Devanagari Font Display
For U++

Basic Char set Unicode 0x0900 to 0x097F.

Extended Joined character's Glyph starts after 0x10000. Number of glyph depends on the font design.

With Linux OS - to correctly display all Indian scripts Harfbuzz is required.

For Roman scripts processing the characters from string sequentially and drawing next to each other renders the font correctly.

Whereas in Indian scripts two things are different.

- Sequence of characters stored in string is not same as displayed.
- Conjuncts are another important element needed to correctly render the Devanagari script. These are joined Glyph.
- In some cases placement of characters is interchanged.

I have created modified TextToSvgPath package to Display Devanagari font properly with following sequence using HarfBuzz.

- Load font file and Create font
- Create hb-buffer and populate.
- Shape it!
  - **It will process the string by organising the character sequence for display. Process zero-width joiner (ZWJ) , zero-width non-joiner (ZWNJ)**
- Get glyph information and positions out of the buffer.
- Render String – using modified hb**RenderCharacterSys** function.

Changes in Files

Draw.h

```
void   Render(FontGlyphConsumer& sw, double x, double y, int ch) const;
void   hbRender(FontGlyphConsumer& sw, double x, double y, int ch) const;
```

FontInt.h

```
void        RenderCharacterSys(FontGlyphConsumer& sw, double x, double y, int ch, Font fnt);
void        hbRenderCharacterSys(FontGlyphConsumer& sw, double x, double y, int ch, Font fnt);
```

Font.cpp

```
void Font::hbRender(FontGlyphConsumer& sw, double x, double y, int ch) const
{
	Mutex::Lock __(sFontLock);
	hbRenderCharacterSys(sw, x, y, ch, *this);
}
```

.

CODE – From

FontFc.cpp

```
void RenderCharacterSys(FontGlyphConsumer& sw, double x, double y, int ch, Font
fnt)
{
      FT_Face face = FTFace(fnt, NULL);
      int glyph_index = FT_Get_Char_Index(face, ch);
      if(glyph_index && FT_Load_Glyph(face, glyph_index, FT_LOAD_DEFAULT) == 0)
            RenderOutline(face->glyph->outline, sw, x, y + fnt.GetAscent(),
            fnt.IsItalic() && !(face->style_flags & FT_STYLE_FLAG_ITALIC));
}

//// Modified for HarfBuzz.
hb_buffer_get_glyph_positions gets the glyph positions from font. Hence no need
to get character index.

void hbRenderCharacterSys(FontGlyphConsumer& sw, double x, double y, int ch,
Font fnt)
{
      FT_Face face = FTFace(fnt, NULL);
//    int glyph_index = FT_Get_Char_Index(face, ch);
      int glyph_index = ch;
      if(glyph_index && FT_Load_Glyph(face, glyph_index, FT_LOAD_DEFAULT) == 0)
            RenderOutline(face->glyph->outline, sw, x, y + fnt.GetAscent(),
            fnt.IsItalic() && !(face->style_flags & FT_STYLE_FLAG_ITALIC));
}
```

