SUGGESTIONS FOR DEVELOPING LAO UNICODE OPENTYPE FONTS

Introduction

The Unicode standard for Lao allocates only one codepoint for each Lao character or mark, but in high quality printed (or written) Lao, the positions of some vowel and tone mark characters relative to the preceding consonant character will vary according to the actual character string, that is, they will be “context-dependent”.

The most obvious example is that most writers prefer to adjust tone-mark heights according to whether or not the tone marks appear above a diacritic vowel, but several other contextual adjustments can also be used to enhance the appearance of the text.

To achieve context-dependent appearance, a font must employ a method of selecting alternate characters, or adjusting character positions, according to the characters around the character being displayed. The OpenType specification describes a very flexible way of implementing this, but unfortunately (at the time of writing), some kinds of OpenType processing are only supported by *Microsoft* for Windows (not yet for Mac OS X or Linux), and have only been applied in a relatively few Windows applications. In particular, **contextual position adjustments** and **diacritic mark anchoring** are not supported on Mac OS X, or by *Adobe* applications on Windows. These limitations have meant that high quality fonts such as **Saysettha OT** and *Microsoft's* **DokChampa**, which are highly optimized for Windows, do not work very well with OS X applications.

It is however possible to create fonts that for both Windows and OS X allow contextually adjusted diacritic positions (in most applications), by using **ligature substitutions**, which are much more widely supported. By using a carefully-ordered sequence of "Required Ligature" substitution rules, and adding appropriate alternate glyphs to the font, it is often possible to simulate the result of more a general contextual position rule.

This approach has allowed the development of the **Saysettha MX** font family (and other MX family fonts) using only the font creation application **FontLab** (http://www.fontlab.com/), since the simpler OpenType substitutions used are fully supported by FontLab. Previous Lao OpenType fonts (such as **Saysettha OT**) required the additional use of *Microsoft's* Visual OpenType Layout Tool (VOLT) to add the necessary OpenType information, which allows much greater control of font behaviour, but is somewhat more difficult to use.

The following notes on the implementation of MX family fonts are provided to help other font designers who may wish to create similar platform-independent OpenType Lao fonts. The notes accompany the sample font **LaoTest MX.ttf**, which can be downloaded at http://www.laoscript.net/downloads/fonts/samplefont.zip together with the font source file **LaoTest MX.vfb** (in FontLab’s database format).

Note: Many different styles for font design and implementation approaches are possible, and these notes are intended as suggestions, not as a prescription for good font design.

Design Principles

Lao Character Shapes and Positions

The basic Lao character shapes (glyphs) should be designed so that the text is still clearly legible even when OpenType corrections are not applied. Unadjusted tone mark characters should avoid colliding with (overstriking) vowel diacritics as much as possible[1].

Glyph Set

The font will include some glyphs that are encoded (as Unicode) and other glyphs that are named but not encoded, and only used when selected by an OpenType substitution according to the immediate context. Contextual substitute glyphs are generally created (from encoded glyphs) as components rather than being drawn separately.

The following table lists the encoded and unencoded glyphs used in the **LaoTest MX** font.

Description of glyph or range	*Unicode range*	*Notes*[2]
Basic Latin (lower-ASCII range)	U+0020 - U+007F	1
Complete Lao Unicode range	U+0E81 - U+0EDD	1,2
ZERO WIDTH SPACE (ZWSP)	U+200B	1
Complete Thai Unicode range	U+0E01 - U+0E5B	3
DOTTED CIRCLE (indicates missing base characters)	U+25CC	4
Latin-1 punctuation and symbols (upper ASCII)	U+00A0 - U+00BF	4
Latin-1 letters and operators (upper ASCII)	U+00C0 - U+FF	5
Lowered tone mark glyphs: ກ່ ກ້ ກ໊ ກ໋	not encoded	6
Superscript vowel and tone mark composites: ກັ່ນ ກິ່ ກີ່ ກຶ່ ກື່ ກົ່ນ ກັ້ນ ກິ້ ກີ້ ກຶ້ ກື້ ກົ້ນ	not encoded	7
Composites of SALA AM with a tone mark: ກ + ◌ຳ => ກຳ, ກ່ + ◌ຳ => ກ່ຳ, ກ້ +◌ຳ => ກ້ຳ	not encoded	8
Alternate composites of SALA AM with a tone mark: ກໍາ, ກໍ່າ, ກໍ້າ	not encoded	8
Lao letter LO SUNG (composite form) ຫຼ	not encoded	9
Lao high-class "digraph" consonants: ຫງ, ຫນ, ຫມ, ຫຣ, ຫລ and ຫວ	not encoded	9
Alternately positioned mai kan, mai kong (and composites): default: ເປັນ, ເຫຼັ້ມ, ເຂົ້າ or adjusted: ກັນ, ຂົ່ມ, ຄົວ	not encoded	10
Duplicated set of consonants and digraphs	not encoded	11

