
Encoding of Myanmar Character Set and Implementation

Dr. Aung Maw
Member, Myanmar IT
Standardization Committee

14 July 2001

Encoding of Myanmar

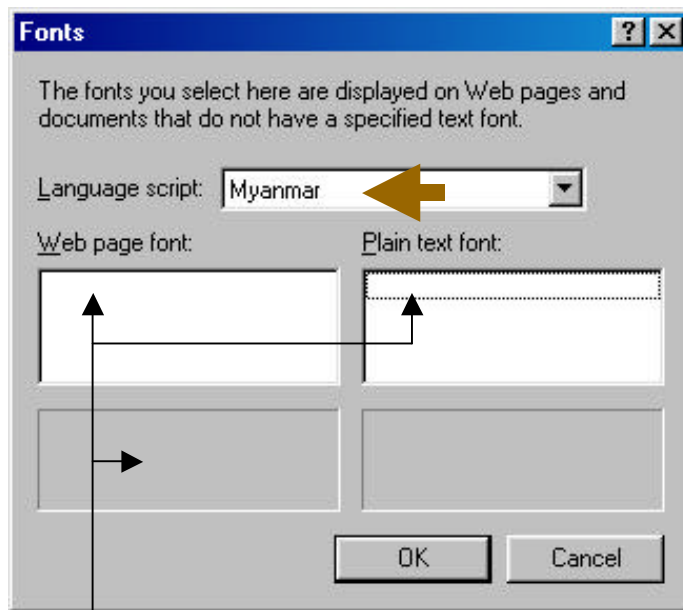
- The Unicode Standard, version 3.0 (August 2000)
- Myanmar Code Range: 1000 – 109F
- Description
 - 1000 – 1021 = Consonants (**u – t**)
 - 1022 – 102A = Independent Vowel (**£? p? o? OD? {? Mo? aMomf**)
 - 102C – 1032 = Dependent Vowel Signs (**-m? -d? -D? -k? -l? a ? -J**)
 - 1036 – 1039 = Various Signs (**-H? -h? -;? -f**)
 - 1040 – 1049 = Digits (**0 – 9**)
 - 102A – 104B = Punctuation (**? /**)
 - 104C – 104F = Various Signs (**ü? í? ãif·? **)



Outcome so far

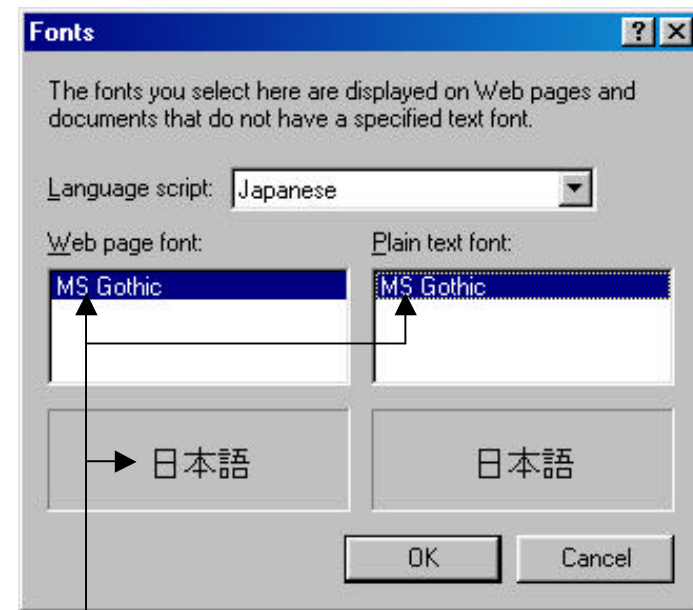
For example - In Microsoft Internet Explorer 5.5

Myanmar



Myanmar OpenType Font

Japanese



Japanese OpenType Font

IMPLEMENTATION

3 Key Technologies in implementation

- Unicode (Encoding of Myanmar Character set)
- Uniscribe (Unicode Script Processor Engine)
- OpenType Font Format

UNICODE – UNISCRIBE – OPENTYPE

from *Microsoft*

Unicode (Encoding of Myanmar)

Excerpt from “ The Unicode Standard, Version 3.0” (ISBN 0-201-61633-5) regarding Myanmar Character Code Table, Standards section.

