How To Type Devanagari

Since a package for typesetting Devanagari, `devnag`, designed by Frans Velthuis, was already available I decided to adapt it to Malayalam-TEXs input conventions.
To start using Devanagari, all you have to do is say

```
\input dngmacs
\input dntrmacs
```

Somewhere near the start of your document. Then you can switch to any of the principle languages that use Devanagari-script by saying `<sanskrit>`, `<hindi>`, `<marathi>`, or `<nepali>`. You will then enter the mode of the indicated language – together called *Devanagari-mode*. You can switch back to normal mode by saying respectively `</sanskrit>`, `</hindi>`, `</marathi>`, or `</nepali>`. If you wish to use Roman transcription for any of those languages, you will have to append `.transcription` to the language name in the start-tag; so Hindi in transcription can be started with `<hindi.transcription>`. No modification of the end-tag is required (or even allowed).
The following table shows the character(s) you have to type to produce a Devanagari character.

OM	ॐ	O...M	ka, qa	क, क़	ka, qa	pa	प	pa
a	अ	a	kha, Kha	ख, ख़	kha, kha	pha, fa	फ, फ़	pha, fa
*	अ		ga, Ga	ग, ग़	ga, ga	ba	ब	ba
aa	आ	ā	gha	घ	gha	bha	भ	bha
i	इ	i	n"a	ङ	ṅa	ma	म	ma
ii	ई	ī	ca	च	ca	ya	य	ya
u	उ	ū	cha	छ	cha	ra	र	ra
uu	ऊ	ū	ja, za	ज, ज़	ja, za	la	ल	la
.r	ऋ	ṛ	jha	झ	jha	va	व	va
.r.r	ॠ	ṝ	n~a	ञ	ña	sha	श	śa
.l	ऌ	ḷ	Ta	ट	ṭa	Sa	ष	ṣa
.l.l	ॡ	ḹ	Tha	ठ	ṭha	sa	स	sa
e	ए	e	Da, Ra	ड, ड़	ḍa, ṛa	ha	ह	ha
ai	ऐ	ai	Dha, Rha	ढ, ढ़	ḍha, ṛha	La	ळ	ḷa
o	ओ	o	Na	ण	ṇa	r"	ॅ	r
au	औ	au	ta	त	ta	k+	क्	k
aM	अं	aṃ	tha	थ	tha	|	।	.
a~	अँ	ã	da	द	da	#	॰	.
aH	अः	aḥ	dha	ध	dha	###	...	...
a"	ऍ	ê	na	न	na	@	°	.
o"	ऑ	ô						

If you want to view or print your document after typing (a part of) it, you will have to do some extra pre-processing*. Saying

`patc -p dng.pat` *input-file*`.dng` *temporary-file*`.dn`
`devnag` *temporary-file*`.dn` *output-file*`.tex`

after pre-processing for Malayalam, if it is used, will do all the neccessary pre-processing for Devanagari. As you can see, you will still have to use Frans Velthuis' pre-processor**. To find out how it should be used exactly, read the manual that comes with `devnag`.

* I admit, if you use several languages in one document, this can get pretty booring.

** As a matter of fact, I have only changed the transcription – all functionality of `devnag`, including `@`-directives to it is still there.

`\<#1>`: typeset glyph number `#1` from the current font.
`\B#1#2`: put `#2` centered below `#1`. This is used, for example, to build ழ from ழ and ◌ in traditional script.
`\M#1#2`: put `#2` right below `#1`. This is used to build, for example, to build ப்ல from ப and ல. Note that if the subscribed character is wider than the character it is subscribed to, the width of the resulting box will be adjusted. This happens when ப்ர is build from ப and ◌.
`\T#1#2`: put `#1` centered on top of `#2`. This is used to position the reepham stroke in traditional script.
`\Z`: switch to the subfont, the font that contains the somewhat smaller glyphs that are subscribed to characters.
`\R#1#2`: decrease the height of box `#2` by `#1`.
`\K#1`: kern `#1` em.
`\X#1`: equivalent to `\hbox#1`.

