Devanagari for TEX
Version 1.2
User Manual

Frans J. Velthuis
©University of Groningen
The Netherlands
May 1991

Contents

1 Introduction

This document describes the use of a package which makes it possible to print characters from the Devanagari script. The package consists of font files, a preprocessor and macro files. It is used with the typesetting system TEX[1] or the macro package LaTeX. It is assumed that the reader is familiar with TEX. The next two sections explain what you need to know about creating Devanagari text. The last section (Advanced Topics) can be skipped on first reading.
A source file for TEX consists of plain ASCII-text. With the devanagari package it has been a design-objective to keep the format of the source text as close as possible to the accepted standards for transliteration. However, symbols like ā and ṛ are not available on keyboards and have to be replaced by other symbols or groups of symbols. The table in the following section shows what has to be entered in order to produce a certain devanagari (Called DN from now on) symbol. A source text which is produced this way has to be processed by a so-called preprocessor before it can be given to TEX as input. This preprocessor takes care of things like determining the shape of vowels, the formation of consonant ligatures etc.
The preprocessor can work in two different modes: Hindi mode or Sanskrit mode. The difference is that in Sankrit mode all the available ligatures are used, while in Hindi mode sometimes a virama stroke is preferred. Furthermore, in Sanskrit mode a virama is automaticaly added at the end of a word if it ends in a consonant.

Acknowledgement: I would especially like to thank Rob Adriaanse, Hans Bakker, Roelf Barkhuis, Henk van Linde and Dominik Wujastyk for their advice and support.

Please send suggestions for improvement etc. to:
E-mail: `velthuis@rc.rug.nl`
Postal Mail:
Frans Velthuis
Nyensteinheerd 267
9736 TV Groningen
The Netherlands

[1] TEX is a trademark of the American Mathematical Society

2 The Characters You Type

The following table shows which character(s) to type in order to produce a character in DN.

`a`	अ or implicit	`ch` or `C`	छ	`r`	र	`f`	फ़	
`aa` or `A`	आ or ा	`j`	ज	`l`	ल	`z`	ज़	
`i`	इ or ि	`jh` or `J`	झ	`L`	ळ	`.kh` or `.K`	ख़	
`ii` or `I`	ई or ी	`~n`	ञ	`v`	व	`.g`	ग़	
`u`	उ or ु	`.t`	ट	`"s`	श	`q`	क़	
`uu` or `U`	ऊ or ू	`.th` or `.T`	ठ	`.s`	ष	`.o`	ॐ	OM
`.r`	ऋ or ृ	`.d`	ड	`s`	स	`.a`	ऽ	avagraha
`.R`	ॠ or ॄ	`.dh` or `.D`	ढ	`h`	ह	`~o`	ऑ	English o
`.l`	ऌ or ॢ	`.n`	ण	`R`	ड़	`~a`	ॅ	English a
`.L`	ॡ or ॣ	`t`	त	`Rh`	ढ़	`.m` or `M`	ं	anusvāra
`e`	ए or े	`th` or `T`	थ	`1`	१	`/`	ँ	candrabindu
`ai` or `E`	ऐ or ै	`d`	द	`2`	२	`.h` or `H`	ः	visarga
`o`	ओ or ो	`dh` or `D`	ध	`3`	३	`|`	।	sentence end
`au` or `O`	औ or ौ	`n`	न	`4`	४	`||`	॥	paragraph end
`k`	क	`p`	प	`5`	५	`@`	॰	abbreviation
`kh` or `K`	ख	`ph` or `P`	फ	`6`	६	`#`	·	elliptical dot
`g`	ग	`b`	ब	`7`	७	`..`	.	period
`gh` or `G`	घ	`bh` or `B`	भ	`8`	८	`~r`	ऱ्	Marathi r
`"n`	ङ	`m`	म	`9`	९			
`c`	च	`y`	य	`0`	०			

Notes:

- For several DN characters there are two alternative ways to produce them. You can type two straight characters or one character with the shift key pressed, according to your preference. In cases where ambiguities might erase, using uppercase vowels is preferable: use `laRAI` instead of `laRaaii` for लड़ाई
- `~a` produces an 'English a' Marathi style. Compare: टैक्सी (Hindi) and टॅक्सी (Marathi).
- Numerals are default printed as 'arabic' numerals. Use the command `\dnnum` somewhere in your input file if you want DN numerals. Every numeral after this command is printed as a DN numeral. The command `\cmnum` switches back to arabic numerals.
- In Hindi mode the character `&` can be put at the end of a word to produce a virāma sign under the final consonant. For example: `pari.sad&` produces परिषद् .
- Take care to write an `a` between consonants if you don't want those consonants to form a consonant group, even if such an `a` is not present in transliteration. So type `karanaa` instead of `karnaa` if you want करना.

