ဝ 102E ဝ
1027	MYANMAR LETTER E
1028	MYANMAR LETTER MON E
1029	MYANMAR LETTER O
102A	MYANMAR LETTER AU

## Dependent vowel signs

102B	MYANMAR VOWEL SIGN TALL AA
102C	MYANMAR VOWEL SIGN AA
102D	MYANMAR VOWEL SIGN I
102E	MYANMAR VOWEL SIGN II
102F	MYANMAR VOWEL SIGN U
1030	MYANMAR VOWEL SIGN UU
1031	MYANMAR VOWEL SIGN E • stands to the left of the consonant
1032	MYANMAR VOWEL SIGN AI
1033	MYANMAR VOWEL SIGN MON II
1034	MYANMAR VOWEL SIGN MON O
1035	MYANMAR VOWEL SIGN E ABOVE

## Various signs

1036	MYANMAR SIGN ANUSVARA
1037	MYANMAR SIGN DOT BELOW = aukmyit • a tone mark
1038	MYANMAR SIGN VISARGA
1039	MYANMAR SIGN VIRAMA = killer (when rendered visibly)
103A	MYANMAR SIGN ASAT = killer (always rendered visibly)

## Dependent consonant signs

103B	MYANMAR CONSONANT SIGN MEDIAL YA
103C	MYANMAR CONSONANT SIGN MEDIAL RA
103D	MYANMAR CONSONANT SIGN MEDIAL WA
103E	MYANMAR CONSONANT SIGN MEDIAL HA

## Consonant

103F	MYANMAR LETTER GREAT SA
------	-------------------------

## Digits

1040	MYANMAR DIGIT ZERO
1041	MYANMAR DIGIT ONE
1042	MYANMAR DIGIT TWO
1043	MYANMAR DIGIT THREE
1044	MYANMAR DIGIT FOUR
1045	MYANMAR DIGIT FIVE
1046	MYANMAR DIGIT SIX
1047	MYANMAR DIGIT SEVEN
1048	MYANMAR DIGIT EIGHT
1049	MYANMAR DIGIT NINE

## Punctuation

104A	MYANMAR SIGN LITTLE SECTION → (devanagari danda - 0964)
104B	MYANMAR SIGN SECTION → (devanagari double danda - 0965)

## Various signs

104C	MYANMAR SYMBOL LOCATIVE
104D	MYANMAR SYMBOL COMPLETED
104E	MYANMAR SYMBOL AFOREMENTIONED
104F	MYANMAR SYMBOL GENITIVE

## Pali and Sanskrit extensions

1050	MYANMAR LETTER SHA
1051	MYANMAR LETTER SSA
1052	MYANMAR LETTER VOCALIC R
1053	MYANMAR LETTER VOCALIC RR
1054	MYANMAR LETTER VOCALIC L
1055	MYANMAR LETTER VOCALIC LL
1056	MYANMAR VOWEL SIGN VOCALIC R
1057	MYANMAR VOWEL SIGN VOCALIC RR
1058	MYANMAR VOWEL SIGN VOCALIC L
1059	MYANMAR VOWEL SIGN VOCALIC LL

## Extensions for Mon

105A	MYANMAR LETTER MON NGA
105B	MYANMAR LETTER MON JHA

105C MYANMAR LETTER MON BBA  
105D MYANMAR LETTER MON BBE  
105E MYANMAR CONSONANT SIGN MON  
MEDIAL NA  
105F MYANMAR CONSONANT SIGN MON  
MEDIAL MA  
1060 MYANMAR CONSONANT SIGN MON  
MEDIAL LA

#### Extensions for S'gaw Karen

1061 MYANMAR LETTER SGAW KAREN SHA  
1062 MYANMAR VOWEL SIGN SGAW KAREN  
EU  
1063 MYANMAR TONE MARK SGAW KAREN  
HATHI  
1064 MYANMAR TONE MARK SGAW KAREN  
KE PHO

#### Extensions for Western Pwo Karen

1065 MYANMAR LETTER WESTERN PWO  
KAREN THA  
1066 MYANMAR LETTER WESTERN PWO  
KAREN PWA  
1067 MYANMAR VOWEL SIGN WESTERN PWO  
KAREN EU  
1068 MYANMAR VOWEL SIGN WESTERN PWO  
KAREN UE  
1069 MYANMAR SIGN WESTERN PWO KAREN  
TONE-1  
106A MYANMAR SIGN WESTERN PWO KAREN  
TONE-2  
106B MYANMAR SIGN WESTERN PWO KAREN  
TONE-3  
106C MYANMAR SIGN WESTERN PWO KAREN  
TONE-4  
106D MYANMAR SIGN WESTERN PWO KAREN  
TONE-5

#### Extensions for Eastern Pwo Karen

106E MYANMAR LETTER EASTERN PWO  
KAREN NNA  
106F MYANMAR LETTER EASTERN PWO  
KAREN YWA  
1070 MYANMAR LETTER EASTERN PWO  
KAREN GHWA