How To Type Tamil

Because all the ingredients where available and easily adaptable to Malayalam-TEX, I decided to add the capability to typeset Tamil. The font used is `wntml10`, available free from Washington State University. All you have to do to be able to type Tamil is say

```
\input tmlmacs
```

somewhere near the beginning of your document. Then you can enter *Tamil-mode* with `<tamil>` and leave it with `</tamil>`.
The following table shows the character(s) you have to type to produce a Tamil character.

a	அ	a	ka	க	ka	va	வ	va
aa, A	ஆ	ā	n"a	ங	ṅa	zha	ழ	ẕa
i	இ	i	ca	ச	ca	La	ள	ḷa
ii, I	ஈ	ī	n~a	ஞ	ña	Ra	ற	ṟa
u	உ	u	Ta	ட	ṭa	n_a	ன	ṉa
uu, U	ஊ	ū	Na	ண	ṇa	ja	ஜ	ja
e	எ	e	ta	த	ta	Sa	ஷ	ṣa
ee, E	ஏ	ē	na	ந	na	sa	ஸ	sa
ai	ஐ	ai	pa	ப	pa	ha	ஹ	ha
o	ஒ	o	ma	ம	ma	kSa	க்ஷ	kṣa
oo, O	ஓ	ō	ya	ய	ya			
au	ஔ	au	ra	ர	ra			
H	ஃ	ḥ	la	ல	la			

When you use Tamil in your document, you have to run it once more through a pre-processor, to replace the transcription for TEX-commands that typeset the Tamil characters. Say

`patc -p tamil.pat` *input-file output-file*

to pre-process Tamil.
If you have lots of text available in another transcription, you can convert it with `patc` to this transcription. two pattern-files are provided. `adami.pat` translates the ADAMI transcription scheme, and `wntml.pat` translates the original transcription used at W.S.U.

in the character string, the second element the sequence of TEX-commands that will produce its graphical representation. If the pre-processor encounters a single character for which no way of typesetting it is given in this table, it will complain, and forget the character.
`@primary` starts a table that defines the TEX-commands for each character and for sequences of characters when they occur at the start of a syllabe. Every entry in this table contains three elements. The first is the character-sequence, the second the TEX-commands that will be placed in front of the syllabe, the third the TEX-commands that will be placed after it. In `mm.scr` this table is very short, since Malayalam script does not modify the first letters of a syllabe (as in Devanagari), but the last ones. The only entry describes the reepham character.
`@secondary` starts a table that defines the TEX-commands for each character and for sequences of characters when they occur at the end of a syllabe. The table has the same format as the `@primary` table, but in `mm.scr` it is much longer, since it defines the glyphs for all the vowel-signs, and for the secondary variants of ൗ, ര, ല, and വ.
`@reepham` tells the pre-processor that a traditional style reepham is used: a little upright stroke centered above the consonant (cluster). This command is needed because otherwise, the pre-processor will center the reepham, incorrectly, above the whole syllabe.

Macros defined by Malayalam-TEX

The traditional Malayalam script is so complicated, that it was impossible to fit all needed glyphs into a single TEX-font. Therefore, a Malayalam font consists of two physical fonts, one with characters at normal size, called the *base-font*, and one with smaller sized characters, called the *sub-font*, that can be used to build conjunct characters on the fly.
One can be tempted to use a normal font for a smaller size for this purpose, but that wouldn't work out as nice, since smaller sized fonts are normally also drawn with thinner lines, while the sub-font is drawn with exactly the same pen as the base-font – and also some other parameters are set in such a way that composing conjuncts will work fine.
The pre-processor uses a macro (`\Z`) to switch to the sub-font, if a glyph from that font is needed. It is not aware of the current style used. This means that `\Z` has to be redefined to select the correct sub-font, depending on current style. This is done behind the scenes when you select a font style.
The macro that selects a font style, sets the base-font, the sub-font, and a corresponding roman font from with several punctuation marks are borrowed, and, when in Malayalam mode, makes the base-font current, otherwise the base-font is made current when Malayalam mode is entered. For a font styles in Malayalam three fonts need to be loaded in as follows:

