Devanāgarī for TEX
Version 2.14.1

Anshuman Pandey

31 December 2007

Contents

List of Tables

1 Introduction

The *Devanāgarī for TeX* (devnag) package provides a way to typeset high-quality Devanāgarī text with TeX. Devanāgarī is a script used for writing and printing Sanskrit and a number of languages in Northern and Central India such as Hindi and Marathi, as well as Nepali. The devnag package was originally developed in May 1991 by Frans Velthuis for the University of Groningen, The Netherlands, and it was the first system to provide support for the Devanāgarī script for TeX.

Several individuals have contributed to the devnag package over the years. Kevin Carmody proposed a method for managing variant glyphs. Marc Csernel revised the preprocessor to handle standard LaTeX commands. Richard Mahoney generated PostScript Type 1 versions of the Devanāgarī fonts. These fonts were later (in version 2.14) replaced with optimized Type 1 fonts created by Karel Píška. François Patte enhanced the LaTeX package by introducing a feature to produce citations in Devanāgarī. Zdeněk Wagner greatly improved the LaTeX package by revising macro definitions and catcodes, eliminating conflicts with other packages, and by introducing support for section headings and captions in Devanāgarī. Rob Adriaanse, Hans Bakker, Roelf Barkhuis, and Henk van Linde provided advice and support to Frans Velthuis when this package was being developed.

The devnag package is presently maintained by the following individuals:

John Smith	jds10@cam.ac.uk
Anshuman Pandey	apandey@u.washington.edu
Dominik Wujastyk	d.wujastyk@ucl.ac.uk
Zdeněk Wagner	zdenek.wagner@gmail.com
Kevin Carmody	i@kevincarmody.org

2 Project Information

The *Devanāgarī for TEX* package is now a project officially housed at Sarovar. The homepage is

`http://devnag.sarovar.org/`

This package is available from the project homepage at Sarovar and from the Comprehensive TEX Archive Network (CTAN). The CTAN path for the package at the primary UK TUG site is

`ftp://ftp.tex.ac.uk/tex-archive/language/devanagari/velthuis/`

Please use the tracking system at the project homepage at Sarovar to submit feature requests, bugs, comments, and questions to the development team.

3 Producing Devanāgarī Text with TEX

Devanāgarī text may be included in any TEX document. There are three steps to producing Devanāgarī text with TEX. First, since TEX does not support Devanāgarī natively, it is necessary to type Devanāgarī using 7-bit (ASCII) roman characters that represent Devanāgarī characters. Secondly, transliterated Devanāgarī text must be entered within devnag-specific delimiters. These delimiters allow the preprocessor to recognize the Devanāgarī sections of the TEX document. Third, the transliterated input must be converted by the preprocessor into a format that TEX understands. The preprocessor scans the document for devnag delimiters. Once it finds a delimiter, the program operates on the text only within the scope of the delimiter. All other document text, with the exception of TEX macros and devnag-specific preprocessor directives, is ignored.

Shown below is a Devanāgarī passage followed by the 7-bit transliterated input that produced it.

धर्मक्षेत्रे कुरुक्षेत्रे समवेता युयुत्सवः।
मामकाः पाण्डवाश्चैव किमकुर्वत संजय॥

```
{\dn dharmak.setre kuruk.setre samavetaa yuyutsava.h | \\
maamakaa.h paa.n.davaa"scaiva kimakurvata sa.mjaya ||}
```

3.1 Macros and Font Definition Files

`dnmacs.tex` This file contains Plain TEX macros for devnag and various font-sizing commands. It must be loaded at the beginning of the document with the command `\input dnmacs`.

`devanagari.sty` This file provides LATEX support for devnag. It must must be loaded in the preamble of the document with the command `\usepackage{devanagari}`. Section 7.9 discusses advanced package options that may be declared with `devanagari.sty`. The associated font definition files `udn*.fd` provide NFSS support for LATEX for the dvng fonts.

`dev.sty` This file is kept for compatibility with old documents. It merely loads `devanagari.sty`. Do not use it in new documents.

`dev209.sty` This file provides legacy support for the obsolete LaTeX 2.09. It should not be used.

3.2 Text Delimiters

The preprocessor recognizes the text it is to act upon by use of delimiters, of which there are two types. The basic delimiter is the `\dn` macro. This delimiter is used by enclosing Devanāgarī text between `{\dn ... }`, eg. `{\dn acchaa}`.

The second delimiter is the `$` character. Devanāgarī text is enclosed between `$ ... $`, eg. `$acchaa$`. The `@dollars` preprocessor directive must be specified to activate this delimiter (section 5.1).

The first delimiter is recommended especially for large blocks of Devanāgarī text. The second delimiter is useful when there is a need to switch often between Devanāgarī and roman text. Any text outside of delimiters is not parsed by the preprocessor.

There are very few restrictions on what may be placed between the delimiters. The 7-bit Velthuis encoding shown in Table 1, all punctuation marks, and all TeX macro commands are acceptable input. The preprocessor will produce a warning for unrecognized input characters and commands.

3.3 Example Input Files

Two sample Devanāgarī documents are bundled with this distribution. Please refer to the contents of these files for examples of producing a Devanāgarī document. The file `misspaal.dn` contains an excerpt from the Hindi short story *Miss Pal* by Mohan Rakesh. The file `examples.dn` contains some advanced examples of Devanāgarī typesetting. Shown below are two small examples of Plain TeX and LaTeX documents with Devanāgarī text.

```
% Sample TeX input file
\input dnmacs
{\dn devaanaa.m priya.h}
\bye
```

```
% Sample LaTeX input file
\documentclass{article}
\usepackage{devanagari}
\begin{document}
{\dn devaanaa.m priya.h}
\end{document}
```

The filename of the TeX document that contains Devanāgarī should be given a `.dn` extension. The preprocessor will produce a filename with a `.tex` extension after processing the input file.

4 Input Encoding

Devanāgarī text is prepared using a 7-bit (ASCII-based) transliterated input encoding in which Devanāgarī characters are represented by Roman characters. The input encoding for devnag was developed by Frans Velthuis with the objective to keep the format of the input source text as close as possible to the accepted scholarly practices for transliteration of Devanāgarī. The Velthuis encoding is widely used and has been adapted by other Indic language TeX packages, and also serves as the basis of other Indic transliteration schemes.

4.1 Supplemental Notes

Attention should be paid to the following points:

1. There are different ways to produce consonant conjuncts. For example, the sequence `ktrya` can be represented as क्‌त्र and as क्‌त्र्य. The creation of conjuncts may be controlled through the use of the preprocessor directives `@sanskrit`, `@hindi`, and `@modernhindi`, and more strictly through the `@lig` directive. Please refer to Table 5 for a list of supported conjuncts.

2. There are two different ways to produce long vowels: typing the short vowel twice or by capitalizing the short vowel, eg. `aa` or `A` for आ.

3. Aspirated consonants may be produced alternately by capitalizing the voiceless counterpart. For example, the standard encoding for भ is `bha`, but it may also be produced by `Ba`; घ is `gha` or `Ga`; etc.

4. For words which have two successive short vowels, a sequence of brackets `{}` may be used to separate the vowels, eg. प्रउग `pra{}uga`, as opposed to प्रौग `prauga` . This is required because the combinations `ai` and `au` represent the dipthongs ऐ and औ.