Kannad

ಅಂಕಾರಯುತವಾಗಿಯೂ ಕೆತ್ತಲ್ಪಟ್ಟಿವೆ ಕನ್ನಡ_ಅಕ್ಷರಮಾಲೆ

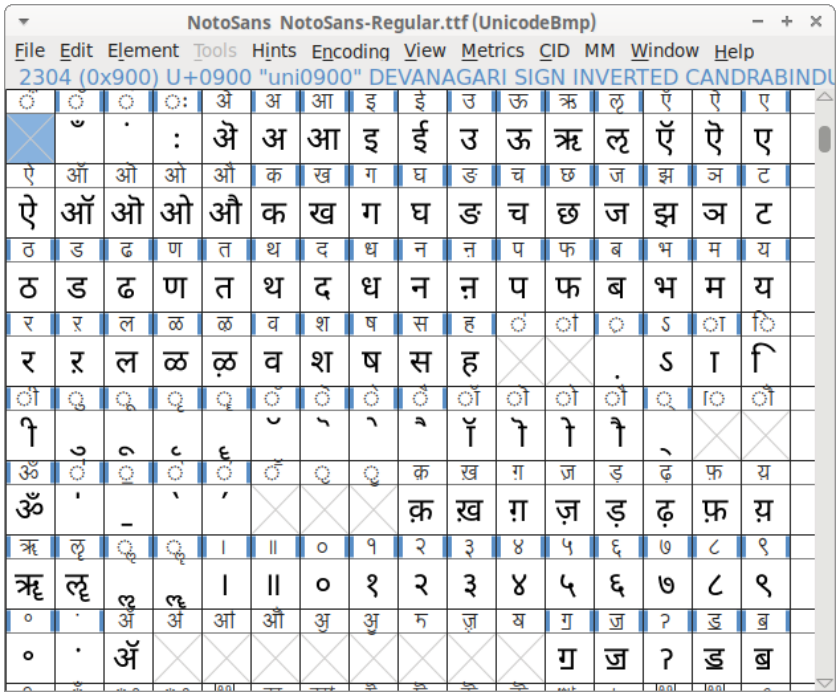
.
File **ToSvg.cpp** from **TextToHbSvgPath** package

```
String TextToHbSvgPath ( double x, double y, const char *text, Font fnt, bool
singleline )
{
	const char *fontfile;
	TextToSvg tts;

	fontfile = "/usr/share/fonts/truetype/noto/NotoSans-Regular.ttf";
	int i1 = Font::FindFaceNameIndex("NotoSans");

	Font fntMarathi ; //= Font(i1,FONT_SIZE);
	fntMarathi = Font(i1,FONT_SIZE);

	LOG ( fontfile );

	FT_Library ft_library;
	FT_Face ft_face;
	FT_Error ft_error;

	if ( ( ft_error = FT_Init_FreeType ( &ft_library ) ) )
		abort();

	if ( ( ft_error = FT_New_Face ( ft_library, fontfile, 0, &ft_face ) ) )
		abort();

	if ( ( ft_error = FT_Set_Char_Size ( ft_face, FONT_SIZE * 64, FONT_SIZE *
64, 0, 0 ) ) )
		abort();

	hb_font_t *hb_font;
	hb_font = hb_ft_font_create ( ft_face, NULL );

	/* Create hb-buffer and populate. */
	hb_buffer_t *hb_buffer;
	hb_buffer = hb_buffer_create ();
	hb_buffer_add_utf8 ( hb_buffer, text, -1, 0, -1 );
	hb_buffer_guess_segment_properties ( hb_buffer );

	/* Shape it! */
	hb_shape ( hb_font, hb_buffer, NULL, 0 );

	/* Get glyph information and positions out of the buffer. */
	unsigned int len = hb_buffer_get_length ( hb_buffer );
	hb_glyph_info_t *info = hb_buffer_get_glyph_infos ( hb_buffer, NULL );
	hb_glyph_position_t *pos = hb_buffer_get_glyph_positions (hb_buffer,
NULL);

	for ( unsigned int i = 0; i < len; i++ )
	{
		hb_codepoint_t gid   = info[i].codepoint;
		fntMarathi.hbRender ( tts, x, y, gid );
		x += pos[i].x_advance/64.0; // move the reference point

		if ( !singleline )
			tts.t << "\n";
	}
	return tts.t;
}
```

