GNU Texinfo font subsystem

Table of Contents

1 Font management

Texinfo's font management macros allow customization of fonts used in the printed document.

1.1 Introduction

In this chapter we introduce the basic concepts of font management in Texinfo.

1.1.1 Font features and attributes

Each font can be characterized by a number of *features*. For example, a font can be characterized by weight (boldness of glyph strokes), slant (inclination of glyph strokes), etc.

Font *attribute* is a particular representation of a feature. For example, the "weight" feature can be represented by attributes "light", "medium", "semibold", "bold", etc.; the "slant" feature by attributes "upright", "slanted", "italic", etc.

Most font features describe physical appearance of a font. However, two special features describe organizational aspects of fonts—font family and font encoding.

A collection of related fonts is grouped into a *font family*. All fonts in a font family generally have the same or similar design, representing variations on that particular design. For example, a font family may include an upright font, an italic font and a bold upright font. Examples of font families: Computer Modern Roman, Times, Helvetica.

Font encoding specifies the set and the order of characters represented within a font. Examples of font encodings: OT1 (Latin upper- and lower-case characters, arabic numerals and some additional glyphs), T1 (characters of Western-European scripts), T2A (characters of Cyrillic scripts). For a guide to TEX font encodings, see http://www.ctan.org/tex-archive/help/Catalogue/entries/encguide.html.

1.1.2 Current font attribute set

At font definition time, each font is associated with a set of attributes. Naturally, only one attribute can be specified per feature, but not every feature has to be represented.

A font then can be selected by specifying a list of attributes. At all times, Texinfo maintains a *current font attribute set*. Texinfo provides macros to initialize this set, as well as to modify it by adding attributes or removing attributes corresponding to certain features. If a font definition is found which matches every attribute in the resulting attribute set, the corresponding font is selected.

For example, let's assume the current font attribute set consists of the following attributes:

'`CMRoman`' "family" feature

'`OT1`' "encoding" feature

'`bold`' "weight" feature

If now we instruct Texinfo to add attributes '`light`' and '`italic`' and to remove the '`encoding`' feature, the font attribute set becomes:

'`CMRoman`'

‘`light`’ replaces the ‘`bold`’ attribute of the “weight” feature

‘`italic`’ “slant” feature

Of course, in order for this font change specification to be successful, a font must be associated previously with such font attribute set.

1.1.3 Font substitution

Sometimes it can be convenient to replace some attributes when a font with a certain set of attributes is requested. For example, a font family may contain slanted but not italic fonts. If we only define slanted fonts, all macros explicitly requesting italic font will fail when such font family is used. But we may decide that it is acceptable to use slanted fonts wherever italic fonts are being used. To achieve this, one solution is to create fake italic font definitions by duplicating definitions of slanted fonts and replacing the ‘`slanted`’ attribute with ‘`italic`’. An alternative, simpler solution is to define font substitution, indicating that all requests for a font with the ‘`italic`’ attribute should be satisfied with a font with the ‘`slanted`’ attribute and all other font attributes unchanged.

It is possible to specify more complex font substitutions which replace/add several attributes and/or remove attributes corresponding to certain features. All defined substitutions are applied in turn, starting with the current font attribute set, with the later substitutions performed on the result of the previous, so that it’s possible to define chains of substitutions. Note that font substitution works with attribute sets; only the final attribute set is used to look up the corresponding font, so intermediate sets do not have to be associated with any font.

Unlike font selection, font substitution is “permissive”, that is, in order for a font substitution to be applied, its attribute set doesn’t need to match the current font attribute set exactly, it can just be a subset of the font attribute set. For example, if a font substitution is defined to apply to the set of two attributes ‘`CMRoman`’ and ‘`OT1`’, the substitution will also apply to any of the following attribute sets:

- ‘`CMRoman`’, ‘`OT1`’, ‘`upright`’;
- ‘`CMRoman`’, ‘`OT1`’, ‘`bold`’;
- ‘`CMRoman`’, ‘`OT1`’, ‘`upright`’, ‘`bold`’;

but to none of the following:

- ‘`CMRoman`’;
- ‘`OT1`’;
- ‘`CMRoman`’, ‘`upright`’;
- ‘`OT1`’, ‘`upright`’.

“Restrictive” font substitutions, where attribute sets have to mach exactly, are (currently?) not supported.