5. The use of uppercase letters to indicate long vowels may be preferred in cases where ambiguity might arise. When encoding a word like कई, the sequence `kaii` will produce the incorrect form कैइ, while `kaI` will yield the correct form कई. A sequence of brackets, as in the previous note, will also produce the correct form, eg. `ka{}ii`.

6. The standard ligatures क्ष, ज्ञ, and त्र are produced by `k.sa`, `j~na`, and `tra`.

7. Candrabindu may be encoded either as a slash, as given in Table 1, or by `~m`.

8. 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 . The underscore character `_` also produces a *virāma*.

9. In many Hindi words an `a` needs to be written between consonants to produce the correct spelling, otherwise a conjunct consonant will be produced. The correct form of Hindi करना is `karanaa`, not `karnaa`, which produces कर्ना.

10. Tab characters in the input file, which previously were treated as fatal errors, are now silently converted to spaces.

5 The Preprocessor

The ANSI C program `devnag.c` is a preprocessor that reads transliterated Devanāgarī input delimited by `\dn` and converts it into a form with which TEX is familiar. To use the preprocessor, `devnag.c` must be compiled into an executable.

The preprocessor handles the details of character placement such as the alignment of vowel diacritics and consonant ligatures. The rest of the layout, however, must be managed by the user. The preprocessor is invoked as

`devnag` *in*[`.dn`] (*out*[`.tex`])

The default file extension for an input file is `.dn` and for an output file `.tex`. The output filename is optional. If an output filename is not specified, the preprocessor will name it after the input file.

For example, typing `devnag hindi` will instruct the preprocessor to read from the file `hindi.dn` and write to the file `hindi.tex`. The program will prompt for the names of the input and output files if they are not given in the command-line.

The preprocessor tries to be safe yet flexible. The extensions are not forced but the input and output file names must be distinct. The exact algorithm (since version 2.14) is:

VOWELS			
a	a	अ	
ā	aa	आ	ा
i	i	इ	ि
ī	ii	ई	ी
u	u	उ	ु
ū	uu	ऊ	ू
r̥	.r	ऋ	ृ
r̥̄	.R	ॠ	ॄ
l̥	.l	ऌ	ॢ
l̥̄	.L	ॡ	ॣ
e	e	ए	े
ai	ai	ऐ	ै
o	o	ओ	ो
au	au	औ	ौ

SIGNS		
anusvāra	.m	ं
candrabindu	/	ँ
visarga	.h	ः
avagraha	.a	ऽ
virāma	&	्
candra a	~a	ॅ
candra o	~o	ॉ
AUM	.o	ॐ
daṇḍā	|	।
double *daṇḍā*	||	॥
'eyelash' *repha*	~r	ऱ
abbreviation	@	॰
ellipsis	#	ऻ
period	..	॥

CONSONANTS		
ka	ka	क
kha	kha	ख
ga	ga	ग
gha	gha	घ
ṅa	"na	ङ
ca	ca	च
cha	cha	छ
ja	ja	ज
jha	jha	झ
ña	~na	ञ
ṭa	.ta	ट
ṭha	.tha	ठ
ḍa	.da	ड
ḍha	.dha	ढ
ṇa	.na	ण
ta	ta	त
tha	tha	थ
da	da	द
dha	tha	ध
na	na	न
pa	pa	प
pha	pha	फ
ba	ba	ब
bha	bha	भ
ma	ma	म
ya	ya	य
ra	ra	र
la	la	ल
va	va	व
śa	"sa	श
ṣa	.sa	ष
sa	sa	स
ha	ha	ह

CONSONANTS		
qa	qa	क़
k̲ha	.kha	ख़
ġa	.ga	ग़
za	za	ज़
ṛa	Ra	ड़
ṛha	Rha	ढ़
fa	fa	फ़
ḷa	La	ळ

DIGITS		
0	0	०
1	1	१
2	2	२
3	3	३
4	4	४
5	5	५
6	6	६
7	7	७
8	8	८
9	9	९

Table 1: The Velthuis Encoding Scheme

1. If the source filename does not have the default extension, it is appended. If such a file does not exist, the preprocessor tries again with the original file name.
2. If the destination filename was given, its extension is checked. It is explicitly forbidden for the destination file to have the `.dn` extension. If the filename is equal to the name of the source file, `.tex` is appended, otherwise the file name is used as is.
3. If the destination filename was not given, it is based upon the source filename. If the source filename contains the `.dn` extension, it is stripped of. Afterwards the `.tex` extension is appended.

The algorithm was designed to prevent typing errors but it is far from foolproof. The filesystem properties are not examined and all filename tests are case insensitive. The file names are not expanded. For instance, if the working directory is `/some/path`, then `x.y`, `./x.y`, `../path/x.y` as well as `/some/path/x.y` refer to the same file but they will be treated as different by the preprocessor. Moreover, the algorithm does not try to follow symlinks neither does it examine inodes in order to discover hard links.

If `devnag` is invoked with the `-v` option, it will display the version number and then exit.

5.1 Preprocessor Directives

The preprocessor creates a TEX file from the input file by acting on two different parts of the input: directives and transliterated input. As it creates a TEX file from the input file, the preprocessor can be told to modify the way in which it operates by means of special commands called directives. Directives are optional commands to the preprocessor that instruct it to process the input text in a given manner, such as permitting hyphenation, suppressing the use of certain ligatures, etc. Directives do not affect typesetting or layout.

Directives must occupy a line by themselves and always begin with the character `@`. Directives may occur anywhere in the document, but not within Devanāgarī delimiters (where `@` is the continuation symbol `°`). Directives specific to a particular passage of text should appear immediately before that passage; directives applying to the entire file should appear just before the first line of actual text to be typeset.

Since `@` is a perfectly legal character in TEX, lines beginning with `@` that do not match any valid `@` command are flagged with a warning, but processing of the file continues. (In the somewhat unlikely event that there is actually a need to have a line of TEX text consisting exactly of, for example, `@hindi`, the preprocessor may be fooled by typing `{}@hindi` or `{@hindi}`.

New "negative" commands have been added to reverse the effect of most existing commands: thus it is now possible to enable or disable specific features for specific passages of text, eg. `@hindi` may be disabled with `@sanskrit`.

In previous releases of `devnag`, the preprocessor would split long lines in its output. The `@obeylines` command was provided to disable this feature. The line-splitting feature has been disabled, so that the preprocessor now outputs lines as they appear in the input file. The `@obeylines` directive is no longer recognized by the preprocessor as a valid directive and will be ignored; it will therefore be typeset as a part of the text of the document.

`@sanskrit` The `@sanskrit` directive is the default mode for the preprocessor. This command may also be used to reinstate the default behavior of the preprocessor after the use of `@hindi` and `@modernhindi`. See Table 5 for a list of conjuncts and the forms produced in `@sanskrit` mode.

`@hindi` The `@hindi` directive switches the preprocessor to Hindi mode. The difference between the Sanskrit and Hindi modes is that in Sanskrit mode the full forms of conjuncts are used, whereas in Hindi mode certain simplified forms are used instead, eg. च्च in place of च्च. See Table 5 for a list of conjuncts and the forms produced by `@hindi`. Additionally, in Sanskrit mode a *virāma* is automatically added at the end of a word if it ends in a consonant, while in Hindi mode the preprocessor assumes the presence of the

inherent vowel *a*. The directive `@hindi`, if used, must precede the `@lig` and `@nolig` commands. See Table 5 for a list of conjuncts and the forms produced in `@hindi` mode.