3 Preparing an Input File

3.1 Preprocessor Commands

The preprocessor that creates a TEX-file from your input file acts on two different parts of this input file. First, there is a special section at the start of the input file containing 'preprocessor commands'. The preprocessor considers a line to be a preprocessor command if it starts with the character `@` and if all the lines before this line do also start with a `@`. At present, the following preprocessor commands are defined:

- `@hindi`
- `@obeylines`
- `@dollars`
- `@dolmode1`
- `@dolmode2`
- `@dolmode3`
- `@lig`
- `@nolig`

The command `@hindi` puts the preprocessor in Hindi mode. The default mode is Sanskrit. The mode definition governs the file as a whole; it is not possible to switch modes somewhere in the middle of an input file.
The other preprocessor commands will be explained later in this and the next section. The order of the commands is not important, except that if a `@hindi` command is present, any `@lig` and `@nolig` commands must come after it.

3.2 Devanagari Text

The second part of the input file on which the preprocessor acts is actual DN text. The remainder of the input file (text in latin script, TEX-commands, etc.) is left untouched. Of course the preprocessor must be told where DN text starts or ends. You can delimit DN text in two ways. Firstly, DN text can be surrounded by `{\dn␣` and `}` . (␣ means a space character). I.e. `{\dn␣acchaa}` produces अच्छा. Secondly, if the preprocessor command `@dollars` is given, DN text can also be delimited by dollars: `$acchaa$`. The second method is convenient when you have to switch often between DN text and text in latin script. For large pieces of text in DN however, the first method is to be preferred, because this way it is easier to determine where DN text ends and where it begins.
What can be typed between DN text-delimiters?

- All characters that produce DN text as described in section 2.
- The characters `!` `` ` `` `'` `(` `)` `*` `-` : ; = ? , [ ]+ which produce the corresponding punctuation marks. (``` `` ``` produces “ and `''` produces ”). Consecutive `-` characters have the same effects as in TEX and LaTeX. If you want to print punctuation marks in latin script other then those mentioned above, you have to type a final `}` or `$` to end the DN text and type the punctuation mark, after which you can switch back to DN text mode.

- The character `%` which produces no output. It has the effect that everything after the `%` upto the end of the line is considered as a comment.
- Left and right braces: `{` and `}` , as long as the number of left braces equals the number of right braces. They produce no output but can be used for grouping.
- TEX or LaTeX commands without arguments or with one argument consisting of text, as long as these commands do not explicitely or implicitely change the current font. The text in the argument is interpreted as DN text. The following commands are legal:

  ```
  \bigskip
  \centerline{acchaa}
  ```

  But these are illegal:

  ```
  \hspace{.1in}
  \begin{center}
  \it hello
  ```

From the above and section 2 it follows that the following characters are illegal between DN-delimiters:
`^ < > x w F N Q S V W X Y Z` and the following characters when they are not used in a combination mentioned in the table on page 2: `"` `.` and `~`

3.3 Plain TEX Commands

The commands for loading the macro-packages, changing the font size etc. differ depending on whether you use Plain TEX or LaTeX. When you prepare an input file for Plain TEX, the following statement has to be included near the beginning of the input file (but after the preprocessor commands): `\input dnmacs`
There are several commands to switch to a different DN font size. Standard (10 points) size is obtained by: `\dnnormal`. `\dnsmall` and `\dnnine` give smaller sizes. (8 and 9 points) The commands to switch to bigger sizes are: (in order of increasing size): `\dnhalf`, `\dnbig`, `\dnlarge` and `\dnhuge`. These commands can be given inside as well as outside DN text.

3.4 LaTeX Commands

If you want to use the Devanagari-fonts within LaTeX, it is necessary to include `dev` as an option in the `\documentstyle` statement. Example: `\documentstyle[dev]{article}`. Within LaTeX the font size is changed automatically for titles, footnotes etc. However you can change the size yourself by using the standard LaTeX size-changing commands. Example: `{\large{\dn acchaa}}`. (The inner pair of braces is necessary for the preprocessor so it can recognize DN text).