`\font\twelvemmf=mm12`	load the base-font
`\font\twelvemmsf=mm12s`	load the sub-font
`\font\twelverm=cmr12`	load the roman font for punctuation marks

An example of a macro to select a style is

`\def\twelvemmr{%`	to select `\twelvemmr`:
`\let\mmcurf=\twelvemmf`	define the current base-font,
`\let\mmcursf=\twelvemmsf`	define the current sub-font,
`\let\mmcurrm=\twelverm`	define the current roman font,
`\mmcbls=15.6pt`	set the current baselineskip,
`\ifmmmode\setmm\fi}`	and if we are in Malayalam mode select it.

If you wish to include more styles, you should define new font selection macro's following this example. The baselineskip should be larger than for roman text, since Malayalam script tends to have taller characters.

The TEX-macros given in `mm.scr` to compose glyphs are all one-letter macros. They are defined in the file `mmmacs.tex`. Care is taken that they do not redefine standard TEX macros, and you should not redefine them in your document. They are the following:

Transcription Files

As stated above, a transcription definition tells the preprocessor how to interprete an input-file. It not only tells it which sequence of Roman characters stands for which Malayalam characters, but also how to separate Malayalam text from English text, which the pre-processor should leave untouched, and also tells the pre-processor how to recognize TEX-commands (in a limited way) and comments – which it strips from the input-file.

A transcription definition is given in a *transcription file*. The general structure of a transcription file is the same as that of a patc file. Inside a transcription file, the following `@`-commands can occur: `@atbegin`, `@atend`, `@table`, and `@end`.

`@atbegin` defines which patterns will start and `@atend` defines which patterns that will end Malayalam mode. You can give several patterns, in double quotes, on the same line as the command is. A quick glance in `mm.trs` will learn you that, apart from the `$`-sign, you can also use `<malayalam>` to start Malayalam and `</malayalam>` to end it. Further, `<mlr>` and `</mlr>` are profided to delimit reformed script, while `<ml>` and `</ml>` delimit traditional script, so that you can use both in the same document, in which case you will have to run the pre-processor twice on the input-file. These delimiters are provided, in case you want to be more clear about the language used, or want to use the `$`-sign for some other purpose. Imagine writing a book about the relationships between Dravidian languages, in which most of the examples are from Tamil, I guess you would prefer to reserve the `$`-signs for Tamil then (but in that case you will have to remove them in the current transcription files).

`@table` starts the table of patterns that are used for transcribing Malayalam characters. Each following non-empty line until the next `@`-command is considered to be an entry in the table. The format is basically the same as that of the `@patterns`-tables of patc, but you may not use pattern-set switching, and an extra action, `=` is added. This action maps a transcription pattern to an internal code. Extreme care is required when changing these codes, since they are hardwired into the pre-processor, that knows that some of them are consonants, and others vowels, and uses that knowledge to perform its task. Also, the same codes are used in the script definition to identify characters. The codes should match in the transcription-table, internally in the pre-processor and in the script-file.

Finally, `@end` tells the preprocessor the transcription file has ended. It will not try the parse the rest, but close immediately after reading this command. If a file ends before the `@end` command is seen, the pre-processor will complain, but that is nothing to bother about.

Script Files

A script files contain a script-definition. A script definition tells the pre-processor how to place graphical elements to represent a string of characters. Because the output of the pre-processor is fed to TEX for actual typesetting, the script file contains TEX-commands to typeset those graphical elements, to place them on top or below of another, etc. The general structure of a script file is the same as that of a transcription file, but, since scripts can be quite complicated, it can (and will) contain much more commands. In a script file you will find the following commands: `@malayalam`, `@atbegin`, `@atend`, `@atbeginsyllabe`, `@atendsyllabe`, `@prebuild`, `@primary`, `@secondary`, `@reepham`, and `@end`.