- **Standards** - There is not yet an official national standard for the encoding of Myanmar/ Burmese. The current encoding was prepared with the consultation of experts from the Myanmar Information Technology Standardization Committee (MITSC) in Yangon. (Rangoon). The MITSC, formed by the government in 1997, consists of experts from the Myanmar Computer Scientists Association, Myanmar Language Commission, and Myanmar Historical Commission.
-

Encoding of Myanmar

- The Myanmar writing system derives from a Brahmi-related script borrowed from South India in about the eighth century for the Mon language.
- Because of its Brahmi origins, the Myanmar script shares the structural features of its Indic relatives, e.g. Devanagari.
- *Hindi, Sanskrit, and Marathi* languages use the *Devanagari* script.
- **Indic Scripts** include Bengali, *Gujarati*, Devanagari, Malayalam, Oriya, Gurmukhi (Punjabi), Tamil, Kannada, Telugu and Myanmar scripts.
- As Myanmar script shares the structural features of so called Indic Scripts, encoding of Myanmar need to follow some **Encoding Principles** of Indic Scripts.
- As with Indic scripts, the Myanmar encoding represents only the basic underlying characters. Multiple glyphs and rendering transformations are required to assemble the final visual form for each syllable.



u + m = um

c + m = cg

* + m = *g

GLYPH

GLYPH is a graphical representation of a character.

FONT is a collection of glyphs.

This is a **CHARACTER**.

Encoding of Myanmar

- Characters or combinations that may appear visually identical in some fonts, such as MYANMAR LETTER WA and MYANMAR DIGIT ZERO, are distinguished by their underlying encoding.



Myanmar Letter WA

Myanmar Digit ZERO

- **Composite Characters** - As is the case in Extended Latin and many other scripts, some Myanmar letters or signs may be analyzed as composites of two or more other characters, and are not encoded separately.

102C ◯◓
102D ◯
102E ◯
102F ◯

1030 ◯
1031 ◯
1032 ◯

atm = a + m
1031+102C

Encoding of Myanmar – **Virama** U+1039

- **Conjunct** - As in other Indic-derived scripts, conjunction of two consonant letters is indicated by the insertion of a virama **U+1039 MYANMAR SIGN VIRAMA** between them; it causes ligation or other rendered combination of the consonants, although the virama itself is not rendered visibly.

rE \dot{A} av; =

ref+w+av; → Storage →

r	e	Virama	w
1019	1014	1039	1010

- **Explicit Virama (Killer)** - The virama **U+1039 MYANMAR SIGN VIRAMA** also participates in some common constructions where it appears as a visible sign, commonly termed killer. In this usage where it appears as a visible diacritic, U+1039 is followed by a **U+200C ZERO WIDTH NON-JOINER** as with Devanagari.

refusnf;yif

→

Storage →

r	e	Virama	ZWNJ
1019	1014	1039	200C

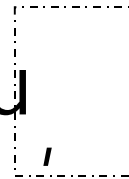
ZWNJ = Zero Width Non-Joiner (from Unicode General Punctuations Table)

Encoding of Myanmar – **Virama** U+1039

- Medial Consonants** - The Myanmar script traditionally distinguishes a set of subscript "medial" consonants: forms of **YA** (**,yifh**), **RA** (**&&pf**), **WA** (**0qGJ**), and **HA** (**[xdk;]**) that are considered to be modifiers of the syllable's vowel, In the Myanmar encoding, the medial consonants are treated as conjuncts; that is, they are coded using the virama .

,yifh

uS = u + , → u

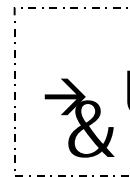


Storage →

u	Virama	,
1000	1039	101A

&&

Mu = u + & → u



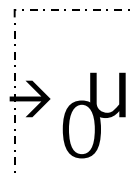
Storage →

u	Virama	&
1000	1039	101B

pf

0qG

uG = u + 0 → u



Storage →

u	Virama	0
1000	1039	101D

J

Encoding of Myanmar – **Virama** U+1039

[xd

k;

rS = r + [→

[r

Storage →

r	Virama	[
1019	1039	101F

- **Kinzi** - The **conjunct form** of U+1004 MYANMAR LETTER NGA is rendered as a superscript sign called kinzi (**F**). Kinzi is encoded in logical order as a **conjunct consonant before** the syllable to which the mark applies.

ocFg & =

o + i f + c g + & Storage →

o	i	Virama	c	m	&
101E	1004	1039	1001	102C	101B

Encoding of Myanmar

- **Signs After Consonants** - Dependent vowels (-m? -d? -D? -k? -l? a ? - J) and other signs (-H? -h? -;? -f) (except kinzi, as noted above) are encoded in logical order **after the consonant** to which they apply, **regardless of where the glyph for the sign happens to be rendered relative to the glyph for the consonant**. In particular, **U+1031 MYANMAR VOWEL SIGN E (a)** is encoded **after** its consonant although in visual presentation it is reordered to appear **before** (to the left of) the consonant form.

armif
→
Storage →

r	m	i	Virama	ZW NJ	a
1019	102C	1004	1039	200C	1031

- **Spacing** - Myanmar does not use any **whitespace** between words. If word boundary indications are desired - for example, for the use of automatic line layout algorithms - the character **U+200B ZERO WIDTH SPACE** should be used to place invisible marks for such breaks. The ZERO WIDTH SPACE can grow to have a visible width when justified.