1.1.4 Other

Scaling factor is an integer equal to magnification ratio times 1000.

1.2 Font definition macros

Font attributes can be defined with the command

`@newfontatrrs` *feature attribute-list*

where *feature* is the font feature to associate the attributes with, *attribute-list* is a comma-separated list of one or more attributes to define. It is not an error to define an already defined attribute, as long as that attribute is associated with the same font feature as before.

For example, the following command

```
@newfontattrs encoding OT1,OMS,OML,OMX
```

defines the classic TEX font encodings. After the above definition, the following command will be valid:

```
@newfontattrs encoding OT1,T1
```

but the following command will generate an error (feature names are case-sensitive):

```
@newfontattrs Encoding OT1
```

A font can be defined with the command

`@newfont` *scale font size lskip*`[`*,reduced,small,smaller*`] [`*attr-list*`]`

Here optional parts are in square brackets. The arguments are as follows:

scale Font's relative scaling factor (see below).

font Font file name (e.g., `cmr10`).

size Design size of the font (see below) specified as a TEX dimension or a number (in which case TEX points are assumed).

lskip Recommended line skip scaling factor which will be multiplied by the selected font size to get the actual line skip.

reduced Scaling factor for a reduced-size font ("one size smaller"), used for acronyms. Default is 909 (10/11).

small Scaling factor for a small-size font ("two sizes smaller"), used for indices, footnotes, small examples, etc. Default is 818 (9/11).

smaller Scaling factor for an even-smaller-size font ("three sizes smaller"), used for superscripts, subscripts, the LATEX logo, etc. Default is 727 (8/11).

attr-list List of attributes to associate with the font. If omitted, attributes from the last `@newfont` command are applied.

This scheme doesn't provide for situation when we want different reduced, small *and* smaller *settings for the same font for different size ranges.*

Different font families have different notions of font size. For example, Bera fonts at 10 pt look much bigger than Computer Modern fonts at 10 pt. When mixing fonts and font families, their sizes must be scaled to achieve visual uniformity. This is what the relative scaling factor (the first argument of `@newfont`) is for—it specifies the scaling factor which needs to be applied to a font to match a corresponding font from the Computer Modern collection of fonts (which are the default fonts of Texinfo).

Maybe it makes better sense to specify scale *for the entire family and not for individual fonts. Not sure if fonts from the same family would ever need different scaling factors. One*

(hypothetical?) case I can think of is when a font family provides fonts at several design sizes, and those design sizes scale differently to the corresponding Computer Modern design sizes, so each design size has to be tweaked individually.

Each font has a *design size*, which is the size in which the designer intended the font to be displayed (the *size* argument of `@newfont`). To produce a font in a size other than the design size, Texinfo can scale a font. Many font families provide fonts in only one design size, usually 10 pt. When fonts are provided in several design sizes, it is best to define all the provided design sizes.

For example, Computer Modern Roman font family provides five design sizes (7 pt, 8 pt, 9 pt, 10 pt and 12 pt) for the italic medium-weight face, but only one design size (10 pt) for the upright caps and small caps face. The font encoding for both faces is OT1. Therefore, part of the definition of the CMRoman font family dealing with italic and caps and small caps faces might look like the following:

```
@newfontattrs family   CMRoman
@newfontattrs encoding OT1
@newfontattrs slant    upright,italic
@newfontattrs caps     normalcaps,capssmallcaps

@newfont 1000 cmti7   7  1350 CMRoman,OT1,italic,normalcaps
@newfont 1000 cmti8   8  1300
@newfont 1000 cmti9   9  1250,1
@newfont 1000 cmti10  10 1200
@newfont 1000 cmti12  12 1150
@newfont 1000 cmcsc10 10 1200 CMRoman,OT1,upright,capssmallcaps
```

1.3 Font selection macros

The `@setfont` command sets the current attribute list and then selects the font associated with that list:

`@setfont{`*`attribute-list`*`}`

If you don't want to specify all attributes but just want to add certain attributes to the current attribute list and/or remove attributes for certain features, use the command

`@modfont{`*`feature-list`*`}{`*`attribute-list`*`}`

Any attributes corresponding to features from *feature-list* will be removed from the current attribute list, attributes from *attribute-list* will be added to it, and the resulting attribute list will be used by Texinfo to select a font.