The `@malayalam` command tells the pre-processor that it is dealing with Malayalam script. It must be given as the first command in a script file. At this time the pre-processor does not understand any other script, but if it will in future, it will be functional, since pre-processor needs to know which script it is dealing with, since not all pecularities of Malayalam script are expressed in general mechanisms defined in a script file.

`@atbegin` defines the sequence that is output every time the pre-processor enters Malayalam mode. At this time the file `mm.scr` it contains a begingroup brace and a TEX-command that selects the current Malayalam font.

`@atend` defines the sequence that is output every time the pre-processor leaves Malayalam mode, currently only a endgroup brace.

`@atbeginsyllabe` and `@atendsyllabe` define the sequences that are output at the begin respectively the end of every syllabe. In `mm.scr` you will find commands that make a separate h-box of every syllabe.

`@prebuild` starts a table that defines a sequence of TEX-commands for each character and for sequences of characters for which the general construction method will fail. Each non-empty line of the table describes how to typeset a sequence of one or more characters. The first element of the line is the character sequence

Advanced Topics

This part of this guide explains how to use the components of Malayalam-TEX separately, and how to modify its tables, for example to change the transcription it understands, or to change the way it composes conjuncts. It should be read carefully, and care should be taken when you tamper with the tables, since several tables depend on each other in delicate ways. If you want to change the transcription only, the best way is to use Patc to translate your transcription to the standard transcripton. To show you how this can be accomplished, I have include the file `ack.pat`, that translates the transcription used on the `alt.culture.kerala` newsgroup to the transcription used with `mm`, assuming that Malayalam text is delimited with `$` in the source already (If they are missing, you should place them manually; it is a non-trivial task to have a computer automatically recognize Malayalam passages in English text). With this file, you can convert texts in this transcription by giving the command:

`patc -p ack.pat` *input-file output-file*

Patc

Patc is a program that can translate patterns occuring in a input file to other patterns in the output file. It is fully table driven, and both the patterns it reads as well as those it outputs are defined in tables or *pattern-files*. It is an ideal tool to switch from one transcription scheme to another. Within the Malayalam-TEX package it is used to translate the easy-to-type transcription of Malayalam I adopted into a sequence of TEX-commands that will produce scientific transcription. The transcription it reads and the commands it outputs are given in a pattern-file called `mm.pat`.
Patc should be called as follows:

`patc [-v] -p` *pattern-file input-file output-file*

The `-v` option tells `patc` to be verbose, and the `-p` option tells it that the next parameter given is the name of a pattern file. The last two parameters tells it to read from *input-file* and to write to *output-file*. The actual format of a pattern-file is described in the documentation that comes with `patc`. (Read this first, the following sections assume knowledge of its contents)

mm

`mm` is the 'actual' pre-processor, it translates the Roman transcription into Malayalam script. It uses the correct shapes for vowels depending on whether they appear initially or after a consonant, and selects the correct conjuncts. This program, too, is table driven.
`mm` is called by:

`mm [-`*option*`] [-t` *transcription-file*`] [-s` *script-file*`]` *input-file* `[`*output-file*`]`

Available options are `-v`, `-V`, and `-h`. The `-v` option tells `mm` to be verbose, if given, the pre-processor will inform you about files read. if you give the `-V` option instead, it will also print a `.` for every syllabe output, and a `+` every time it is forced to use a virama instead of a conjunct. When `-h` is given, `mm` will do nothing but print a help message.
The `-t` option tells `mm` to read the *transcription definition* from the file given in the next parameter. The transcription definition defines which Roman characters are used to transcribe Malayalam characters. If it is ommitted, `mm` will use `mmr.trs`. The `-s` option tells `mm` to read the *script definition* from the file given in the next parameter. The script definition defines which glyphs from the font are to be used to represent Malayalam characters in various contexts. If it is ommitted, `mm` will use `mmr.scr`, which defines reformed script.
The last two parameters tell `mm` where to read its input from and where to put its output. Default, the name of the input file ends with `.mm` and the name of the output file with `.tex`. If the name of the output file is ommitted, `mm` will construct it from the input file name by replacing the extension of it with `.tex`.
`mm` will look for it's transcription and script files in the current directory first, and then in the directory pointed at by the environment variable `MMDIR`.