#### Extension for Geba Karen

1071 MYANMAR VOWEL SIGN GEBA KAREN I

#### Extensions for Kayah

1072 MYANMAR VOWEL SIGN KAYAH OE  
1073 MYANMAR VOWEL SIGN KAYAH U  
1074 MYANMAR VOWEL SIGN KAYAH EE

#### Extensions for Shan

1075 MYANMAR LETTER SHAN KA  
1076 MYANMAR LETTER SHAN KHA

1077 MYANMAR LETTER SHAN GA  
1078 MYANMAR LETTER SHAN CA  
1079 MYANMAR LETTER SHAN ZA  
107A MYANMAR LETTER SHAN NYA  
107B MYANMAR LETTER SHAN DA  
107C MYANMAR LETTER SHAN NA  
107D MYANMAR LETTER SHAN PHA  
107E MYANMAR LETTER SHAN FA  
107F MYANMAR LETTER SHAN BA  
1080 MYANMAR LETTER SHAN THA  
1081 MYANMAR LETTER SHAN HA  
1082 MYANMAR CONSONANT SIGN SHAN  
MEDIAL WA  
1083 MYANMAR VOWEL SIGN SHAN AA  
1084 MYANMAR VOWEL SIGN SHAN E  
1085 MYANMAR VOWEL SIGN SHAN E ABOVE  
1086 MYANMAR VOWEL SIGN SHAN FINAL Y  
1087 MYANMAR SIGN SHAN TONE-2  
1088 MYANMAR SIGN SHAN TONE-3  
1089 MYANMAR SIGN SHAN TONE-5  
108A MYANMAR SIGN SHAN TONE-6  
108B MYANMAR SIGN SHAN COUNCIL TONE-2  
108C MYANMAR SIGN SHAN COUNCIL TONE-3  
108D MYANMAR SIGN SHAN COUNCIL  
EMPHATIC TONE

#### Extensions for Rumai Palaung

108E MYANMAR LETTER RUMAI PALAUNG FA  
108F MYANMAR SIGN RUMAI PALAUNG  
TONE-5

#### Shan digits

1090 MYANMAR SHAN DIGIT ZERO  
1091 MYANMAR SHAN DIGIT ONE  
1092 MYANMAR SHAN DIGIT TWO  
1093 MYANMAR SHAN DIGIT THREE  
1094 MYANMAR SHAN DIGIT FOUR  
1095 MYANMAR SHAN DIGIT FIVE  
1096 MYANMAR SHAN DIGIT SIX  
1097 MYANMAR SHAN DIGIT SEVEN  
1098 MYANMAR SHAN DIGIT EIGHT  
1099 MYANMAR SHAN DIGIT NINE

#### Extensions for Khamti Shan

109A MYANMAR SIGN KHAMTI TONE-1  
109B MYANMAR SIGN KHAMTI TONE-3

#### Extensions for Aiton and Phake

109C MYANMAR VOWEL SIGN AITON A  
109D MYANMAR VOWEL SIGN AITON AI

#### Shan symbols

109E MYANMAR SYMBOL SHAN ONE  
109F MYANMAR SYMBOL SHAN  
EXCLAMATION

	A9E	A9F
0	ခ	ဝ
1	ဆ	ရီ
2	သ	ဇ
3	ဆ	ဒိ
4	င	ဇ
5	ံ	ဂ
6	ံ	ဌ
7	ဂ	ရ
8	ဌ	ဂ
9	က	ယ
A	တ	လ
B	လ	တ
C	မ	တ
D	တ	ပ
E	တ	င
F	ရ	

### **Shan Pali Consonants**

A9E0 MYANMAR LETTER KHAMTI GHA  
A9E1 MYANMAR LETTER KHAMTI CHA  
A9E2 MYANMAR LETTER KHAMTI JHA  
A9E3 MYANMAR LETTER KHAMTI NNA  
A9E4 MYANMAR LETTER KHAMTI BHA

### **Combining Mark**

A9E5 MYANMAR SIGN SHAN SAW

### **Punctuation**

A9E6 MYANMAR MODIFIER LETTER SHAN  
REDUPLICATION

### **Consonants**

A9E7 MYANMAR LETTER TAI LAING NYA  
A9E8 MYANMAR LETTER TAI LAING FA  
A9E9 MYANMAR LETTER TAI LAING GA  
A9EA MYANMAR LETTER TAI LAING GHA  
A9EB MYANMAR LETTER TAI LAING JA  
A9EC MYANMAR LETTER TAI LAING JHA  
A9ED MYANMAR LETTER TAI LAING DDA  
A9EE MYANMAR LETTER TAI LAING DDHA  
A9EF MYANMAR LETTER TAI LAING NNA

### **Digits**

A9F0 MYANMAR TAI LAING DIGIT ZERO  
A9F1 MYANMAR TAI LAING DIGIT ONE  
A9F2 MYANMAR TAI LAING DIGIT TWO  
A9F3 MYANMAR TAI LAING DIGIT THREE  
A9F4 MYANMAR TAI LAING DIGIT FOUR  
A9F5 MYANMAR TAI LAING DIGIT FIVE  
A9F6 MYANMAR TAI LAING DIGIT SIX  
A9F7 MYANMAR TAI LAING DIGIT SEVEN  
A9F8 MYANMAR TAI LAING DIGIT EIGHT  
A9F9 MYANMAR TAI LAING DIGIT NINE