3.5 Calling the Preprocessor

After an input file is completed, it has to be processed by the preprocessor before it can be fed to TEX or LaTeX. The preprocessor is called `devnag`. To process an input file, type:

```
devnag <inputfile> <outputfile>
```

The default file extension for an input file is `.DN` and for an output file `.TEX`. If no output filename is specified then the name of the output file will be the same as that of the input file. Example: `devnag hindi` will cause `devnag` to read from the file `hindi.dn` and write to the file `hindi.tex`. If you just type: `devnag` then you will be prompted for the names of the input and output file.

4 Advanced Topics

4.1 DN Delimiters

As said before, you can use dollar-signs as well as `{\dn␣` and `}` to delimit DN text. If you use dollar-signs, these signs cannot be used for other purposes, such as printing a dollar-sign or switching to mathematical mode. However, dollar-signs can be printed by means of `\char36` (in TEX) or `\symbol{36}` (in LaTeX). Switching to mathematical mode can be accomplished in an alternative way in LaTeX by using `\(` and `\)` as delimiters of mathematical mode.
The preprocessor does the following with text between DN-delimiters:
1. It translates the Devanagari-text into TEX-macro's.
2. It takes care of the switch to a Devanagari-font.
3. It takes care of adjustment of various TEX-parameters. (hyphenation, line-distance etc.)
One can imagine situations in which not all of these functions are needed or desired. The behaviour of the preprocessor after seeing a dollar-delimiter can be defined with the preprocessor commands `@dolmode1`, `@dolmode2` and `@dolmode3`. The preprocessor command `@dolmode1` activates functions 1,2 and 3. This is equivalent to `@dollars`. The preprocessor command `@dolmode2` activates the funtions 1 and 2, while the command `@dolmode3` only activates function 1. Using `{\dn#` and `}` to delimit DN-text has the same effect as using dollars after specifying the `@dolmode3` command, i.e. only function 1 is activated.
When the preprocessor sees `{\dn␣` it always switches to DN mode, without considering the context. This means that you cannot use the string '`{\dn␣`' within a 'verbatim'-environment. This also means that you should not 'hide' this string within the definition of another macro or command. For instance, the following construction will fail:

```
\def\bigdn{\dn \dnbig }
{\bigdn acchaa}
```

It should also be noted that the file-structure within DN text will not be preserved by the preprocessor. This means that single end-of-lines may appear in other places than in the original file. Because a space and an end-of-line are the same for TEX, this will in general not influence the final output. However, it is possible to leave the carriage returns in the same place before and after processing by the preprocessor by using the `@obeylines` preprocessor command. This command should not be confused with the `\obeylines` command, which is a TEX-command.

4.2 Hyphenation

Hyphenation is not often seen in devanagari texts. When it occurs, it is nearly always between parts of a compound word. In versions of TEX before version 3.0 it is not possible to use two seperate hyphenation tables within one text file. For these reasons hyphenation is not applied to DN text. This means that TEX will have more problems finding suitable line-breakpoints in DN text than in 'normal' text. Therefore, when there are large pieces of DN text, it is advised to make the line-width not too small. Because hyphenation is turned off, things like 'discretionary hyphens' (`\-`) and `\hyphenation`-commands are useless with DN text.

4.3 Line Distance

Because of the subscripts and superscripts, DN text needs more distance between lines than 'latin' text. The parameter in TEX that controls the distance between lines is called `\baselineskip`. This parameter is increased for DN text in the macro-files. However, TEX determines what the line distance whithin a paragraph has to be at the end of that paragraph. This may cause problems if a paragraph contains a mixture of DN text and 'latin' text and the paragraph does not end with DN text. The value of `\baselineskip` at the end of the paragraph will be the value for 'latin' text and the lines will be too close to each other. There are several solutions. In the first place you can give `\baselineskip` an explicit value before the paragraph ends. (See the macro-files for the values of `\baselineskip` for the different font-sizes.) Alternatively, you can add a piece of DN text at the end which contains just a paragraph end:

```
{\dn \par}
```

Even when a paragraph has only DN text, you have to take care that the paragraph end is included in the DN text. This means that the final `}` or `$` that ends the DN text must come *after* the empty line or the `\par` commmand that forms the paragraph end.