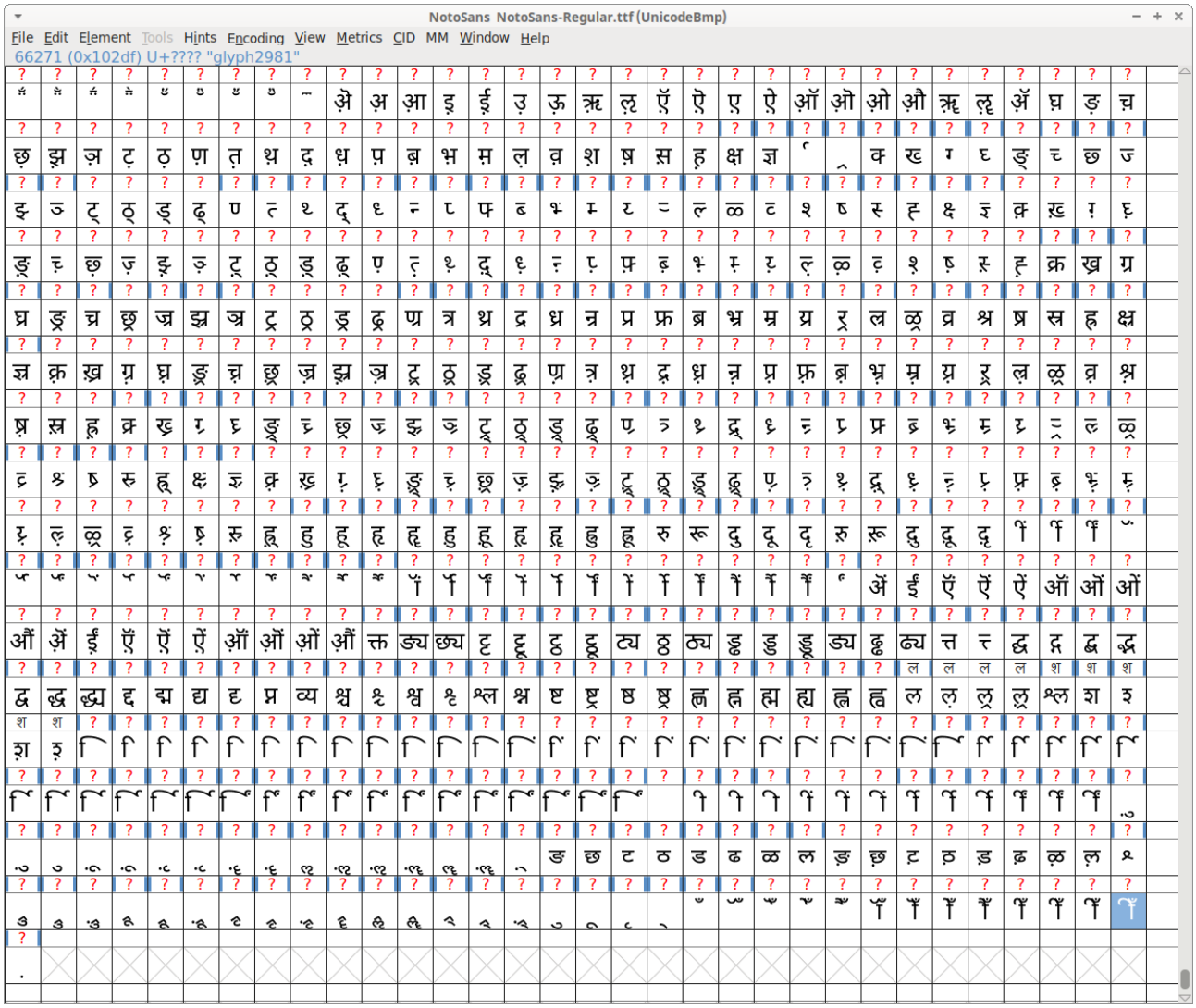
Sample Text
/* सन्निहित अन्तर्राष्ट्रीय प्रणाली सम्पत्ति धर्म अनुच्छेद स्थिति अंतर्राष्ट्रीयकरण गतिविधि
वेब की सम्पूर्ण क्षमता के उपयोग की दिशा में अग्रणी */
.

Basic Char Set

.



Extended Char set (Conjuncts - Joined Glyph and half Glyph)



.