The cillu letters (ക്, ണ്, ന്, ര്, and ല്) will be used automatically at the end of words, as well as anusvaram (ം). If a full letter with candrakala is desired, you should place `<<+` after the final letter, e.g. `t<<+` yields ത് and `m<<+` yields മ്.
The usage of the `$`-sign by Malayalam-TEX overrides its normal meaning in TEX. To enter and leave normal math-mode, you will have to type `\math`, for display math you just say `\Math`. The $-sign itself is produced by `\dollar`.

Malayalam transcription

Malayalam Text can also be printed in scientific transcription. When you want a Malayalam passage to appear in scientific transcription, you have to switch to *Malayalam-transcription mode*, which is started and ended with `$$`. The transcription used follows, but for a few exceptions, the standardized transcription of the National Library in Calcutta.
If you want to use a capital letter in Malayalam-transcription mode, you can precede the letter with a `^`. This will be ignored in Malayalam mode. If the need may arise to typeset a whole passage of transcribed Malayalam in capital letters, the trick can be done by enclosing it between `^^` and `^^`.

Choosing Fonts

With Malayalam-TEX you can use, as in English, several font-styles: regular, *slanted*, and **bold**. There is no *italic*, since italic fonts, which have a different design are unknown in Malayalam typography, but the font is available in two flavours, the normal style, using the same line thickness every where in the character, and a 'calligraphic' version, which shows differences in line thickness in the character, as if it was written with a calligraphic pen. Of course, the fonts are profided in several sizes: 6, 8, 10, 12, and 17 points high.
Every style and size is given a macro, which you have to enter to use a certain style and size. For example, if you want to use 12pt Malayalam calligraphic slanted, you say `\twelvemmcsl`. If you use it in normal mode, the font-change will be in effect the next time you switch to Malayalam mode, in Malayalam mode, it works immediately.
Available fonts are:

```
\sixmmr        \eightmmr
\mmr           \mmsl          \mmb           \mmc           \mmcsl         \mmcb
\twelvemmr     \twelvemmsl    \twelvemmb     \twelvemmc     \twelvemmcsl   \twelvemmcb
\seventeenmmr  \seventeenmmb  \seventeenmmc  \seventeenmmcb
```

A sample of each font-style appears in the Appendix.

Using the Pre-Processor

When you have finished typing (a part of) your document, you cannot feed it to TEX immediately. First you have to run it trough the pre-processor, which will replace the Malayalam in transcription you entered with instructions for TEX to typeset the correct glyphs. The pre-processor takes care of placing the vowel-signs around consonants, and using the correct conjuncts for consonants without intervening vowels.
To pre-process you document, you type:

`ml` *input-file output-file*

This will process the file for traditional script. If you want to use the nowadays normal reformed script, say:

`mlr` *input-file output-file*

Actually, the commands `ml` and `mlr` are two small command files, which call two programs with various parameters that do the actual pre-processing. The first is called `patc`, which takes care of Malayalam-transcription mode, the second is `mm` which task it is to do the contextual analysis for the Malayalam script. If you don't use Malayalam transcription in your document, you can run only `mm` to save some time.

Preparing Your Document

When you want to use Malayalam-TEX, all you have to do is say

```
\input mmmacs
```

somewhere near the beginning of your document, before the first occurance of Malayalam text, and have the file `mmmacs.tex` in a directory where it can be found by TEX. This file will load the fonts and define the commands necessary. After this, you can use the commands as described below.

How To Type Malayalam

Since on most computers it is impossible to enter Malayalam directly, Malayalam text needs to be entered in transcription. Traditionally, Western scholars have used a transcription with various accents, like ḍ, ś or ẓ to represent Malayalam characters, but even these cannot be reproduced on most computers. To solve this problem, Malayalam-TEX uses a transcription that only uses the characters of the standard ASCII character set.

The following table shows the character(s) you have to type in order to produce a Malayalam character. The third column shows the scientific transcription of each character.