`@modernhindi` This directive switches the preprocessor to Hindi mode, similar to `@hindi`, but uses far fewer Sanskrit-style ligatures. Conjuncts are created from half-consonant forms wherever possible. See Table 5 for a list of conjuncts and the forms produced in `@modernhindi` mode.

`@dollars` / `@nodollars` In addition to the `{\dn ... }` delimiters, Devanāgarī text can also be delimited by dollar signs, eg. `$acchaa$`. The directive `@dollars` instructs the preprocessor to switch to dollar mode and to recognize `$` as a delimiter. In dollar mode, the dollar sign cannot be used for other purposes, such as printing a dollar sign or switching to math mode. Dollar signs may be printed by through low-level font commands, eg. `\char36` in Plain TEX or `\symbol{36}` in LATEX. Switching to math mode when `@dollars` is active is accomplished by using `\(` and `\)` as math delimiters.

`@dolmode0` / `@dolmode1` / `@dolmode2` / `dolmode3` When `@dollars` is active, the behavior of the preprocessor can be modified further through the `@dolmode0`, `@dolmode1`, `@dolmode2`, and `@dolmode3` directives.

```
@dolmode0   $acchaa$   →   $acchaa$
@dolmode1   $acchaa$   →   \dn aQCA
@dolmode2   $acchaa$   →   \pdn aQCA
@dolmode3   $acchaa$   →   aQCA
```

Alternately, `@dolmode3` can be mimicked using the macro `\dn#`, `{\dn# acchaa}` → `aQCA`. Also, the Plain TEX macro `\pdn` changes the current font into Devanāgarī in the current size. LATEX automatically adjusts the font sizing for Devanāgarī to the document font size.

`@lig` / `@nolig` Certain conjuncts may be enabled or disabled by using the directives `@lig` and `@nolig`. The former enables conjuncts while the latter disables them. Supported conjuncts are assigned codes and are showned in Table 5. For example, the command `@nolig 2` produces क्त instead of क्त from the input `kta`.

More than one conjunct may be specified with a `@lig` or `@nolig` command, eg. `@lig 20 43 90`. There is no limit to the number of `@lig` or `@nolig` directives issued within a document. However, when a certain conjunct is disabled, all other conjunct combinations involving the disabled conjunct are also disabled. For example, if conjunct 3 क्न (`kna`) is disabled then conjunct 10 क्न्य (`knya`) will also be disabled.

Some basic Devanāgarī conjuncts like क्ष, ज्ञ and त्र cannot be disabled and are not shown in Table 5. Also, most two element ligatures involving *ra*, eg. क्र, ग्र, and प्र cannot be disabled.

`@hyphen` / `@nohyphen` These directives control hyphenation of Devanāgarī text. They provide the ability to enclose a section of particularly dense text between `@hyphen` and `@nohyphen` without affecting other parts of text in the document. To type a hyphen in a Devanāgarī document simply type `-`. For example, typing `{\dn dive-dive}` will produce दिवे-दिवे. Please refer to section 7.3 for more information.

`@tabs` / `@notabs` The `@tabs` directive instructs the preprocessor to recognize the `&` character as a TEX tabular character, not as a method of encoding *virāma*. The command `@notabs` resets this feature. If `&` appears word medially, eg. `.sa.t&"siraa.h`, it will be processed as a *virāma* even if `@tabs` is specified. This avoids possible incompatibility issues with legacy documents. The `_` character now doubles as a way of producing *virāma*, particularly if `@tabs` is used, eg. `pari.sad_`

`@vconjuncts` / `@novconjuncts` A change has been introduced in the way text like `{\dn .sa.t"siraa.h}` is processed; specifically, instances where an *i* vowel is associated with a consonant sequence containing a *virāma*. The previous behavior was to treat the consonant sequence as if it were a normal conjunct by placing the vowel diacritic before the sequence as a whole, eg. षट्ि शराः. The majority opinion seems to be that this is undesirable, and that the vowel symbol should follow the consonant to which the *virāma*

is attached, eg. षट्शिराः. This is now the default behavior of the preprocessor, but the `@vconjuncts` directive has been implemented to reinstate the previous output method. The command `@novconjuncts` resets this feature.

5.2 Protecting Text from Conversion

The preprocessor will convert all text found in a Devanāgarī environment. Text may be protected from the preprocessor using the < and > delimiters. In the example below, the font command between the angle brackets will be ignored by the preprocessor, but will be removed from the output file.

For example, with `{\dn dharmak.setre <\font\zzz=dvng10 at 18pt> kuruk.setre}` the preprocessor will operate on `dharmak.setre` and `kuruk.setre`, but will ignore `\font\zzz=dvng10 at 18pt` because it occurs within the < and > delimiters.

5.3 Embedding Roman Text within Devanāgarī Text

The font commands `\rsize` (Plain TEX) and `\NormalFont` (LATEX) put Roman text within Devanāgarī Text. This is useful in conjuction with the text protection delimiters < and >. Examples:

Plain TEX: `{\dn naama {\rsize <John>}}`

LATEX: `{\dn naama {\NormalFont <John>}}`

By default, Roman text using `\rsize` and `\NormalFont` is printed in a Computer Modern font whose size matches the current Devanāgarī font. Non-Devanāgarī punctuation marks, such as comma, question mark, and exclamation mark, and Arabic numerals are automatically printed in this font. To change this font, redefine the `\rsize` or `\NormalFont` command.

5.4 Breaking Pre-Defined Conjuncts

The preprocessor will automatically produce predefined ligatures from certain sequences of consonants. Placing the + character between two consonants prevents any predefined ligature representing those consonants from being produced. Instead a conjunct will be created from half-forms, or, if half-forms do not exist, full forms stopped with *virāma* will be used.

For example, the sequence `kha` will produce ख. Using + to break the sequence – `k+ha` – will create क्ह.

The use of + is similar to the use of `{}`. For example, write `t{}ha` to produce त्ह, if desired instead of `tha` थ.

The + character can be used independently of the `@lig`/`@nolig` directives, and it can disable any conjunct. Note that the + character only disables single occurrences of a conjunct. To disable all occurrence of a conjunct use the `@lig` directive.

5.5 Supported LATEX Commands

The preprocessor recognizes some LATEX macros with arguments. The following command types are legal within delimiters.

- Font commands: Standard TEX size-changing commands, eg. `\small`, `\large`, `\huge`.
- Environments, including the three table environments: `tabular`, `supertabular`, and `longtable`. Note: To use table environments within delimited text, the `@tabs` directive must be specified in order to

enable the use of the ampersand as a tab marker instead of a marker for *virāma*. Refer to Section 5.1 for more details on preprocessor directives.