---

[1] Unlike most *Lao Script for Windows* fonts, neither **Phetsarath OT**, developed by a Lao government agency as an official Lao Unicode font, nor Microsoft's **DokChampa** font follow this principle, so have significant display deterioration when used in applications without OpenType support.

[2] See following page.

Notes on recommended glyphs used for Lao fonts:

1. Essential glyph or range of glyphs.
2. Not all code points are defined. Required code points are indicated on the Unicode chart for Lao http://www.unicode.org/charts/PDF/U0E80.pdf.
3. Not essential, but helps Windows applications to recognize that the font is a complex script. Actual Thai character glyphs are not needed unless the font is intended for use with Thai as well as Lao.
4. Strongly recommended.
5. Recommended, but not essential.
6. Each tone mark will be replaced by a lowered equivalent mark *except* when following a superscript vowel.
7. Vowel, tone mark sequences are replaced by composites for **mai eek** and **mai thoo** to allow the positioning of the tone marks to be correctly defined with respect to the superscript vowel.
8. See **Sala AM** (below, under **Special Notes**).
9. To simplify further contextual substitutions, a single glyph replaces each two glyph sequence.
10. The two superscript marks **mai kan** and **mai kong** are used both as vowel signs and as vowel modifiers. They will normally be approximately centered over the boundary between the initial and final consonant of a syllable unless they are following a prefix vowel. When following a prefix vowel, they are usually approximately centered over the immediately preceding base consonant. The MX series fonts use contextual substitutes with the marks (or composites) slightly to the right of the default position.
11. See **Consonant Duplication** (below, under **Special Notes**).

Encoding

The glyphs selected by contextual substitution do not need to be given Unicode values, as they do not represent stored characters, but only display time variants. However in some fonts some of the substitutes have been encoded in the Unicode “Private Use Area”, since before the availability of OpenType it was only possible to make contextual adjustments by changing the stored or displayed character stream. But as this requires non-standard encoding, it is better not to make use of PUA encoding in stored text, and for new fonts it is probably better not to encode the variant glyphs as Unicode code points at all.

When setting font header parameters, the **Supported codepages** list should include **1252 Latin 1** and **874 Thai**. Including the Thai codepage helps applications such as *Microsoft* Word to recognize that the character set is from a **Complex script**, and set the font accordingly.

The **Supported Unicode ranges** tab requires **Basic Latin**, **Thai** and **Lao** to be checked. Other ranges selected by Fontlab's automatic selection of code ranges may also be checked.

OpenType Features

Ligature substitutions (and other kinds of OpenType processing of glyph sequences) are grouped together into “features” which are either enabled by default, or else may be enabled and disabled by the user (in applications such as *Adobe* **InDesign**). In order to develop a font that would be usable on both Windows and Mac OS X, with as many applications as possible, a search was made for a registered feature that would be recognized and processed by default even in those applications which do not permit user selection of OpenType features. In that way, text using such a font will be enhanced even with applications do not allow user selection of OpenType features (provided that they support OpenType rendering).

The registered feature found to be most widely supported and enabled by default is the **Required Ligatures** (**rlig**) feature, so the main substitutions used in the MX series fonts have been defined for this feature[3].