`a`	അ or implicit	a	`ka`	ക	ka	`pa`	പ	pa
`aa`	ആ or ാ	ā	`kha`	ഖ	kha	`pha, fa`	ഫ	pha, fa
`i`	ഇ or ി	i	`ga`	ഗ	ga	`ba`	ബ	ba
`ii`	ഈ or ീ	ī	`gha`	ഘ	gha	`bha`	ഭ	bha
`u`	ഉ or ു	u	`n"a`	ങ	ṅa	`ma`	മ	ma
`uu`	ഊ or ൂ	ū	`ca`	ച	ca	`ya`	യ	ya
`.r`	ഋ or ൃ	ṛ	`cha`	ഛ	cha	`ra`	ര	ra
`.r.r`	ൠ or ൄ	ṝ	`ja`	ജ	ja	`Ra, t_a`	റ	ṟa, ṯa
`.l`	ഌ	ḷ	`jha`	ഝ	jha	`RRa, t_t_a`	ൿ	ṯa
`.l.l, .L`	ൡ	ḹ	`n~a`	ഞ	ña	`la`	ല	la
`e`	എ or െ	e	`Ta`	ട	ṭa	`va`	വ	va
`ee`	ഏ or േ	ē	`Tha`	ഠ	ṭha	`sha`	ശ	śa
`ai`	ഐ or ൈ	ai	`Da`	ഡ	ḍa	`Sa`	ഷ	ṣa
`o`	ഒ or ൊ	o	`Dha`	ഢ	ḍha	`sa`	സ	sa
`oo`	ഓ or ോ	ō	`Na`	ണ	ṇa	`ha`	ഹ	ha
`au`	ഔ or ൗ	au	`ta`	ത	ta	`La`	ള	ḷa
`au"`	ൗ or ൊൗ	au	`tha`	ഥ	tha	`zha`	ഴ	ẓa
`M`	ം	ṃ	`da`	ദ	da	`+`	്	
`H`	ഃ	ḥ	`dha`	ധ	dha	`u+`	്	ŭ
			`na, n_a`	ന	na, n̲a			

Malayalam text can only be entered when in *Malayalam mode*. Malayalam mode starts and ends with a `$`-sign.

Punctuation marks can be typed in Malayalam the same way as in English. They will show up as expected. A nice ornamental star ✲ can be produced with `\ornstar`.

Figures in Malayalam mode will appear as European figures. If you want Malayalam figures, you will have to say `\mmfigures` first. After that `1234567890` will appear as ൧൨൩൪൫൬൭൮൯൦. To switch back to European figures, say `\rmfigures`.

More precise control over the formation of ligatures can be reached with the characters `>>` and `<<`. `>>` will prefent the formation of a ligature, and a virama will be used between the consonants instead. `<<` will force the pre-processor to use the secondary variant for the following character. This can be used to show vowel-signs or secondary letters on themselves. `[]` will produce a dotted circle ◌, to make tables like the above, and, last but not least, `>` will disambiguate the transcription if needed.

Introduction

Malayalam-TEX is a package to typeset Malayalam with TEX*. It can be used to typeset in both the traditional and reformed script. The complete package includes a pre-processor and fonts in various styles (regular, bold, slanted, calligraphic) and various sizes.

Malayalam Text is entered in an easily adaptable ASCII transcription. This text is then converted to commands for TEX to typeset the Malayalam by a pre-processor, which takes care of applying the correct conjunct letters and placing the vowel signs on the right position. The same ASCII transcription can also be used to derive scientific transcription of the text, which will save scholars lots of time spend in typing the macros for accented letters themselves.

This document assumes a working knowledge of TEX, and will not try to explain things already explained in the TEX-book. It can roughly be divided in three parts. The first part explains how to use Malayalam-TEX in a standard fashion. The second part goes deeper into the internals, so that you are able to change the behaviour of Malayalam-TEX. This part can be skipped if you are happy with the standard settings. Some appendices complete this manual.

* Malayalam-TEX will work only with TEX3.0 or later versions of TEX.