Uniscribe (Unicode Script Processor)

- The Unicode Script Processor (**USP10.DLL**), new to Windows 2000; is a collection of API's that enable a text-layout client to format **complex scripts**. The **Unicode Script Processor**, also referred to as "**Uniscribe**" supports the **complex rules** found in scripts such as Arabic, Indian, Thai and Myanmar. Uniscribe also handles scripts written from right-to-left, such as Arabic or Hebrew, and supports the mixing of scripts.
 - Uniscribe is composed of multiple "shaping engines." These shaping engines contain the **layout knowledge** for particular scripts (for example, *Arabic, Hebrew, Thai, Hindi, Tamil, Myanmar*). **Uniscribe provides** -
 - **character-to-glyph mapping;**
 - **dx,dy positioning;**
 - **line breaking at word boundaries;**
 - **hit testing and**
 - **cursor positioning.**
 - Using Uniscribe, clients need only manage a backing store of Unicode character codes, typed by the user in "logical order" (as defined by Unicode). Text-layout clients do not need to maintain any other buffer or mapping table to track character order, and the backing store never changes as a result of layout operations.
-

Uniscribe (Unicode Script Processor)

- Clients of Uniscribe include:
 - **Win32 API's**
 - **plain text applets**
 - **edit controls**
 - ***RichEdit 3.0***
 - ***Wordpad***
 - **Office9x and above**
 - **IE4.0 and above**
 - **FrontPage Express**
 - **Outlook Express**
 - **USP10.DLL (the Uniscribe library) ships with Windows 2000**
 - **Internet Explorer v.4.0cs and greater**
 - **USP10.DLL may also be used on NT4, Windows 95 and Windows 98 systems.**
-

OpenType Font Format

- In TrueType Font, there is a one-to-one relationship between an encoded character and the glyph that represents it. Systems and applications that make use of such fonts do not need to make a distinction between **character processing** and **glyph processing**.
 - The Unicode Standard is strictly concerned with character processing, and presumes that Unicode text strings will be input and stored in a simple sequence defined as 'logical order'.
 - The Unicode Standard also presumes the **existence of rendering systems** above the Unicode text string that will **reorder codepoints** and affect sophisticated glyph processing to shape the rendering of the text through **glyph substitution and positioning features**.
 - **In OpenType Font - all the information controlling the substitution and relative positioning of glyphs during glyph processing is contained within the font itself.**
 - In OpenType Font there are 2 internal tables. These are the **GSUB** and **GPOS** tables that contain instructions for, respectively, **glyph substitution** and **glyph positioning**.
 - **Glyph substitution** involves replacing one or more glyphs with one or more different glyphs *representing the same text string*. The backing string of Unicode characters is not changed, only the visual representation.
 - These substitutions may be required (as part of script rendering), recommended as default behavior, or activated at the discretion of the user; they may also be contextual, active only when preceded or followed by a certain glyph or sequence of glyphs, or contextually chained so that one substitution affects another.
-

OpenType Font Format - OpenType Layout Services library (OTLS)

- The **OpenType Layout Services library (OTLS)** is a set of helper functions that serve a text processing client by retrieving information from fonts and guiding the operating system in rendering text.
- The client and OTLS work together to layout text, using some, all or none of the OpenType Layout features defined within a font, as decided by the application developer.
- The client can use OTLS functions to query a font about what layout feature it supports and with what script and language systems they are associated.
- OTLS is designed to expose the full functionality of OpenType fonts to an application, so it is a powerful assistant in implementing support for even the most complicated aspects of Windows glyph processing: for instance, GSUB layout features that are designed to present the user with a choice of variant glyphs.

**Unicode → Encoding of Myanmar Character Set →
OpenType Layout Format (Myanmar.TTF) → Uniscribe
(USP10.dll) → Myanmar Language Script and OpenType
font in future release of Microsoft Windows.**

Unicode -- OpenType -- Uniscribe

Implementation information (on Internet)

<http://www.microsoft.com/typography>