- Spaces: `\hspace`, `\hspace*`, `\vspace`, `\vspace*`, `\addvspace`, `\setlength`, `\addtolength`, `\enlargethispage`, `\enlargethispage*`, and `\\[`*n*`]`. Plain TEX commands may also be used when placed between brackets: `{\hskip }`, `{\vskip }`, `{\vadjust }`, and `{\kern }`.
- Counters: `\setcounter`, `\stepcounter`, `\addtocounter`, and `\refstepcounter`. Page numbering in Devanāgarī is also available through the `dev` counter.
- Boxes and rules: `\parbox`, `\makebox`, `\framebox`, `raisebox`, and `\rule`.
- References: `\label`, `\ref`, `\pageref`, `\index`, `\cite`, and `\bibitem`. If the argument of the `\index` command is in Devanāgarī, it will appear in Devanāgarī in the index file.
- File commands: `\input` and `\include`.
- Roman text may also be embedded within Devanāgarī delimited text as long as the Roman does not exceed the length of one line. Use `{\rm ... }` to produce embedded Roman text.

5.6 Using Custom LATEX Commands

LATEX users take advantage of numerous commands from various packages. All packages will never be supported directly but solution is easy. As a matter of fact, all control sequences without arguments or taking Devanāgarī text as argument can be used without problem. Suppose the you want to colorize text ॐ तत्सत् using the `\textcolor` command from the COLOR package. The bare word *red* must not be converted. Conversion can be suppressed by mechanism described in section 5.2. One can write e. g.:

```
{\dn \textcolor<{red}>{.o tatsat}}
```

It is, however, preferable to separate meaning and form. If a macro, e. g. `\myemph` is defined, it is possible to decide later how the text will be emphasized. The text will then be entered as follows:

```
\def\myemph#{\textcolor{red}}
{\dn \myemph{.o tatsat}}
```

Using such an approach one does not need any escaping mechanism at all.

6 Devanāgarī Fonts

The devnag package provides three font families in addition to the Standard family: Bombay, Calcutta, and Nepali. All families are available in regular, oblique, bold, bold oblique, and pen shapes and weights. The Bombay, Calcutta, and Nepali families provide variant glyphs which are predominant regional forms for certain characters, as shown in Table 4.

- `\dnbombay` switches to the Bombay family
- `\dncalcutta` switches to the Calcutta family
- `\dnnepali` switches to the Nepali family
- `\dnoriginal` switches back to the default regular family
- `\dnpen` switches to the Pen family
- `\dnpenbombay` switches to the Bombay Pen family
- `\dnpencalcutta` switches to the Calcutta Pen family

- \dnpennepali switches to the Nepali Pen family

The oblique, bold, and bold oblique shapes and weights are produced using standard LaTeX macros. Oblique is obtained with either of the \textit{} or \itshape font-changing commands. Bold is obtained with either \textbf{} or \bfseries. Bold oblique requires a combination of the bold and oblique commands, such as \bfseries\itshape.

To use bold, oblique, and bold oblique varieties in Plain TeX, use the macros \dnbf and \dnit. The regional families are accessed using the macro commands described above. See dnmacs.tex for further information.

Font size may be controlled in LaTeX with the standard font sizing commands. In Plain TeX, font size may be controlled with the following macros: \dnsmall, \dnnine, \dnnormal, \dnhalf, \dnbig, \dnlarge, and \dnhuge. See dnmacs.tex for further information.

6.1 Bombay-Style Fonts

The family name for the Bombay Devanāgarī fonts is dnb. To access this family, use the command \dnbombay after the \dn macro. Standard LaTeX font commands like \fontfamily{dnb} and \usefont{U}{dnb}{}{} may be used to access the Bombay fonts, however, these commands conflict with the preprocessor. Access to the Bombay family within Devanāgarī environments should be restricted to the \dnbombay macro. Use the command \dnoriginal to return to the standard Devanāgarī font.

6.2 Calcutta-Style Fonts

The family name for the Calcutta Devanāgarī fonts is dnc. In roder to access this family, use the command \dncalcutta after the \dn macro. Standard LaTeX font commands like \fontfamily{dnc} and \usefont{U}{dnb}{}{} may be used to access the Calcutta fonts, however, these commands conflict with the preprocessor. Access to the Calcutta family within Devanāgarī environments should be restricted to the \dncalcutta macro. Use the command \dnoriginal to return to the standard Devanāgarī font.

6.3 Nepali-Style Fonts

The family name for the Nepali Devanāgarī fonts is dnn. To access this family, use the command \dnnepali after the \dn macro. Standard LaTeX font commands like \fontfamily{dnn} and \usefont{U}{dnn}{}{} may be used to access the Nepali fonts, however, these commands conflict with the preprocessor. Access to the Nepali family within Devanāgarī environments should be restricted to the \dnnepali macro. Use the command \dnoriginal to return to the standard Devanāgarī font.

6.4 Devanāgarī Pen Fonts

The Devanāgarī Pen family is a simple modification of the Standard face created by Tom Ridgeway, which resembles Devanagari written with a pen. Standard Pen fonts are available as the family dnp and may be accessed within Devanāgarī environments with the command \dnpen. The Pen family for the Bombay style are available as the family dnpb, and may be accessed with the command \dnpenbombay. The Pen family for the Calcutta style is called dnpc and may be accessed with the command \dnpencalcutta. The Pen family for the Nepali style is called dnpn and may be accessed with the command \dnpennepali. Use the command \dnpen to return to the standard Devanāgarī Pen font.

6.5 Default Devanāgarī Font (LaTeX Only)

The Devanāgarī package provides options `bombay`, `calcutta`, `nepali`, `pen`, `penbombay`, `pencalcutta`, and `pennepali` so as to set the corresponding font as the default one. It may seem that using `\dncalcutta` at the beginning of the document is sufficient. However, as we will show later in this document, the Devanāgarī package may create automatically some captions as well as a running head. When producing such texts, LaTeX is set to use Roman fonts and the automatic text switches to Devanāgarī just by `\dn`. You would thus see अध्याय in the normal text but अध्याय in automatic captions which is undesirable. The package options inform which font style should be used as default.

It is also possible to change the default font by defining macro `\dnfamilydefault`.

The font switching commands described in the previous subsections can be used for local changes of the style.

6.6 PostScript Type 1

The package now includes Type 1 fonts created by Karel Píška using an analytic fit. The fonts included here are a subset of his release of Indic Type 1 fonts that are available from CTAN:fonts/ps-type1/indic and licensed under GPL.

An accurate analytic fit of outline contour curves taken from the METAPOST output helps to avoid artifacts produced by an autotracing bitmap approach. It allows to keep preciseness of calculations and to produce the outline fonts faithful, optimal (to minimize their space amount) and hinted. Therefore the results are not only more precise than fonts presented earlier but also occupy a smaller place even if they include hinting additionally. The distribution contains the PFB and TFM files both with accurate glyph widths. The user UniqueID are present to distinguish fonts during download process in PostScript printers and other RIP devices.

The AFM files have be considered as derived files not usable for TFM creation, because the AFM2TFM program has a feature to round the glyph widths and is not able to reproduce the original metrics. Use these files only with tools that explicitly require AFM.

To use the Type 1 fonts with `dvips` and pdfTeX, it is in modern TeX distributions sufficient to run `mktexlsr` or `texhash` and then issue:

```
updmap --enable MixedMap=dvng.map
```

If you do not have `updmap`, you must edit the local dvips `psfonts.map` file to contain a reference to `dvng.map`; or copy the contents of `dvng.map` into `config.ps`.

Detailed installation instructions can be found in the README file in the root directory of the CVS working copy or in the `doc/generic/velthuis/` directory in the release package.