1.3.1 Relative font scaling

By default, all fonts are scaled to match the Computer Modern fonts, and the Computer Modern fonts come out at their "natural" sizes. This happens when base font scaling factor is set to 1000, the relative scaling factor of the Computer Modern fonts. You can set a different base scaling factor using the command

`@fontbasescale` *`scale`*

If *scale* is omitted, the current font's relative scaling factor will be used.

1.4 Font substitution macros

Each font substitution consists of three sets: filter (set of attributes), removed features (set of features) and added attributes (set of attributes). When selecting a font, Texinfo applies the list of defined substitutions to the current attribute list, and uses the resulting attribute list to selects a font.

examines each substitution in turn, applying those whose filter matches the current attribute list (i.e., those whose filter contains each attribute from the list) to the result of the previous substitutions on the current attribute list. The resulting attribute list as used to select a font.

To add a substitution to the head of the substitution list, use the command

```
@fontsubstpre =filter -removed-features +added-attributes
```

The following command adds a substitution to the tail of the substitution list:

```
@fontsubstpost =filter -removed-features +added-attributes
```

1.5 Font collections

`@declarefontcollection`

`@fontcollection`

1.5.1 Font styles

Font styles are a way to apply one of the defined font families to the text. It is possible to specify font styles for the various elements of a document individually, such as body text, page headings and footings, table of contents, indexes, and chapter, section, subsection and sub-subsection titles.

1.5.2 Font style selection

Below are the styles defined by Texinfo, with the corresponding default meanings and commands which select them. All the style commands take a single argument in braces and typeset it according to font attributes specified for the style.

‘`serif`’ serifed fonts (CMRoman), applied with `@serif`.

‘`sans`’ sans serif fonts (CMSans), applied with `@sansserif`.

‘`mono`’ monospace fonts (CMMono), applied with `@t`.

‘`default`’ fonts used in absence of any style switches, and applied with `@r` (CMRoman).

‘`math`’ fonts used in math mode (CMMath), no explicit switches.

For example, the command

```
@sansserif{text}
```

typesets ‘`text`’ using the ‘`sans`’ font style (which results in a sans serif font by default).

1.5.3 Font styles for document elements

It is possible to customize each of the above styles separately for each element of the document. Texinfo associates styles with the following elements:

'`*`'	Default, used for body text. Attributes from this 'element' are also inherited by other elements, unless those elements redefine them.
'`heading`'	
'`footing`'	Page headings and footings.

```
@fontaxes   * mono    .
@fontaxes   * math    .

@fontsize   * 11
@fontweight * m
@fontshape  * n
```

After you have finished specifying font attributes for the styles, you should activate them with `@setfonts`. To revert to the default styles, call `@resetfonts`.

Below is a complete example of a document style specification.

```
@c Use Computer Modern fonts with no axes (the defaults).
@unsetfonts

@c The body text will be 12pt medium-weight upright font.
@fontsize * 12

@c Page headings are in a smaller italic font; footings are in
@c smaller upright.
@fontsize  heading 10
@fontshape heading it
@fontsize  footing 10

@c TOC is in the default face and smaller; indexes are in the default
@c face and even smaller.
@fontsize toc   11
@fontsize index 10

@c Font attributes for the title.  We are not going to use @serif,
@c @sans, etc., so we don't care about non-default styles.
@fontfamily title * SomeFunkyFontFamily
@fontaxes   title * of,pf
@fontsize   title 20
@fontweight title bx
@fontshape  title it

@c Use sans fonts for chapters, sections and subsections.  We exchange
@c the meanings of @sans and @serif.

@fontfamily chapter *     CMSans
@fontfamily chapter serif CMSans
@fontfamily chapter sans  CMRoman
@fontfamily chapter math  CMBrightMath
@fontsize   chapter 17

@fontfamily section *     CMSans
@fontfamily section serif CMSans
@fontfamily section sans  CMRoman
```

```
@fontfamily section math  CMBrightMath
@fontsize   section 14

@fontfamily subsection *     CMSans
@fontfamily subsection serif CMSans
@fontfamily subsection sans  CMRoman
@fontfamily subsection math  CMBrightMath
@fontsize   subsection 12

@c Don't forget to activate the styles.
@setfonts
```

1.6 Input encodings

1.7 Font encodings

Index