

# CD-I Font Format Specification

---

All of the glyphs in a font module are numbered using an unsigned sixteen bit integer. This glyph number is required to access a glyph in a font. Since some character code standards have codes which are sixteen bit quantities, this format provides three methods of interpreting (e.g. translating to glyph numbers) the character data passed to the text functions, eight bit, seven/fifteen bit, and sixteen bit. The eight bit method is usable for most alphanumeric type languages. When the eight bit method is being used, only those glyphs in a font which are numbered from 0 to 255 are accessible. In the seven/fifteen method, the most significant bit of a character code is checked to determine whether the codes is a seven bit quantity (one byte) or a fifteen bit quantity (two byte). When the most significant bit is reset (i.e. to zero), then a seven bit quantity is represented. All seven bit quantities map to glyph numbers 0 to 127 and all fifteen bit quantities map to glyph numbers 32768 to 65535. This method is useful because it can support ISO 646 (or ISO 8859-1: \$00 - \$7F) code and shifted JIS Kanji code in the same data stream. In the sixteen bit mode, all character codes map directly to glyph numbers. Each font supports a subrange of the possible range of glyph numbers. Since all character codes map uniquely to glyph numbers, each font, in effect, supports a subrange of character codes. The glyph is selected from the font that supports the required subrange. If two or more fonts contain glyphs with the same glyph number(s), the font with the lowest font number will be used.

Text in virtually any size, style, or font is supported by the software. These attributes are controlled by a font module, which is an OS-9 data module that contains the bitmap images for each displayable character.

A font module consists of six parts, a module header, the font data section, the glyph offset table, the glyph data table, the font bitmap, and the module CRC. The module header and module CRC are standard parts of an OS-9 memory module.

*microware*

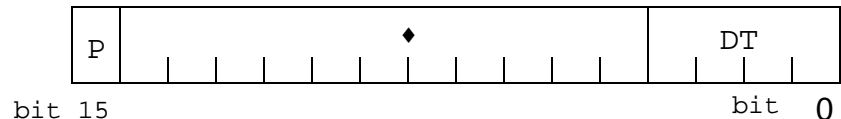
At the module's entry point, (which is declared in the module header) is the font data section. This data section is usually placed immediately following the module header and contains general information about the font. The format of the font data section is as follows.

**Table 0-1 Structure of Text Font Data Section**

Offset	Length	Name	Description
0	2	font_type	Font type & flags
2	2	fnt_width	Maximum Glyph cell width <sup>1</sup>
4	2	fnt_height	Glyph cell height <sup>1</sup> (ascent+descent)
6	2	fnt_ascent	Ascent <sup>1</sup> of character cell above baseline
8	2	fnt_descent	Descent <sup>1</sup> of character cell below baseline
10	2	fnt_pxlsz	Pixel size in bits
12	2	fnt_frstch	First character value of font
14	2	fnt_lastch	Last character value of font
16	4	fnt_lnlcn	Line length of first font bitmap in bytes <sup>2</sup>
20	4	fnt_offstbl	Offset <sup>3</sup> to glyph offset table
24	4	fnt_databl	Offset <sup>1</sup> to glyph data table
28	4	fnt_maploff	Offset <sup>1</sup> to first bitmap <sup>2</sup>
32	4	fnt_map2off	Offset <sup>1</sup> to second bitmap <sup>2</sup> RGB only

1. Value in pixels.
2. fnt\_maploff and fnt\_map2off must reference addresses which are long word aligned. fnt\_lnlcn must be divisible by 4.
3. Offset values are taken from the beginning of the font data section.

## Font Data Type/Flags Format:



P = Proportional/mono-spaced flag  
 0 = Monospace  
 1 = Proportional spaced  
 DT = Data type  
 0 = Single-bit  
 1 = Double-bit  
 2 = Quad-bit  
 3 = CLUT4  
 4 = CLUT7  
 5 = CLUT8  
 6-8 = Reserved  
 9 = RGB555  
 10-15 = Reserved

The glyph offset table is usually placed immediately following the font data section. This table is required in all fonts. It is merely a one dimensional array of sixteen bit integers which represent the horizontal distance in pixels from the left edge of the font bitmap to the left edge of the glyph. There must be one entry in the glyph offset table for each glyph in the range specified by the first glyph number and last glyph number in the font data section. If a particular glyph number in that range is not displayable, then its entry in the glyph offset table should be set to -1.

The glyph data table is usually placed immediately following the glyph offset table. There must be one entry in the glyph data table for each glyph in the range specified by the font data section, whether displayable or not. Each entry in the glyph data table is formatted as shown in the table below.

**Table 0-2 Structure of each Entry in the Glyph Data Table**

Offset	Length	Description
0	1	Character width in pixels
1	3	Reserved

The font bitmap is usually placed immediately following the glyph data table. The font bitmap is a rectangular array of pixels of the same form as a drawmap. Its height is the same as the character cell height. All of the glyphs for the font are placed in the font bitmap in sequential order from left to right.