7 Special Topics

7.1 Delimiter Scope

The LaTeX font-size commands may be used within Devanāgarī delimited text, however, as a general rule, the font-size command should follow the `\dn` delimiter, otherwise the font definition commands of `\dn` will be overridden. For example, items 1, 2, and 3 below produce the correct forms, but 4 does not:

1. `{\dn \large acchaa}` → अच्छा
2. `{\dn {\large acchaa}}` → अच्छा
3. `{\large {\dn acchaa}}` → अच्छा
4. `{\large \dn acchaa}` → अचचह्अअ

7.2 Line Spacing

Due to the super- and subscript characters of the Devanāgarī script, the default line spacing (leading) often needs to be increased to prevent the crowding of lines. The parameter `\baselineskip` (`\linespread` for LaTeX) controls the line spacing.

TeX determines and adjusts the value of `\baselineskip` after it finishes processing a paragraph. If a paragraph contains a mixture of Devanāgarī and Roman text, and ends with Roman text, then TeX will set the value of `\baselineskip` according to the Roman text. This may result in crowding of Devanāgarī text.

There are, however, solutions to this. An explicit value can be assigned to `\baselineskip` before the paragraph ends. The macro file `dnmacs.tex` shows examples of the value of `\baselineskip` at different font sizes. Default line spacing is also set in `devanagari.sty`. Alternately, 'dummy' devnag text containing `\par` can be placed at the end of the paragraph, eg. `{\dn \par}`.

Even when a paragraph has only devnag text, the paragraph-end command must be included within devnag text, meaning that the closing delimiter, which ends the devnag text, must follow the empty line or the `\par` command that forces the paragraph to end.

7.3 Hyphenation

The devnag package does more or less all that needs to be done from the point of view of hyphenating Sanskrit in Devanāgarī through the `@hyphen` and `@nohyphen` directives, which are discussed in section 5.1. If hyphenation is off, then there are no hyphens, and very stretchy inter-word space. This is acceptable for ragged-right settings or for text in verse form, but may produce poor results in right-justified prose text, especially if the given passage contains long compound words. If hyphenation is on then discretionary hyphens are set between all syllables.

7.4 Captions and Date Formats (LaTeX only)

The language modules of the babel package change captions texts and date formats. Although devanagari.sty is not a babel module, similar mechanism is implemented here. Macros `\datehindi` and `\datemodernhindi` enable Europian style Hindi date generated by the standard `\today` command. The "traditional" and "modern" variants comtain the same names of the months, they differ only in the ligatures used. You should therefore use `\datemodernhindi` in documents processed with `@modernhindi`. The names of the months used in the definition of `\datemodernhindi` are summarized in Table 2.

1	जनवरी	7	जुलाई
2	फ़रवरी	8	अगस्त
3	मार्च	9	सितम्बर
4	अप्रैल	10	अक्तूबर
5	मई	11	नवम्बर
6	जून	12	दिसम्बर

Table 2: Names of the months in the definition of `\datemodernhindi`

If you use `\datehindi` or `\datemodernhindi`, you can also use the macros `\hindidatearabic` and `\hindidatedevanagari` to control whether the day number in `\today` is printed in Arabic or Devanāgarī, respectively. The `\hindidatedevanagari` mode is the default. The macros `\cmnum` and `\dnnum` have no effect in this case.

Macro	Caption
\abstractname	सारांश
\appendixname	परिशिष्ट
\bibname	संदर्श ग्रन्थ
\ccname	
\chaptername	अध्याय
\contentsname	विषय-सूची
\enclname	
\figurename	चित्र
\headpagename	पृष्ठ
\headtoname	
\indexname	सूची
\listfigurename	चित्रों की सूची
\listtablename	तालिकाओं की सूची
\pagename	पृष्ठ
\partname	खण्ड
\prefacename	प्रस्तावना
\refname	हवाले
\tablename	तालिका
\seename	देखिए
\alsoname	और देखिए
\alsoseename	और देखिए

Table 3: Modern Hindi captions

The captions are similarly switched to Hindi by `\captionshindi` or `\captionsmodernhindi`, respectively. Again the texts differ only in the ligatures used. The captions for the modern Hindi variant are given in Table 3.

The macros for the LETTER class are left intentionally empty. The idea of the babel package is to prepare a universal template for business letters using a set of macros. The header of the letter would make use of the `\headtoname` macro which will produce "To: Mr. Kumar" in English letters and "Komu: pan Kumar" in Czech letters. If we simply defined `\headtoname` to को, the universal template would put it before the name which would be wrong. Hindi requires different word order, namely श्री कुमार को. The universal templates are thus useless in Hindi and the letter template must be redesigned almost from scratch. It therefore makes no sense to define the letter macros.

Two package options are provided: `hindi` and `modernhindi`. If used, they cause the `\dn` command to switch the caption text and date format as well. The date format and captions may be switched back by macros `\dateenglish`, `\dateUSenglish`, and `\captionsenglish`.

7.5 Customizing the date and captions (LaTeX only)

The user might prefer different caption texts. If just a few texts are to be changed, they can simply be redefined in the main document, for instace by:

```
\def\indexname{{\dn anukrama.nikaa}}
```

This redefinition must appear **after** `\captionshindi` or `\captionsmodernhindi` was invoked.

It is also possible to change all definitions. The source texts in the Velthuis transliteration can be found in the `documentation` directory (`$TEXMF/doc/generic/velthuis`) in file `captions.dn` with some suggested

variants in comments. You can either put modified definitions to your main document (after devanagari.sty) or to a package of your own. Remember that the preprocessor will not see your package, you must preprocess it separately. Your package must either reside in the same directory as your document or in some directory which is searched by LaTeX. In the latter case you will have to rebuild the database by running mktexlsr or texhash in many TeX distributions.

Do not put your packages to standard distribution directories. You may lose them when upgrading your TeX distribution.

Customization of the captions texts is easier in the babel module. The module is described in section 9.

7.6 Using देवनागरी in Sections and References (LaTeX only)

All macros necessary for typesetting Devanāgarī text are robust. The section/chapter titles as well as figure and table captions can contain Devanāgarī words. However, the font is changed to the standard document font before the title is typeset. It is therefore mandatory to use `\dn` even if the section title appears inside the `\dn` environment. Thus, `\chapter{{\dn mis paal}}` will be printed correctly while `{\dn\chapter{mis paal}}` will always create garbage text. Section numbers as well as page numbers will be printed in Roman numerals.

7.7 Devanāgarī and Arabic Numerals

Except when they are represented by commands, numerals are printed as Arabic numerals by default. The command `\dnnum` switches to Devanāgarī numerals. The command `\cmnum` switches back to Arabic numerals.

Numerals represented by commands are printed as Devanāgarī numerals by default. If you want to use `\cmnum` or `\dnnum` to control whether numerals within commands are printed as Arabic or Devanāgarī, use the command `\rn` around the numeral command. If the command contains a mix of letters and numerals, you may need to redefine the command with `\rn` around the numeral part.

Without `\rn`, numerals in commands are always printed as Devanāgarī, even if `\cmnum` is in effect. For example, in Plain TeX, the macro `\folio` contains the current page number. To make this respect `\cmnum`, use a command such as `{\dn p.r.s.tha \rn{\folio}}`. LaTeX users can use the techniques described in the next section.

