Bitmap Distribution Format

Version 2.1

X Window System Standard

X Version 11, Release 6.7

The Bitmap Distribution Format (BDF), Version 2.1, is an X Consortium standard for font interchange, intended to be easily understood by both humans and computers.

File Format

Character bitmap information will be distributed in an USASCII-encoded, human-readable form. Each file is encoded in the printable characters (octal 40 through 176) of USASCII plus carriage return and linefeed. Each file consists of a sequence of variable-length lines. Each line is terminated either by a carriage return (octal 015) and linefeed (octal 012) or by just a linefeed.

The information about a particular family and face at one size and orientation will be contained in one file. The file begins with information pertaining to the face as a whole, followed by the information and bitmaps for the individual characters.

A font bitmap description file has the following general form, where each item is contained on a separate line of text in the file. Tokens on a line are separated by spaces. Keywords are in upper case, and must appear in upper case in the file.

1. The word STARTFONT followed by a version number indicating the exact file format used. The version described here is 2.1.
2. Lines beginning with the word COMMENT may appear anywhere between the STARTFONT line and the ENDFONT line. These lines are ignored by font compilers.
3. The word FONT followed by either the XLFD font name (as specified in part III) or some private font name. Creators of private font name syntaxes are encouraged to register unique font name prefixes with the X Consortium to prevent naming conflicts. Note that the name continues all the way to the end of the line and may contain spaces.
4. The word SIZE followed by the *point size* of the characters, the *x resolution*, and the *y resolution* of the device for which these characters were intended.
5. The word FONTBOUNDINGBOX followed by the *width in x*, *height in y*, and the x and y displacement of the lower left corner from the *origin*. (See the examples in the next section.)
6. Optionally, the word STARTPROPERTIES followed by the number of properties (p) that follow.
7. Then come p lines consisting of a word for the *property name* followed by either an integer or string surrounded by double-quote (octal 042). Internal double-quote characters are indicated by using two in a row.

Properties named FONT_ASCENT, FONT_DESCENT, and DEFAULT_CHAR should be provided to define the logical font-ascent and font-descent and the default-char for the font. These properties will be removed from the actual font properties in the binary form produced by a compiler. If these properties are not provided, a compiler may reject the font or may compute (arbitrary) values for these properties.

8. The property section, if it exists, is terminated by ENDPROPERTIES.
9. The word CHARS followed by the number of character segments (*c*) that follow.
10. Then come *c* character segments of the form:
    a. The word STARTCHAR followed by up to 14 characters (no blanks) of descriptive *name* of the glyph.
    b. The word ENCODING followed by one of the following forms:
        i. <n> – the glyph index, that is, a positive integer representing the character code used to access the glyph in X requests, as defined by the encoded character set given by the CHARSET_REGISTRY-CHARSET_ENCODING font properties for XLFD conforming fonts. If these XLFD font properties are not defined, the encoding scheme is font-dependent.
        ii. −1 <n> – equivalent to form above. This syntax is provided for backward compatibility with previous versions of this specification and is not recommended for use with new fonts.
        iii. −1 – an unencoded glyph. Some font compilers may discard unencoded glyphs, but, in general, the glyph names may be used by font compilers and X servers to implement dynamic mapping of glyph repertoires to character encodings as seen through the X protocol.
    c. The word SWIDTH followed by the *scalable width* in x and y of character. Scalable widths are in units of 1/1000th of the size of the character. If the size of the character is *p* points, the width information must be scaled by *p*/1000 to get the width of the character in printer's points. This width information should be considered as a vector indicating the position of the next character's origin relative to the origin of this character. To convert the scalable width to the width in device pixels, multiply SWIDTH times *p*/1000 times *r*/72, where *r* is the device resolution in pixels per inch. The result is a real number giving the ideal print width in device pixels. The actual device width must of course be an integral number of device pixels and is given in the next entry. The SWIDTH y value should always be zero for a standard X font.
    d. The word DWIDTH followed by the width in x and y of the character in device units. Like the SWIDTH, this width information is a vector indicating the position of the next character's origin relative to the origin of this character. Note that the DWIDTH of a given ‘‘hand-tuned’’ WYSIWYG glyph may deviate slightly from its ideal device-independent width given by SWIDTH in order to improve its typographic characteristics on a display. The DWIDTH y value should always be zero for a standard X font.
    e. The word BBX followed by the width in x (*BBw*), *height* in y (*BBh*), and x and y displacement (*BBox*, *BBoy*) of the lower left corner from the *origin* of the character.
    f. The optional word ATTRIBUTES followed by the attributes as 4 *hex-encoded* characters. The interpretation of these attributes is undefined in this document.
    g. The word BITMAP.
    h. *h* lines of *hex-encoded* bitmap, padded on the right with zeros to the nearest byte (that is, multiple of 8).
    i. The word ENDCHAR.
11. The file is terminated with the word ENDFONT.

Metric Information

Figures 1 and 2 best illustrate the bitmap format and character metric information.

BBw = 9, BBh = 22, BBox = -2, BBoy = -6
DWIDTH = 8 0
SWIDTH = 355 0
“+” = character origin and width

Figure 1: An example of a descender

BBh = 6, BBw = 4, BBox = +2, BBoy = +12
DWIDTH = 5 0
SWIDTH = 223 0

Figure 2: An example with the origin outside the bounding box

An Example File

The following is an abbreviated example of a bitmap file containing the specification of two characters (the j and quoteright in figures 1 and 2).

```
STARTFONT 2.1
COMMENT This is a sample font in 2.1 format.
FONT -Adobe-Helvetica-Bold-R-Normal--24-240-75-75-P-65-ISO8859-1
SIZE 24 75 75
FONTBOUNDINGBOX 9 24 -2 -6
STARTPROPERTIES 19
FOUNDRY "Adobe"
FAMILY "Helvetica"
WEIGHT_NAME "Bold"
SLANT "R"
SETWIDTH_NAME "Normal"
ADD_STYLE_NAME ""
PIXEL_SIZE 24
POINT_SIZE 240
RESOLUTION_X 75
RESOLUTION_Y 75
SPACING "P"
AVERAGE_WIDTH 65
CHARSET_REGISTRY "ISO8859"
CHARSET_ENCODING "1"
MIN_SPACE 4
FONT_ASCENT 21
FONT_DESCENT 7
COPYRIGHT "Copyright (c) 1987 Adobe Systems, Inc."
NOTICE "Helvetica is a registered trademark of Linotype Inc."
ENDPROPERTIES
CHARS 2
STARTCHAR j
ENCODING 106
SWIDTH 355 0
DWIDTH 8 0
BBX 9 22 -2 -6
BITMAP
0380
0380
0380
0380
0000
0700
0700
0700
0700
0E00
0E00
0E00
0E00
0E00
1C00
1C00
1C00
```

```
1C00
3C00
7800
F000
E000
ENDCHAR
STARTCHAR quoteright
ENCODING 39
SWIDTH 223 0
DWIDTH 5 0
BBX 4 6 2 12
ATTRIBUTES 01C0
BITMAP
70
70
70
60
E0
C0
ENDCHAR
ENDFONT
```