Some OpenType-aware applications do not yet recognize the Lao language (**lao**) OpenType script tag, but for such applications, lookups affecting the Lao Unicode character set will often be processed correctly if defined for the Latin script (**latn**). The MX series fonts duplicate all lookups under both script tags so that both types of applications will apply the OpenType processing.

Special Notes

Sala AM

Microsoft's processing of tone-mark - SALA AM sequences breaks SALA AM (U+0EB3) into two components, a ring (NIGGAHIT, U+0ECD) and SALA AA (U+0EB2), then re-orders the sequence so that the tone mark follows the ring component and is positioned with respect to that component:

ກ + ◌ຳ => ກໍ + າ, ກ່ + ◌ຳ => ກໍ + ◌່ + າ, ກ້ +◌ຳ => ກໍ + ◌້ + າ

This processing allows exact placement of the ring with respect to the preceding base character, but can only be used with applications that use *Microsoft's* OpenType support, on Windows, as neither Apple nor Adobe support character decomposition.

Two different conventions are followed for SALA AM, with the “ring” either centered over the base consonant, or approximately aligned with the right hand side of that consonant:

ກຳ, ກ່ຳ, ກ້◌ຳ(“standard rendering”) or ກໍາ, ກໍ່າ, ກໍ້າ (“alternate rendering”)

In (carefully) hand-written Lao, the first convention is more common, but for consonants that have ascenders, the ring can appear either before, behind or above the ascender. In typewritten Lao, both ways were usually possible.

In the MX series fonts, if the characters are typed in standard Unicode order (consonant + tone mark + sala am, or consonant + niggahit + tone mark + sala aa), the first (composite) form of each group will normally be displayed, whether on Windows or Mac OS X.

In applications that support user-selection of OpenType features[4], the alternate form can be displayed simply by enabling either the Stylistic Set 01 (**ss01**) or Stylistic Alternates (**salt**) feature.

Allowing the alternate rendering by enabling Stylistic Set 01 (**ss01**), as well as Stylistic Alternates (**salt**), has been implemented since Adobe InDesign CS2 does not support the use of the Stylistic Alternates (**salt**) feature, and also so that it can be enabled independently of Stylistic Set 02 (**ss02**) on Mac OS X applications, since on current versions of OS X different Stylistic Set features cannot be applied to the same text.

---

[3] Contact support@laoscript.net for a list of features supported by different applications.

[4] In applications which do not allow user control of OpenType features, the second form can also be produced by entering, for example: ກ + ◌ໍ + ◌່ + **ZWSP** + າ => ກໍ່າ.

Consonant Duplication

Every consonant and digraph has been duplicated (as a component glyph) in the MX series fonts to work around the single-character context limit in Apple's processing of contextual substitutions.

The original and duplicate consonants are used as follows:

(a) each digraph sequence is always replaced by a single glyph

(b) any consonant (or composed digraph) glyph *that follows a prefix vowel* is replaced by an identical, *but renamed*, copy of that glyph

(c) when **mai kan** or **mai kong** follow an “ordinary” glyph, it will be replaced by its right-shifted alternate, but if it follows one of the renamed consonants, it will remain positioned above the consonant (as illustrated previously).

Use of OpenType to support an old writing convention

In older Lao documents, syllables in which ຍ (LAO LETTER NYO, U+0E8D) would occur finally are often written instead using ຽ (LAO SEMIVOWEL SIGN NYO, U+0EBD). The MX series fonts allow the same text to be displayed either way, by enabling or disabling the Stylistic Set 2 (**ss02**) feature, which changes syllable-final ຍ to ຽ at display time, without affecting the stored text.

Version 2 Revision

The original MX-series fonts (released in 2008) used the *Contextual Ligatures* (**clig**) registered feature to implement the main position adjustments needed to produce good-quality Lao text. In the updated Version 2 fonts, the *Required Ligatures* (**rlig**) feature is used instead, since this feature is processed and enabled by default on more platforms, and in particular, when used in web pages as an embedded font viewed by browsers on Windows, MacOS or iIOS (iPhone or iPad).

Dr. John M. Durdin
Lao Script for Windows
http://www.laoscript.net/

March 2012