Arabic numerals are printed in the font specified by the commands `\rsize` (Plain TeX) or `\NormalFont` (LaTeX). This font is also used for non-Devanāgarī punctuation marks. By default, this is a Computer Modern font whose size matches the current Devanāgarī font. To change this font, redefine the `\rsize` or `\NormalFont` command.

7.8 Devanāgarī Page Numbers and Other Counters (LaTeX only)

Changing page numbers and other counters to print Devanāgarī numerals is possible with the `devanagari` numeral style. Page numbers can be set to Devanāgarī with the following declaration:

```
\pagenumbering{devanagari}
```

The default is `\pagenumbering{arabic}`.

You can use the command `\devanagari` as an alternative to `\arabic` to control the style of a counter anywhere in the document. For example, `\devanagari{page}` prints the current page number in Devanāgarī numerals, while `\arabic{page}` prints it in Arabic numerals.

You can change the automatic display of other counters to Devanāgarī by redefinition macros. For example, to change section numbering to Devanāgarī, redefine `\thesection`:

```
\renewcommand\thesection{\devanagari{section}}
```

The macros \cmnum and \dnnum have no effect on counters, unless you also use the command \rn. To make page numbering respect the settings of \cmnum and \dnnum, use \rn as follows:

```
\renewcommand\thepage{\dn p.r.s.tha \rn{\arabic{<page>}}}
```

Note that page is enclosed within angle brackets. This is required because it is within the scope of the \dn command, and the preprocessor does not recognize counter names as commands.

Devanāgarī page numbering unfortunatelly conflicts with some versions of the HYPERREF package. The source of the problem has not be resolved yet. Use this feature with caution.

7.9 Category Codes

TEX assigns a category code (\catcode) to each character. For example, normal characters are assigned to category 11, and because the backslash belongs to category 0, it is treated as the first character of macro commands.

The fonts in the devnag package use characters with codes below 32. In previous releases of the package the category of these characters was to 11. However, these catcode assignments caused conflicts with some packages and with tables where tab characters were used. Most of these problems could be solved at the macro level, but unfortunately not all of them. The most serious problem is that words like वक्त do not get correctly transfered from section titles to the table-of-contents.

A modification of the preprocessor was necessary to resolve this issue. As a result, a change of character categories is no longer needed. The output of the revised preprocessor is compatible with previous releases of devanagari.sty. This fix solves the table-of-contents problem, but not the conflicts. The new devanagari.sty is still able to process files generated by the previous versions of the preprocessor.

To indicate which version of the preprocessor was used to process a given Devanāgarī file, a string is written to the beginning of the output TEX file. The macro definition \DevnagVersion is written to the first line of the output file and indicates the preprocessor version. If the whole document is present in a single file, the definition will appear before reading the macro package. The package then changes its behaviour according to the existence or non-existence of the above mentioned macro. If the macro is defined, no categories are changed. If the macro is undefined, the devanagari package assumes that it is processing an output from an older version of the preprocessor and the categories of the characters are changed.

The nocatcodes option is intended for use with files produced by the old preprocessor. This option changes the categories only within the \dn environment, not globally for the document. The catcodes option instructs the package to change the categories globally. This does not, however, change the categories as a part of the \dn command. If you assume that the categories can be changed somewhere in the middle of the document and you wish to set them properly by \dn, you can use the compat option. The macro \UnDevCatcodes changes catcodes back to the normal values within the \dn environment.

8 Using Devanāgarī in XƎLATEX

This topic does not fully fit here but it is described in this manual because the module for babel, which is also included in this package, can be used both in the traditional TEX and XƎTEX.

Using Devanāgarī in XƎTEX does not require any preprocessor. Instead you should use a Unicode OpenType font and enter Devanāgarī directly in your text editor. You do not install such fonts for XƎTEX, it takes the fonts installed in your operating system. OpenType fonts can have different features, the most important of them is

"Script". You have to specify "Script=Devanagari" when loading the font so that conjuncts are properly formed and i-matras are moved before the consonant. In XeLaTeX (the LaTeX format in XeTeX) you can easily achieve it by:

```
\usepackage{fontspec}
...
\fontspec[Script=Devanagari]{fontname}
```

Replace "fontname" with the name of the font. You can query the font name in your operating system. The FONTSPEC package maps the standard LaTeX font switching commands to the XeTeX primitives. Look into its manual for more details.

Unicode assigns special codes to Devanāgarī numerals. Automatically created number, e. g. the page number created by `\thepage`, will be printed in Arabic numerals. XeTeX contains the TECkit library for mapping characters. If you specify (as another option in the square brackets of the font switching command) "Mapping=devanagarinumerals", all Arabic numerals will be automatically converted to the corresponding Devanāgarī numerals.

Although the text may be entered directly in Devanāgarī, it may be useful to process the source text in the Velthuis Devanāgarī by XeLaTeX. This can also be achieved by TECkit remapping. Two mapping files are available: Mapping=velthuis-sanskrit and Mapping=velthuis. The latter is intended for Hindi. They differ in the only feature that the Sanskrit map will append a virama if the word ends in a consonant while the Hindi map will not. Both maps convert numerals too. These maps are still rather experimental. Examples of their usage can be found in files `xetex-misspaal.tex` and `xetex-examples.tex` which are distributed in the document directory of this package. Under the TEXMF tree their directory should be `doc/generic/velthuis`.

Remember that Velthuis transliteration employs the tilde character which normally denotes a nonbreakable space. If you want to utilise TECkit maps, you have to change its `\catcode` to 12 inside the Devanāgarī text, otherwise words like विज्ञापन will be printed incorrectly.

9 Using Hindi with babel

Since the DEVNAG package version 2.14.1 a new module for babel is released. This unifies the interface for multiligual documents where Hindi is one of the languages. The module is available for use both in traditional LaTeX and XeLaTeX. Usage in plain TeX is not yet supported but the module should work in plain XeTeX. Usage with the traditional LaTeX does require the full DEVNAG package. The module is built upon its macros as well as the preprocessor and fonts.

9.1 Installation

The Hindi module is not yet included in the standard babel distribution, you have to install it yourself. Fortunatelly installation is quite simple. It s not necessary to plug `hindi.dtx`, which you find in the documentation directory, to babel and generate all files. Instead the relevant files have been generated for you and are installed automatically with this package. The only thing you have to do is to inform LaTeX and XeLaTeX that the Hindi module is available. It is done in the `language.dat` file. Since we have no hyphenation patterns for Hindi[1], just append the following line to the end of the file:

```
hindi zerohyph.tex
```

[1] Some time ago hyphenation patterns for Sanskrit for XeLaTeX have been released on CTAN. Their usability for Hindi in XeLaTeX should be evaluated.

If you do not have permission to edit this file, you can make its copy in directory `tex/generic/config` under `texmf-var` which is usually world writable. Save the file and generate the formats. Unofficial but tested installation script can be found at `http://icebearsoft.euweb.cz/tex/csh_babel.php`.

9.2 Usage

If you want to use Hindi as a babel module, put the following command into the preamble of your document:

```
\usepackage[hindi]{babel}
```

Do not load `devanagari.sty`. The above mentioned syntax is valid both for LaTeX and XeLaTeX. The babel module knows that `devanagari.sty` is required in LaTeX and will load it automatically. Moreover, babel will not work correctly with older version of the DEVNAG package. It therefore employs version checking.