### **Consonants**

A9FA MYANMAR LETTER TAI LAING LLA  
A9FB MYANMAR LETTER TAI LAING DA  
A9FC MYANMAR LETTER TAI LAING DHA  
A9FD MYANMAR LETTER TAI LAING BA  
A9FE MYANMAR LETTER TAI LAING BHA

	AA6	AA7
0	က	◌်
1	ဂ	က
2	ဃ	ဂ
3	ဣ	ဃ
4	ဥ	ဣ
5	ဧ	ဥ
6	ဧ	ဧ
7	ဧ	ဧ
8	ဧ	ဧ
9	ဧ	ဧ
A	ဧ	ဧ
B	ဧ	◌်
C	ဧ	◌်
D	ဧ	◌်
E	ဧ	ဧ
F	ဧ	ဧ



**Khamti Consonants**

AA60 MYANMAR LETTER KHAMTI GA  
AA61 MYANMAR LETTER KHAMTI CA  
AA62 MYANMAR LETTER KHAMTI CHA  
AA63 MYANMAR LETTER KHAMTI JA  
AA64 MYANMAR LETTER KHAMTI JHA  
AA65 MYANMAR LETTER KHAMTI NYA  
AA66 MYANMAR LETTER KHAMTI TTA  
AA67 MYANMAR LETTER KHAMTI TTHA  
AA68 MYANMAR LETTER KHAMTI DDA  
AA69 MYANMAR LETTER KHAMTI DDHA  
AA6A MYANMAR LETTER KHAMTI DHA  
AA6B MYANMAR LETTER KHAMTI NA  
AA6C MYANMAR LETTER KHAMTI SA  
AA6D MYANMAR LETTER KHAMTI HA  
AA6E MYANMAR LETTER KHAMTI HHA  
AA6F MYANMAR LETTER KHAMTI FA  
AA70 MYANMAR LETTER KHAMTI  
REDUPLICATION  
AA71 MYANMAR LETTER KHAMTI XA  
AA72 MYANMAR LETTER KHAMTI ZA  
AA73 MYANMAR LETTER KHAMTI RA

**Khamti Extensions**

AA74 MYANMAR LOGOGRAM KHAMTI OAY  
AA75 MYANMAR LOGOGRAM KHAMTI QN  
AA76 MYANMAR LOGOGRAM KHAMTI HM

**Aiton Extensions**

AA77 MYANMAR SYMBOL AITON  
EXCLAMATION  
AA78 MYANMAR SYMBOL AITON ONE  
AA79 MYANMAR SYMBOL AITON TWO  
AA7A MYANMAR SYMBOL AITON RA

**Pa'o Karen Tone Mark**

AA7B MYANMAR SIGN PAO KAREN TONE

**Tai Laing Tone Marks**

AA7C MYANMAR SIGN TAI LAING TONE-2  
AA7D MYANMAR SIGN TAI LAING TONE-5

**Shwe Palaung Letters**

AA7E MYANMAR LETTER SHWE PALAUNG CHA  
AA7F MYANMAR LETTER SHWE PALAUNG SHA

## References

- Bechert, et al 1979, *Burmese Manuscripts, Part I* Wiesbaden.
- Department of the Myanmar Language Commission 1993, *Myanmar – English Dictionary* Ministry of Education, Union of Myanmar.
- Hosken, Martin 2011, “Proposal to add minority characters to Myanmar script ” (ISO/IEC JTC1/SC2/WG2 N3976R)
- Okell, John 1994, *Burmese: An Introduction to the Script* SOAS, London.
- Sai Kam Mong 2004, *The History and Development of The Shan Scripts* Silkworm Books, ChiangMai Thailand
- Stribley, Keith “Collation of Myanmar (Burmese) in Unicode” (unpublished manuscript, 2009)  
<http://www.thanlwinsoft.org/ThanLwinSoft/MyanmarUnicode/Sorting/>
- Stern, Theodore “Three Pwo Karen Scripts: A Study of Alphabet Formation” (Anthropological Linguistics, Vol 10, No. 1, 1968)
- The Unicode Consortium 2006, *The Unicode Standard, Version 5.0* Addison-Wesley, Massachusetts.

## **Afterward**

As a researcher, it is impossible to create a document such as this without the help of many people. There are too many people to name them all, but two people, Keith Stribley and Ngwe Tun, stand out as those who have worked to encourage me to make this document as accurate and useful as possible and have provided valuable information and insight.

This research has been undertaken with the support of Payap University and now as part of the work of the Payap University Linguistics Institute.

This document consists entirely of text conformant to Unicode 5.2 and was typeset using a version of OpenOffice with Graphite support. This has enabled me to use only one font for all the Myanmar script samples: Padauk. The various stylistic variants are enabled through the use of the features mechanism Graphite offers.