4.4 Ligature Commands

It is possible to enable or disable the use of ligatures for consonant groups. This is done by means of the preprocessor commands `@lig` and `@nolig`. The former enables ligatures while the latter disables them. Ligatures are specified by means of numbers. The table on the following page shows the ligatures which are available together with their numbers. It possible to specify more then one ligature with a `@lig` or `@nolig` command. In that case the numbers must be seperated by spaces. There also has to be a space between the command and the first number. Example:

```
@lig 20 43 90
```

There is no limit to the number of `@lig` or `@nolig` preprocessor commands in a file. When the use of a ligature for a consonant group is disabled, the consonants are printed after each other and a virama (hal) sign is used. When a ligature for a certain consonant group is disabled, then all the ligatures for consonant groups which are formed by adding characters to this consonant group are also disabled. Example: If ligature 3 is disabled then ligature is 10 is also disabled. Some 'standard' ligatures like ज्ञ cannot be disabled and are not in the table. In Sanskrit mode, all available ligatures are enabled. For Hindi mode the table shows which ligatures are enabled.

Table of Ligatures

#	consonants	lig.	hindi	#	consonants	lig.	hindi	#	consonants	lig.	hindi
1	क क	क्क	yes	36	ञ च	ञ्च		71	द र य	द्र्य	yes
2	क त	क्त	yes	37	ञ ज	ञ्ज		72	द व य	द्व्य	yes
3	क न	क्न		38	ट क	ट्क	yes	73	ध न	ध्न	
4	क म	क्म		39	ट ट	ट्ट	yes	74	न न	न्न	yes
5	क य	क्य	yes	40	ट ठ	ट्ठ	yes	75	प त	प्त	
6	क ल	क्ल		41	ट य	ट्य	yes	76	प न	प्न	
7	क व	क्व		42	ठ य	ठ्य	yes	77	प ल	प्ल	
8	क त य	क्त्य	yes	43	ड ग	ड्ग		78	ब न	ब्न	
9	क त व	क्त्व	yes	44	ड घ	ड्घ		79	ब ब	ब्ब	
10	क न य	क्न्य		45	ड ड	ड्ड	yes	80	ब व	ब्व	
11	क र य	क्र्य	yes	46	ड म	ड्म		81	भ न	भ्न	
12	क व य	क्व्य		47	ड य	ड्य	yes	82	म न	म्न	
13	क त र य	क्त्र्य	yes	48	ड ग य	ड्ग्य		83	म ल	म्ल	
14	घ न	घ्न		49	ड घ र	ड्घ्र		84	ल ल	ल्ल	
15	ङ क	ङ्क	yes	50	ड र य	ड्र्य	yes	85	व न	व्न	
16	ङ ख	ङ्ख	yes	51	ढ य	ढ्य	yes	86	व व	व्व	
17	ङ ग	ङ्ग	yes	52	त त	त्त	yes	87	श च	श्च	
18	ङ घ	ङ्घ	yes	53	त न	त्न		88	श न	श्न	
19	ङ ङ	ङ्ङ	yes	54	द ग	द्ग		89	श ब	श्ब	
20	ङ न	ङ्न	yes	55	द घ	द्घ		90	श ल	श्ल	
21	ङ म	ङ्म	yes	56	द द	द्द	yes	91	श व	श्व	
22	ङ य	ङ्य	yes	57	द ध	द्ध	yes	92	ष ट	ष्ट	
23	ङ क त	ङ्क्त		58	द न	द्न		93	ष ठ	ष्ठ	
24	ङ क य	ङ्क्य	yes	59	द ब	द्ब	yes	94	ष ट य	ष्ट्य	
25	ङ क ष	ङ्क्ष		60	द भ	द्भ	yes	95	ष ट व	ष्ट्व	
26	ङ ख य	ङ्ख्य	yes	61	द म	द्म		96	ष ट र य	ष्ट्र्य	
27	ङ ग य	ङ्ग्य	yes	62	द य	द्य	yes	97	स न	स्न	
28	ङ घ य	ङ्घ्य	yes	63	द व	द्व	yes	98	स र	स्र	yes
29	ङ घ र	ङ्घ्र	yes	64	द ग र	द्ग्र		99	ह ण	ह्ण	yes
30	ङ क त य	ङ्क्त्य		65	द घ र	द्घ्र		100	ह न	ह्न	yes
31	ङ क ष व	ङ्क्ष्व		66	द द य	द्द्य	yes	101	ह म	ह्म	yes
32	च च	च्च		67	द द व	द्द्व	yes	102	ह य	ह्य	yes
33	च ञ	च्ञ		68	द ध य	द्ध्य	yes	103	ह र	ह्र	yes
34	छ य	छ्य	yes	69	द ध व	द्ध्व	yes	104	ह ल	ह्ल	yes
35	ज र	ज्र	yes	70	द भ य	द्भ्य	yes	105	ह व	ह्व	yes