You may enter more comma separated languages in the square brackets. The last language will be the default. You can then switch languages by standard babel commands as `\selectlanguage{hindi}` and

```
\begin{otherlanguage}{hindi}
...
\end{otherlanguage}
```

If traditional LaTeX is used, you must still insert the source text inside the `{\dn }` group, otherwise the preprocessor will not recognize it.

9.3 Language attributes

Package `devanagari/sty` can be loaded with some options. When using the babel module this functionality is available only indirectly by specifying these options in the `documentclass` declaration. This is not preferred because these options should not be treated as document properties. Instead these options were reimplemented as babel language attributes. The attributes are selected by invoking:

`\languageattribute{hindi}{`⟨*list of attributes*⟩`}`

The command must appear after loading babel.

9.3.1 Attribute modernhindi

This attribute switches to the Modern Hindi captions. Remember that the preprocessor does not understand the language attributes and babel cannot read the preprocessor directives. You must therefore correctly insert either `@hindi` or `@modernhindi` directive in addition to the language attribute. The attribute has no effect in XeLaTeX because conjunct building is defined in the tables inside the OpenType font.

9.3.2 Font style attributes

The module implements attributes `bombay`, `calcutta`, `nepali`, `pen`, `penbombay`, `pencalcutta`, `pennepali` that serve the same purpose of the package options with the same name. These attributes are connected with the Velthuis fonts and are thus unavailable in XeLaTeX.

9.4 Customizing captions

The babel module makes use of the same definitions of date and captions as shown in tables 2 and 3. Their redefinition within the framework of babel is easy. Suppose that we want to change the title of Index to अनुक्रमणिका as we did in section 7.5. We achieve it by placing the following command to the preamble after loading babel:

```
\addto\captionshindi{\def\indexname{{\dn anukrama.nikaa}}}
```

The same can be done for `\cationsmodernhindi`. Notice that the command will be seen by the preprocessor.

If you want to do the same in XeLaTeX, enter the changed text directly in Unicode:

```
\addto\captionshindi{\def\indexname{अनुक्रमणिका}}
```

10 Vedic Macros

These macros put Vedic intonation marks above and below individual Devanagari letters and construct other characters generally used only in Vedic text.

There are two groups of these macros, one for Rig Veda, and one for Sama Veda. To use the Rig Veda macros, you must first enter the command `\dnveda` at some point after `\input dnmacs` in plain TeX or after `\usepackage{devanagari}` in LaTeX, and to use the Sama Veda accents, you must first type `\dnsamaveda`.

Both of these modes redefine standard macro names already used in Plain TeX and LaTeX. In Rig Veda mode the macros `\_`, `\=`, `\|`, and `\~` are redefined, while in Sama Veda mode, `\^` and `\@` are redefined. If your document already uses these macros in their original sense, then use `\dnveda` or `\dnsamaveda` only within `\dn` mode. Otherwise, use `\dnveda` or `\dnsamaveda` once at the beginning of the document.

This approach to macro names has been used because, when intonation marks are needed, they are needed very frequently and are inserted into parts of words, so the macro names should be very short and symbolic.

10.1 Rig Veda Macros

10.1.1 Anudatta (low) tone macro _, variable width

This macro takes one argument, the text letter. Example: `\dnveda ... {\dn \_{a}gnim}`

This macro may be combined with `\|` for a pluta mark: `\_{\|{3}}`.

The anudatta mark produced by this macro is nearly as wide as the letter and thus varies in width from one letter to another.

10.1.2 Anudatta (low) tone macro \=, fixed width

This macro takes one argument, the text letter. Example: `\dnveda ... {\dn \={a}gnim}`

This macro may be combined with `\|` for a pluta mark: `\={\|{3}}`.

The anudatta mark produced by this macro has a fixed width and is centered under the letter.

10.1.3 Svarita (rising) tone macro \|

This macro takes one argument, the text letter. Example: `\dnveda ... {\dn \|{ii}Le}`

This macro may be combined with `\|` for a pluta mark: `\|{\_{3}}`.

10.1.4 Pada separator macro \~

This macro inserts a pada separtor between two Devanagari letters.

Example: `\dnveda ... {\dn na\_{ra}\~maa}`

10.2 Usage Samples

This subsection provides two small usage samples of Vedic macros.

अग्निमीळे पुरोहितं यज्ञस्य देवमृत्विजम् । होतारं रत्नधातमम् १

The text above was typeset by:

```
{\dn\dnnum\Large\dncalcutta
{\dnveda

\_{a}gni\|{mii}Le \_{pu}ro\|{hi}ta.m \_{ya}j~na\|{sya}
\_{de}va\_{m.r}tvi\|{ja}m | \
ho\|{taa}ra.m ra\_{tna}dhaa\|{ta}mam \quad 1

}}
```

२ ३ १ २ ३१२ ३ २ ३ १ २
१ अग्न आ याहि वीतये गृणानो हव्यदातये
१ २र ३ १ २
नि होता सत्सि बर्हिषि १

The text above was typeset by:

```
\def\samaindent{\parindent=1.0in}
\def\dnitem#1{\noindent\llap{#1\space}\leftskip\parindent}
{\dn\dnnum\dncalcutta
{\dnsamaveda\samaindent
\dnitem{1} \^{a}{2}\^{gna}{3} \^{aa}{1} \^{yaa}{2}hi
\^{vii}{3}\^{ta}{1}\^{ye}{2} g.r\^{.naa}{3}\^{no}{2}
\^{ha}{3}\^{vya}{1}\^{daa}{2}taye \\
\^{ni}{1} ho\^{taa}{2ra} satsi \^{ba}{3}\^{rhi}{1}\^{.si}{2} \quad 1

}}
```

	ORIGINAL	BOMBAY	CALCUTTA	NEPALI
a	अ	अ	অ	
r̥	ऋ	ऋ	ঋ	
r̥̄	ॠ	ॠ	ৠ	
l̥	ऌ	ऌ	ঌ	
l̥̄	ॡ	ॡ	ৡ	
cha	छ	छ	ছ	
jha	झ	झ	ঝ	ञ
ṇa	ण	ण	ণ	
la	ल	ल	ল	
śa	श	श	শ	
1	१	१	১	१
5	५	५	৫	
8	८	८	৮	
9	९	९	৯	९

	ORIGINAL	BOMBAY	CALCUTTA	NEPALI
kṣa	क्ष	क्ष	ক্ষ	
kṣ-	क्ष	क्ष	ক্ষ	
ṅkṣa	ङ्क्ष	ङ्क्ष	ঙ্ক্ষ	
ṅkṣva	ङ्क्ष्व	ङ्क्ष्व	ঙ্ক্ষ্ব	
chya	छ्य	छ्य	ছ্য	
jña	ज्ञ	ज्ञ	জ্ঞ	
jñ-	ज्ञ	ज्ञ	জ্ঞ	
jh-	झ	झ	ঝ	ञ
ṇ-	ण	ण	ণ	
ṇṇa	ण्ण	ण्ण	ণ্ণ	
lla	ल्ल	ल्ल	ল্ল	
ś-	श	श	শ	
-ya	य	य	য	
hṇa	ह्ण	ह्ण	হ্ণ	

Table 4: Standard and Variant Devanāgarī Characters

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

Table 5: Supported Devanāgarī Ligatures

Regular

Original (dvng)	अ ऋ छ श ल ५ ८ ण ह्न झ क्ष ण्ड श्ल
Bombay (dvnb)	अ ऋ छ श ल ५ ८ ण ह्न झ क्ष ण्ड श्ल
Calcutta (dvnc)	অ ঋ ছ শ ল ৫ ৮ ণ হ্ন ঝ ক্ষ ণ্ড শ্ল
Nepali (dvnn)	अ ऋ छ श ल ५ ८ ण ह्न झ क्ष ण्ड श्ल

Oblique

Original (dvngi)	*अ ऋ छ श ल ५ ८ ण ह्न झ क्ष ण्ड श्ल*
Bombay (dvnbi)	*अ ऋ छ श ल ५ ८ ण ह्न झ क्ष ण्ड श्ल*
Calcutta (dvnci)	*অ ঋ ছ শ ল ৫ ৮ ণ হ্ন ঝ ক্ষ ণ্ড শ্ল*
Nepali (dvnni)	*अ ऋ छ श ल ५ ८ ण ह्न झ क्ष ण्ड श्ल*

Bold

Original (dvngb)	**अ ऋ छ श ल ५ ८ ण ह्न झ क्ष ण्ड श्ल**
Bombay (dvnbb)	**अ ऋ छ श ल ५ ८ ण ह्न झ क्ष ण्ड श्ल**
Calcutta (dvncb)	**অ ঋ ছ শ ল ৫ ৮ ণ হ্ন ঝ ক্ষ ণ্ড শ্ল**
Nepali (dvnnb)	**अ ऋ छ श ल ५ ८ ण ह्न झ क्ष ण्ड श्ल**

Bold Oblique

Original (dvngbi)	***अ ऋ छ श ल ५ ८ ण ह्न झ क्ष ण्ड श्ल***
Bombay (dvnbbi)	***अ ऋ छ श ल ५ ८ ण ह्न झ क्ष ण्ड श्ल***
Calcutta (dvncbi)	***অ ঋ ছ শ ল ৫ ৮ ণ হ্ন ঝ ক্ষ ণ্ড শ্ল***
Nepali (dvnnbi)	***अ ऋ छ श ल ५ ८ ण ह्न झ क्ष ण्ड श्ल***

Pen

Original (dvpn)	*अ ऋ छ श ल ५ ८ ण ह्न झ ण्ड श्ल*
Bombay (dvpb)	*अ ऋ छ श ल ५ ८ ण ह्न झ ण्ड श्ल*
Calcutta (dvpc)	*অ ঋ ছ শ ল ৫ ৮ ণ হ্ন ঝ ণ্ড শ্ল*
Nepali (dvpnn)	*अ ऋ छ श ल ५ ८ ण ह्न झ ण्ड श्ल*

Table 6: Devanāgarī Font Specimens

Devanagari Regular

बहुत से छोटे छोटे राजाओं की बोल चाल का ढंग भी, *जिस समय वे वाइसराय* से मिलने आए थे, **संक्षेप के साथ** लिखने के योग्य है। कोई तो दूर ही से *हाथ जोड़े* आए, **और दो एक ऐसे थे कि** जब एडिकांग के बदन ***झुकाकर इशारा करने*** पर भी उन्हों ने **सलाम न किया** तो एडिकांग ने *पीठ पकड़ कर* उन्हें धीरे से झुका दिया।

Devanagari Bombay

बहुत से छोटे छोटे राजाओं की बोल चाल का ढंग भी, *जिस समय वे वाइसराय* से मिलने आए थे, **संक्षेप के साथ** लिखने के योग्य है। कोई तो दूर ही से *हाथ जोड़े* आए, **और दो एक ऐसे थे कि** जब एडिकांग के बदन ***झुकाकर इशारा करने*** पर भी उन्हों ने **सलाम न किया** तो एडिकांग ने *पीठ पकड़ कर* उन्हें धीरे से झुका दिया।

Devanagari Calcutta

बहुत से छोटे छोटे राजाओं की बोल चाल का ढंग भी, *जिस समय वे वाइसराय* से मिलने आए थे, **संक्षेप के साथ** लिखने के योग्य है। कोई तो दूर ही से *हाथ जोड़े* आए, **और दो एक ऐसे थे कि** जब एडिकांग के बदन ***झुकाकर इशारा करने*** पर भी उन्हों ने **सलाम न किया** तो एडिकांग ने *पीठ पकड़ कर* उन्हें धीरे से झुका दिया।

Devanagari Nepali

बहुत से छोटे छोटे राजाओं की बोल चाल का ढंग भी, *जिस समय वे वाइसराय* से मिलने आए थे, **संक्षेप के साथ** लिखने के योग्य है। कोई तो दूर ही से *हाथ जोड़े* आए, **और दो एक ऐसे थे कि** जब एडिकांग के बदन ***झुकाकर इशारा करने*** पर भी उन्हों ने **सलाम न किया** तो एडिकांग ने *पीठ पकड़ कर* उन्हें धीरे से झुका दिया।

Devanagari Pen Regular

बहुत से छोटे छोटे राजाओं की बोल चाल का ढंग भी, जिस समय वे वाइसराय से मिलने आए थे, संक्षेप के साथ लिखने के योग्य है। कोई तो दूर ही से हाथ जोड़े आए, और दो एक ऐसे थे कि जब एडिकांग के बदन झुकाकर इशारा करने पर भी उन्हों ने सलाम न किया तो एडिकांग ने पीठ पकड़ कर उन्हें धीरे से झुका दिया।

Devanagari Pen Bombay

बहुत से छोटे छोटे राजाओं की बोल चाल का ढंग भी, जिस समय वे वाइसराय से मिलने आए थे, संक्षेप के साथ लिखने के योग्य है। कोई तो दूर ही से हाथ जोड़े आए, और दो एक ऐसे थे कि जब एडिकांग के बदन झुकाकर इशारा करने पर भी उन्हों ने सलाम न किया तो एडिकांग ने पीठ पकड़ कर उन्हें धीरे से झुका दिया।

Devanagari Pen Calcutta

बहुत से छोटे छोटे राजाओं की बोल चाल का ढंग भी, जिस समय वे वाइसराय से मिलने आए थे, संक्षेप के साथ लिखने के योग्य है। कोई तो दूर ही से हाथ जोड़े आए, और दो एक ऐसे थे कि जब एडिकांग के बदन झुकाकर इशारा करने पर भी उन्हों ने सलाम न किया तो एडिकांग ने पीठ पकड़ कर उन्हें धीरे से झुका दिया।

Devanagari Pen Nepali

बहुत से छोटे छोटे राजाओं की बोल चाल का ढंग भी, जिस समय वे वाइसराय से मिलने आए थे, संक्षेप के साथ लिखने के योग्य है। कोई तो दूर ही से हाथ जोड़े आए, और दो एक ऐसे थे कि जब एडिकांग के बदन झुकाकर इशारा करने पर भी उन्हों ने सलाम न किया तो एडिकांग ने पीठ पकड़ कर उन्हें धीरे से झुका दिया।

Table 7: Examples of Devanāgarī Faces